



SIMATIC HMI

WinCC V7.0 SP1

MDM - WinCC: Tools (SmartTools,
User Archive, interfaces)

System Manual

SmartTools

1

User archive

2

COM Provider in the Layout
Editor

3

OPC - Open Connectivity

4

Print of the Online Help

11/2008

Legal information

Warning notice system

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

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

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

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

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

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

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

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation for the specific task, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be adhered to. The information in the relevant documentation must be observed.

Trademarks

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

Disclaimer of Liability

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

Table of contents

1	SmartTools.....	9
1	Resources.....	9
1.1	Overview.....	9
1.2	Tag Export/Import.....	10
1.2.1	Tag Export/Import.....	10
1.2.2	Installation of Export/Import Tags.....	10
1.2.3	Operation.....	11
1.2.4	The Tag Export/Import tool.....	12
1.2.5	Connections.....	13
1.2.6	File structures.....	13
1.2.7	Tag.....	14
1.2.8	Appendix.....	16
1.3	Tag simulator.....	18
1.3.1	The Tag simulator.....	18
1.3.2	Using the tag Simulator.....	19
1.3.3	Functions of the simulator.....	19
1.3.4	Installing the simulator.....	20
1.3.5	Adding/deleting tags.....	21
1.3.6	Parameter assignment of the functions.....	21
1.3.7	Activating/deactivating tags.....	22
1.3.8	Display of the tags.....	22
1.3.9	Loading/saving simulation data.....	23
1.3.10	FAQ.....	23
1.4	Dynamic Wizard Editor.....	23
1.4.1	Dynamic Wizard Editor.....	23
1.4.2	Overview.....	23
1.4.3	Installation of the Dynamic Wizard Editor.....	26
1.4.4	Structure.....	27
1.4.4.1	Structure.....	27
1.4.4.2	Toolbar.....	28
1.4.4.3	Editor window.....	29
1.4.4.4	Help Editor.....	30
1.4.4.5	Output window.....	31
1.4.5	Structure of a Dynamic Wizard function.....	32
1.4.5.1	Structure of a Dynamic Wizard function.....	32
1.4.5.2	Dynamic Wizard dialog.....	32
1.4.5.3	Integrating header files and DLLs.....	33
1.4.5.4	Language-dependent definitions.....	33
1.4.5.5	Wizard flags.....	34
1.4.5.6	Property list.....	35
1.4.5.7	System interface.....	36
1.4.5.8	Global variables.....	37
1.4.5.9	Options list.....	37
1.4.5.10	Trigger list.....	40

1.4.5.11	Display of parameter assignment.....	42
1.4.5.12	Wizard Functions for parameter input.....	43
1.4.5.13	Wizard functions for generating dynamics.....	68
1.4.5.14	Wizard WinCC functions.....	80
1.4.5.15	Wizard progress functions.....	84
1.4.5.16	Wizard Windows functions.....	87
1.4.6	Examples.....	92
1.4.6.1	Examples.....	92
1.4.6.2	Demo Wizard.....	93
1.4.6.3	Dynamic motor.....	95
1.5	Documentation Viewer.....	98
1.5.1	WinCC Documentation Viewer.....	98
1.5.2	Installing WinCC Documentation Viewer.....	98
1.5.3	Description.....	99
1.5.4	Creating the .emf file(s).....	100
1.6	WinCC CrossReferenceAssistant.....	101
1.6.1	WinCC CrossReferenceAssistant.....	101
1.6.2	Installation of the CrossReferenceAssistant.....	101
1.6.3	General.....	102
1.6.4	Known functions (script management).....	103
1.6.5	Project selection.....	106
1.6.6	File selection.....	107
1.6.7	Conversion.....	108
1.6.8	Expanded settings.....	109
1.7	WinCC Communication Configurator.....	111
1.8	WinCC Configuration Tool.....	112
1.8	Resources.....	112
1.8.1	Introduction.....	112
1.8.2	System requirements.....	113
1.8.3	Installation of the ConfigurationTool.....	113
1.8.4	Interface.....	114
1.8.4.1	Interface.....	114
1.8.4.2	Toolbar.....	114
1.8.4.3	Dropdown menu.....	115
1.8.4.4	Pop-up menu.....	117
1.8.4.5	Status bar.....	118
1.8.5	Operation of the WinCC configuration tool.....	118
1.8.5.1	Operation of the configuration tool.....	118
1.8.5.2	Creating a new project folder.....	121
1.8.5.3	Worksheets.....	131
1.8.5.4	Dialogs.....	189
1.8.5.5	Processing configured data.....	195
1.8.5.6	Creating objects by means of the pop-up menu of the tag table.....	211
1.8.5.7	Changing the data storage location.....	227
1.8.6	Diagnostics.....	228
1.8.6.1	Diagnostics.....	228
1.8.6.2	Error column.....	229
1.8.7	Tips and tricks.....	231
1.8.7.1	Tips and tricks.....	231
1.8.7.2	Example of a quantity structure in the configuration tool.....	231

1.8.7.3	Changing the worksheet.....	233
1.8.7.4	Row limitation.....	235
1.8.7.5	Address generation.....	235
1.8.7.6	VBA macros.....	239
1.8.7.7	Special characters.....	239
1.8.7.8	Address strings for Simatic S7 Protocol Suite.....	240
1.8.7.9	Packages.....	242
1.9	WinCC Archive ConfigurationTool.....	243
1.9	Resources.....	243
1.9.1	Introduction.....	243
1.9.2	System requirements.....	244
1.9.3	Installation of the Archive ConfigurationTool.....	245
1.9.4	Quick start.....	245
1.9.5	Operating WinCC Archive.....	252
1.9.5.1	Operating WinCC Archive.....	252
1.9.5.2	Creating an archive folder.....	253
1.9.5.3	Configuring a process value archive.....	259
1.9.5.4	Configuring a compressed archive.....	276
1.9.5.5	Checking the archive data.....	284
1.9.5.6	Creating, modifying and deleting.....	289
2	User archive.....	299
2	Resources.....	299
2.1	Introduction to User Archive.....	299
2.2	The Editor "User Archive".....	300
2.2.1	Introduction to Editor "User Archive".....	300
2.2.1.1	The Editor "User Archive".....	300
2.2.1.2	The WinCC UserArchiveControl.....	301
2.2.1.3	The standard function of the WinCC script language.....	301
2.2.1.4	Application areas for the User Archive.....	302
2.2.1.5	Functional Scope of User Archives.....	304
2.2.2	The Editor "User Archive".....	306
2.2.2.1	Structure of the "User Archive" editor.....	306
2.2.2.2	The menus of the "User Archive" editor.....	306
2.2.2.3	The toolbar of the "User Archive" editor.....	315
2.2.2.4	The table window of the "User Archive" editor.....	316
2.2.3	Configuration.....	317
2.2.3.1	Configuration.....	317
2.2.3.2	Configuration of a User Archive.....	317
2.2.3.3	Modification Configuration of User Archives.....	339
2.2.3.4	Configuration of views.....	340
2.2.3.5	Example of a User Archive.....	353
2.2.4	Data exchange using SIMATIC S5 / S7.....	354
2.2.4.1	SIMATIC message frame interface.....	354
2.2.4.2	Data Exchange with S5 and S7 via WinCC Tags.....	354
2.2.4.3	Data exchange with S5 and S7 via raw data tags.....	355
2.2.4.4	Data format differences between WinCC and S5/S7.....	360
2.2.5	Appendix.....	361
2.2.5.1	Appendix.....	361
2.2.5.2	The SQL language.....	361
2.2.5.3	Alphabetical list of SQL keywords.....	362

2.2.5.4	Quantity framework.....	363
2.2.5.5	Performance to read and write tags.....	364
2.2.5.6	Performance of the screen opening times.....	364
2.3	User Archive Scripts.....	365
2.3.1	Standard Functions of the WinCC Script Language.....	365
2.3.2	Configuration of Actions.....	366
2.3.3	Functions for Editing and Displaying User Archives.....	366
2.3.4	Handles for the Configuration of User Archives.....	368
2.3.5	Handles for the Runtime Functions.....	368
2.3.6	Example of a Script.....	371
2.3.7	Appendix.....	379
2.3.7.1	Description of the Standard Functions of the WinCC Script Language.....	379
2.3.7.2	Standard functions for user archives.....	379
2.3.7.3	Functions for Configuring User Archives.....	382
2.3.7.4	General Runtime Functions.....	394
2.3.7.5	Archive-specific Runtime Functions.....	400
2.4	WinCC UserArchiveControl.....	424
2.4	Resources.....	424
2.4.1	WinCC UserArchiveControl.....	424
2.4.2	Configuring the UserArchiveControl.....	425
2.4.2.1	How to configure the UserArchiveControl.....	425
2.4.2.2	How to define the contents of the UserArchiveControl.....	427
2.4.2.3	How to configure the display for the table.....	430
2.4.2.4	How to configure the toolbar and the status bar	434
2.4.2.5	How to export runtime data.....	438
2.4.2.6	How to define the effect of the online configuration.....	439
2.4.2.7	How to make the toolbar for the UserArchiveControl dynamic.....	441
2.4.3	Operation in runtime.....	442
2.4.3.1	Operating the UserArchiveControl in runtime.....	442
2.4.3.2	To process the data in the UserArchiveControl:.....	445
2.4.3.3	How to select the data of the user archive.....	446
2.4.3.4	How to sort the display of user archive data.....	448
2.5	Before WinCC V7: WinCC User Archives Table Element.....	449
2.5	Resources.....	449
2.5.1	User Archives Table Element.....	449
2.5.2	Functionality.....	449
2.5.3	Configuration of User Archives Table Element.....	451
2.5.3.1	Configuring a User Archives Table Element.....	451
2.5.3.2	Place the User Archive Table Element in a process screen.....	451
2.5.3.3	Define properties of the User Archives Table Element.....	453
2.5.3.4	Delete the User Archives Table Element.....	454
2.5.4	Properties of WinCC User Archives Table Element.....	455
2.5.4.1	Properties of WinCC User Archives Table Element.....	455
2.5.4.2	"General" tab.....	456
2.5.4.3	"Columns" tab.....	458
2.5.4.4	"Toolbar" Tab.....	459
2.5.4.5	"Status Bar" Tab.....	460
2.5.4.6	"Filter/ Sorting" tab.....	461
2.5.4.7	"Fonts" Tab.....	463
2.5.4.8	"Colors" tab.....	464
2.5.5	Configuring a Form View.....	465

2.5.5.1	Configuring a Form View.....	465
2.5.5.2	Insert "Text" form field.....	467
2.5.5.3	Insert "Edit" form field.....	468
2.5.5.4	Insert "Button" form field.....	469
2.5.5.5	Edit form fields subsequently.....	470
2.5.5.6	Delete form fields.....	470
2.5.6	User Archives Table Elements in Runtime.....	470
2.5.6.1	Table of User Archives Table Element.....	470
2.5.6.2	The User Archives Table Element Form.....	471
2.5.6.3	Toolbar of WinCC User Archives Table Element.....	473
2.5.6.4	Operating the Control using Dynamized Objects.....	477
2.5.6.5	List of properties for the User Archives Table Element.....	480
2.5.6.6	Overview of the dynamizable properties in the layout.....	483
3	COM Provider in the Layout Editor.....	485
3	Resources.....	485
3.1	COM Provider in the Layout Editor.....	485
3.2	Working with COM Server Objects.....	485
3.3	How to Output Data from a COM Server in a Report.....	486
3.4	Example of an integration of a COM server.....	487
3.5	Details of the COM Interface for Reporting.....	488
4	OPC - Open Connectivity.....	493
4	Resources.....	493
4.1	OPC - Open Connectivity.....	493
4.2	Functionality of OPC.....	493
4.3	OPC Specifications.....	494
4.4	Compatibility.....	495
4.5	Using OPC in WinCC.....	496
4.6	WinCC OPC XML DA Server.....	498
4.6.1	Functionality of WinCC OPC XML DA server.....	498
4.6.2	Installation.....	500
4.6.2.1	Installation.....	500
4.6.2.2	Installing the Internet Information Service (IIS).....	500
4.6.2.3	Installing Microsoft .NET Framework.....	501
4.6.2.4	Installing WinCC OPC XML DA Server.....	502
4.7	WinCC OPC DA server.....	502
4.7.1	Functionality of the WinCC OPC DA Server.....	502
4.7.2	Using Multiple OPC DA Servers.....	503
4.7.3	Querying the OPC DA Server Name.....	504
4.7.4	Examples of OPC DA Connections.....	506
4.7.4.1	WinCC - WinCC Connection.....	506
4.7.4.2	WinCC - SIMATIC NET FMS OPC Server Connection.....	510
4.7.4.3	WinCC - SIMATIC NET S7-OPC Server Connection.....	512
4.7.4.4	WinCC - Microsoft Excel Connection.....	518
4.8	WinCC OPC HDA server.....	524

4.8.1	Functionality of the WinCC OPC HDA server.....	524
4.8.2	Data Structure of a WinCC OPC HDA Server.....	525
4.8.2.1	Data Structure of a WinCC OPC HDA Server.....	525
4.8.2.2	Overview of the supported attributes.....	526
4.8.2.3	Overview of the supported assemblies.....	527
4.8.2.4	Overview of the supported functions.....	528
4.8.2.5	Time Format of a WinCC OPC HDA Server.....	528
4.8.3	Quality codes.....	530
4.8.4	Supported Write-Accesses.....	531
4.8.5	Example of an OPC HDA Connection.....	533
4.8.5.1	Example of an OPC HDA Connection.....	533
4.8.5.2	HDA server browser.....	534
4.8.5.3	How to Configure Access to a WinCC Archive Tag Using the HDA Server Browser.....	536
4.8.5.4	Reading Values of WinCC Archive Tags.....	537
4.8.6	Special features of the OPC HDA server in WinCC for acyclic logging.....	539
4.9	WinCC OPC A&E Server.....	541
4.9.1	Functionality of the WinCC OPC A&E server.....	541
4.9.2	Mapping of the WinCC Message System on OPC A&E.....	543
4.9.2.1	Mapping of the WinCC Message System on OPC A&E.....	543
4.9.2.2	Mapping the WinCC message classes and message types.....	544
4.9.2.3	Mapping the WinCC message priority.....	545
4.9.2.4	Attributes of the WinCC Message System.....	545
4.9.2.5	Acknowledgement theory.....	547
4.9.3	Quality Codes for OPC A&E.....	549
4.9.4	Example of an OPC A&E Connection.....	549
4.9.4.1	Example of an OPC A&E Connection.....	549
4.9.4.2	How to Configure Access to the WinCC Message System.....	550
4.9.5	OPC A&E server with hierarchical access.....	552
4.9.5.1	Functionality of the OPC A&E server.....	552
4.9.5.2	OPC A&E Server of WinCC V6.2 SP2 or higher.....	553
4.9.5.3	Mapping the WinCC Message System on OPC A&E.....	556
4.9.5.4	Quality Codes for OPC A&E.....	562
4.9.6	Reading archived messages.....	563
4.9.6.1	Accessing archived events.....	563
4.9.6.2	Syntax for accessing archived messages using OPC.....	564
4.9.6.3	Read methods for archived messages.....	565
4.9.6.4	Identifying archived messages.....	567
4.10	Commissioning.....	568
4.10.1	OPC Commissioning.....	568
4.10.2	Configuring Windows.....	568
4.10.2.1	This is how you configure Windows accounts for the use of WinCC OPC.....	568
4.10.2.2	How to adapt the Windows firewall settings.....	569
4.10.3	XML.....	570
4.10.3.1	Commissioning - OPC XML.....	570
4.10.3.2	Defining the Security Settings with IIS.....	570
4.10.3.3	How to set the correct version of ASP.NET.....	572
4.10.3.4	How to Test the Installation.....	572
4.10.4	Trace.....	574
Index		575

SmartTools

1 Resources

1.1 Overview

Contents

Smart Tools is a collection of programs which are useful when working with WinCC. It contains the following programs and files:

- Tag Simulator
- Tag Export/Import
- Dynamic Wizard Editor
- Documentation Viewer
- WinCC CrossReferenceAssistant
- OnlOff and OnlOn
- Communication Configurator
- WinCC Configuration Tool
- WinCC Archive ConfigurationTool

Note

Smart Tools are supplementary tools. Bear in mind that they may have an impact on how WinCC works, such as on its runtime behavior and memory requirements.

As far as user friendliness and functionality are concerned, the same criteria as for the WinCC Basis software do not necessarily apply.

See also

WinCC Communication Configurator (Page 111)

Tag Export/Import (Page 10)

The Tag simulator (Page 18)

Dynamic Wizard Editor (Page 23)

WinCC Documentation Viewer (Page 98)

WinCC CrossReferenceAssistant (Page 101)

1.2 Tag Export/Import

1.2.1 Tag Export/Import

Short description

The program exports all connections, all data structures and all tags from an open project into corresponding ASCII files. These can then be imported into a second project. The ASCII format allows processing the data with a spreadsheet program before reimporting them.

1.2.2 Installation of Export/Import Tags

Introduction

Tag Export/Import can be installed in two different ways:

This program can only be used in a meaningful manner, if WinCC has been installed as well.

Requirements

In WinCC multi-user projects the Smart Tool "Tag Export/Import" cannot be used for a client without its own project.

Procedure

1. During WinCC setup, select "WinCC V7.0 complete" from the "Programs" dialog.
WinCC is installed together with SmartTools, the WinCC ConfigurationTool and WinCC Archive ConfigurationTool.

The Tag Export/Import application is started by selecting "SIMATIC" > "WinCC" > "Tools".

Alternative Procedure

You can also install the Tag Export/Import application from the WinCC DVD.

1. Switch to the WinCC DVD directory "InstData\WinCC\setup\Products\SC_SMARTTOOLS".
2. Double-click setup.exe.
3. From the "Components" dialog select "VarExpImp".
4. Click "Continue". Follow the instructions in the dialog boxes.

1.2.3 Operation

EXPORT

1. First start WinCC and open the project from which to export tags. Start "VAR_EXIM.EXE"
2. Select path and name of the file to which to export. Initially only the name of the file without extension is needed.
3. Select the "Export" mode.
4. Press "Execute". Confirm the information in the message box.
5. Wait until "End Export/Import" is displayed in the status bar.
6. You can view the created files using the respective buttons "tag" (tags), "con" (connections), "dex" (structures) and "diag" (logbook).



Empty groups are not exported.

The underscore (_) is reserved for naming. This is why file names must not contain underscores.

IMPORT

1. First start WinCC and open the project into which to import tags.
2. All channel drivers into which the connections are to be imported must be available in the project. If required, add missing drivers to the project.
3. Start "VAR_EXIM.EXE"
4. Select path and name of the file from which to import. Initially only the name of the file without extension is needed. When using the selection dialog, click on one of the three exported files.
5. Select the "Import" or "ImportOverwrite" mode. In the "ImportOverwrite" mode tags already existing in the target project are overwritten with imported tags bearing the same name. In the "Import" mode a message is written into the logbook file instead and the tag in the target project remains unchanged.
6. Press "Execute". Confirm the information in the message box.
7. Wait until "End Export/Import" is displayed in the status bar.
8. View the created data in the WinCC tag management.



In both modes import is not possible with WinCC Runtime enabled.

The following sections describe the technical details of the Tag Export/Import. However, these information are not necessarily required for the standard case of a target computer with the same system configuration as the WinCC system used for exporting. Knowledge of the WinCC variable structure is only required if you want to create new tags or modify existing tags via the ASCII files.

1.2.4 The Tag Export/Import tool

The "Tag Export/Import" tool is an independent application based on the WinCC API. The tool may be used for exporting all WinCC tags of a project into ASCII files as well as for importing tags, e.g. into a second project. In doing so, three files are created.

- `[name]_cex.csv` for the logical connections
- `[name]_dex.csv` for the structural descriptions
- `[name]_vex.csv` for the tag descriptions

In the files, a header is generated informing about the data created. During import the three files are automatically read in.

The file `[name]_cex.csv` is imported first, as the tags can only be created if the associated connection already exists.

This is followed by the data structures defined in the file `[name]_dex.csv`. These are user-defined data types which have to be known before creating a tag of this type.

Afterwards the tag definitions from the file `[name]_vex.csv` are read in.

Tag groups cannot be created independently from a tag. If a group does not exist, it is automatically created together with the tag. This is why no group file is generated when exporting. If groups are defined in your project which contain no tag, these groups are not exported.

When creating a tag, the addressing defining the physical location of the tag in the automation system is configured, among others. This address depends on the connection channel and the connected automation system. When configuring via WinCC Explorer the user is presented a channel dependent input screen. When editing an import file of "Tag Export/Import" these characteristics have to be taken into account.

In any case, exported tags are only to be imported in a second WinCC project if this system has the same configuration as the system on which the data were exported. When data are imported into a project, possibly already existing tags must be known. If required, the addresses have to be corrected manually.

On the other hand, the import of tags from systems with another channel configuration may be possible in some cases. The success of such a procedure depends on channel and AS!

The exported structure types, however, are hardware-independent. All channel or connection specific parameters such as Connection, GroupName, address information are not considered. These parameters are only defined when creating a structure tag.



Points to note about exporting the address string in szSpecific

The address string is checked by the channel specific software when creating the tag. These channel DLLs expect a certain syntax which must not undergo country-specific changes. This is why the address strings are placed in additional " " parantheses when exporting, which are later removed during import. This is to make sure tools like Excel do not change the address information (list separator problem).

1.2.5 Connections

A connection can only be imported if a corresponding channel driver has been configured. In addition, the parameters in the file `[name]_cex.csv` must be suited to the configured channel driver.

Should the target computer use a different channel than the system from which the exported data originate, this is the first change which has to be made to the exported data.

A simple procedure to determine the required connection data:

Configure all connections on the target computer and start the export process. This allows you to get the parameters of your target computer from the file `[name]_cex.csv`.



Example of a logical connection:

#Conname	Unit	Common	Specific	Flag
VerbAS1	Industrial Ethernet	General	""	0
VerbAS2	Profibus FMS	General	""	0

The second line shows the logical connection of an Industrial Ethernet driver from the Simatic S7 Protocol Suite.

The third line contains the logical connection of a Profibus FMS driver.

Under #Conname the name of the logical connection is given. In the exported file you find this logical connection name in the tag data. The logical connection is used for the communication with the automation system to access the external tags in the process.

1.2.6 File structures

Structure of the file `[name]_dex.csv`.

#Structure Name	Type ID	Creator ID	Project Path	
Control_1	1001	2500	C:\Testdaten\Proj.mcp	
#Varname	C.Vartype	C.CreatorID	C.VarLength	etc.
NewTag1	...	...	...	...
NewTag2	...	...	...	...
NewTag3	...	...	...	...
etc.	...	...	...	...

#:	Comment character
#Structure Name:	The next line contains the name of the file structure with structure-specific parameters. The indication of the project path only serves to document from which project the data were exported. The data are imported automatically into the currently opened project.
#Varname:	The next lines contain the elements of the file structure until a new line #Structure Name is detected or no further lines are defined. One line contains all parameters required for defining a tag.



Remark for WinCC API users:

The column heading of the first line shows the names of the parameters in the corresponding file structures of the API calls. The data may then be clearly interpreted in an Excel table.

If a name is given with a letter followed by a point this is to facilitate the assignment to the API calls.



Example:

C.nnnnnn	contained in Common substructure
P.nnnnnn	contained in Protocol substructure
L.nnnnnn	contained in Limits substructure
S.nnnnnn	contained in Scaling substructure

1.2.7 Tag

Tags and connections consist of a general part and a specific part. The specific part is always provided by the channel DLL. Even though this part may be missing during configuration (it must be defined prior to activation), all objects whose specific part is missing will be ignored during import. The missing parts will be replaced by '*' during export.

Importing tags of a user-defined type.

A predefined WinCC tag is identified by a data type with a value between 1 and 18.

A user-defined structure type receives as data type the TypID which was assigned during creation of the data structure by the data manager. This TypID is greater than 1000.

A structure type is determined by its name and its data type.

The name of the data structure is identical on the computer from which it was exported and the computer on which it was imported. The TypID, however, does not have to or will not be identical.

In order to create a tag of the type structure type, an assignment must be made from TypID to structure name.



Example:

The structure types are exported to the file *[name]_dex.csv*.

#Structure Name	Type ID	Creator ID	Project Path		
ExternStr1	1046	0	G:\Testdaten\...		
#Varname	C.Vartype	C.CreatorID	C.VarLength	C.VarProperty	...
Tag1	1	0	2	...	...
Tag2	2	0	1	...	...
Tag3	3	0	1	...	...
Tag4	4	0	2	...	...

The tags are exported to the file *[name]_vex.csv*.

#Varname	Conn	Group	Spec	Flag	CType
InstExStr1	VerbLp	GruLp	DB200,DBB10	0	1046
InstExStr1.Tag1	VerbLp	GruLp	DB200,D10.0	0	1
InstExStr1.Tag2	VerbLp	GruLp	DB200,DBB10	0	2
InstExStr1.Tag3	VerbLp	GruLp	DB200,DBB10	0	3
InstExStr1.Tag4	VerbLp	GruLp	DB200,DBB10	0	4

In the file *[name]_dex.csv* a structure type with name ExternStr1 and TypID 1046 is defined.

In the file *[name]_vex.csv* a structure tag of type ExternStr1 with name InstExStr1 is defined. The assignment that this represents a structure type ExternStr1 is made by the value in column CType, which contains TypID 1046 of this structure type.

In order to import a structure tag, the structure type must also be contained in the file *[name]_dex.csv*, and tags of this type in the file *[name]_vex.csv*.

Tag Import/Export remembers the Name/TypID assignment to quantify the structure tags even if the TypID is different on the target computer.

If you wish to import tags of the type "user-defined" without reading the data structure during import (e.g. no file *[name]_dex.csv*), the TypID of the target computer of this data structure must be manually edited in the csv file. In this case, the TypID will be determined by using the exported file *[name]_dex.csv* of the target computer, as described above.

The tag properties are displayed in column CPro as decimal value in the exported file *[name]_vex.csv*. The following tag properties exist:

Tag property	Decimal value	Hexadecimal value
Internal tag with project-wide updates	2	2
Internal tag with computer-specific updates	8194	2002
External tag	4	4

For example, if internal tags with computer-specific updates are to be exported from WinCC but internal tags with project-wide updates are to be imported in WinCC, the value of the tag property in column CPro may be changed in the export file from 8194 to 2 for the respective tags. Subsequently, the modified export file is saved and imported in WinCC.

1.2.8 Appendix

Tag fields

Varname: char szVarName[MAX_DM_VAR_NAME +1];

Conn: char szConnection[MAX_DM_CONNECTION_NAME +3];

Group: char szGroupName[MAX_DM_GROUP_NAME +1];

Spec: char szSpecific[MAX_DM_VAR_SPECIFIC +1];

Flag: DWORD dwFlags;

Common

Ctyp: DWORD dwVarType; // Variable type

CLen: DWORD dwVarLength; // Variable length

CPro: DWORD dwVarProperty; // Variable property internal/external

CFor: DWORD dwFormat; // Format conversion

Protocol:

P1 : BOOL bTopLimitErr; // error in top limit

P2 : BOOL bBottomLimitErr; // error in bottom limit

P3 : BOOL bTransformationErr; // transformation error

P4 : BOOL bWriteErr; // write error

P5 : BOOL bWriteErrApplication;

P6 : BOOL bWriteErrProcess;

Limits values

LF1: double dTopLimit;

LF2: double dBottomLimit;


```
LF3: double dStartValue;  
LF4: double dSubstituteValue;
```

Limits flags

```
L1 : BOOL bTopLimit; // use substitute value on top limit  
L2 : BOOL bBottomLimit; // use substitute value on bottom limit  
L3 : BOOL bStartValue; // use substitute value on start  
L4 : BOOL bConnectionErr; // use substitute value on error connection  
L5 : BOOL bTopLimitValid; // value top limit is valid  
L6 : BOOL bBottomLimitValid; // value bottom limit is valid  
L7 : BOOL bStartValueValid; // value start is valid  
L8 : BOOL bSubstValueValid; // value substitute is valid
```

Values for the field Ctyp

```
BIT 1  
SBYTE 2  
BYTE 3  
SWORD 4  
WORD 5  
SDWORD 6  
DWORD 7  
FLOAT 8  
DOUBLE 9  
TEXT_8 10  
TEXT_16 11  
RAW 12  
ARRAY 13  
STRUCT 14  
BITFIELD_8 15  
BITFIELD_16 16  
BITFIELD_32 17  
TEXTREF 18
```

Description of connection fields

Conname: char szConnection[MAX_DM_CONNECTION_NAME +3];

Unit: char szUnitName [MAX_DM_UNIT_NAME+1];

Common: char szCommon [MAX_DM_CON_COMMON +1]

Specific: char szSpecific [MAX_DM_CON_SPECIFIC +1] ;

1.3 Tag simulator

1.3.1 The Tag simulator

Short description

The **tag simulator** is used to simulate internal tags and process tags.

A typical area of application for the **tag simulator** is the testing of a configuration without connected process peripherals or with connected process peripherals but without a running process.

Without a connected process, only internal tags may be simulated.

In case of connected process peripherals, the process tags may be directly supplied with values by the **tag simulator**. This allows you to perform a function test of the HMI system using the original hardware.

The updating time for tag values is one second. Any changes take effect only upon activation of the functions or changes of the project folder.

A maximum number of 300 tags may be configured.

Another possible application for the **tag simulator** is the implementation of a project for demonstration purposes.

There is often no system connection for the presentation of the HMI system. In these cases, the simulation will control the internal tags.

A detailed description of the tag simulator is to be found in the respective online help.



WARNING

The tag simulator writes the process values into the connected automation system. This means that a possible reaction of connected process peripherals has to be taken into account.

1.3.2 Using the tag Simulator

A typical area of application for the tag simulator is the testing of a configuration without connected process peripherals or with connected process peripherals but without a running process.

Simulation of process tags without process peripherals

Without a connected process, only internal tags may be simulated. In order to simulate the process off-line, the following procedure is recommended:

1. Make a backup copy of your project by copying your project folder and renaming it to, for example, xxx_sim. Use this backup copy as your test object. Open WinCC with this copied project.
2. Use the functions "Cut" and "Paste" to add the tags to be simulated to the internal tags. Do **not** use "Copy" and "Paste", otherwise WinCC Explorer will automatically generate an extension to the tag name to ensure that tag names are unique in the project. The address information of tags declared as internal tags will thus be lost.
3. With the aid of the simulator, the tags can now be supplied with values.
4. Once the test phase is over, you can continue your work with the original project.

Simulation of process tags With connected process peripherals

In case of connected process peripherals, the process tags may be directly supplied with values by the tag simulator. This allows you to perform a function test of the HMI system using the original hardware, such as e.g.:

- Checking limit value levels, message output.
- Testing the continuity of alarms, warnings, error messages and checking status displays.
- Presetting, reading and modifying digital and analog inputs and outputs.
- Alarm simulation.

1.3.3 Functions of the simulator

Introduction

The simulation provides the configurator with six different functions. These functions allow to supply the configured objects with realistic values.

To test different cases, the simulation offers 6 functions. Each tag can be assigned one of these 6 functions.

Sine

As periodic non-linear function.

Oscillation

For simulating jumps of a reference variable.

Random numbers

The "Random numbers" function makes randomly created values available to the user.

Increment

An up counter beginning with the minimum value again after reaching the maximum value.

Decrement

A down counter beginning with the maximum value again after reaching the minimum value.

Slider

A slider allowing the user to set a fixed value.

1.3.4 Installing the simulator

The tag simulator can be installed in two different ways:

Procedure

1. During WinCC setup, select "WinCC V7.0 complete" from the "Programs" dialog.
WinCC is installed together with SmartTools, the WinCC ConfigurationTool and WinCC Archive ConfigurationTool.

Start the tag simulator by selecting "SIMATIC" > "WinCC" > "Tools".

Alternative Procedure

You can also install the tag simulator application from the WinCC DVD.

1. Switch to the WinCC DVD directory "InstData\WinCC\setup\Products\SC_SMARTTOOLS".
2. Double-click setup.exe.
3. Select the entry "WinCC Tag Simulator" in the "Components" dialog.
4. Click "Continue". Follow the instructions in the dialog boxes.

Starting the simulator

The simulator Simulation.exe can either be started via Windows Explorer, or you can enter it in the startup list of WinCC Explorers, whereby it will automatically start whenever a project is activated.

An activated project in WinCC Explorer is required for the proper operation of the simulator. If the simulator has been added to the startup list of the project, this prerequisite is satisfied automatically.

1.3.5 Adding/deleting tags

Adding new tags

Using the menu commands "Edit/New Tag", tags can be added to the simulation. For this purpose, the tag selection dialog of the WinCC Explorer will be called, in which the desired tags from the active project can be selected. If new tags are to be created, this can be performed in the tag selection dialog as well. By confirming the selection with the "OK" button, the previously selected tag is entered in the "Properties..." tab of the simulator. There you may specify how the tag value is to be changed.

For final acceptance into the simulator the tag tab control must be clicked before the next tag is added.

The configured tag simulation can be saved in a configuration file with the file extension "sim".

Removing tags

If a tag is to be removed from the simulator list, it must be selected and deleted by clicking the menu item "Edit/Delete Tag". The selected tag will then be deleted from the list of tags to be simulated without confirmation dialog.

1.3.6 Parameter assignment of the functions

The function parameters can be set individually for each tag.

Sine wave

For the sine function the value range can be set by means of the **Amplitude** parameter.

The zero point for the value range can be defined with **Offset**.

The period is set by means of the parameter **Period of oscillation** (set value * cycle time).

Oscillation

The parameter **Setpoint** is used to define the value kept after the transient reaction.

The parameter **Overshoot** specifies how much the values may deviate from the setpoint value if the attenuation is set to zero.

The parameter **Period of oscillation** defines the time interval. After the time interval has elapsed, the oscillation begins anew.

Random numbers

The parameters **Lower limit** and **Upper limit** specify the interval for the random numbers.

Increment

The parameters **Start value** and **Stop value** specify the interval for the up counter.

Decrement

The parameters **Start value** and **Stop value** specify the interval for the down counter.

Slider

The parameters **Start value** and **Stop value** define the adjustment range of the slider.

1.3.7 Activating/deactivating tags

For a smooth transition from offline configuration to online configuration the tags can be individually activated and deactivated via an own check box.

If the tag is enabled, the simulation calculates the values and transfers them to the WinCC Explorer.

If the check box is not activated, the simulation does not transfer values to the WinCC Explorer.

1.3.8 Display of the tags

In the configuration phase, tag control is facilitated for the configurator by displaying the following information in the Tags tab:

- current WinCC project
- name of the tag
- assigned function
- status (active / inactive)
- present value

Upon selecting the name of a tag, the other parameters are entered in the "Properties" tag.

1.3.9 Loading/saving simulation data

The simulation data can be saved so that they are available when restarting the simulator. This is done by selecting the menu items "File/Save" or "File/Save as...".

A simulation configuration already saved is loaded by means of the menu commands "File/Open".

When starting the simulator the last used configuration related to a WinCC project is automatically loaded.

1.3.10 FAQ



Error with DM-API, DLL not found

Errors occurring when calling the simulator in connection with DLLs are due to a missing path instruction in the file AUTOEXE.BAT. Check this file for the following entry in the path instruction:

SET PATH = ...;<WinCC drive>:\<WinCC directory>\bin

e.g.: SET PATH=C:\SIEMENS\WINCC\BIN;

1.4 Dynamic Wizard Editor

1.4.1 Dynamic Wizard Editor

Short description

The **Dynamic Wizard Editor** is a tool for creating own Dynamic Wizards. Dynamic Wizards allow frequently reoccurring configuration processes to be automated.

1.4.2 Overview

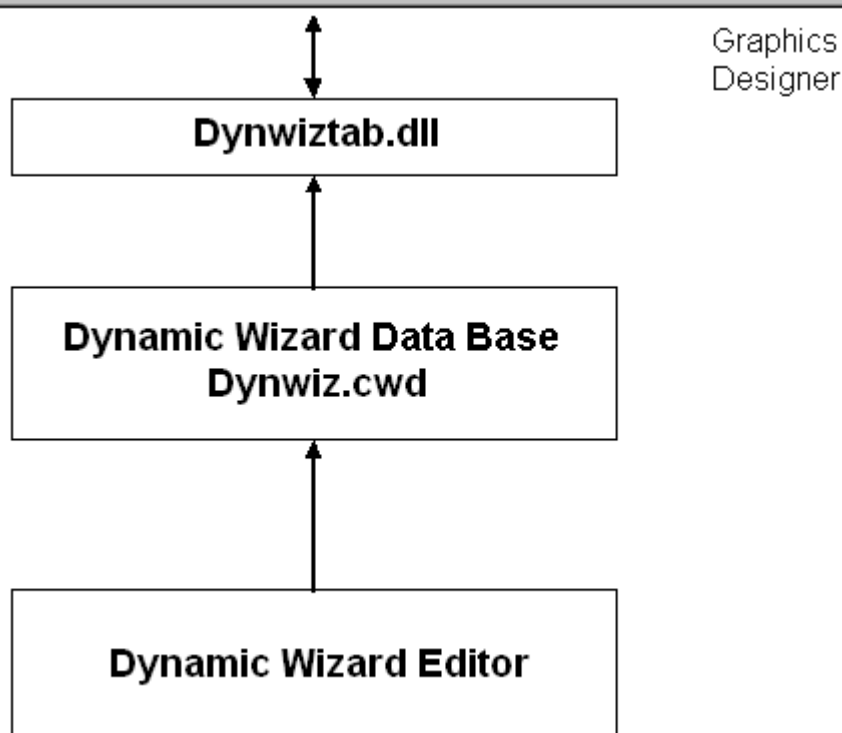
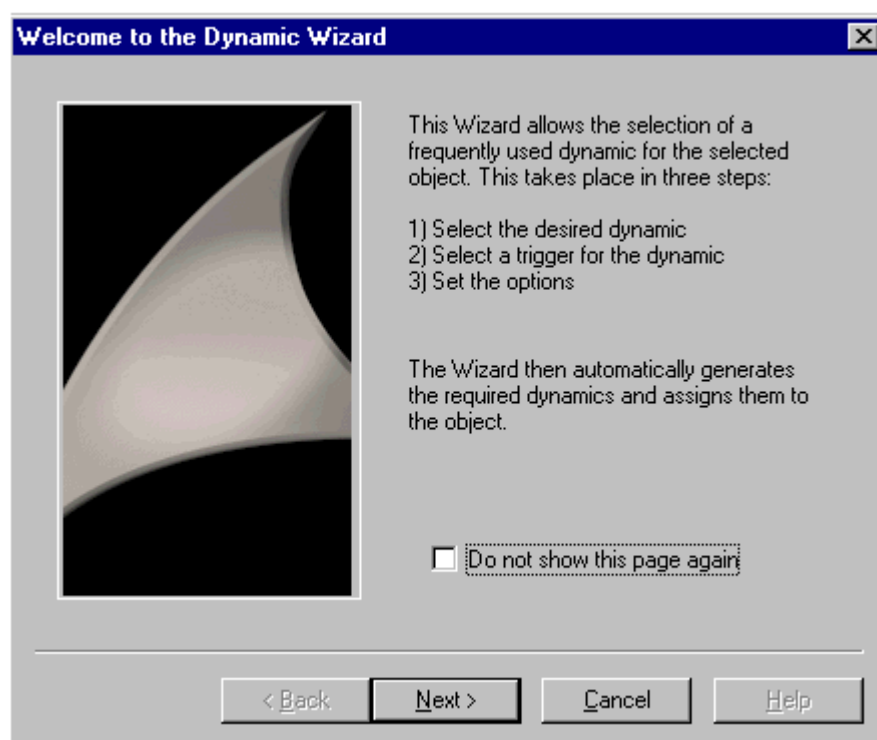
Introduction

The Dynamic Wizard brings additional functionality to Graphics Designer. It assists the user in handling frequently reoccurring configuration sequences. This reduces the configuration effort and the risk of configuration errors.

1.4 Dynamic Wizard Editor

The Dynamic Wizard consists of a variety of Dynamic Wizard functions. A large number of Dynamic Wizard functions are already supplied. And these can be further extended with functions that you create yourself.

An editor is included for creating your own Dynamic Wizard functions. This editor is the program dynwizedit.exe.



All of the Dynamic Wizard functions are stored in a data basis (...\\WinCC\\wscript\\Dynwiz.cwd) on the hard disk. The Dynamic Wizard has a standardized display and user interface for selecting and parameterizing the Dynamic Wizard functions. After a Dynamic Wizard function has been selected, Dynamic Wizard will load it into memory and start it.

Interface between Dynamic Wizard and Dynamic Wizard function

The link between Dynamic Wizard and the Dynamic Wizard function is made using a system interface in the Dynamic Wizard function. The structure of this interface is defined. This interface contains information, which Dynamic Wizard can evaluate.

Essential content of the interface:

Reference to the process function

The process function is the intended main function of the Dynamic Wizard. It contains the "service", which a Dynamic Wizard function provides for the user, e.g. the creation of an action on a graphic object.

The options list defines the parameters, which are required for the process function. It also defines how they are specified using the dialog's user interface.

The trigger list List defines the triggers to be linked to the generated object. It also defines how they are specified using the dialog's user interface.

See also

Trigger list (Page 40)

Options list (Page 37)

1.4.3 Installation of the Dynamic Wizard Editor

The Dynamic Wizard Editor can be installed in two different ways:

Procedure

1. During WinCC setup, select "WinCC V7.0 complete" from the "Programs" dialog.
WinCC is installed together with SmartTools, the WinCC ConfigurationTool and WinCC Archive ConfigurationTool.

Start the Dynamic Wizard Editor by selecting "SIMATIC" > "WinCC" > "Tools".

Alternative procedure

You can also install the Dynamic Wizard Editor from the WinCC DVD.

1. Switch to the WinCC DVD directory "InstData\WinCC\setup\Products\SC_SMARTTOOLS".
2. Double-click setup.exe.
3. Select "Dynamic Wizard Editor" in the "Components" dialog.
4. Click "Next". Follow the instructions in the dialog boxes.

1.4.4 Structure

1.4.4.1 Structure

The Dynamic Wizard Editor comprises the following elements:

Menu bar

The menu bar contains the functions of the Dynamic Wizard Editor. The menu bar is always visible.

Toolbar

The tool bar can be made visible as needed and dragged to any place on the screen with the mouse.

Editor window

The Editor window is only visible if a Dynamic Wizard function has been opened to be edited or a new one is created. Each function will be opened in its own editing window. Several editing windows can be open at the same time.

Output window

The output window can be made visible as required. It shows the result of the functions "Create CWD", "Read Wizard Script" and "Compile Script".

Status bar

The status bar can be made visible as required. It informs about the keyboard setting and the position of the cursor in the editing window.

Dynamic Wizard

With the Dynamic Wizard, you can dynamize an object using C actions. When executing wizards, preconfigured C actions and trigger events are defined and transferred to the object properties.

See also

Output window (Page 31)

Editor window (Page 29)

Toolbar (Page 28)

1.4.4.2 Toolbar

Introduction

The toolbar allows you to execute actions more rapidly. You do not have to make several selections through the menus until you reach the required function.



Icons

Icon	Description
	Creates a new Dynamic Wizard function.
	Opens an existing Dynamic Wizard function (*.wnf).
	Saves the Dynamic Wizard function.
	Cuts the selected text and copies it to the clipboard.
	Copies the selected text to the clipboard.
	Pastes the contents of the clipboard at the location of the cursor.
	Prints the contents of the current editing window.
	Displays additional information on the Dynamic Wizard Editor.
	Creates the Dynamic Wizard data (CWD). This function reads in all existing wizard scripts for the currently set language and prepares them for processing in the Dynamic Wizard. The file created is to be found in the WinCC installation path (Installation path\wscripts\dynwiz.cwd).
	Reads in the wizard scripts and makes them available to the Dynamic Wizard.
	Sets the language for which the wizard script is configured. The languages known to WinCC are made available here, regardless of the installed languages. Changing the Wizard language does not affect the overall system and the configuration interface.
	Changes the object. The Dynamic Wizard, which is also available in the Editor for test purposes, depends on the different properties of an object in Graphics Designer. This function allows switching to an existing object in an existing picture so that a new or existing wizard script can be tested in the Editor. The newly set object causes the Dynamic Wizard to be adapted in a way that only wizard scripts suitable for this object are displayed.

Icon	Description
	Shows all Dynamic Wizard scripts for the selected language. Moreover, wizard scripts existing in the dialog can be deleted from the list.
	Opens the Help Editor.
	Compiles the script.

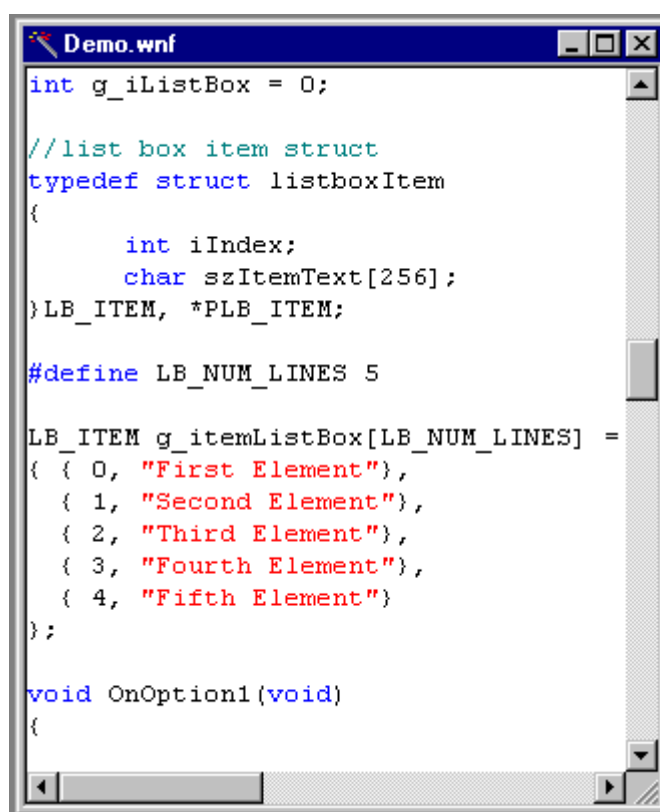
See also

Help Editor (Page 30)

1.4.4.3 Editor window

Introduction

The editor window serves to create and edit Dynamic Wizard functions.



Color coding

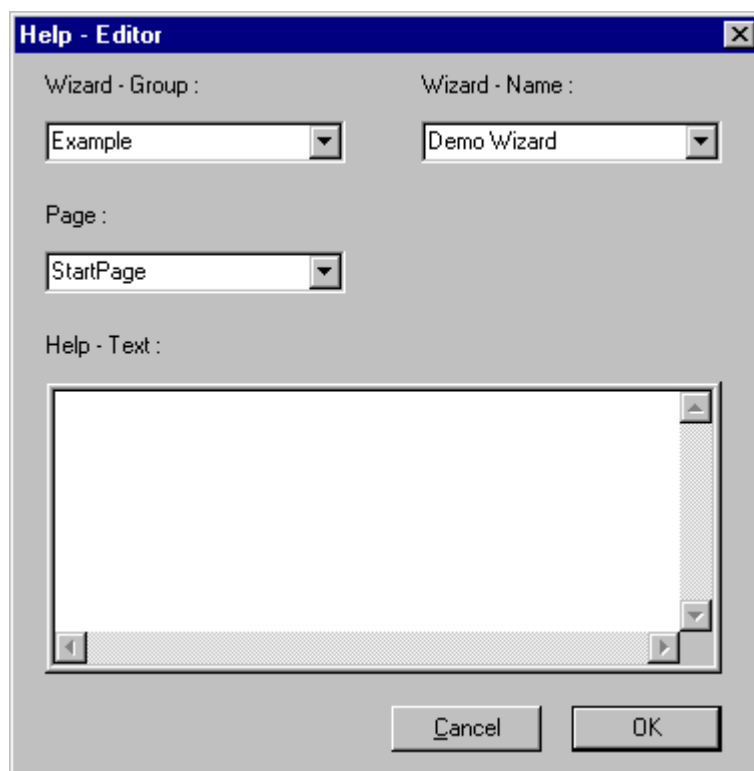
The C code is displayed with the following colors:

Color	Significance	Example
blue	Keywords	#define, void
green	Comments	// this is a comment
red	Strings	"First Element"
black	other C code	OnOption1

1.4.4.4 Help Editor

Introduction

Within this dialog a help text can be entered for each page created via the wizard script. Only help texts for Dynamic Wizards already created can be entered.



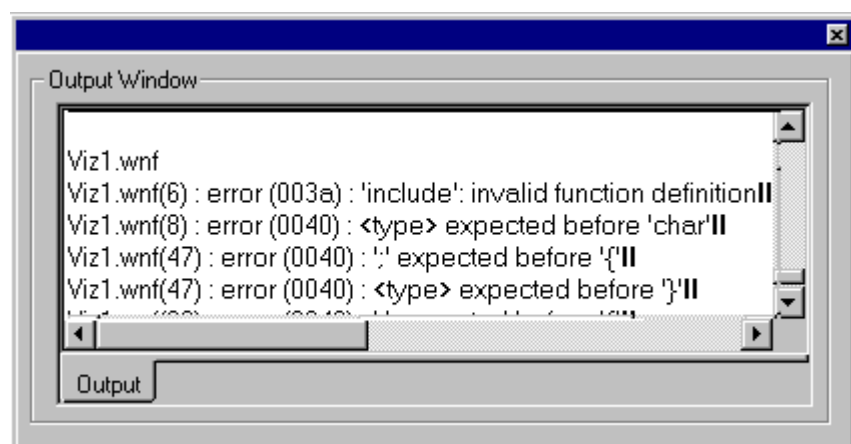
Elements of the Help Editor

Element	Description
Wizard group	This field is used to specify the group (=tab) containing the Dynamic Wizard.
Wizard name	This field is used to select the Dynamic Wizard for which a help text is to be created.
Page	This field is used to select the dialog page for which the help text is to be created.
Help text	In this field the help text will be entered.

1.4.4.5 Output window

Introduction

The output window shows the result of the functions "Create CWD", "Read Wizard Script" and "Compile Script".



The output window helps to find errors in the scripts.

If there is an error in the script, the following message appears:

...\\WinCC\\wscripts\\wscripts.deu\\DemoWiz1.wnf(6):error(003a): 'include': invalid function definition

	Description
...\\WinCC\\wscripts\\wscripts.deu\\	Directory containing the wnf file.
DemoWiz1.wnf(6)	File name and line number where the error occurs
error(003a): 'include': invalid function definition	Error number and error description.

1.4.5 Structure of a Dynamic Wizard function

1.4.5.1 Structure of a Dynamic Wizard function

Introduction

A Dynamic Wizard function must have a certain specified structure. It corresponds to the required components.

1. Integrating header files and DLLs
2. Language-dependent definitions
3. Wizard flags
4. Property list
5. System interface
6. Global variables
7. Options list
8. Trigger list
9. Display of parameter assignment

See also

Display of parameter assignment (Page 42)
Trigger list (Page 40)
Options list (Page 37)
Global variables (Page 37)
System interface (Page 36)
Property list (Page 35)
Wizard flags (Page 34)
Language-dependent definitions (Page 33)
Integrating header files and DLLs (Page 33)

1.4.5.2 Dynamic Wizard dialog

Introduction

Each Dynamic Wizard option has its specific functionality. However, due to the predefined function structure all functions have a similar sequence and a similar dialog interface. The Dynamic Wizard dialog consists of several dialog pages.

- "Welcome to the Dynamic Wizard" dialog
- "Select trigger" dialog

- "Set options" dialog
- "Finished !" dialog

1.4.5.3 Integrating header files and DLLs

Introduction

A header file contains declarations of constants, data types, tags and functions.

The header files are integrated into the function by means of an `#include` instruction. The most important file to be integrated is the `dynamic.h` file, in which, among others, the functions for designing the Dynamic Wizard interface are declared.

```
// *****
//**      Integration of Header-Files      **
// *****
#include "dynamic.h"
```

DLL files (Dynamic Link Library) are executable routines which can be loaded by a program if this program requires so.

To be able to use DLL files, they are integrated into the function by means of a `#pragma` instruction.

```
// *****
//**      Integration of Dlls      **
// *****
#pragma code ("pdllcsapi.dll")
#include "pdllcsapi.h"
#pragma code {}
```

In the Dynamic Wizard Editor the following paths are defined:

WinCC header files: ...\\WinCC\\aplib\\

WinCC DLLs: ...\\WinCC\\bin\\

Should the files be stored in another directory, the complete path is to be specified in the `#include-` and `#pragma` instructions.

1.4.5.4 Language-dependent definitions

Introduction

The Dynamic Wizard standard functions exist in the three languages German, English, and French. When the language is changed in the WinCC Explorer, the respective language version is also selected for the Dynamic Wizard functions.

Within the paths

..\\WinCC\\wscripts\\wscripts.deu

..\WinCC\wscripts\wscripts.enu

..\WinCC\wscripts\wscripts.fra

a WNF file must exist for each wizard function.

Upon creation, all language-dependent definitions should be arranged in this section. This facilitates the creation of other language versions.

```
// *****
//                               Language-Dependent Definitions                               //
// *****
//                               German                                                       //
//-----
include    "defdeu.h"

char* DynWizGroupName      = "WinCC C-Kurs";
char* DynWizDynamicName    = "Motor dynamisieren";
char* DynWizToDoOption1    = "Wählen Sie die gewünschte Strukturvariable:";
//-----
//                               Englisch                                                       //
//-----
#include    "defenu.h"

char* DynWizGroupName      = "WinCC C-Course";
char* DynWizDynamicName    = "Make a Motor Dynamic";
char* DynWizToDoOption1    = "Select the desired Structure Tag:";
//-----
//                               French                                                       //
//-----
#include    "deffra.h"

char* DynWizGroupName      = "Cours de C WinCC";
char* DynWizDynamicName    = "Dynamiser moteur";
char* DynWizToDoOption1    = "Sélectionnez la variable de structure:";
```

1.4.5.5 Wizard flags

Introduction

These Flags are used to define configuration type for which the Dynamic Wizard function applies.

```
WIZARD_FLAGS(WIZARD_FLAG_OCX | WIZARD_FLAG_ALL_PROJECT_TYPES)

BEGIN_PROPERTY_SCHEME
END_PROPERTY_SCHEME
```

Flags

FLAG	
WIZARD_FLAG_OCX	For all OCX files
WIZARD_FLAG_ALL_PROJECT_TYPES	For all projects
WIZARD_FLAG_SINGLEUSER_PROJECT	For single-user projects only
WIZARD_FLAG_MULTICLIENT_PROJECT	For client projects
WIZARD_FLAG_MULTIUSER_PROJECT	For clients without project data only

1.4.5.6 Property list

Introduction

The property list defines the object types for which a Dynamic Wizard function can be used. This is done by specifying a list of object properties. If an object has at least one of the listed properties, the Dynamic Wizard function can be applied to it.

```
//*****
//** Objektauswahl mittels Objekteigenschaften **
//*****

BEGIN_PROPERTY_SCHEME
  {"BackColor", VT_I4},
END_PROPERTY_SCHEME
```

Each entry in the property list consists of two parameters:

- Property name, e.g., Backcolor in the English version.
- WinCC Data type

If an empty property list is used, the Dynamic Wizard function can be applied to all object types. In any case, there must be a property list, even if it is empty.

1.4.5.7 System interface

Introduction

The system interface is used to define the properties of the new Dynamic Wizard function.

```

BEGIN_DYNAMICS
{
    DynWizGroupName,           // 1. Parameter
    DynWizDynamicName,         // 2. Parameter
    NULL,                      // 3. Parameter
    "logo16.bmp",              // 4. Parameter
    DynWizHelpText,            // 5. Parameter
    {                          // 6. Parameter
        // "OnOption1",
        // "OnOption2",
        NULL
    },
    "OnGenerate",              // 7. Parameter
    "OnShowGenerateInfo",      // 8. Parameter
    {                          // 9. Parameter
        // PREDEFINED_MACRO,
        // {DynWizTrigger1Text, OnTrigger1},
        {NULL, NULL}
    },
},
END_DYNAMICS

```

Parameter description

1. The first parameter defines the tab on which the Dynamic Wizard function is to appear.
2. The second parameter defines the name under which the Dynamic Wizard function is to appear.
3. The third parameter is always NULL.
4. The fourth parameter designates the name of the icon to be used for the Dynamic Wizard function.
5. The fifth parameter is a help text with a more detailed description of the functionality of the Dynamic Wizard function.
6. The sixth parameter is a list with the names of the functions created for the individual option pages. This list is to be concluded by a NULL entry. A maximum of five option pages can be created. Additional information on this topic may be found under "Options list".
7. The seventh parameter is the name of the process function called after clicking on the "Complete" button. The process function is the intended main function of the Dynamic Wizard. It contains the "service", which a Dynamic Wizard function provides for the user, e.g. the creation of an action on a graphic object.

8. The eighth parameter is the name of the function which summarizes the setting on the option pages and displays them for the user before the latter clicks on the Complete button. Additional information on this topic may be found under "Display of parameter assignment".
9. The ninth parameter is a list of the triggers to be displayed on the trigger page. For the most common applications, macros are available to fill this trigger list. Additional information on this topic may be found under "Trigger list".

1.4.5.8 Global variables

Introduction

For each parameter to be set on the option pages a global variable must be defined. This is to make sure that the set parameters are known in all created functions and can be utilized.

Data transfer between system functions is only possible by means of global variables. This is always required when trigger and/or option parameters have to be transferred to the process function.

```
// *****  
//      Definition of Global Tags  
// *****  
  
char g_Demo_Typ ="Demo"
```

1.4.5.9 Options list

Introduction

Options are parameters needed for the functionality of the Dynamic Wizard function. Options do not require a trigger.

Options are defined in the options list of the system interface. For each option, the options list contains the name of the assigned options function, e.g. "OnOption1".

```

BEGIN_DYNAMICS
{
DynWizGroupName,
DynWizDynamicName,
NULL,
"logo16.bmp",
DynWizHelpText,
// *****
//           Optionenliste
// *****
{
  "OnOption1",
  "OnOption2",
  NULL
},
  "OnGenerate",
  "OnShowGenerateInfo",
  { // Triggerliste
    { NULL, NULL }
  },
},
END_DYNAMICS

```

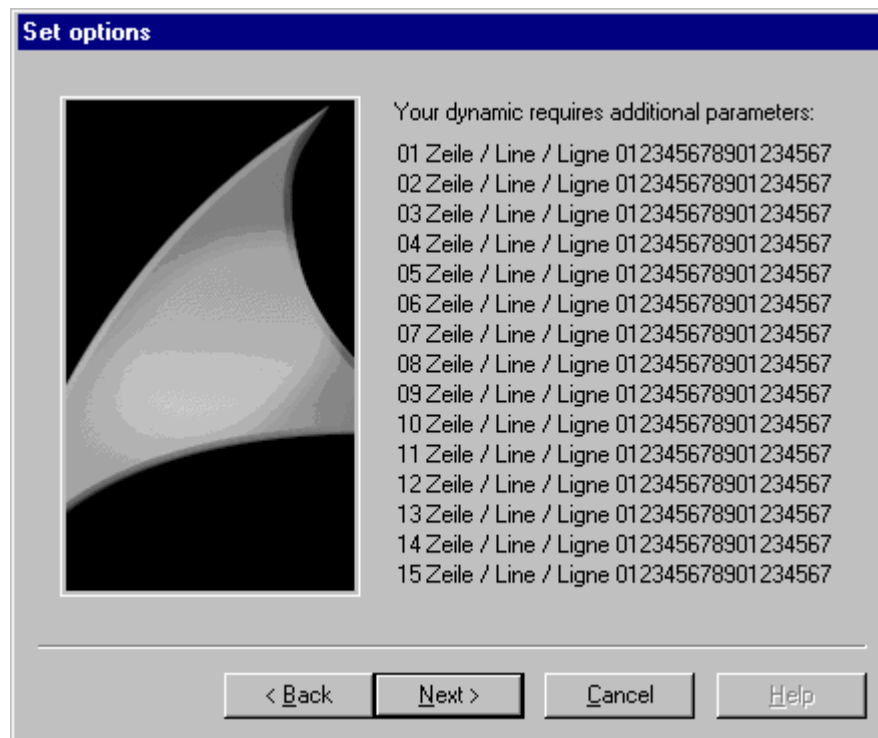
The options list is concluded by a NULL pointer. A maximum of five options may be defined in the list.

Option functions

The Dynamic Wizard calls the option functions successively, according to their sequence in the options list. For each option function the "Set options" dialog is displayed where the function programs its specific entry.

There are wizard system functions available for programming the entry. Additional information on this topic may be found under "Wizard system functions."

The "Set options" dialog makes available a defined area for arranging static texts, input fields and other input boxes.



In the "Set options" dialog, the area is completely filled with lines 1 to 15.

The corresponding option function is as follows:

```
//-----
//      Option-Funktion OnOption1
//-----

void OnOption1(void)
{
    CreateStatic(0, 0, "01 Zeile / Line / Ligne 012345678901234567");
    CreateStatic(0, 15, "02 Zeile / Line / Ligne 012345678901234567");
    CeateStatic(0, 30, "03 Zeile / Line / Ligne 012345678901234567");
    CeateStatic(0, 45, "04 Zeile / Line / Ligne 012345678901234567");
    CreateStatic(0, 60, "05 Zeile / Line / Ligne 012345678901234567");
    CreateStatic(0, 75, "06 Zeile / Line / Ligne 012345678901234567");
    CreateStatic(0, 90, "07 Zeile / Line / Ligne 012345678901234567");
    CreateStatic(0, 105, "08 Zeile / Line / Ligne 012345678901234567");
    CreateStatic(0, 120, "09 Zeile / Line / Ligne 012345678901234567");
    CreateStatic(0, 135, "10 Zeile / Line / Ligne 012345678901234567");
    CreateStatic(0, 150, "11 Zeile / Line / Ligne 012345678901234567");
    CreateStatic(0, 165, "12 Zeile / Line / Ligne 012345678901234567");
    CreateStatic(0, 180, "13 Zeile / Line / Ligne 012345678901234567");
    CreateStatic(0, 195, "14 Zeile / Line / Ligne 012345678901234567");
    CreateStatic(0, 210, "15 Zeile / Line / Ligne 012345678901234567");
}
```

1.4.5.10 Trigger list

Introduction

Triggers are only required in connection with actions on a graphic object.

Triggers are defined in the trigger list of the system interface. The trigger list contains an entry for each trigger.

```

BEGIN_DYNAMICS
{
  DynWizGroupName,
  DynWizDynamicName,
  NULL,
  "logo16.bmp",
  DynWizHelpText,
  "OnOption1",
  "OnOption2",
  NULL
},
"OnGenerate",
"OnShowGenerateInfo",
{
  //*****
  //          Trigger list
  //*****
  { "Mouse click" , "OnTriggerMC" },
  { "Pressing left mouse key" , "OnTriggerLMDown" },
  { "Releasing left mouse key" , "OnTriggerLMUp" },
  { "Pressing right mouse key" , "OnTriggerRMDown" },
  { "Releasing right mouse key" , "OnTriggerRMUp" },
  { NULL, NULL }
},
},
END_DYNAMICS

```

The entry consists of two parameters. The first parameter is the designation of the trigger which is shown in the interface, such as Click left mouse button. The second parameter gives the name of the assigned trigger function.

The trigger list is concluded by a NULL pointer pair. A maximum of 50 triggers may be defined in the list.

For the triggers used most frequently, predefined macros are available.

Macro	
JCR_TRIGGERS	Trigger events DECLARE_JCR_TRIGGERS Mouse click, Left mouse button, Right mouse button
JCR_ZYCL_TRIGGERS	cyclic triggers DECLARE_JCR_ZYKL_TRIGGERS Picture cycle, Window cycle, Upon change, 250 ms, 500 ms, 1 second, 2 seconds, 5 seconds, 10 seconds, 1 minute, 5 minutes, User cycle 1, User cycle 2, User cycle 3, User cycle 4, User cycle 5
JCR_ACTION_TRIGGERS	Action triggers DECLARE_JCR_ACTION_TRIGGERS

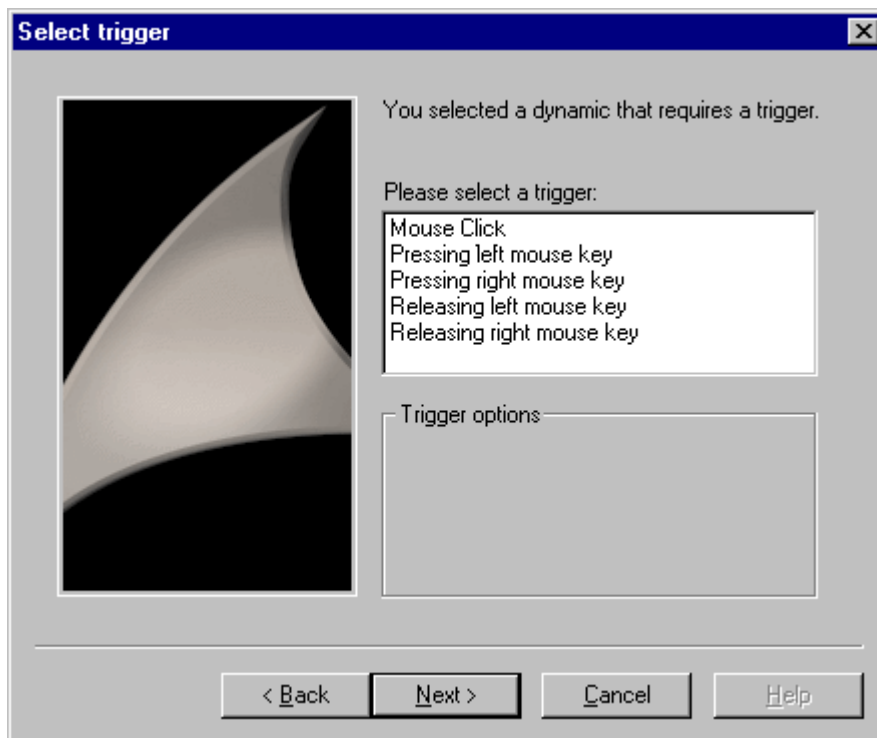
```

...
...
BEGIN_PROPERTY_SCHEME
END_PROPERTY_SCHEME
BEGIN_DYNAMICS
{
    "System Functions",
    "Exit WinCC Runtime",
    NULL,
    "logo16.bmp",
    "Exits WinCC Runtime and switches to \r\nthe DESIGN Mode.",
    { NULL, NULL, },
    "OnGenerate",
    "OnShowGenerateInfo",
    {
        JCR_TRIGGERS,
    },
}
END_DYNAMICS

DECLARE_JCR_TRIGGERS
...
...

```

The "Select trigger" dialog is created from the trigger list. All trigger designations are displayed in a list box for selection.



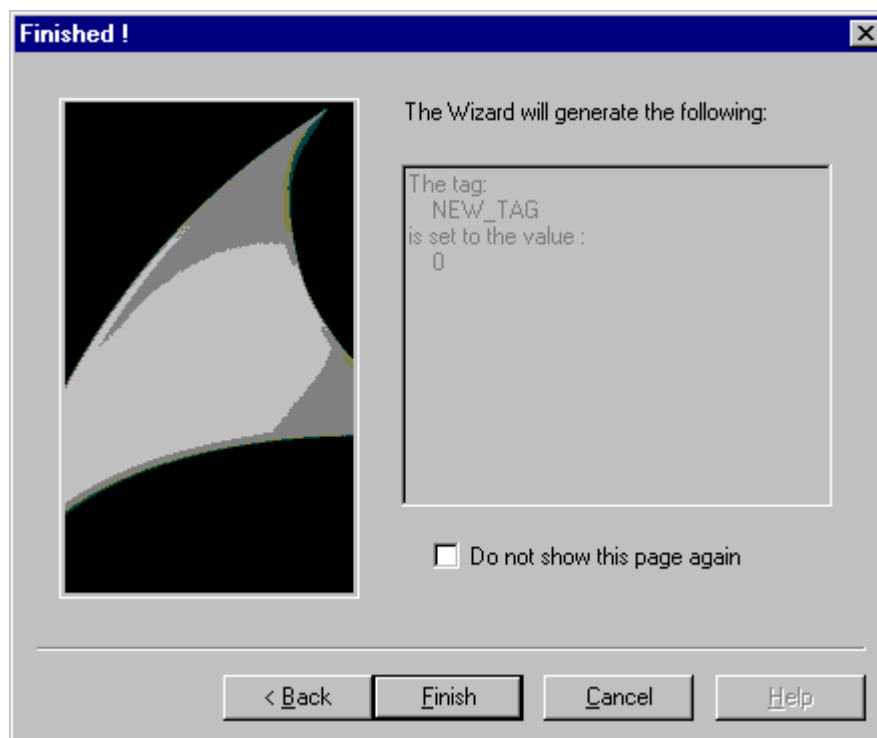
After selecting a trigger, the Dynamic Wizard calls the assigned trigger function.

1.4.5.11 Display of parameter assignment

Introduction

Trigger and option parameters can be displayed in the "Finished !" dialog. This allows the user to recheck the parameterization and change it, if required.

In the display field of the "Finished !" page, a text can be displayed using the Windows function SetWindowText The height of the display field is 12 lines.



1.4.5.12 Wizard Functions for parameter input

CreateStatic

Introduction

In the "Set options" dialog a static text is displayed for the x,y coordinates.

Syntax

HWND CreateStatic (int x, int y, char* "Text")

Parameters

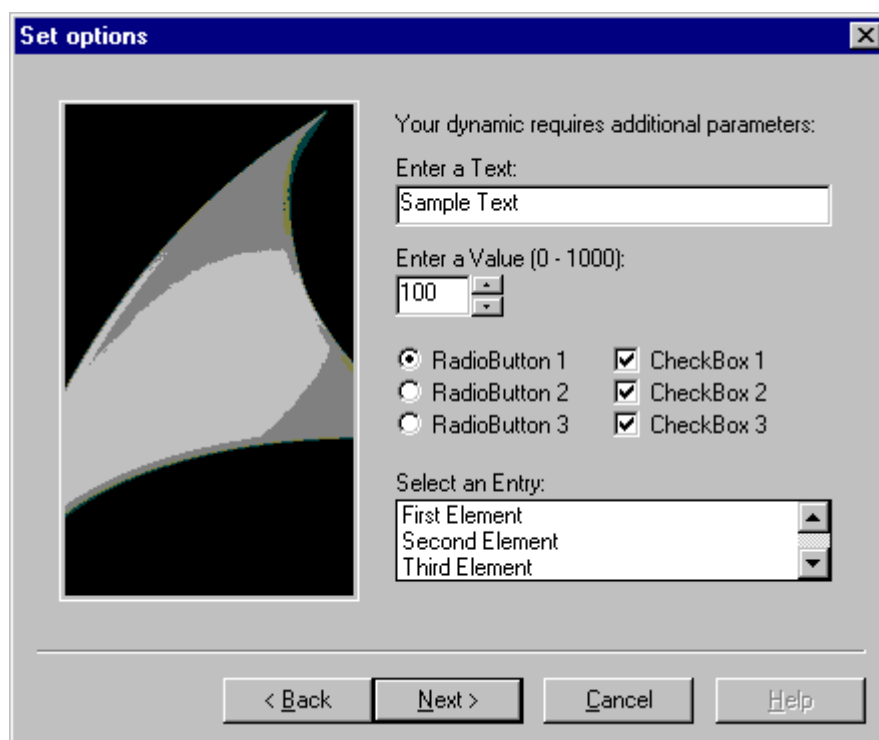
Parameters	Description
int x	Shows the value of the x coordinate.
int y	Shows the value of the y coordinate.
char* Text	Shows the displayed text.

Return value

	Return value
HWND	returns the object handle.

Example

The following excerpt from the file "Demo.wnf" shows the use of this function.



```
char* DynWizEditStatic = "Enter a text:";
...
..
void OnOption1(void)
{
    static BOOL bFirst = TRUE;
    HWND hWnd = NULL;
    .....
    if (bFirst == TRUE)
    {
        strcpy(g_szEdit, DynWizEdit);
```

```

bFirst = FALSE;
}
//Static text
CreateStatic(0,5,DynWizEditStatic);
.....
.....
}

```

CreateEdit

Introduction

In the "Set options" dialog an input field is displayed for the x,y coordinates. A text can be typed in this input field.

Syntax

HWND CreateEdit (int x, int y, char* pText)

Parameters

Parameters	Description
int x	Shows the value of the x coordinate.
int y	Shows the value of the y coordinate.
char* pText	Pointer to an input buffer. The input buffer may have a predefined value. This is displayed in the input field.

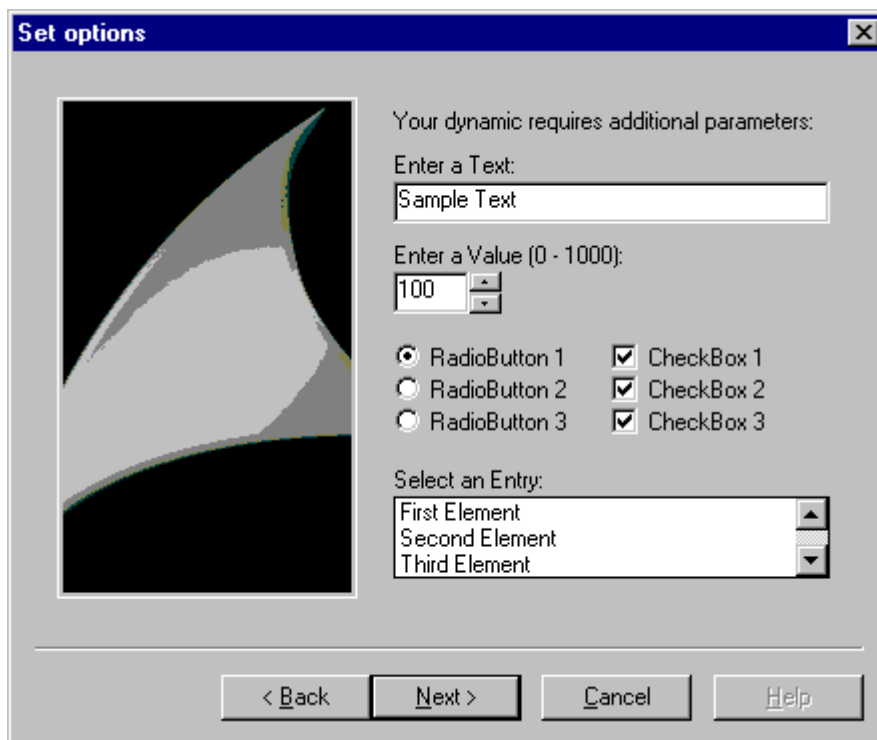
Return value

	Return value
HWND	Returns the object handle
pText	Input buffer contains the entered text.

Example

The following excerpt from the file "Demo.wnf" shows the use of this function.

An input field is displayed in the "Set options" dialog of the "Demo Wizard".



```
char* DynWizEditStatic = "Enter a text:";
char* DynWizEdit = "Sample text";
...
..
char g_szEdit[256];
void OnOption1(void)
{
    static BOOL bFirst = TRUE;
    HWND hWnd = NULL;
    .....
    if (bFirst == TRUE)
    {
        strcpy(g_szEdit, DynWizEdit);
        bFirst = FALSE;
    }
    //Static text for the input field
    CreateStatic(0, 5, DynWizEditStatic);
```

```
//Input field
hWnd = CreateEdit(0,20,g_szEdit)
GetWindowRect(GetParent(hWnd), &rect);
MoveWindow(hWnd,0,20,(rect.right-rect.left),21,TRUE);

.....
.....
}
```

CreateSpinEdit

Introduction

In the "Set options" dialog an input field with controls is displayed for the x,y coordinates. This input field is used to enter an integer value into an entry variable.

Syntax

HWND CreateSpinEdit (int x, int y, int* pValue, int Min, int Max, int Base)

Parameters

Parameters	Description
int x	Shows the value of the x coordinate.
int y	Shows the value of the y coordinate.
int* pValue	Pointer to an integer entry variable. The entry variable may have a predefined default value.
int Min	Lower limit for the input value
int Max	Upper limit for the input value
int Base	Input number format: 10 = decimal input 16 = hexadecimal input

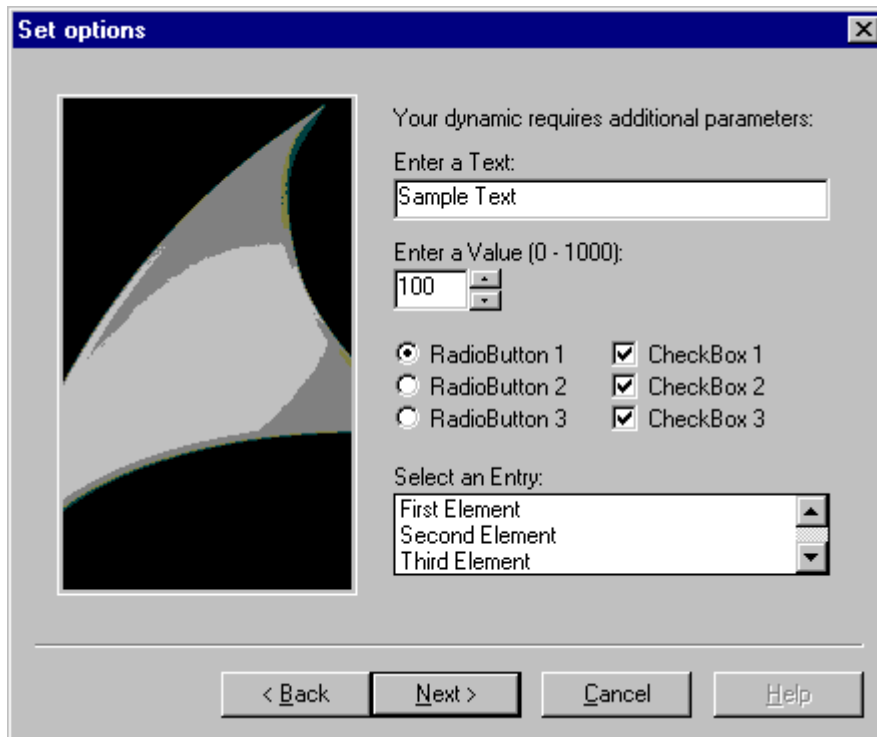
Return value

	Return value
HWND	Returns the object handle.
pValue	Entry variable contains the entered value.

Example

The following excerpt from the file "Demo.wnf" shows the use of this function.

An input field with controls is displayed in the "Set options" dialog of the "Demo Wizard". You can select a value between 0 and 1000 there.



```
char* DynWizSpinStatic= "Enter a value (0 - 1000):";
char* DynWizEdit = "Sample text";
...
...
char g_szEdit[256];
void OnOption1(void)
{
    static BOOL bFirst = TRUE;
    HWND hWnd = NULL;
    .....
    if (bFirst == TRUE)
    {
        strcpy(g_szEdit,DynWizEdit);
        bFirst = FALSE;
    }
    ...
}
```



```

...
//Static text for the input field with controls
CreateStatic(0,50,DynWizSpinStatic);
...
//Input field with controls
hWnd = CreateSpinEdit(0,65,&g_iSpinEdit,0,1000,10);
MoveWindow(hWnd,0,65,(rect.right-rect.left)/4,21,TRUE);

...
...
}

```

CreateListBox

Introduction

In the "Set options" dialog a selection field is displayed for the x,y coordinates. The selection field allows listing several entries. By clicking the mouse, one entry can be selected.

Syntax

HWND CreateListbox (int X, int Y, char* Headline, int NumLines, int* pSelect)

Parameters

Parameters	Description
int x	Shows the value of the x coordinate.
int y	Shows the value of the y coordinate.
char* Headline	Header of the selection field
int NumLines	Number of lines in the selection field. The following must be specified: NumLines = Number of lines + 1 (1 ≤ NumLines ≤ 16)
int* pSelect	Pointer to the result variable

Return value

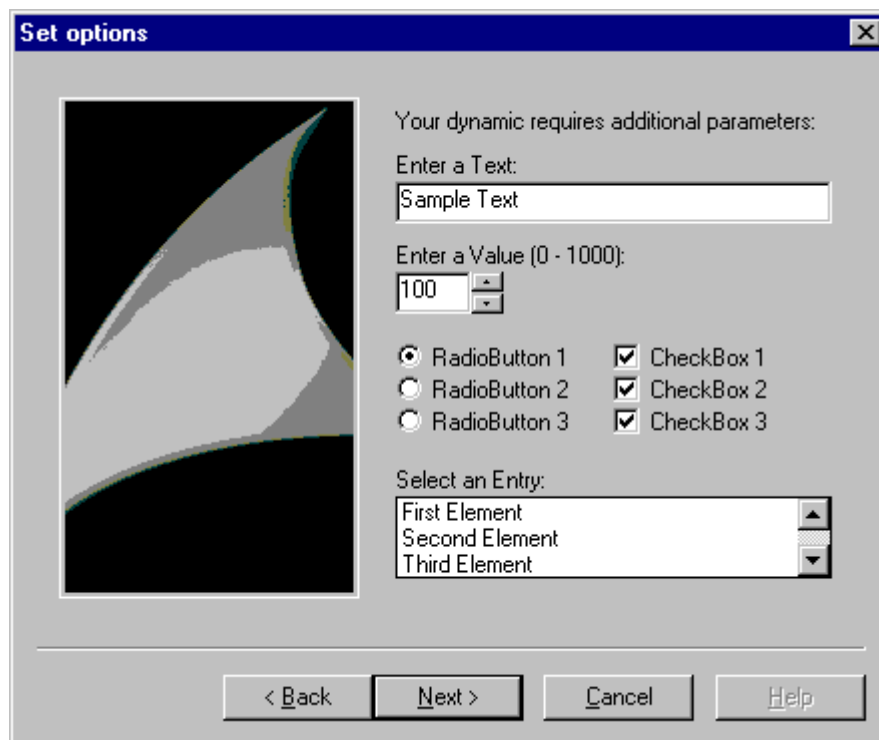
	Return value
HWND	Returns the object handle.
pSelect	Number of the selected entry. The number is the index in the list (beginning with 0).

Example

The following excerpt from the file "Demo.wnf" shows the use of this function. In the "Set options" dialog of the "Demo Wizard" a selection field is displayed. The possible size of the selection field is three lines. As more than three entries exist, a scroll bar is displayed.

Note

With the function "CreateListbox" only the selection field itself is created. The line content must be entered using the function "SendMessage".



```
char* DynWizListStatic= "Select an entry:";

...

int g_iListBox = 0;

//Type definition of the elements in the selection field
typedef struct listBoxItem
{
    int iIndex;
    char szItemText[256];
}LB_ITEM, *PLB_ITEM;
```

```

#define LB_NUM_LINES 5

LB_ITEM g_itemListBox[LB_NUM_LINES] =
{
{ 0, "First Element"},
{ 1, "Second Element"},
{ 2, "Third Element"},
{ 3, "Fourth Element"},
{ 4, "Fifth Element"}
};

void OnOption1(void)
{
static BOOL bFirst = TRUE;
HWND hWnd = NULL;

.....
if (bFirst == TRUE)
{
strcpy(g_szEdit,DynWizEdit);
bFirst = FALSE;
}

...
...

//Static text for the selection field
CreateStatic(0,162,DynWizListStatic);

...

//Selection field
hWnd = CreateListbox(0,177,"Headline",LB_NUM_LINES,&g_iListBox);
MoveWindow(hWnd,0,177,(rect.right-rect.left),50,TRUE);

//With the function "CreateListbox" only the box itself is created. The line content must be
entered using //the function "SendMessage".
for (i=0; i<LB_NUM_LINES; i++)
{
SendMessage(hWnd, LB_INSERTSTRING, (WPARAM)-1,
(LPARAM)g_itemListBox[i].szItemText);
}

```

```
}
```

CreateCheckBox

Introduction

In the "Set options" dialog a check box is displayed for the x,y coordinates. This check box allows enabling an option. In a dialog, several check boxes can be used.

Syntax

HWND CreateCheckBox (int x, int y, char* Text, BOOL* pSelect)

Parameters

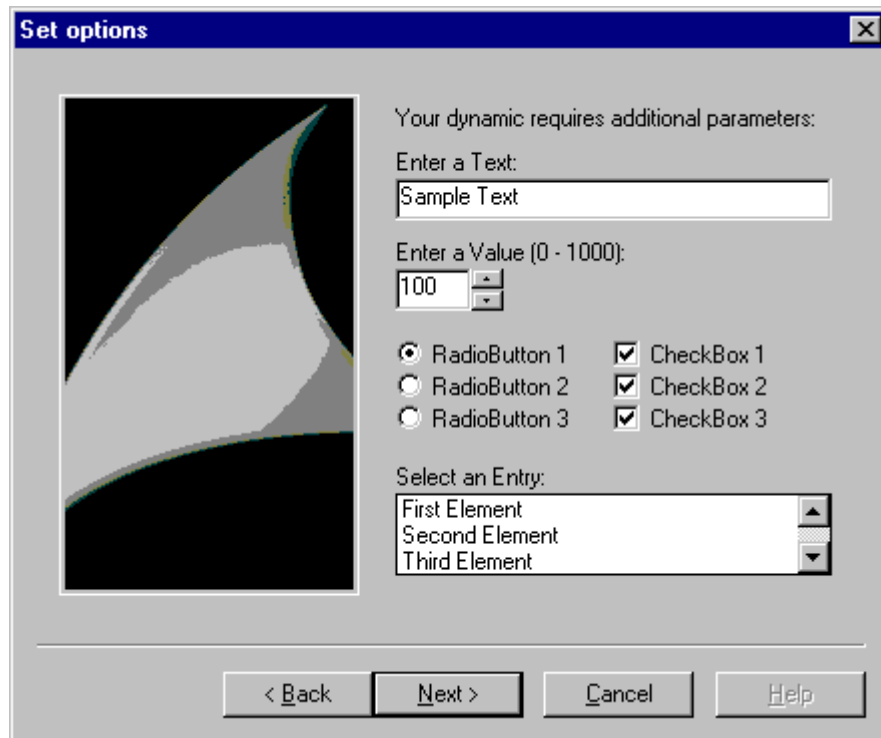
Parameters	Description
int x	Shows the value of the x coordinate.
int y	Shows the value of the y coordinate.
char* Text	Text displayed right of the check box.
BOOL* pSelect	Pointer to the result variable. The result variable should be preassigned a default value (True/False).

Return value

	Return value
HWND	Returns the object handle.
pSelect	Activation status FALSE = not activated TRUE = activated

Example

The following excerpt from the file "Demo.wnf" shows the use of this function. In the "Set options" dialog of the "Demo Wizard" three check boxes are displayed, each one representing an option. Each option can be independently activated.



```

BOOL g_bCheck1 = TRUE;
BOOL g_bCheck2 = TRUE;
BOOL g_bCheck3 = TRUE;

void OnOption1(void)
{
    static BOOL bFirst = TRUE;
    HWND hWnd = NULL;
    .....
    if (bFirst == TRUE)
    {
        ...
    }
    ...
}

```

```
...  
//Check box  
iMid = (rect.right-rect.left)/2 ;  
  
CreateCheckBox(iMid,100,"CheckBox 1",&g_bCheck1);  
CreateCheckBox(iMid,116,"CheckBox 2",&g_bCheck2);  
CreateCheckBox(iMid,132,"CheckBox 3",&g_bCheck3  
}
```

CreateFrame

Introduction

In the "Set options" dialog a rectangular border is displayed. The left upper corner of the frame is defined by the x,y coordinates. The right lower corner of the frame is identical to the right lower corner of the option window.

Syntax

HWND CreateFrame (int x, int y, char* Title)

Parameters

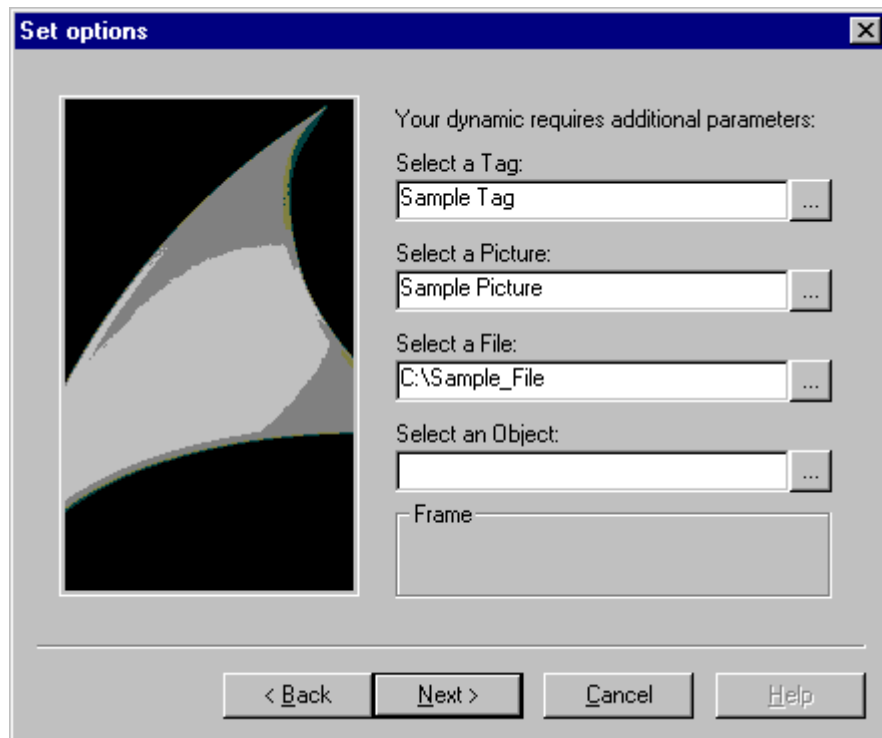
Parameters	Description
int x	Shows the value of the X coordinate.
int y	Shows the value of the Y coordinate.
char* Title	Label at the top edge of the rectangle

Return value

	Return value
HWND	Returns the object handle.

Example

The following excerpt from the file "Demo.wnf" shows the use of this function. In the "Set options" dialog of the "Demo Wizard" a frame bearing the title "Frame" is displayed.



```
void OnOption2(void)
{
    //Frame
    CreateFrame(0,150,"Frame");
}
...
...
```

CreateRadioButton

Introduction

In the "Set options" dialog a radio button is displayed for the x,y coordinates. This radio button allows enabling an option.

Using radio buttons is only useful if there are several of them in a dialog. There is only one radio button active at a time.

Syntax

HWND CreateRadioButton (int x, int y, char* Text, BOOL* pSelect)

Parameters

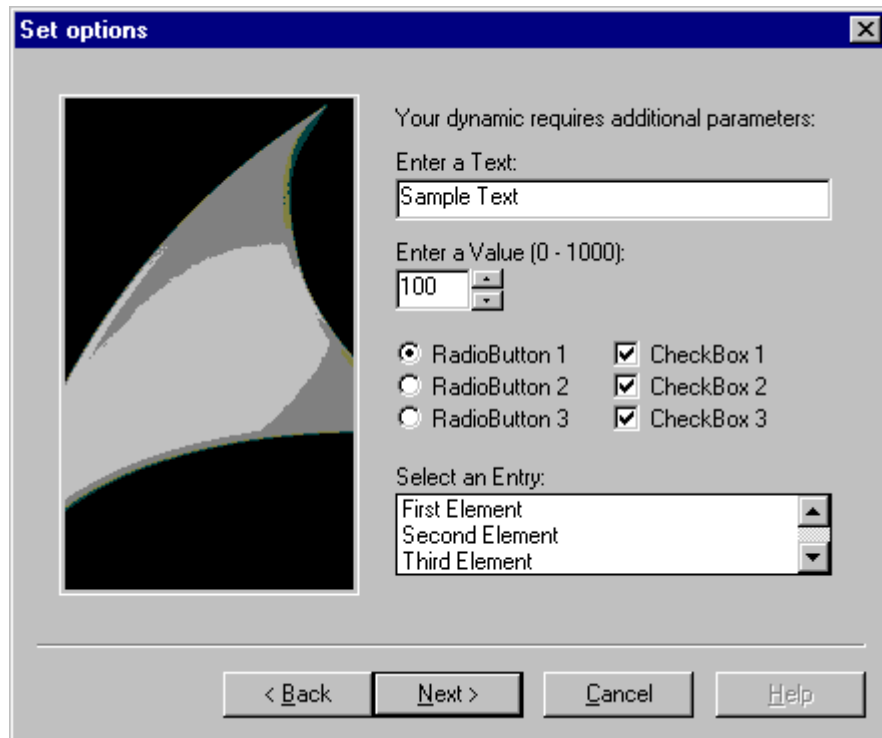
Parameters	Description
int x	Shows the value of the X coordinate.
int y	Shows the value of the Y coordinate.
char* Text	Name of the option activated with the radio button. The text is displayed to the right of the radio button.
BOOL* pSelect	Pointer to the result variable. The result variable should be preassigned a default value (True/False).

Return value

	Return value
HWND	Returns the object handle.
pSelect	Activation status: FALSE = not activated TRUE = activated

Example

The following excerpt from the file "Demo.wmf" shows the use of this function. In the "Set options" dialog of the "Demo Wizard" three radio buttons are displayed, each one representing an option. Only one option can be activated.



```

BOOL g_bOption1 = TRUE;
BOOL g_bOption2 = FALSE;
BOOL g_bOption3 = FALSE;

void OnOption1(void)
{
    static BOOL bFirst = TRUE;
    HWND hWnd = NULL;
    .....
    if (bFirst == TRUE)
    {
        ...
    }
    ...
}

```

...

//Radio buttons

```
CreateRadioButton(0,100,"RadioButton 1",&g_bOption1);
```

```
CreateRadioButton(0,116,"RadioButton 2",&g_bOption2);
```

```
CreateRadioButton(0,132,"RadioButton 3",&g_bOption3);
```

```
}
```

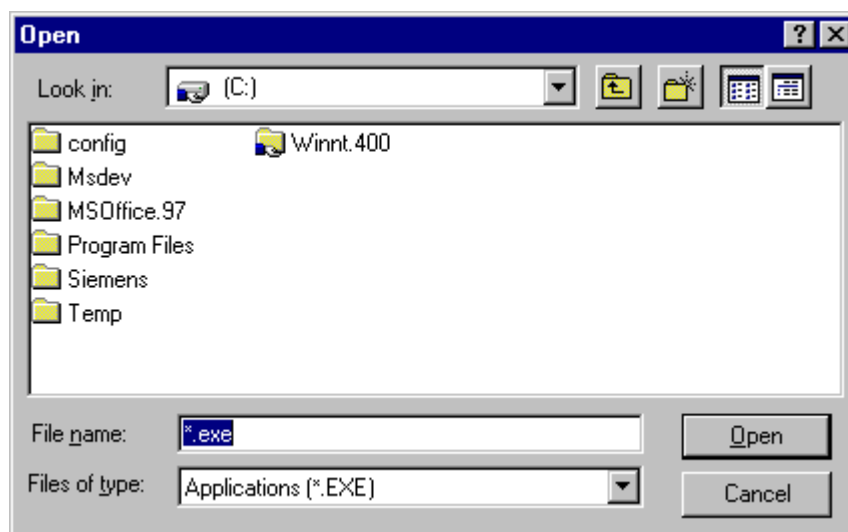
CreateFileBrowser

Introduction

In the "Set options" dialog an input field with a Browse button is displayed for the x,y coordinates. A file name can be typed in this input field.



Clicking the Browse button opens a file selection dialog.

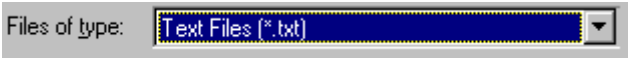
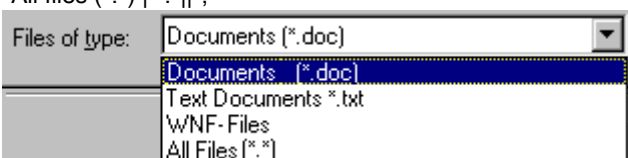


Syntax

HWND CreateFileBrowser (int x, int y, DWORD Flags, char* Filter, char* Dateiname)

Parameters

Parameters	Description
int x	Shows the value of the x coordinate.
int y	Shows the value of the y coordinate.

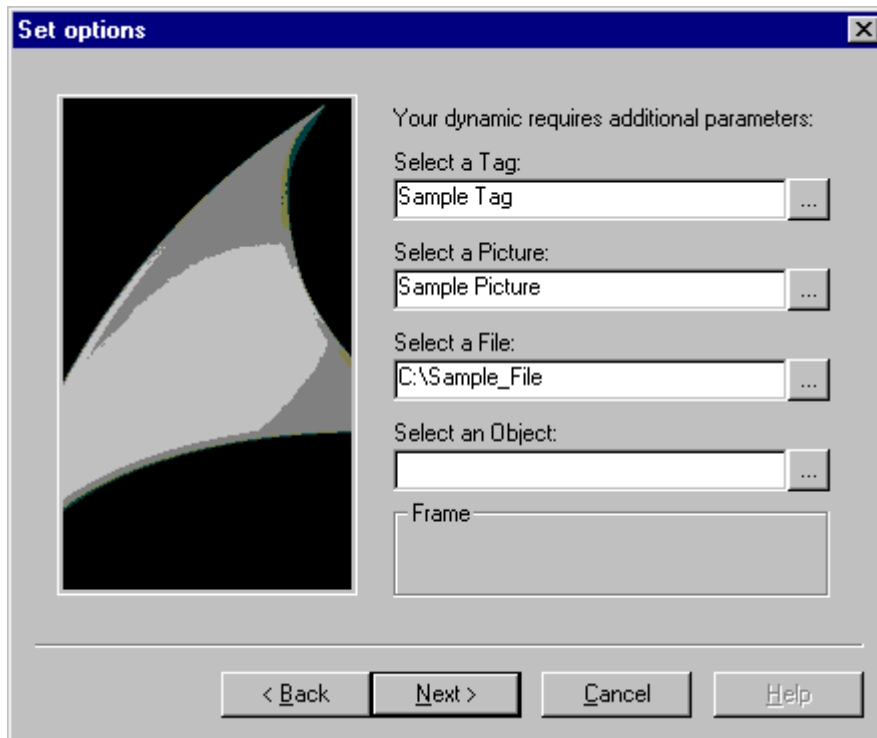
Parameters	Description
DWORD Flags	Control flag of the selection window: FB_WITHPATH = File name with path FB_SAVE_AS = Instead of the Open dialog the "Save as" dialog appears.
char* Filter	Filter for the data type display in the selection field of the file selection dialog. By specifying the extension, the data types to be shown in the selection field are defined. A filter consists of a string pair. The first string is the filter name. The second string is the filter function in the format *.typ, 'typ' being a file extension. The selection field only shows files with this extension. 1. and second string are separated by . Several filters can be stringed by . The last filter is limited by . Examples: char* Filter1 = "Graphic pictures (*.PDL) *.PDL ";  char* Filter2 = "Documents (*.doc) *.doc " "Text files *.txt *.txt " "WNF files *.wnf " "All files (*.*) *.* ";  No blanks are allowed at the end of the filter function.
char* File name	Input buffer for the file name. A path name can be defined as default value. This standard value has the following effect: The path name is displayed in the input window by default. Clicking on the Browse button sets the path in the file selection dialog. If the file name has the extension '*.typ', all files of this type are shown in the selection field of the selection dialog.

Return value

	Return value
HWND	Returns the object handle.
File name	Input buffer contains the file name.

Example

The following excerpt from the file "Demo.wnf" shows the use of this function. In the "Set options" dialog of the "Demo Wizard" an input field with a Browse button is displayed. Clicking on the Browse button opens the file selection dialog.



```
char* DynWizFileBrowserStatic = "Select a file:";
char* DynWizFileBrowser = "C:\\Sample file";
char* DynWizFilter = "Text files (*.txt) | *.txt|"
    "All files (*.*) | *.*||";
...
char g_szFileBrowser[256];
...
void OnOption2(void)
{
    static BOOL bFirst = TRUE;
    HWND hWnd = NULL;
    RECT rect;
    ...
    if (bFirst == TRUE)
    {
```

```

...
strcpy(g_szFileBrowser, DynWizFileBrowser);
First = FALSE;
}
...
...
// Static text for the input field with Browse button
CreateStatic(0, 95, DynWizFileBrowserStatic);
//File selection dialog
hWnd =
CreateFileBrowser(0, 110, FB_WITHPATH, DynWizFilter, g_szFileBrowser);
MoveWindow(hWnd, 0, 110, (rect.right-rect.left), 21, TRUE);
}

```

CreateVarBrowser / CreateVarBrowserEx

Introduction

In the "Set options" dialog an input field with a Browse button is displayed for the x,y coordinates. A tag name can be typed in this input field. Clicking on the Browse button opens the WinCC tag selection dialog. The function "CreateVarBrowserEx" allows the additional parameterization of a tag filter. This filter limits the tags displayed in the tag selection dialog. Filtering is possible by data type, tag group, tag name and connection.

Syntax

HWND CreateVarBrowser (int x, int y, char* VarName)

HWND CreateVarBrowserEx (int x, int y, LPDM_VARFILTER VarFilter, char* VarName)

Parameters

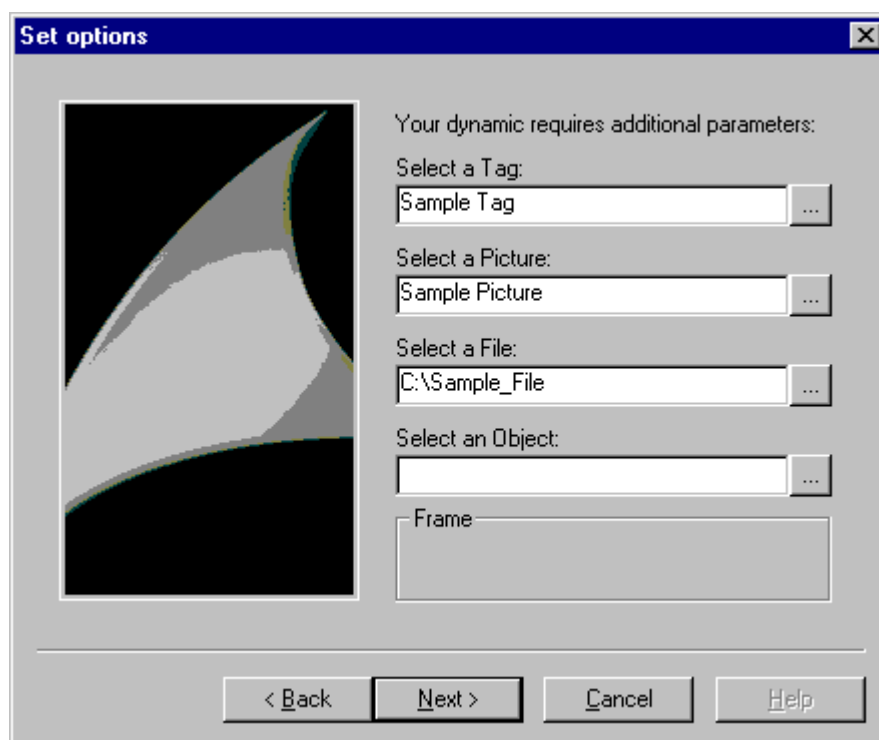
Parameters	Description
int x	Shows the value of the x coordinate.
int y	Shows the value of the y coordinate.
LPDM_VARFILTER VarFilter	Optional indication of a pointer to the tag filter. When specifying a NULL pointer no filter is active. The tag filter must be defined by means of the structure DM_VARFILTER. Additional information on this topic may be found in the WinCC ODK documentation.
char* VarName	Contains the tag name. The tag name may have a predefined default value. This entry is always displayed.

Return value

	Return value
HWND	Returns the object handle.
VarName	Input buffer contains the tag name

Example

The following excerpt from the file "Demo.wnf" shows the use of this function. In the "Set options" dialog of the "Demo Wizard" an input field with a Browse button is displayed. Clicking on the Browse button opens the WinCC tag selection dialog.



```
char* DynWizVarBrowser = "Sample tag";
char* DynWizPicBrowserStatic = "Select a picture:";
...
char g_szVarBrowser[256];
...
void OnOption2(void)
{
    static BOOL bFirst = TRUE;
    HWND hWnd = NULL;
```

```

RECT rect;

...

if (bFirst == TRUE)
{
    ...

    strcpy(g_szVarBrowser, DynWizVarBrowser);
    First = FALSE;
}

...

...

// Static text for the input field with Browse button
CreateStatic(0, 95, DynWizFileBrowserStatic);

//Tag selection dialog
hWnd =
CreateFileBrowser(0, 110, FB_WITHPATH, DynWizFilter, g_szFileBrowser);
GetWindowRect(GetParent(hWnd), &rect);
MoveWindow(hWnd, 0, 110, (rect.right-rect.left), 21, TRUE);
}

```

CreatePackageBrowser/CreatePackageBrowserEx

Introduction

In the "Set options" dialog an input field with a Browse button is displayed for the x,y coordinates. A name can be typed in this input field. The package browser is enabled by clicking on the Browse button at the right edge of the input field. The flag or the ProgID defines the type of data to be shown from the package.

With the function "CreatePackageBrowserEx" a ProgID can be transferred instead of a flag.

Syntax

HWND CreatePackageBrowser (int x, int y, DWORD flags, char* Name)

HWND CreatePackageBrowserEx (int x, int y, char* ProgID, char* Name)

Parameters

Parameters	Description
int x	Shows the value of the x coordinate.
int y	Shows the value of the y coordinate.

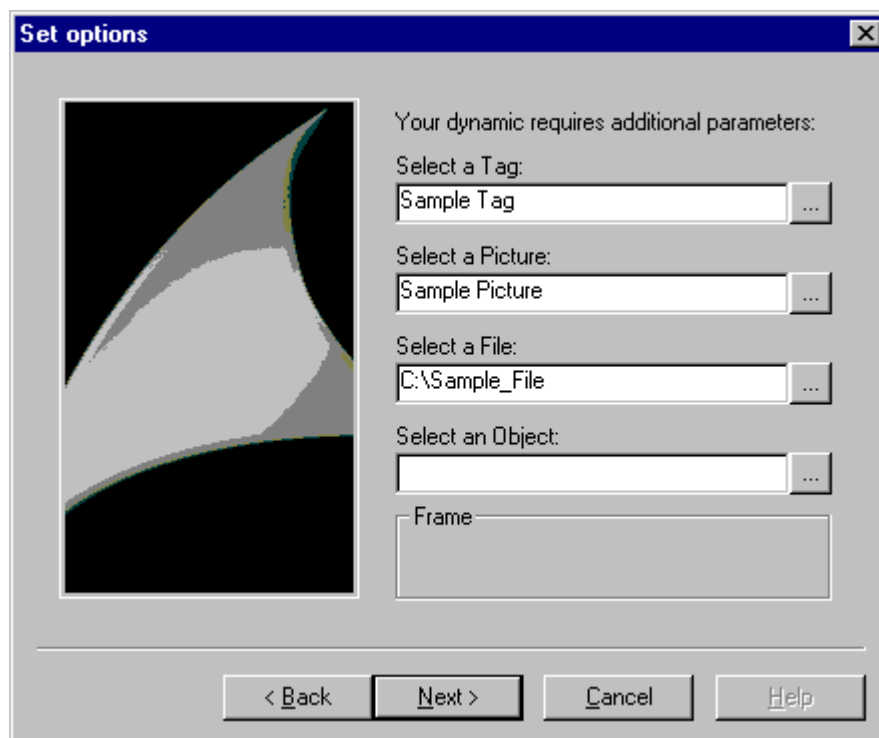
Parameters	Description
DWORD flags	Currently only PB_PICTURE can be used. This enables picture selection.
char* ProgID	Programmatic ID of the component used for building the selection. By transferring "WinCC.CCFileASOStub.1" the picture selection is addressed.
char* Name	Contains the name. The name may have a predefined default value. This entry is always displayed.

Return value

	Return value
HWND	Returns the object handle.
Name	Input buffer contains the name

Example

The following excerpt from the file "Demo.wnf" shows the use of this function. In the "Set options" dialog of the "Demo Wizard" an input field with a Browse button is displayed. Clicking on the Browse button opens the picture selection dialog.



```
char* DynWizPicBrowserStatic = "Select a picture:";
char* DynWizPicBrowser = "Sample picture";
```



```

...
char g_szPicBrowser[256];
...
void OnOption2(void)
{
    static BOOL bFirst = TRUE;
    HWND hWnd = NULL;
    RECT rect;
    ...
    if (bFirst == TRUE)
    {
        ...
        &#9;strcpy(g_szPicBrowser,DynWizPicBrowser);

        First = FALSE;
    }
    ...
    ...

    // Static text for the input field with Browse button
    CreateStatic(0,50,DynWizPicBrowserStatic);

    //Picture selection dialog
    hWnd = CreatePackageBrowser(0,65,PB_PICTURE,g_szPicBrowser);
    MoveWindow(hWnd,0,65,(rect.right-rect.left),21,TRUE);
}

```

CreateObjectBrowser

Introduction

In the "Set options" dialog an input field with a Browse button is displayed for the x,y coordinates. An object or property name can be typed in this input field. Clicking on the Browse button opens a selection dialog. In this selection dialog an object or property name can be searched and selected.

Syntax

HWND CreateObjectBrowser (int x, int y, char* Title, DWORD flags, char* ObjectName)

Parameters

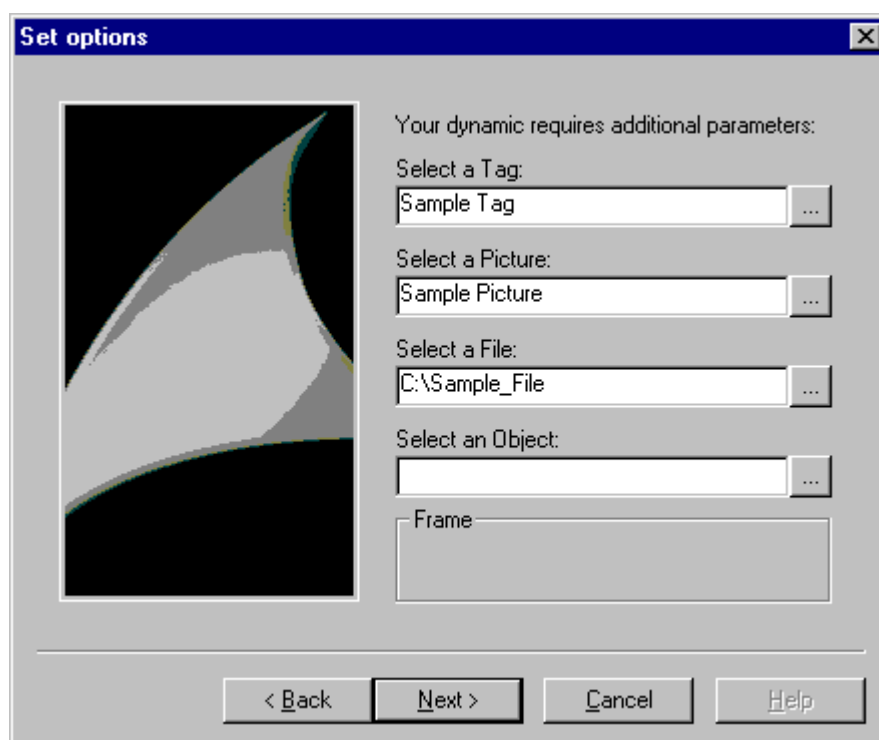
Parameters	Description
int x	Shows the value of the x coordinate.
int y	Shows the value of the y coordinate.
char* title	Label of the selection dialog.
DWORD flags	Two different flags can be transferred: OB_OBJECTS Display of all objects OB_PROPERTIES Additionally the property selection is offered.
char* ObjectName	Input buffer for the object or property name. The input buffer can be set to a default value.

Return value

	Return value
HWND	Returns the object handle.
ObjectName	Input buffer contains the object or property name

Example

The following excerpt from the file "Demo.wnf" shows the use of this function. In the "Set options" dialog of the "Demo Wizard" an input field with a Browse button is displayed. Clicking on the Browse button opens the window object selection dialog.



```
char* DynWizObjectBrowserStatic = "Select an object:";
char* DynWizObjectBrowser = "Object";
char* DynWizObject = "Window object selection";
;
...
char g_szObjectBrowser[256];
...
void OnOption2(void)
{
    static BOOL bFirst = TRUE;
    HWND hWnd = NULL;
    RECT rect;
    ...
    if (bFirst == TRUE)
    {
        ...
        strcpy(g_szObjectBrowser, DynWizObjectBrowser);
        First = FALSE;
    }
    ...
    ...

    // Static text for the input field with Browse button
    CreateStatic(0, 50, &#9;CreateStatic(0, 140, DynWizObjectBrowserStatic);
);

//Window selection dialog
hWnd =
CreateObjectBrowser(0, 155, DynWizObject, OB_OBJECTS, g_szObjectBrowser)
;

MoveWindow(hWnd, 0, 155, (rect.right-rect.left), 21, TRUE);
}
```

1.4.5.13 Wizard functions for generating dynamics

GenerateBLOB

Introduction

The GenerateBLOB function (BLOB = Binary Large Object) creates an action which can be appended to a graphic object property. An action comprises 3 parts.

Prologue: This is the header of the C function.

Example:

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName, UINT nFlags, int x, int y)
{
```

The prologue depends on the trigger starting the action (in the above example: clicking the left mouse button).

Epilogue: This is the end of the C function made up of the '}' symbol.

Core: This part contains the actual functionality of the C function. Example:

```
ProgramExecute("notepad.exe");
```

This function creates and compiles the C code of the action. The compilation results in a P code. This code is interpreted and processed by the WinCC runtime system. In case of an incorrect C code no P code is created.

The function creates a BLOB in which the parts of the action (C code, P code, trigger...) are stored. Before the end of the wizard function the BLOB has to be deleted again. More detailed information on deleting the BLOB function may be found under "DeleteBLOB".

Syntax

AP_BLOB GenerateBLOB (char* Prolog, char* Epilog, char* Format, ...)

Parameters

Parameters	Description
char* Prologue	Prologue of the action as ASCII string.
char* Epilogue	Epilogue of the action as ASCII string.
char* Format	Core of the action as ASCII string or format string according to the standard function "printf".

Note

The C codes are created by means of the C function `sprintf`. The parameter is processed as format string, i.e. format control characters (e.g. `\ % "`) are evaluated. If these are to be transferred into the C code (e.g. as format string for a `printf` call in an action), they must be provided with a `\`.

Example:

`\` → `\\`

`%` → `\%`

`"` → `\"`

Return value

The function returns a structured tag of the type `AP_BLOB` with the following structural components:

Structural component	Return value
<code>DWORD dwPCodeSize</code>	Length of the created P code in bytes
<code>LPVOID lpPCode</code>	Pointer to the created P code
<code>int nErrors</code>	Number of compiler errors
<code>int nWarnings</code>	Number of compiler warnings

Example

The following excerpt from the file "Execute Programm.wnf" shows the use of this function. The wizard function creates a C script which starts another application (in this example: `notepad.exe`).

```
...
...
void OnGenerate(void)
{
    PCMN_ERROR pError;
    AP_BLOB *blob;
    char code[500];
    char sError[500];

    ..
    Slash2Db1Slash(g_Picture, strlen(g_Picture));
    ..
    sprintf(code, "%sProgramExecute(\"%s\");", ifcode, g_Picture);
    ..
    //Prologue
```

```
blob = GenerateBLOB("#include \"apdefap.h\"\\r\\n\"
\"void OnClick(char* lpszPictureName,\" \"char*lpszObjectName,char*
lpszPropertyName,\"
\"UINT nFlags,int x, int y) {\",
//Epilogue
\"}\",
//Core
code);

BEGIN_JCR_BLOBERRORS

SetAction(NULL, blob, g_Trigger);

END_JCR_BLOBERRORS

DeleteBLOB(blob);
}
```

Created C script

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName,
char* lpszObjectName,
char* lpszPropertyName,
UINT nFlags, int x, int y)
{
ProgramExecute("notepad.exe");
}
```

DeleteBLOB**Introduction**

The GenerateBLOB function creates a BLOB. At the end of the wizard function the BLOB has to be deleted again. The BLOB is deleted with the DeleteBLOB function.

Syntax

```
void DeleteBLOB (AP_BLOB* blob)
```

Parameters

Parameters	Description
AP_BLOB* blob	Pointer to the result variable of the "GenerateBLOB" function.

Example

```
DeleteBLOB(blob);
```

SetAction**Introduction**

An action is appended to the selected graphic object at the specified trigger.

If the trigger is an event, it is directly specified as calling parameter.

If the trigger is the dynamization of a property, it must be entered into the BLOB beforehand using the functions AddVarTrigger or AddTimeTrigger.

Note

If the action is not to be appended to the selected object but to another one, the API function PDLCSSetAction must be used. Further information on the function PDLCSSet Action may be found in the WinCC ODK manual.

Syntax

```
BOOL SetAction (char* Property, AP_BLOB* Blob, DWORD Trigger )
```

Parameters

Parameters	Description
char* Property	Property name. Always use the English name of the property. For a trigger on an event a NULL pointer has to be transferred.
AP_BLOB* Blob	Pointer to the result variable of the "GenerateBLOB" function.
DWORD TriggerID	ID of the trigger: NOTDEFINED = trigger is entered in the BLOB MOUSECLICK = mouse click MOUSEBUTTONDOWN = click left mouse button MOUSEBUTTONUP = release left mouse button MOUSERBUTTONDOWN = click right mouse button MOUSERBUTTONUP = release right mouse button KEYBOARDDOWN = press key KEYBOARDUP = release key OBJECTCHANGE = object change PROPERTYCHANGE = property change PICTUREOPEN = picture selection PICTURECLOSE = closing of picture

Return value

	Return value
BOOL return value	TRUE = Function completed successfully. FALSE = Function not completed successfully.

Example

See example in GenerateBLOB function.

See also

GenerateBLOB (Page 68)

AddTimeTrigger

Introduction

The function supplements the action with a trigger of the type "cyclic trigger".

Syntax

BOOL AddTimeTrigger (AP_BLOB* Blob, char* Name, DWORD TriggerType, DWORD GraphCycleType, DWORD CycleID)

Parameters

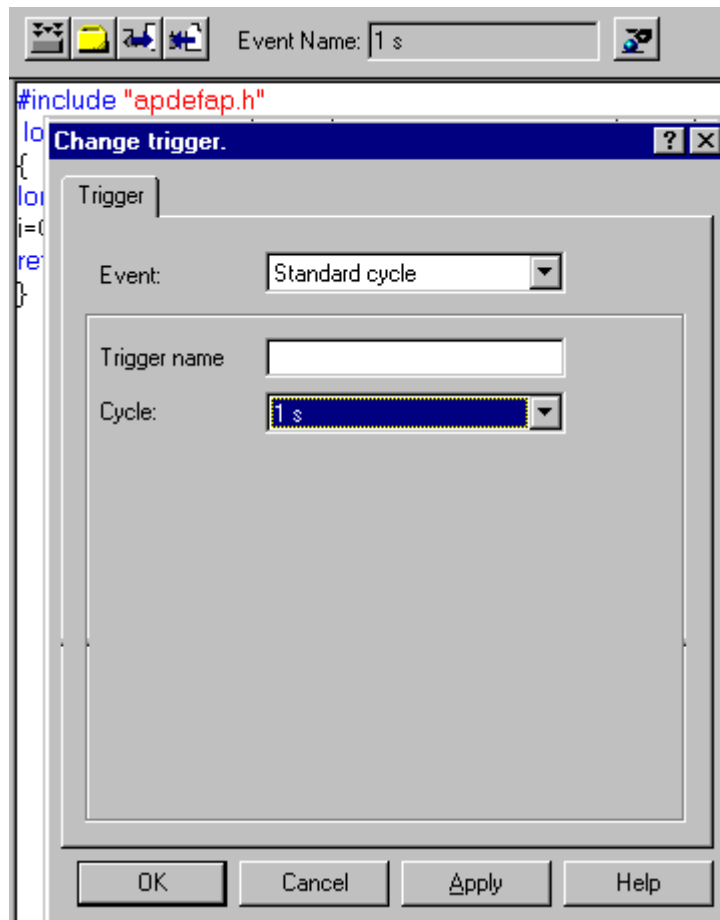
Parameters	Description
AP_BLOB* Blob	Pointer to the result variable of the "GenerateBLOB" function.
char* Name	Name of the event. This can be any ASCII string. The name is shown as event name in the action window.
DWORD TriggerType	Type of cyclic trigger: 2 = time cycle (standard cycle) 4 = graphic object cycle
DWORD GraphCycleType	Type of graphic object cycle: 2 = window cycle 1 = picture cycle
DWORD CycleID	Trigger cycle: 0 = upon change 1 = 250 ms 2 = 500 ms 3 = 1 s 4 = 2 s 5 = 5 s 6 = 10 s 7 = 1 min 8 = 5 min 9 = 10 min 10 = 1 h 11 = user cycle 1 12 = user cycle 2 13 = user cycle 3 14 = user cycle 4 15 = user cycle 5

Return value

	Return value
BOOL	TRUE = Function completed successfully. FALSE = Function not completed successfully.

Example

The time passing between two action triggerings is 1s.



```
BOOL FctRet;  
..  
FctRet = AddTimeTrigger(blob, "1 sec", 2, 0, 3);
```

AddVarTrigger /AddVarTriggerEx

Introduction

The function supplements the action with a trigger of the type "tag trigger".

Syntax

```
BOOL AddVarTrigger (AP_BLOB* Blob, char* EventName, char* VarName )
```

BOOL AddVarTriggerEx (AP_BLOB* Blob, char* EventName, char* VarName, DWORD CycleID)

Parameters

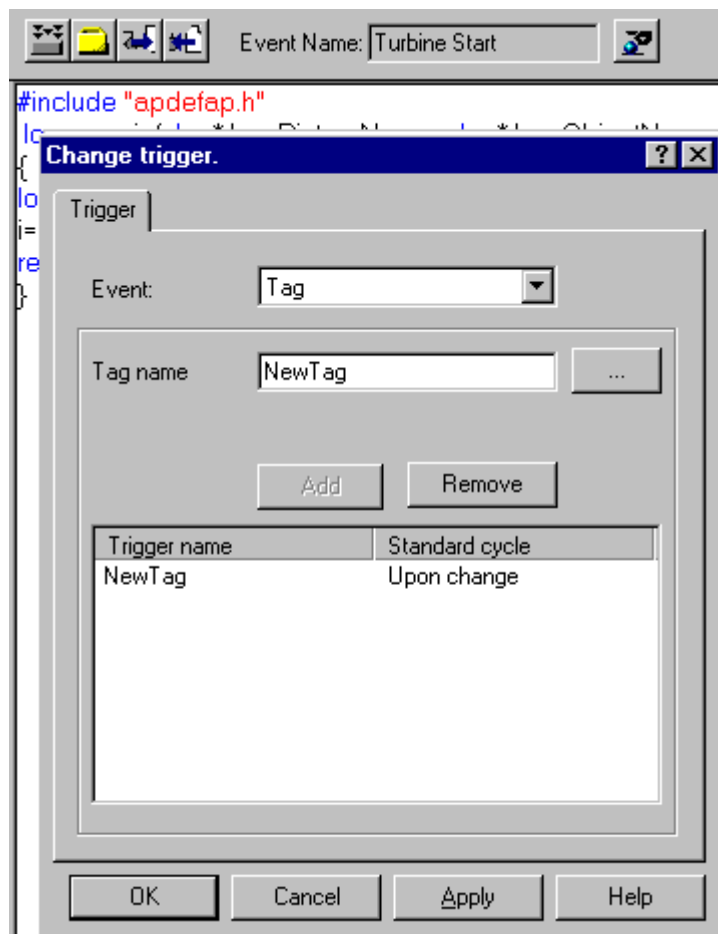
Parameters	Description
AP_BLOB* Blob	Pointer to the result variable of the "GenerateBLOB" function.
char* EventName	Name of the event. This can be any ASCII string. The name is shown as event name in the action window.
char* VarName	Name of the WinCC tag initiating the triggering or participating in it.
DWORD CycleID	Trigger cycle: 0 = upon change 1 = 250 ms 2 = 500 ms 3 = 1 s 4 = 2 s 5 = 5 s 6 = 10 s 7 = 1 min 8 = 5 min 9 = 10 min 10 = 1 h 11 = user cycle 1 12 = user cycle 2 13 = user cycle 3 14 = user cycle 4 15 = user cycle 5 For the function AddVarTrigger the value CycleID = 4 ist predefined.

Return value

	Return value
BOOL	TRUE = Function completed successfully. FALSE = Function not completed successfully.

Example

You have entered the tag 'StartTurbine1' as trigger in the trigger type "Tag". The action is started as soon as the value of one of these tags changes.



```
BOOL FctRet
```

```
FctRet = AddVarTriggerEx(blob,"Turbine Start","StartTurbine1",0);
```

SetValidateFct

Introduction

The name of a check function is communicated to the Dynamic Wizard. The check function allows checking the options and trigger parameters. In case of a negative test result a reentry can be initiated.

The check function is called when the button "Continue" is clicked in the "Select options" or "Set trigger" dialogs. In case of a positive test result the dialog is closed on the next page

appears. In case of a negative test result the dialog remains active. A continuation is only possible after entering the correct parameters.

The check function takes effect as soon as it is set in the Dynamic Wizard. It is also effective for subsequent option pages. If no or another check function is to take effect, either a dummy function (with positive test result) or another check function must be set.

Syntax

BOOL SetValidateFct (LPCSTR FctName)

Parameters

Parameters	Description
LPCSTR FctName	Name of the check function as ASCII string.

Return value

	Return value
BOOL	Result of the test TRUE = positive test result. FALSE = negative test result.

Example

The following excerpt from the file "Instanzobjekt.wnf" shows the use of this function.

The wizard function has been expanded with a check function.

```
...  
...  
// Validation option 1  
BOOL ValidateOpt1(void)  
{  
    // Property selected  
    return (strcmp(g_NewInst, ""));  
}  
  
void OnOption1(void)  
{  
    HWND hWnd;  
    RECT rect;
```

```
DM_VARFILTERdmFilter = {DM_VARFILTER_TYPE, 1, NULL, NULL, NULL,
NULL };

SetValidateFct("ValidateOpt1");

sprintf(g_NewInst, "");

..
}
```

EnumProperty/EnumPropertyEx

Introduction

The EnumProperty function lists the object properties of an object. The EnumPropertyEx function allows specifying the object properties to be listed.

Syntax

BOOL EnumProperty (char* FName, LPVOID pltem, DWORD dwFlags);

BOOL EnumPropertyEx (LPCTSTR Projectname, LPCTSTR Picturename, LPCTSTR Objectname, char* FName, LPVOID pltem, DWORD dwFlags);

Parameters

Parameters	Description
LPCTSTR Projectname	Pointer to the name of the project, including directory and file extension.
LPCTSTR Picturename	Pointer to the name of the picture whose objects are to be listed. Distinction is made between upper and lower case.
LPCTSTR object name	Pointer to the name of the object
char* FName	Name of your callback function called once for each object property.
LPVOID pltem	Pointer to application-specific data passed on to the callback function. This pointer is not evaluated by the function but made available in the callback function again.
DWORD dwFlags	dwFlags specifies the property types to be listed. Presently the following specifications are possible: PropertyHasDynamic (Value: 0x0001)	Only object properties with dynamics are enumerated. PropertyHasEvents (Value: 0x0002)	Only object properties with events are enumerated. PropertyIsDynamicable (Value: 0x0003)	Only object properties that can be made dynamic are enumerated.

Return value

	Return value
BOOL	TRUE = Object properties of an object type listed FALSE = Error

Example

The following excerpt from the file "Dynamic Property.wnf" shows the use of this function.

```
...
...
// Callback function
BOOL EnumFct(char *property, VARTYPE vt, LPVOID pItem)
{
    sprintf(g_prop[SendMessage((HWND)pItem, LB_INSERTSTRING, (WPARAM)-1,
        (LPARAM)property)], property);
    return TRUE;
}

void OnOption1(void)
{
    HWND hWnd, LBHwnd;
    RECT rect;
    static BOOL bFirst = TRUE;

    if(bFirst)
    {
        ...
    }
    ...

    CreateStatic(0, 10, "Properties of the current object :");
    LBHwnd=CreateListbox(0, 30, g_Headline, 8, &g_indexProperty);
    EnumProperty("EnumFct", LBHwnd, 3);
    GetWindowRect(GetParent(LBHwnd), &rect);
    ...
```

1.4.5.14 Wizard WinCC functions

GetProjectName

Introduction

The path of the current WinCC project is determined.

Syntax

LPCSTR GetProjectName (void)

Return value

	Return value
LPCSTR	Pointer to the ASCII string of the MCP file

Example

```
LPCSTR Name;
```

```
Name = GetProjectName();
```

The function provides e.g. the following result: C:\Siemens\WinCC\WinCCProjects
\Example.mcp

GetPictureName

Description

The name of the current picture (*.pdl) is determined.

Syntax

LPCSTR GetPictureName (void)

Return value

	Return value
LPCSTR	Pointer to the ASCII string of the PDL file

Example

```
LPCSTR Name;
```

```
Name = GetPictureName();
```

The function provides e.g. the following result: TurbineControl.PDL

GetDefaultWNFPath

Description

The path of the current WNF directory is determined.

Syntax

```
LPCSTR GetDefaultWNFPath ( void )
```

Return value

	Return value
LPCSTR	Pointer to the ASCII string of the path name

Example

```
LPCSTR Name;
```

```
Name = GetDefaultWNFPath();
```

The function provides e.g. the following result: C:\Siemens\WinCC\wscripts\wscripts.deu\

GetObjectName

Introduction

The name of the selected graphic object in the current picture is determined.

Syntax

```
LPCSTR GetObjectName ( void )
```

Return value

	Return value
LPCSTR	Pointer to the ASCII string of the object name

Example

```
LPCSTR Name;
```

```
Name = GetObjectName();
```

The function provides e.g. the following result: Button1

InsertXRefSection**Description**

The function inserts into the transferred source code a section according to the Xref notation, so that the transferred tags and picture names are entered as define.

Syntax

```
BOOL InsertXRefSection (char * SourceCode, char* TagName[], int TagCount, char* PictName[], int PictCount)
```

Parameters

Parameters	Description
char *SourceCode	CodeBuffer in which to insert the Xref section
char *TagName[]	NULL or field of tag names inserted into the Xref section.
int TagCount	Number of tag names in the field DayName[]
char *PictName[]	NULL or field of picture names inserted into the Xref section
int PictCount	Number of picture names in the field PictName[]

Return value

	Return value
BOOL	The result value indicates whether the function has been completed successfully.
char *TagName[]	Field of the defines for the tags passed at the same position
char *PictName[]	Field of the defines for the pictures passed at the same position

Example

```
char* szPictureArray[1];
char szPictName[255];
char szSourceCode[1100];

strcpy(szPictName, "Newpdl.pdl");
szPictureArray[0] = szPictName;
strcpy(szSourceCode, "");
InsertXrefSection(szSourceCode, NULL, 0, szPictureArray, 1);
```

The function returns the following result:

szSourceCode:

```
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction "DMTagName"
// next TagID : 1
// WINCC:TAGNAME_SECTION_END
// WINCC:PICNAME_SECTION_START

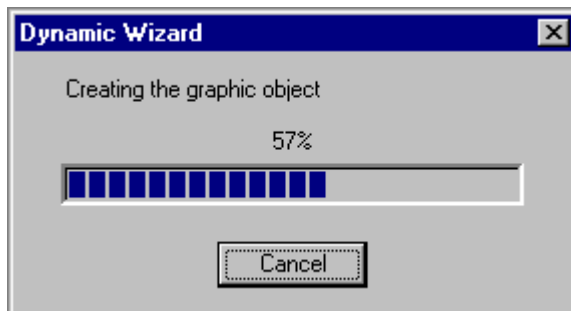
// syntax: #define PicNameInAction "PictureName"
// next PicID : 1
#define PIC_0 " Newpdl.Pdl"
// WINCC:PICNAME_SECTION_END
szPictureArray[0]: "PIC_0"
```

1.4.5.15 Wizard progress functions

Wizard progress functions

Introduction

The progress functions serve to show the progress (in %) of a procedure in a "progress box".



Upon creation of the progress bar (CreateProgressDlg) a start and end value and an increment are specified. The start value corresponds to 0% progress and the end value to 100% progress. The increment defines the progress change steps.

Typically, the start value is =0 and the increment =1. The end value corresponds to the number of executed actions in the process.

During the procedure the progress is incremented (Progress_StepIt) or set to a defined value (Progress_SetPos).

At the end of the procedure the progress display has to be removed again (DestroyProgressDlg)

It is possible to display a text in the progress bar (Progress_SetStatus), such as "Creating graphic objects". This can also be changed during processing to differentiate various partial procedures.

In most cases it is not possible to divide the procedure in a way that allows a chronologically linear progress display. However, this is not really necessary. Displaying the progress as such is sufficient.

See also

DestroyProgressDlg (Page 86)

Progress_SetPos (Page 86)

Progress_StepIt (Page 86)

Progress_SetStatus (Page 85)

CreateProgressDlg (Page 85)

CreateProgressDlg

Introduction

A progress bar shows the progress of a processing procedure from 0 to 100%.

Syntax

PROGRESS_DLG CreateProgressDlg (int nLower, int nUpper, int nStepInc)

Parameters

Parameters	Description
int nLower	Progress start value (corresponds to 0 %)
int nUpper	Progress end value (corresponds to 100 %)
int nStepInc	Progress increment

Return value

	*Return value
PROGRESS_DLG	Object handle

Progress_SetStatus

Description

A text is entered as header into the progress bar.

Syntax

void Progress_SetStatus (PROGRESS_DLG hDlg, char* ActionName)

Parameters

Parameters	Description
PROGRESS_DLG hDlg	Object handle
char* ActionName	Header text

Progress_Stepl

Description

The progress of a processing procedure is incremented by one step.

Syntax

```
void Progress_Stepl (PROGRESS_DLG hDlg )
```

Parameters

Parameters	Description
PROGRESS_DLG hDlg	Object handle

Progress_SetPos

Description

In the progress bar the progress is set to a defined value. The value must lie between start and end value.

Syntax

```
void Progress_SetPos (PROGRESS_DLG hDlg, int nPos )
```

Parameters

Parameters	Description
PROGRESS_DLG hDlg	Object handle
int nPos	Progress value

DestroyProgressDlg

Introduction

The progress bar will be closed.

Syntax

void DestroyProgressDlg (PROGRESS_DLG hDlg)

Parameters

Parameters	Description
PROGRESS_DLG hDlg	Object handle

1.4.5.16 Wizard Windows functions

Wizard Windows functions

Introduction

The following is a short description of Windows functions which must or can be used in connection with the Wizard system functions (especially with the window functions for parameter input).

More detailed information may be found in the Programmer's Reference of Microsoft Developers Studio /Win32 SDK.

See also

MessageBox (Page 91)

ShowWindow (Page 91)

GetWindow (Page 90)

SendMessage (Page 89)

MoveWindow (Page 89)

GetWindowRect (Page 88)

GetParent (Page 87)

GetParent

Introduction

The handle of the parent window is determined for a window, e.g. the handle of the options window.

Syntax

HWND GetParent (HWND hWnd)

Parameters

Parameters	Description
HWND hWnd	Handle of the window for which to determine the parent window

Return value

	Return value
HWND	Handle of the parent window NULL = No parent window exists.

GetWindowRect

Introduction

Size and coordinates of a window are determined, e.g. the size of the options window.

Syntax

BOOL GetWindowRect (HWND hWnd, LPRECT lpRect)

Parameters

Parameters	Description
HWND hWnd	Window handle
LPRECT lpRect	Pointer to a structured result variable

Return value

	Return value
BOOL	TRUE = Function completed successfully. FALSE = Function not completed successfully.
LPRECT lpRect	Structured result variable of the LPRECT structure with the structural components: LONG left: X coordinate of the left upper corner LONG top: Y coordinate of the left upper corner LONG right: X coordinate of the right lower corner LONG bottom: Y coordinate of the right lower corner:

See also

Adding the "Motor.wnf" script to the database (Page 96)

CreateEdit (Page 45)

MoveWindow

Introduction

Position and dimension of a window are changed, e.g. the position and the size of the input fields in the options window.

Syntax

BOOL MoveWindow (HWND hWnd, int x, int y, int nWidth, int nHeight, BOOL bRepaint)

Parameters

Parameters	Description
HWND hWnd	Window handle
int x	X coordinate of the left upper corner
int y,	Y coordinate of the left upper corner
int nWidth	Width
int nHeight	Height
BOOL bRepaint	TRUE = The window is redrawn.

Return value

	Return value
BOOL	TRUE = Function completed successfully. FALSE = Function not completed successfully.

See also

Creating the Dynamic Wizard function for the Motor (Page 96)

SendMessage

Introduction

A message is sent to a window. The function is used for filling a selection field, for example.

Syntax

LRESULT SendMessage (HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)

Parameters

Parameters	Description
HWND hWnd	Window handle
UINT Msg,	Message type: LB_INSERTSTRING = Insert text in a ListBox
WPARAM wParam	1. message parameter: -1 = The text is appended at the end.
LPARAM lParam	2. message parameter: Pointer to the text

Return value

	Return value
LRESULT	Object handle

GetWindow

Introduction

The handle of a window is determined which has a certain relation to another window (original window).

Syntax

GetWindow (HWND hWnd, UINT uCmd)

Parameters

Parameters	Description
HWND hWnd	Handle of the original window
UINT uCmd	Relation GW_HWNDFIRST = upper window

Return value

	Return value
HWND	Handle of the found window or NULL

ShowWindow

Introduction

The display type of a window is specified.

Syntax

ShowWindow (HWND hWnd, int nCmdShow)

Parameters

Parameters	Description
HWND hWnd	Window handle
int nCmdShow	Display status of the window SW_HIDE = not visible

Return value

	Return value
BOOL	TRUE = Window was visible FALSE = Window was not visible

MessageBox

Introduction

The function serves to display a message for the user if an error has occurred or a user action is required.

The message is displayed with a user specific text, title and button.

Syntax

int MessageBox (HWND hWnd, LPCTSTR lpText, LPCTSTR lpCaption, UINT uType)

Parameters

Parameters	Description
HWND hWnd	Handle of the parent window NULL = Message has no parent window.
LPCTSTR lpText	Message text

Parameters	Description
LPCTSTR lpCaption	Title text
UINT uType	Box type MB_OK = Message with an 'OK' button MB_OKCANCEL = Message with the buttons 'OK' and 'Cancel'

Return value

	Return value
int	ID of the button clicked: IDOK = 'OK' button clicked IDCANCEL = 'Cancel' button clicked

Example



```
int RetMsg;
```

```
RetMsg = MessageBox (NULL, "Error calling the API functions", "System error", MB_OK);
```

1.4.6 Examples

1.4.6.1 Examples

Introduction

In the context of this description two examples of Dynamic Wizard functions are given:

- Demo Wizard
- Dynamic motor

See also

Dynamic motor (Page 95)

Demo Wizard (Page 93)

1.4.6.2 Demo Wizard

Demo Wizard

Introduction

In the "Demo.wnf" file, a Dynamic Wizard called "Demo Wizard" is created. This Dynamic Wizard shows the basic functions available to make the entry of data convenient for the user. However, the Demo Wizard does not perform an action.

See also

Adding the "Demo.wnf" script to the database (Page 94)

Creating the help text (Page 93)

Creating the Dynamic Wizard function for the Demo Wizard (Page 93)

Creating the Dynamic Wizard function for the Demo Wizard

Requirements

A WinCC project must be open.

Procedure

1. In the Windows Explorer copy the "Demo.wnf" file from the directory "Siemens\ WinCC\ documents\ german" to the directory "Siemens\ WinCC\ wscript\ wscript.deu".
2. Start the Dynamic Wizard Editor.
3. In the File menu of the Dynamic Wizard Editor select "Open". The file selection dialog box opens.
4. Mark the "Demo.wnf" file. Click "Open". The "Demo.wnf" file is shown in an editor window.
5. Click the icon  in the toolbar to compile the script. The result is displayed in the output window.

See also

Adding the "Demo.wnf" script to the database (Page 94)

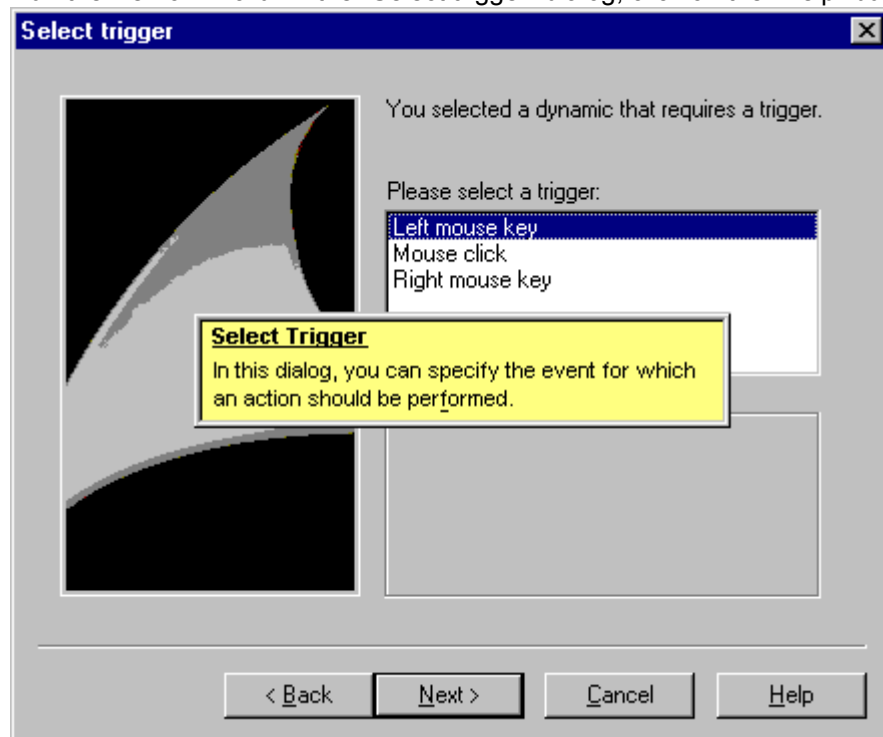
Creating the help text

Introduction

In this section a help for the dialog "Select trigger" is created.

Procedure

1. Click the icon  in the toolbar. The help editor opens.
2. In the "Wizard - Group" field, select "Example".
3. In the "Wizard - Name" field, select "Demo Wizard".
4. In the "Page" field, select "TriggerPage".
5. In the "Help - Text" field, enter the following text: "Select trigger
In this dialog you can specify the result for which the action is to be executed."
6. Close the help editor by clicking on the "OK" button.
7. Run the Demo Wizard. In the "Select trigger" dialog, click on the "Help" button.



Adding the "Demo.wnf" script to the database

Introduction

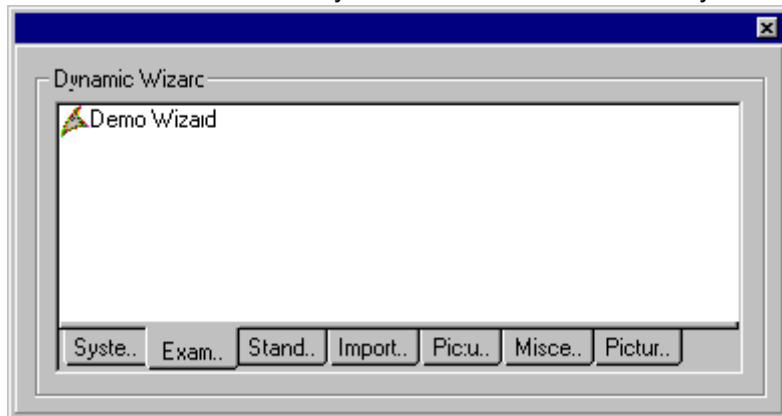
To be able to use the Dynamic Wizard function "Demo.wnf" in the Graphics Designer, it must be integrated into the database of the Dynamic Wizard.

To do this, the following steps are necessary,

1. Importing the Wizard scripts
2. Creating the cwd file

Procedure

1. Click the icon  in the toolbar. The file selection dialog box opens.
2. Select the "Demo.wnf" file. Click "Open".
3. Click the icon  in the toolbar to newly create the database.
4. In the "View" menu of the Dynamic Wizard Editor select "Dynamic Wizard".



5. Click on the "Example" tab. Double-click on the entry "Demo Wizard".

1.4.6.3 Dynamic motor

Dynamic motor

Introduction

In the "Motor.wnf" script file, a Dynamic Wizard called "Make Motor Dynamic" is created.

Note

This was created especially to make a user object called Motor dynamic and cannot be applied to any other kind of object.

See also

Making the customized object "Motor" dynamic (Page 97)

Adding the "Motor.wnf" script to the database (Page 96)

Creating the Dynamic Wizard function for the Motor (Page 96)

Creating the Dynamic Wizard function for the Motor

Requirements

A WinCC project must be open.

Procedure

1. In Windows Explorer open the "Motor.zip" Winzip file in the directory "Siemens\ WinCC\ documents\ german".
2. Extract the "Motor.wnf" file into the directory "..\WinCC\wscripts\wscripts.deu".
3. Extract the "Motor_dyn.pdl" file into the directory "..\WinCC\WinCCProjects\Name of the WinCCProject\GraCs".
4. Start the Dynamic Wizard Editor.
5. In the File menu of the Dynamic Wizard Editor select "Open". The file selection dialog box opens.
6. Mark the "Motor.wnf" file. Click "Open". The "Motor.wnf" file is displayed in an editor window.
7. Click the icon  in the toolbar to compile the script. The result is displayed in the output window.

See also

Adding the "Motor.wnf" script to the database (Page 96)

Adding the "Motor.wnf" script to the database

Introduction

To be able to use the Dynamic Wizard function "Motor.wnf" in the Graphics Designer, it must be integrated into the database of the Dynamic Wizard.

To do this, the following steps are necessary,

1. Importing the Wizard scripts
2. Creating the cwd file

Procedure

1. Click the icon  in the toolbar. The file selection dialog box opens.
2. Select the "Motor.wnf" file. Click "Open".
3. Click the icon  in the toolbar to newly create the database.

See also

Making the customized object "Motor" dynamic (Page 97)

Making the customized object "Motor" dynamic

Introduction

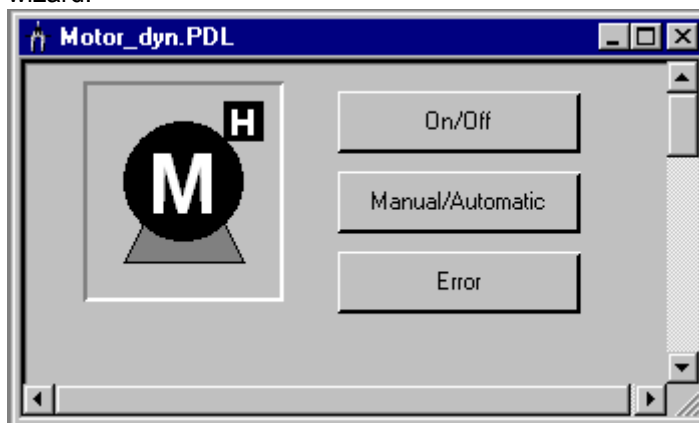
The Motor customized object is linked to a WinCC structure tag of the structure type "MotorStruct" via the "Dynamic Motor" dynamic wizard. In this context, various C-actions and tag connections are created on this object. This wizard cannot be used on other object types.

Requirements

- Create an internal tag "T08i_course_wiz_selected" of the data type "Text tag 8-bit character set".
- Create a structure with the name "MotorStruct" and three internal elements called "Active", "Hand" and "Error" of the data type BIT.
- Create an internal tag named "STR_Course_wiz1" of the data type "MotorStruct".

Procedure

1. Open the Graphics Designer. Select the "Open" item in the "File" menu. Select the "Motor_dyn.pdl" picture in the file selection dialog.
2. Select the Motor customized object. The "Example" tab offers the "Dynamic Motor" wizard.



3. Start the Dynamic Wizard. Click on the "Continue" button in the "Welcome to the Dynamic Wizard" dialog. The "Set options" dialog opens up.
4. Click on the Browse button in the "Set options" dialog. The tag selection dialog opens. Select "STR_Course_wiz1" as the structure tag. Close the dialog by clicking on the "OK" button.
5. Click on the Continue button in the "Set options" dialog. The "Finished!" dialog opens. Close the dialog by clicking on the "OK" button.

6. Save the picture. Start the Graphics Designer Runtime.
7. The buttons can be used to simulate the tag values of the selected motor.

See also

Creating the structure and the structure tag (Page 98)

Creating the structure and the structure tag

Introduction

This section illustrates how to configure the "MotorStruc" structure and the "STR_Course_wiz1" structure tag. The structure and the structure tag are used in the example "Dynamic motor".

Procedure

1. Select "New structure type" from the structure types context menu. The structure properties dialog will be displayed.
2. Rename the structure into "MotorStruc". Click "New element" and create the internal tag "Active" of the data type BIT.
3. Click "New element" and create the internal tag "Hand" of the data type BIT.
4. Click "New element" and create the internal tag "Error" of the data type BIT. Close the dialog by clicking on the "OK" button.
5. In the navigation frame, click the plus sign in front of the icon for tag management. Select "New tag" from the internal tags context menu. Create a WinCC tag "STR_Course_wiz1" of the data type "MotorStruc".

1.5 Documentation Viewer

1.5.1 WinCC Documentation Viewer

Short description

Print jobs of the WinCC report system can be redirected to a file. In case of large amounts of data one file is created for each report page.

The **WinCC Documentation Viewer** allows to display and print these files.

1.5.2 Installing WinCC Documentation Viewer

The WinCC Documentation Viewer can be installed in two different ways:

Procedure

1. During WinCC setup, select "WinCC V7.0 complete" from the "Programs" dialog.
WinCC is installed together with SmartTools, the WinCC ConfigurationTool and WinCC Archive ConfigurationTool.

Start the WinCC Documentation Viewer by selecting "SIMATIC" > "WinCC" > "Tools".

Alternative procedure

You can also install the WinCC Documentation Viewer from the WinCC DVD.

1. Switch to the WinCC DVD directory "InstData\WinCC\setup\Products\SC_SMARTTOOLS".
2. Double-click setup.exe.
3. Select the entry "WinCC Documentation Viewer" in the "Components" dialog.
4. Click "Continue". Follow the instructions in the dialog boxes.

Note

If a WinCC project is activated, only the "emf" files of this project can be viewed and printed out. If WinCC is not active, all "emf" files can be opened and printed out with the WinCC Documentation Viewer.

1.5.3 Description

Introduction

Print jobs can be redirected to a file. In case of large amounts of data one file is created for each report page.

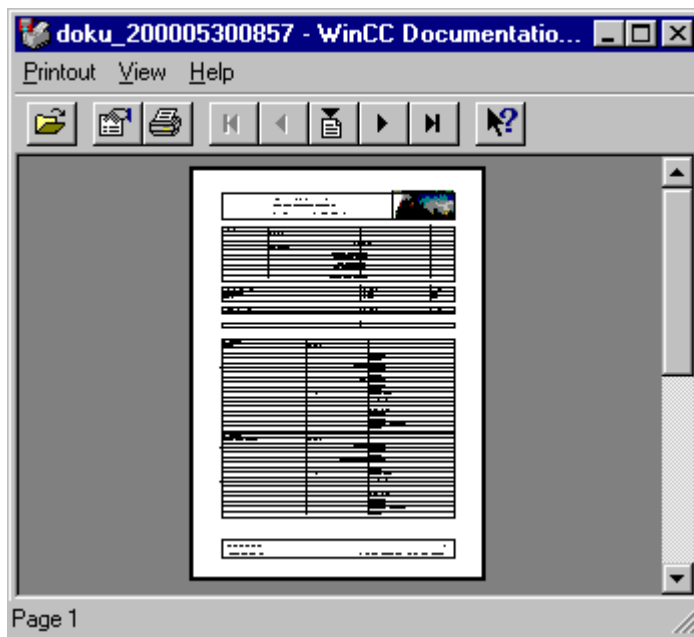
The WinCC Documentation Viewer allows to display and print these files.

Note

If a WinCC project is already activated when starting the WinCC Documentation Viewer, only the "emf" files of this project can be viewed and printed out.

If WinCC has been opened but not activated when starting the Viewer, all "emf" files can be opened and printed out with the Viewer.

Upon deactivating runtime, the Viewer is closed in any case.



The WinCC Documentation Viewer consists of three areas.

The top border of the screen contains the menu bar. The menu items are described in the direct help.

The toolbar is directly under the menu bar. Frequently used functions, such as page up and page down, are loaded as icons on this bar. The functions of the individual icons are described in the direct help.

The window shows the current document. The display can be enlarged in two steps by clicking.

The screen is limited at the bottom by the status bar displaying information on the current operation.

1.5.4 Creating the .emf file(s)

Introduction

Print jobs can be redirected to a file. In case of large amounts of data one file is created for each report page. The print output is routed to one or several .emf files. The files are stored with the name Page <nnnnnn>.emf in the path, <nnnnnn> representing a five-character consecutive number.

The path name is composed as follows: from the project path (e.g. "C:\VFSWinCC\PRT\") and <storage> + <YYYYMMDDHHMM> (YYYY = year, MM = month, DD = day, HH = hour, MM = minute).

If you enter "PDdata" in the "Storage" field, the following path structure is created for the print job within the project directory.



Procedure

1. Choose the "Project documentation setup" command from the File menu in the WinCC editors.
2. Click on the "Printer setup" tab in the "Print job properties" dialog.
3. Activate the "File (*.emf)" check box on the "Printer setup" tab. If you do not wish a simultaneous output to the printer, deactivate the "Printer" check box.
4. In the "Storage" field, enter the name of the path in which the file is to be stored. Close the dialog by clicking on the "OK" button.
5. Select the "Print project documentation" item in the "File" menu. The print output is routed to one or several .emf files. The files are stored with the name Page <nnnnnn>.emf in the path, <nnnnn> representing a five-character consecutive number.

1.6 WinCC CrossReferenceAssistant

1.6.1 WinCC CrossReferenceAssistant

Short description

WinCC CrossReferenceAssistant is a tool which searches scripts for picture names and tags and supplements the scripts so that the WinCC component **Cross Reference** finds the picture names and tags and lists them in the cross reference list.

1.6.2 Installation of the CrossReferenceAssistant

The WinCC CrossReferenceAssistant has a German, English and French user interface.

Procedure

1. During WinCC setup, select "WinCC V7.0 complete" from the "Programs" dialog.
WinCC is installed together with SmartTools, the WinCC ConfigurationTool and WinCC Archive ConfigurationTool.
- Start the WinCC CrossReferenceAssistant by selecting "SIMATIC" > "WinCC" > "Tools".

Alternative procedure

It is also possible to install the WinCC CrossReferenceAssistant from the WinCC DVD.

1. Switch to the WinCC DVD directory "InstData\WinCC\setup\Products\SC_SMARTTOOLS".
2. Double-click setup.exe.
3. Select the "CrossReferenceAssistant" entry in the "Components" dialog.
4. Click "Continue". Follow the instructions in the dialog boxes.

1.6.3 General

WinCC is able to create CrossReference lists. To ensure the tags in the function calls are recognized properly when creating these lists, WinCC was extended by a configuration rule which provides the following:

To be able to search and replace the tag and picture names used in the C actions, the script must be written as follows:

At the start of the script, all tags and picture names must be declared in two sections. Within the sections no further instructions must be entered.

The sections are structured as follows:

```
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction DMTagName
// next TagID : 1
#define ApcVarName1 "VarName1"
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction PictureName
// next PicID : 1
#define ApcPictureName1 "PictureName1"
#define ApcPictureName2 "PictureName2"
#define ApcPictureName3 "PictureName3"
// WINCC:PICNAME_SECTION_END
```

Calling the standard functions for the reading or writing of the tags must then be done through the defined tags and pictures.

```
GetTagDWord (ApcVarName1);
OpenPicture(ApcBildname1);
SetPictureName( ApcPictureName2, "PictureWindow1",ApcPictureName3);
```

If the configuration rule is not followed, no CrossReference lists can be created because the tag and picture references in the scripts cannot be resolved.

With the aid of the WinCC CrossReferenceAssistant all function calls known in the Script Management are replaced by the format described above. Only project functions, pictures and actions are converted.

The Runtime environment for the WinCC CrossReferenceAssistant is WinCC. If WinCC is not running or the project to be converted is not loaded, WinCC is started by the WinCC CrossReferenceAssistant or the project is loaded.

See also

Known functions (script management) (Page 103)

1.6.4 Known functions (script management)

The following functions are known to the Wizard by default and are implemented during conversion:

Functions with tags as parameters:

GetTagBit()
GetTagByte()
GetTagChar()
GetTagDouble()
GetTagDWord()
GetTagFloat()
GetTagRaw()
GetTagSByte()
GetTagSDWord()
GetTagSWord()
GetTagWord()

SetTagBit()
SetTagByte()
SetTagChar()
SetTagDouble()
SetTagDWord()
SetTagFloat()
SetTagRaw()
SetTagSByte()
SetTagSDWord()
SetTagSWord()
SetTagWord()

GetTagBitWait()
GetTagByteWait()

GetTagCharWait()
GetTagDoubleWait()
GetTagDWordWait()
GetTagFloatWait()
GetTagRawWait()
GetTagSByteWait()
GetTagSDWordWait()
GetTagSWordWait()
GetTagWordWait()

SetTagBitWait()
SetTagByteWait()
SetTagCharWait()
SetTagDoubleWait()
SetTagDWordWait()
SetTagFloatWait()
SetTagRawWait()
SetTagSByteWait()
SetTagSDWordWait()
SetTagSWordWait()
SetTagWordWait()

GetTagBitState()
GetTagByteState()
GetTagCharState()
GetTagDoubleState()
GetTagDWordState()
GetTagFloatState()
GetTagRawState()
GetTagSByteState()
GetTagSDWordState()
GetTagSWordState()
GetTagWordState()

SetTagBitState()

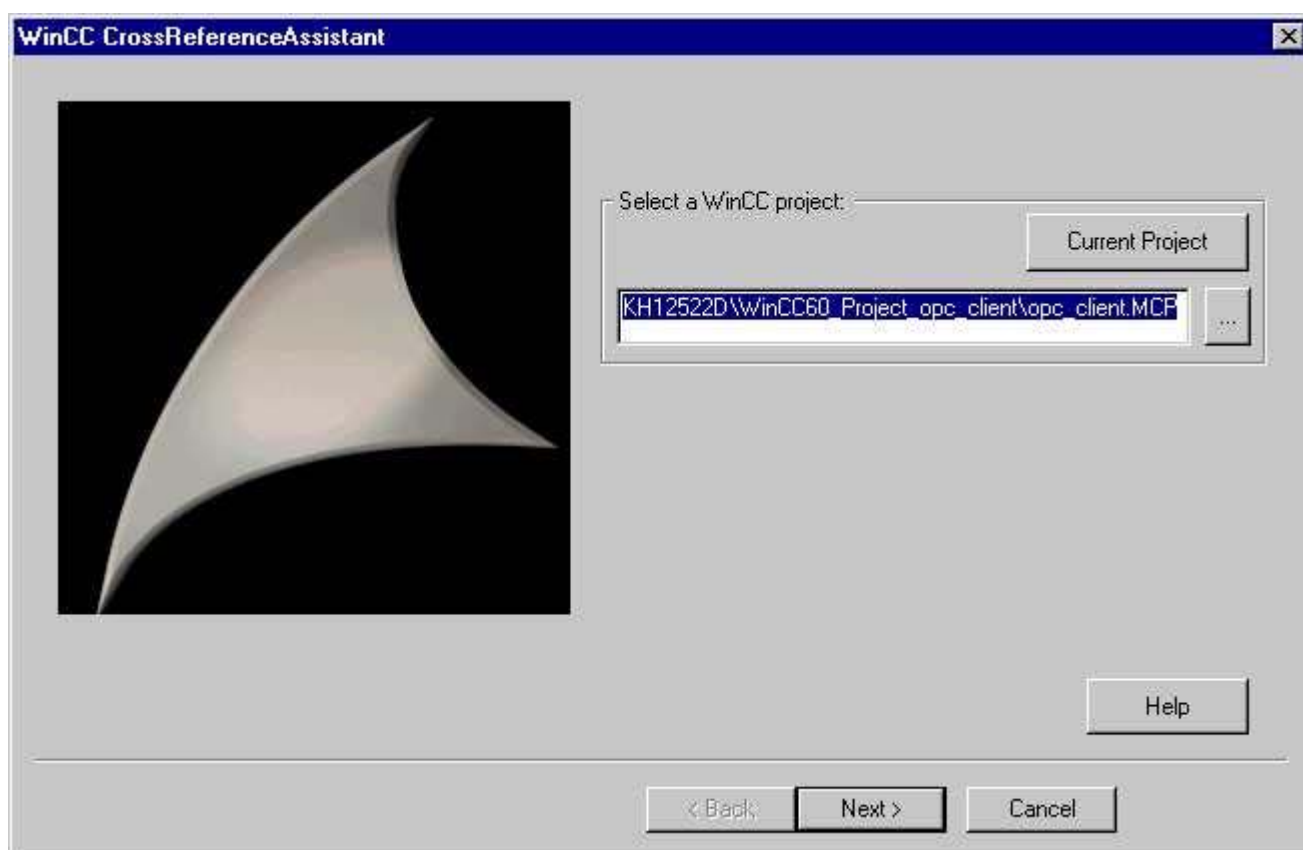
SetTagByteState()
SetTagCharState()
SetTagDoubleState()
SetTagDWordState()
SetTagFloatState()
SetTagRawState()
SetTagSByteState()
SetTagSDWordState()
SetTagSWordState()
SetTagWordState()

GetTagBitStateWait()
GetTagByteStateWait()
GetTagCharStateWait()
GetTagDoubleStateWait()
GetTagDWordStateWait()
GetTagFloatStateWait()
GetTagRawStateWait()
GetTagSByteStateWait()
GetTagSDWordStateWait()
GetTagSWordStateWait()
GetTagWordStateWait()

SetTagBitStateWait()
SetTagByteStateWait()
SetTagCharStateWait()
SetTagDoubleStateWait()
SetTagDWordStateWait()
SetTagFloatStateWait()
SetTagRawStateWait()
SetTagSByteStateWait()
SetTagSDWordStateWait()
SetTagSWordStateWait()
SetTagWordStateWait()

Functions with picture names as parameters:

SetPictureName()
GetPictureName()
GetVisible()
SetVisible()
GetLink()
SetLink()
Set_Focus()
OpenPicture()
GetLinkedVariable()

1.6.5 Project selection

Clicking "..." opens the OpenFile dialog box which enables you to select any project. When clicking "**Current project**", the WinCC CrossReferenceAssistant tries to import and display the project currently loaded in WinCC. If WinCC is not running or no project is loaded, it is started or the required project is loaded.

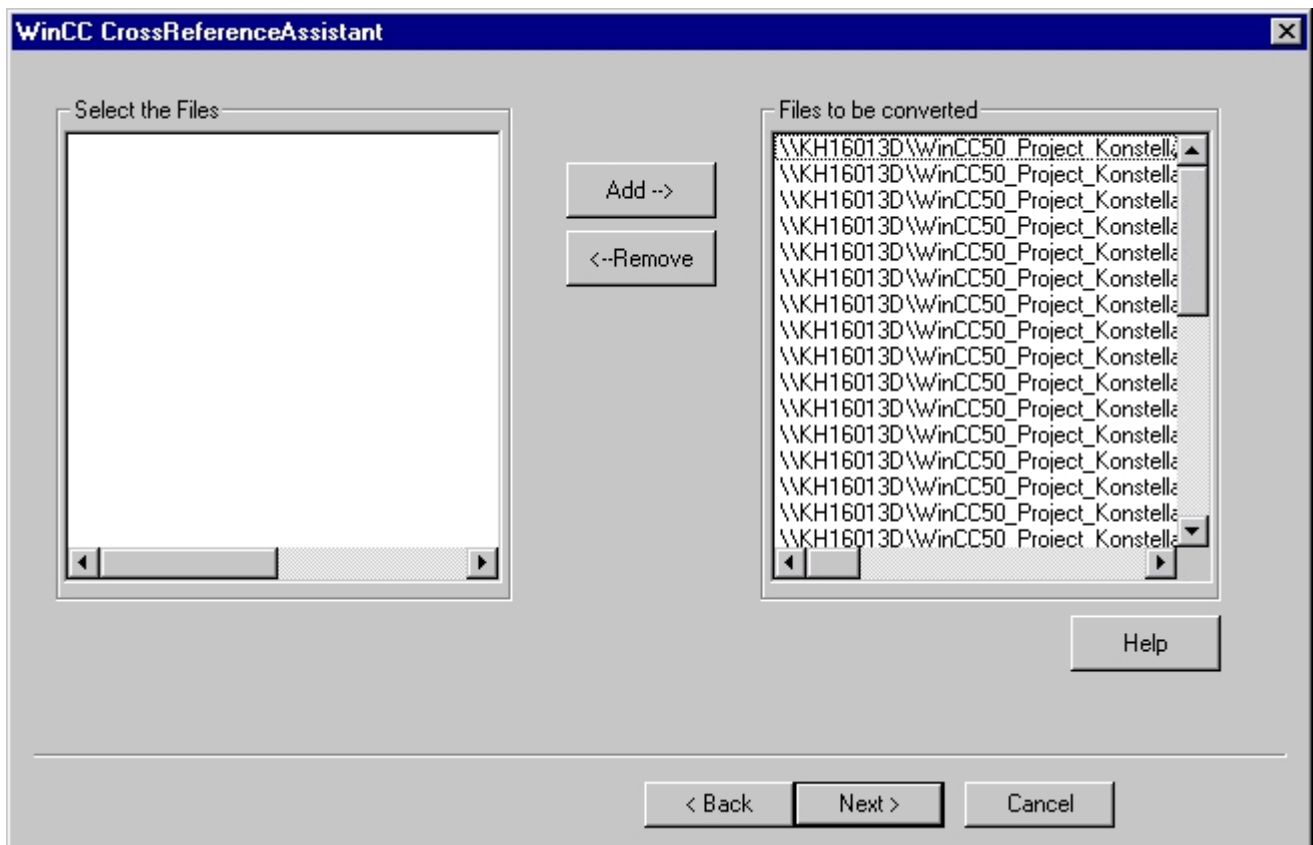
If a different project has been loaded, it is closed and the required project is loaded. This process may take some time.

As soon as text is entered in the input line at the **"Select a WinCC project"** prompt, you can click the button **"Next >"**. Then the specified project is checked to see whether it is a valid WinCC project. If the project is not valid, the focus is set on the input line and a message window opens with an explanation of the corresponding error.

"Clicking **Cancel** exits the WinCC CrossReferenceAssistant.

1.6.6 File selection

All pictures, project functions and C actions belonging to the project are displayed in the right list of the dialog box. In the default setting, all files belonging to the project are converted.



The user can decide to exclude certain files from the conversion to possibly add them later. Deleting files from the conversion list is done by (multiple) selection of the corresponding files in the **"Files to be converted"** list and clicking the **"<--Remove"** button.

Deleted files are displayed in the left list and can be added to the conversion again. To do so, these files must be selected in the **"Select files"** list. By clicking **"Add-->"** they are then added to the right list **"Files to be converted"**.

After selecting the files, click **"Next >"**. The specified files are then read and analyzed.

„Clicking "< Previous" takes you back to the project selection. Clicking "Cancel" exits the WinCC CrossReferenceAssistant.

See also

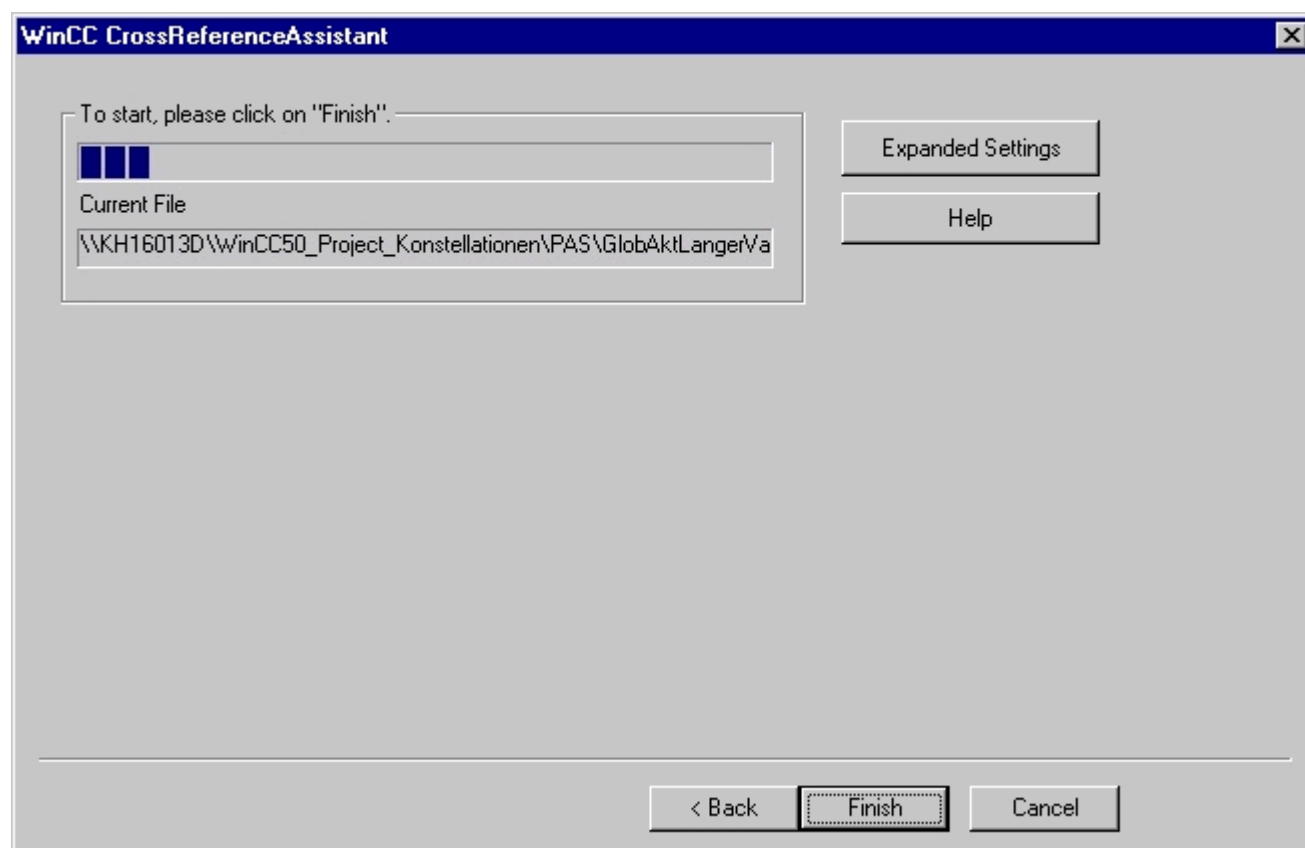
Project selection (Page 106)

1.6.7 Conversion

Introduction

The last page of the Wizards one the one hand enables you to make "Advanced Settings" (see Advanced Settings), and on the other hand to view the progress and the file currently being processed after starting the conversion.

Description



Clicking "< Back" takes you back to the File Selection. Clicking "Cancel" exits the WinCC CrossReferenceAssistant.

To start script conversion, click "Finish". After conversion begins, you cannot go back ("< Back"), or click "Advanced Settings".

During conversion, a progress bar displays what percentage of the conversion has been completed. You can also see which file is being currently converted.

The conversion is performed as follows: The scripts are examined for function calls which expect picture or tag parameters. If such a function is found in the script, the character string passed on as parameter is replaced by a define (see configuration rules).

A script management file checks which functions expect picture or tag parameters. This is why all of these functions must be entered in this file and thereby introduced to the system. The script conversion can also be used to extend the list of these functions with project functions and standard functions which also expect picture and tag parameters (advanced settings).

When the conversion is finished, a summary is displayed which provides information about how many functions, pictures and scripts in the pictures and how many tags have been converted.

If an error occurs, you can find more detailed information about the error cause by viewing the log file created during conversion. This file is located in the project directory and is called CCCrossReferenceAssistant.log.

See also

Expanded settings (Page 109)

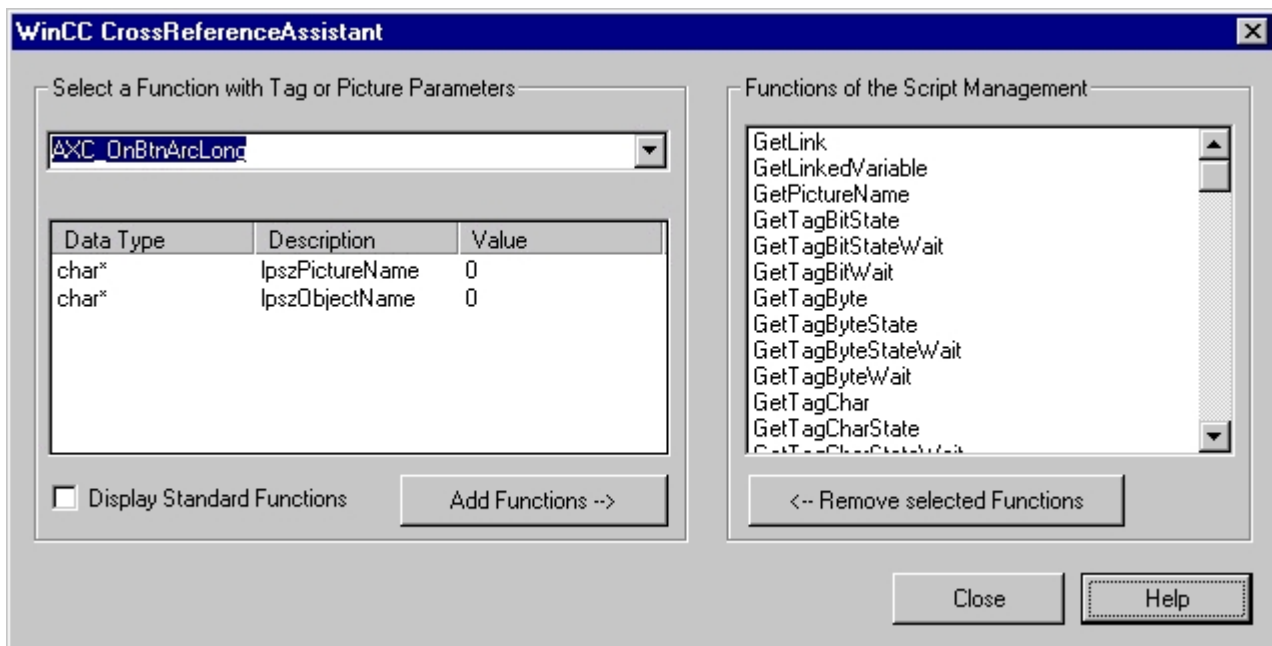
General (Page 102)

File selection (Page 107)

1.6.8 Expanded settings

If required you can activate functions you created yourself.

In the "**Select a function with tag and picture parameters:**" list, all project functions are displayed. If the "**Display standard functions**" check box is activated, the default functions in WinCC are also displayed.



The user can select a function from the function list which expects a tag or a picture as parameter at a certain position. All functions selected here are included in a project-specific script management file.

The CrossReferenceAssistant only recognizes those functions as functions with picture and tag parameters which have been introduced to the system as such. To make sure calls of customized functions expecting tag parameters can be adapted in accordance with the configuration rules, these must be included in the script management in the course of the conversion.

The **"Functions of the script management"** dialog box displays all functions already added to the script management. When this dialog is displayed, the standard and project configuration files are read out and the common contents of both files are displayed.

To name a function expecting a tag or picture parameter, you must first select it from the **"Select a function with tag or picture parameters:"** combination field.

Using the parameter list you can then define whether the special parameter represents a tag or a picture. When clicking "..." a popup menu opens which lets the user choose whether the selected parameter is a tag or a picture.

This process must be repeated for all parameters to which one of the criteria applies.

"Add function -->" confirms the input and adds the selected function to the list on the right side of the dialog box. If you make a mistake, you can undo it by selecting the functions to be deleted in the **"Functions of the script management"** list and then removing them from the list by clicking **"<-- Remove selected functions"**.

When clicking **"Close"** the group information is written to the configuration files, the modified information is taken into account during conversion and the dialog is closed.

See also

General (Page 102)

1.7 WinCC Communication Configurator

The WinCC Communication Configurator (CCCommunicationConfigurator.exe) is a tool allowing to set WinCC communication parameters to the available network environment in a simple manner.

It is expedient to use WinCC Communication Configurator whenever there is no Ethernet LAN with a transfer rate of 100 MBit/s. We also recommend using the configurator when the connection is sporadically unstable as a result of high-load situations (such as no connection to the data servers, I/O fields without a displayed value).

The WinCC communication is configured with standard parameters so that it reacts very sensitively to communication errors, e.g. in order to report any faults which occur to the user quickly or also to ensure a short "failover" time to the redundant server in the case of a client computer.

On a network having low transfer rates or a high network/CPU loading, the stability of the logical network connections of WinCC is affected by this error-sensitive behavior, since the expected feedback times cannot be achieved in the lower-level mechanisms of lifebeat monitoring.

The Communication Configurator adapts the communication parameters to the existing scenario so that an optimum balance is reached between error sensitivity and connection stability.

Note

If the WinCC Communication Configurator is used in this manner, this must be done on the WinCC client and on the WinCC server.

The Communication Configurator modifies only settings for WinCC communication, but not the parameterizations of the operating system communication links.

Field / option	Description
100 MBit/s	Applicable to Ethernet LANs with a transfer rate of 100 MBit/s (default setting).
10 MBit/s	For Ethernet LANs with a transfer rate of 10 MBit/s.
1 MBit/s	For networks with a transfer rate of 1 MBit/s.
0.1 MBit/s	For networks or communications links with a transfer rate of 0.1 MBit/s. This setting is suitable for links using ISDN (MultiLink), ISDN and modem.
Check box "Server pings client"	Checks connection to client using server.
"Default" button	Sets setting to default value "Ethernet LAN (100 MBit/s)"

Note

We recommend the use of clients with a transfer rate of at least 128 KBit/s.

1.8 WinCC Configuration Tool

1.8 Resources

1.8.1 Introduction

Introduction

The WinCC configuration tool provides a simple, high-performance option for configuring bulk data in WinCC. Microsoft Excel serves as the user interface. This enables users to create a WinCC project in Microsoft Excel and use the advantages Microsoft Excel offers with regard to operation.

This chapter provides an overview of the functionality of the WinCC configuration tool and of its operation.

Overview

The WinCC configuration tool provides a simple, high-performance option for configuring bulk data in WinCC.

Microsoft Excel serves as the user interface. Due to its table structure, it is particularly suitable for the handling and representation of data from WinCC. It also offers a wide range of editing options (including autofill, etc.). Moreover, experienced users can extend the edition options by creating VBA programs (macros).

The configuration tool allows you to set up a new WinCC project and to configure it from the very beginning from within Excel. In addition, you can also read in existing WinCC projects and process them in Excel. The local configuration data are used for this purpose. Only clients without a project of their own can be read in remotely. The computer name of the client on which the ConfigurationTool is running must be entered in the project on the server. The right for "Configure remote" must have been assigned.

The configuration is done in a special type of Excel workbook known as a WinCC project folder. It contains various types of worksheets that are used to configure specific types of WinCC objects. The configuration tool allows you to configure data from the data manager, alarm logging, tag logging and the text library.

Note

In the ConfigurationTool, you can only edit links or tags of channels that are included by default in WinCC.

1.8.2 System requirements

Introduction

The configuration tool has the same requirements as WinCC and Microsoft Excel. You have however the option to use the configuration tool without WinCC. In this case, you can of course not write data to WinCC.

System requirements

- Windows XP SP3 / Windows Server 2003 (R2) SP2 / Windows Vista SP1
- Microsoft Excel XP, Office 2003, Office 2007. Visual Basic for applications must be installed.
- Internet Explorer 6 or 7

Note

You will need the "Office Assistant" in Excel, so that the warning messages of the configuration tool will be output.

See also

Example of a quantity structure in the configuration tool (Page 231)

1.8.3 Installation of the ConfigurationTool

The WinCC Configuration Tool can be installed in two different ways.

Procedure

1. During WinCC setup, select "WinCC V7.0 complete" from the "Programs" dialog. WinCC is installed together with SmartTools, the WinCC ConfigurationTool and WinCC Archive ConfigurationTool.

Start the WinCC ConfigurationTool by selecting "SIMATIC" > "WinCC" > "Tools".

Alternative procedure

It is also possible to install the WinCC ConfigurationTool from the WinCC DVD.

1. Switch to the WinCC DVD directory "InstData\WinCC\setup\Products\ConfigurationTool".
2. Double-click setup.exe.
3. Follow the instructions in the dialog boxes.
The WinCC ConfigurationTool is installed.

1.8.4 Interface

1.8.4.1 Interface

Introduction

The configuration tool features a number of new functions. It comes with its own toolbar and system menu entry. New menu options are also added to the Excel menu. The following chapters describe the above menu options and functions in more detail.

See also

Status bar (Page 118)

Toolbar (Page 114)

Pop-up menu (Page 117)

Dropdown menu (Page 115)

1.8.4.2 Toolbar

Introduction

The toolbar of the configuration tool contains the following elements.



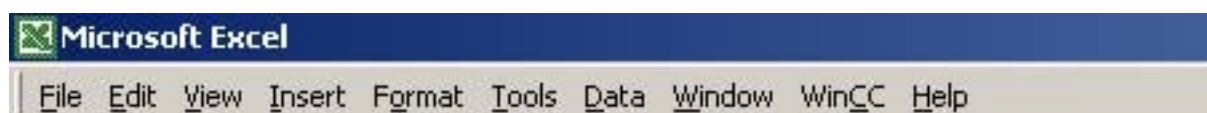
	Create project folder The "New project folder" wizard is opened. This wizard allows you to set up new project folders.
	Switching languages Opens the "Select language" dialog.
	Help Opens the online help of the configuration tool.
	Create WinCC project The "New WinCC project" wizard is opened. With this wizard, you can create a new WinCC project based on an existing project folder. The project folder is then linked to the new WinCC project.
	Establish project connection This option is only available, if the active project folder is already assigned to a WinCC project and the WinCC project is not active.

	Adding tables The "Add table" dialog is opened. In this dialog, you can add new worksheets to the project folder.
	Write to WinCC The "Write" dialog is opened. In this dialog, you can write all data from the project folder to WinCC.

1.8.4.3 Dropdown menu

Introduction

The dropdown menu contains the options available under the respective menu headings. The configuration tool adds a separate menu item to the Excel menu.



Elements of the dropdown menu

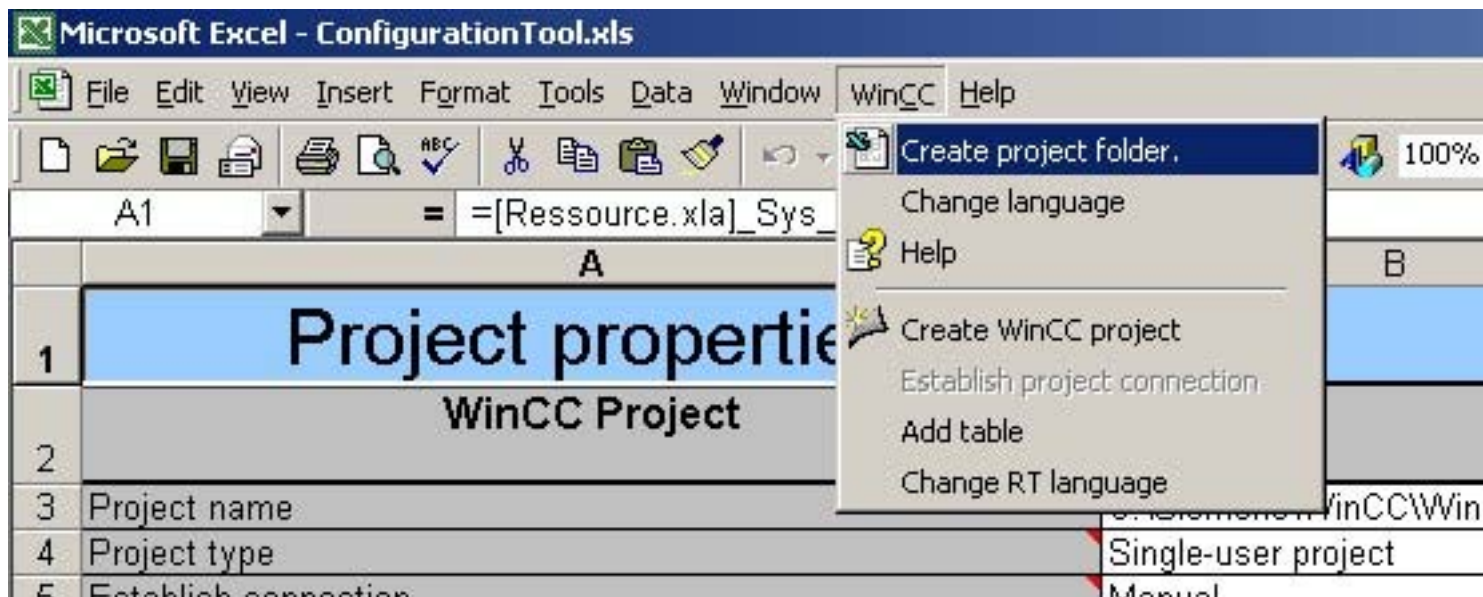
Depending on the connection status and project folder type, different options are available in the dropdown menu.

No project folder active



Create project folder	The "New project folder" wizard is opened. This wizard allows you to set up a new project folder.
Switching languages	Opens the "Select language" dialog.
Help	Opens the online help of the configuration tool.

A project folder linked to a WinCC project is active



Create WinCC project	The "New WinCC project" wizard is opened. With this wizard, you can create a new WinCC project. The project folder is then linked to the new WinCC project.
Establish project connection	Establishes a link between the project folder and the associated WinCC project.
Adding tables	The "Add table" dialog is opened. In this dialog, you can add new worksheets to the project folder.
Change RT language	Opens the "Change RT language" dialog.
Write objects	Depending on the type of worksheet that is active, additional menu options might be available. These are primarily menu options used to write objects that can be configured on the active worksheet.

Configuring several languages in the configuration tool

In the WinCC configuration tool you can change the WinCC interface language and the runtime language just as in WinCC:

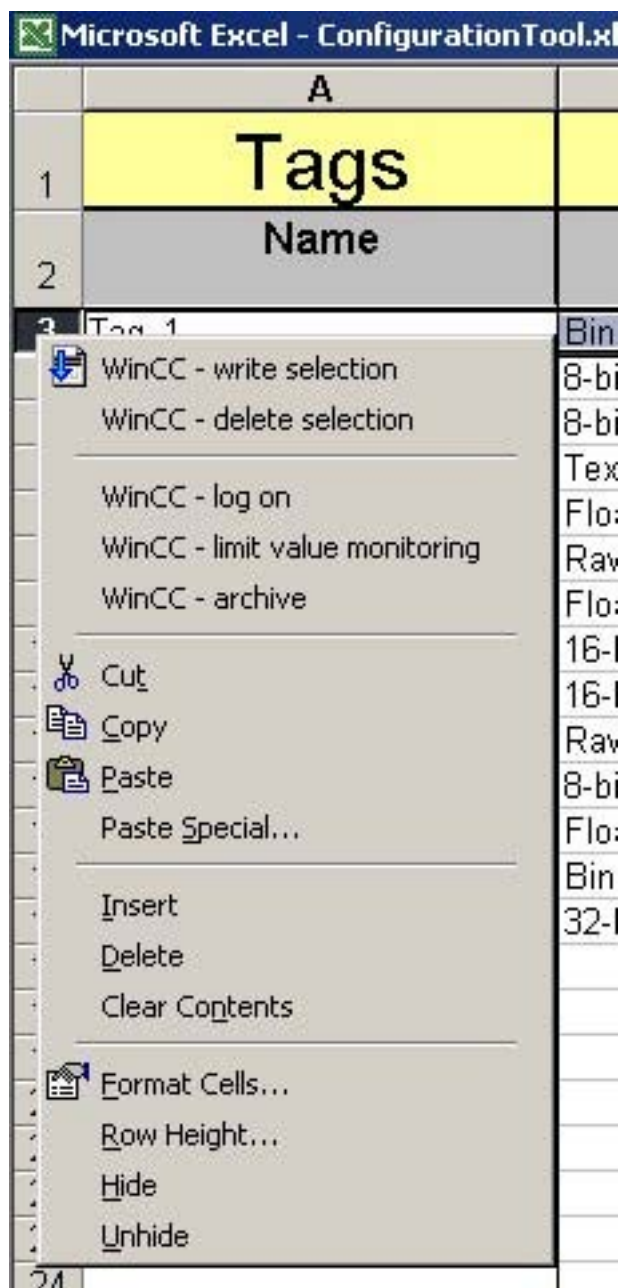
- To change the interface language, select the menu option "Change language".
- To change the runtime language, select the menu option "Change RT language". The menu item is available only if the project folder is linked to a WinCC project.

For example, the user text blocks and the message texts depend on the runtime language. When switching the runtime language, the texts are displayed in the selected language and you can extend them.

1.8.4.4 Pop-up menu

Introduction

The configuration tool adds two additional menu options to the row pop-up menu of Excel. The row pop-up menu is only available if entire rows in the worksheet are selected.



WinCC - write selection	This menu item is only available, if there is a link to the associated WinCC project. With the menu option "WinCC - write selection", all selected objects are written to WinCC. This is the only option allowing to write individual objects to WinCC.
WinCC - delete selection	This menu option "WinCC - delete selection" is used to delete selected objects from the project folder and WinCC. Only this menu allows deleting individual objects from WinCC.

1.8.4.5 Status bar

Introduction

In the configuration tool the status bar of Excel shows information relating to WinCC. If a project is opened in WinCC, the status bar shows its path and the project name of the WinCC project currently open in WinCC.

1.8.5 Operation of the WinCC configuration tool

1.8.5.1 Operation of the configuration tool

Introduction

Generally, all functions offered by Excel can be used without restriction. Exceptions here are the sort and delete functions as they are used by the configuration tool itself.

The configuration tool assists you in the entry of data, providing you with a range of useful function.

Dropdown list boxes

Dropdown list boxes can be accessed by double-clicking the respective row in the table. Many parameters are entered as text. By entering parameters using dropdown list boxes, you ensure a valid value is entered.



	A	B
1	Tags	
2	Name	Data type
3	Tag_1	Binary tag
4	Tag_2	8-bit value with sign
5	Tag_3	8-bit value with sign
6	Tag_4	8-bit value without sign
7	Tag_5	16-bit value with sign
8	Tag_6	16-bit value without sign
9	Tag_7	32-bit value with sign
10	Tag_8	32-bit value without sign
11	Tag_9	Floating point number 32-bit IEEE

Autofilling

With autofill, which is an Excel standard function, you can enter many objects in a very efficient way. The configuration tool supports the autofill operation in two different ways. You can specify the autofilling method you wish to use in the project properties table.

Microsoft Excel - ConfigurationTool.xls		
	A	B
1	Project properties	
2	WinCC Project	
3	Project name	C:\Siemens\WinCC\Win
4	Project type	Single-user project
5	Establish connection	Manual
6	Connection status	Connected
7		
8	Data input	
9	Use default values	Yes
10		
11	Add-in	
12	Max. number of lines	
13		

- **Autofilling with default values**
Autofilling with default values has the advantage that the parameters provided automatically by the configuration tool can be set in the respective default value tables. In the tag table, for example, you need only select the tag name and the data type of the tag; the other parameters are then loaded automatically from the default value table. If the values in the default value table are chosen with care, this tool can help you save a considerable amount of time.
- **Autofilling without default values**
Autofilling without default values has the advantage that the parameters fields are not automatically filled by the configuration tool. Instead, each value in the cell is checked and is not overwritten, if it is valid.
This option is for example used for copying entire objects.

Note

For more information on the autofill tool, please refer to the Microsoft Excel help.

Unique name

The configuration tool checks object names for uniqueness. If an already used name is detected, you can configure the system to automatically generate a new unique name. The original name is complemented with a consecutive number.

If you wish to generate object names that are numbered consecutively, we recommend using the Excel standard function designed for this purpose (enter first name, e.g. Tag_1 and then autofill).

Input Prompts

The configuration tool checks each data input. If invalid data is entered, it is automatically corrected. Each alteration performed by the configuration tool is commented by an input prompt. This is displayed by the Office Assistant. The input prompt need not be acknowledged but disappears automatically with the next input.

Note

If the Office Assistant is not installed, it is not possible to use the input prompt.

1.8.5.2 Creating a new project folder

Creating a new project folder

Introduction

The configuration data of a WinCC project is managed in a project folder. It contains various types of table sheets which serve to configure specific types of WinCC objects. A new project folder can be created using the toolbar or drop-down menu.

Procedure

1. Click the  button in the toolbar. The "New project folder" wizard is opened.
2. The following 3 options are available on the first page of the wizard:
 - No connection**
A project folder will be created which is not assigned to any WinCC project.
 - Establish connection to new project**
A new project folder is created. A new WinCC project is created and assigned to the project folder.
 - Establish connection to existing project**
A new project folder is created. It is assigned to an existing WinCC project. The WinCC project data is read out.



3. Select the required option and press the "Complete" button. For the options "Establish connection to new project" and "Establish connection to existing project", press the "Continue" button.
4. The procedures for this are described in the chapters below.

See also

- Creating a project folder with a connection to an existing WinCC project (Page 124)
- Creating a project folder with a connection to a new WinCC project (Page 126)
- Creating a project folder without a connection (Page 123)

Creating a project folder without a connection

Introduction

The configuration tool allows you to set up a project folder that is not assigned to a WinCC project. A project folder that is not linked to a WinCC project can be configured without any restrictions. It is however not possible to write data to WinCC. In order to write configured data to WinCC, you must create a new WinCC project.

Procedure

1. Click the  button in the toolbar. The "New project folder" wizard is opened.
2. Select the option "No connection". Click "Complete" to close the wizard.



See also

Creating a new WinCC project (Page 129)

Creating a project folder with a connection to an existing WinCC project

Introduction

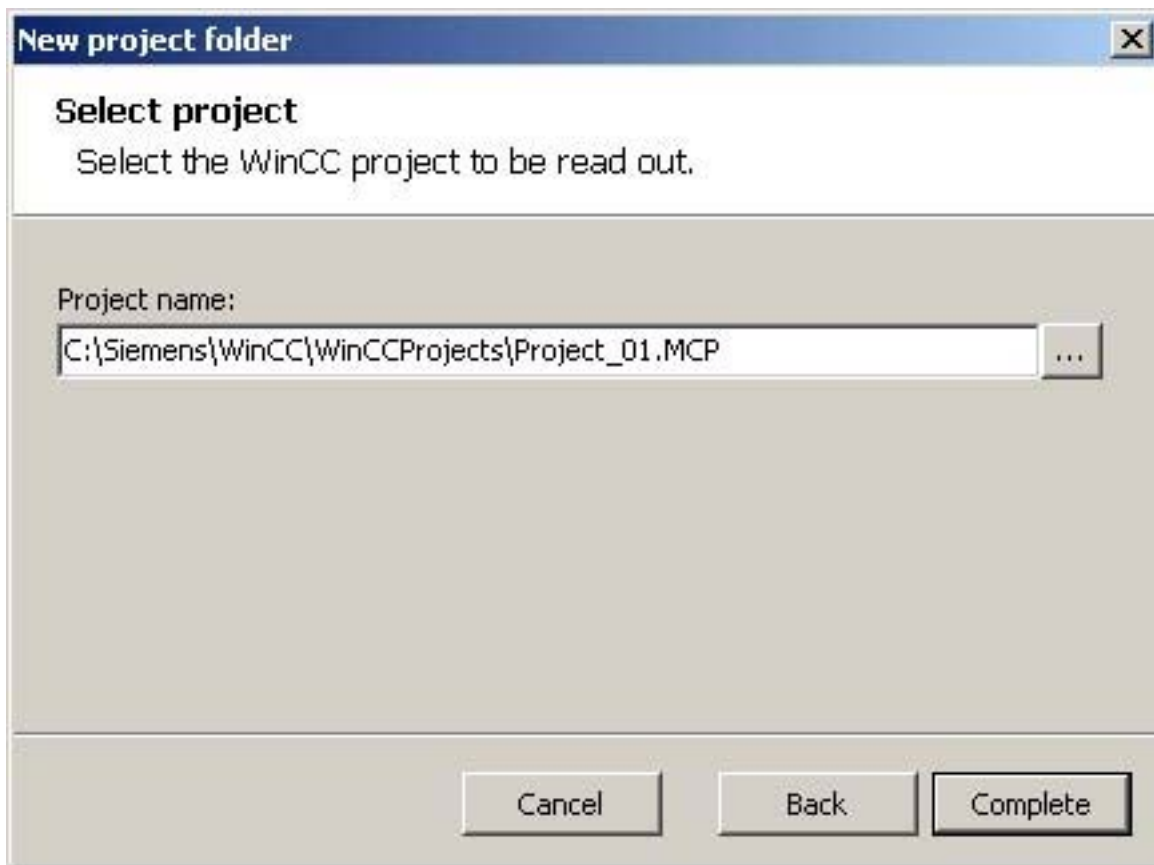
The configuration tool offers you the option to create a new project folder and to assign it to an already existing WinCC project. The data of the WinCC project are then read to the new project folder.

Procedure

1. Click the  button in the toolbar. The "New project folder" wizard is opened.
2. Select the option "Establish connection to existing project". Click "Continue" to open the second page of the wizard.



3. Select the desired WinCC project. Click "Complete" to close the wizard.



Note

When reading large WinCC projects, Microsoft Excel might display an OLE message box. This box is automatically acknowledged by the configuration tool. While the OLE message box is displayed, performance might be impaired.

Creating a project folder with a connection to a new WinCC project

Introduction

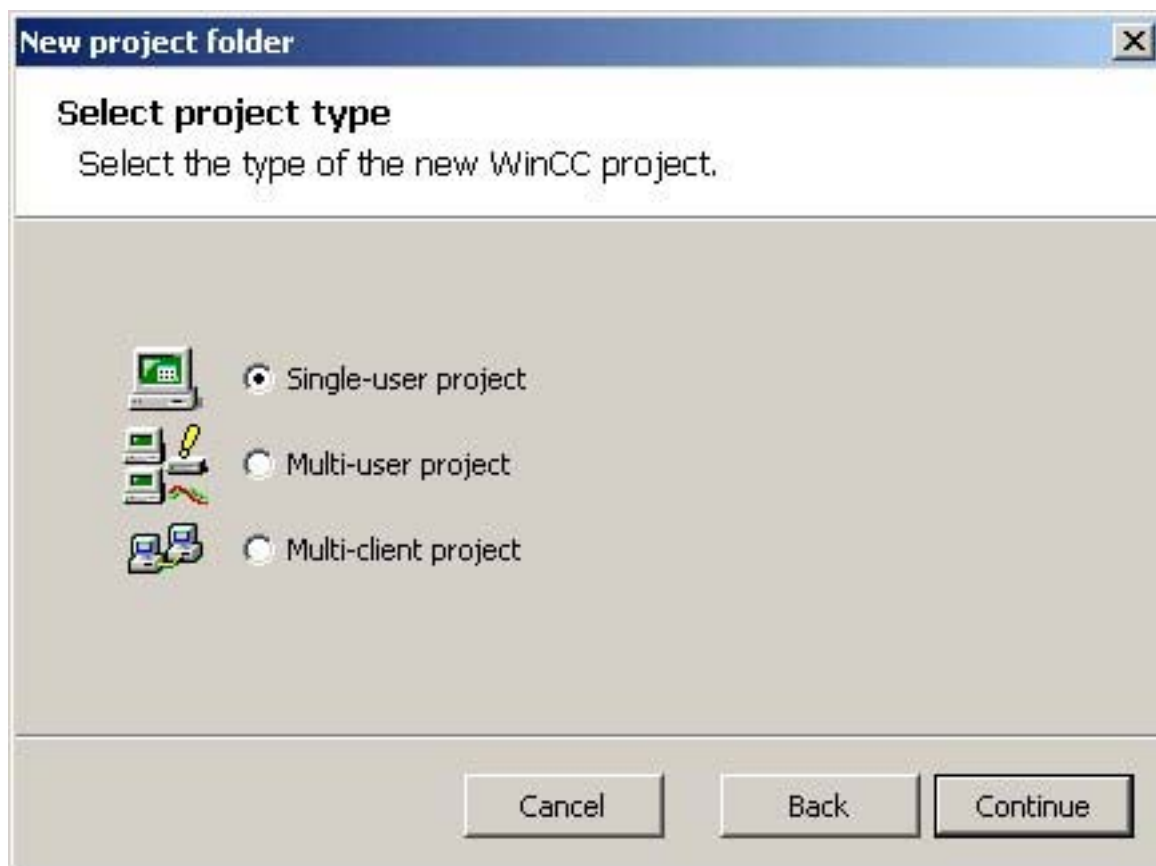
The configuration tool offers you the option to create a new project folder and to assign it to a new WinCC project. The data already available in the project are read.

Procedure

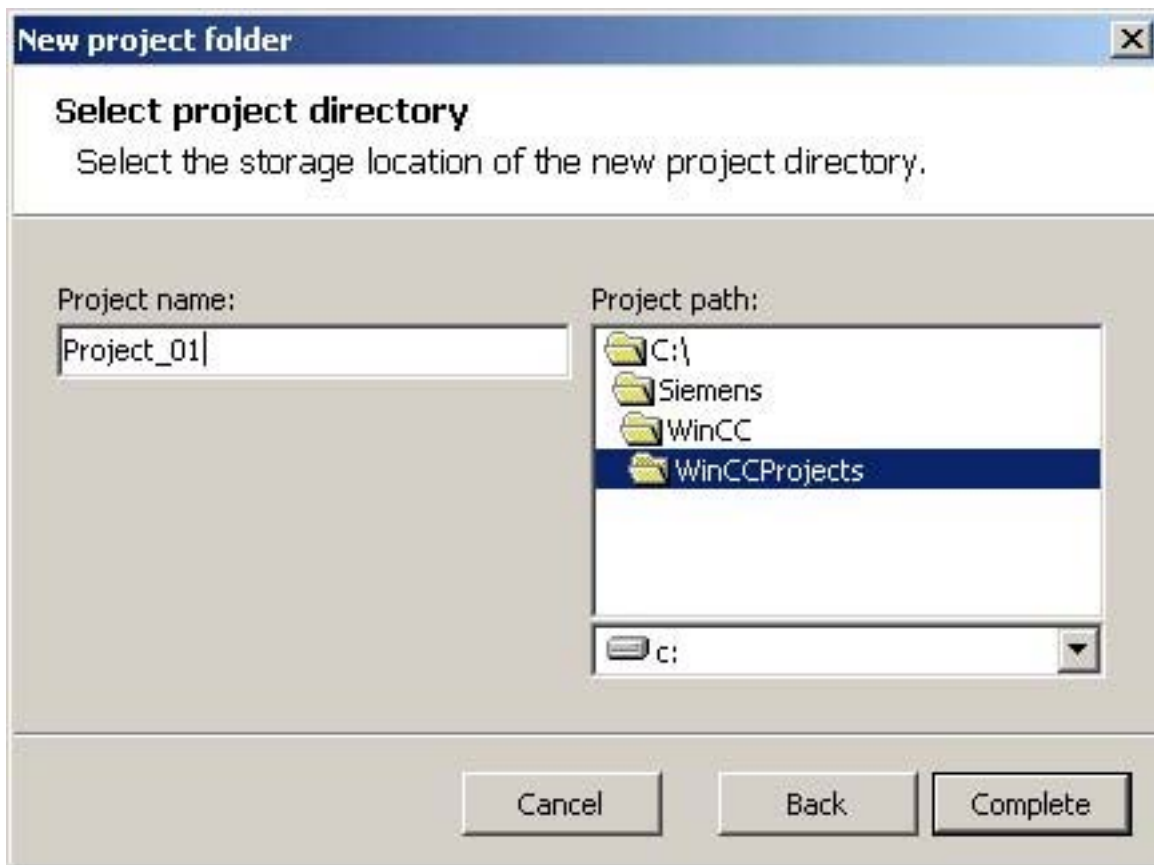
1. Click the  button in the toolbar. The "New project folder" wizard is opened.
2. Select option "Establish connection to new project". Click "Continue" to open the second page of the wizard.



3. Select the type of the new WinCC project. Click "Continue" to open the third page of the wizard.



4. On the third page of the wizard, select the location in which the new WinCC project is to be saved. Enter a name for the new WinCC project. Also specify the folder in which the project folder is to be set up. Click "Complete" to close the wizard.



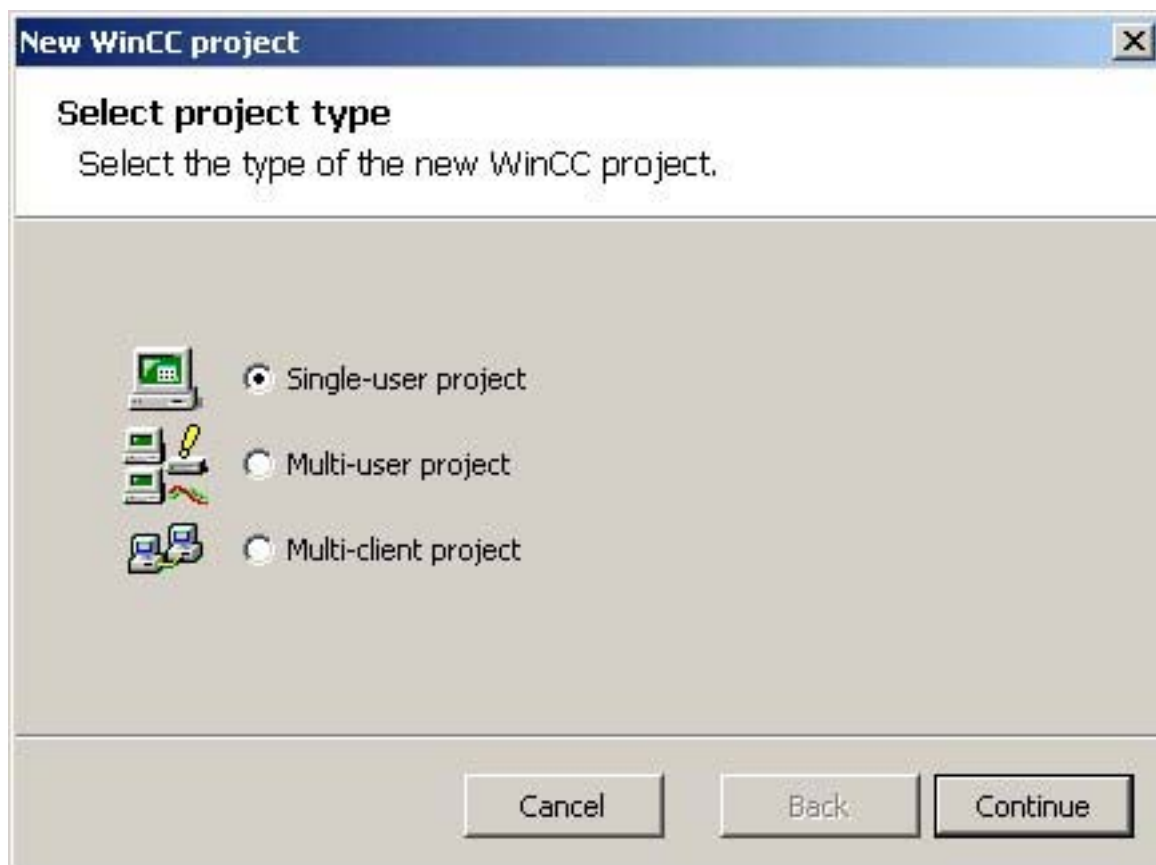
Creating a new WinCC project

Introduction

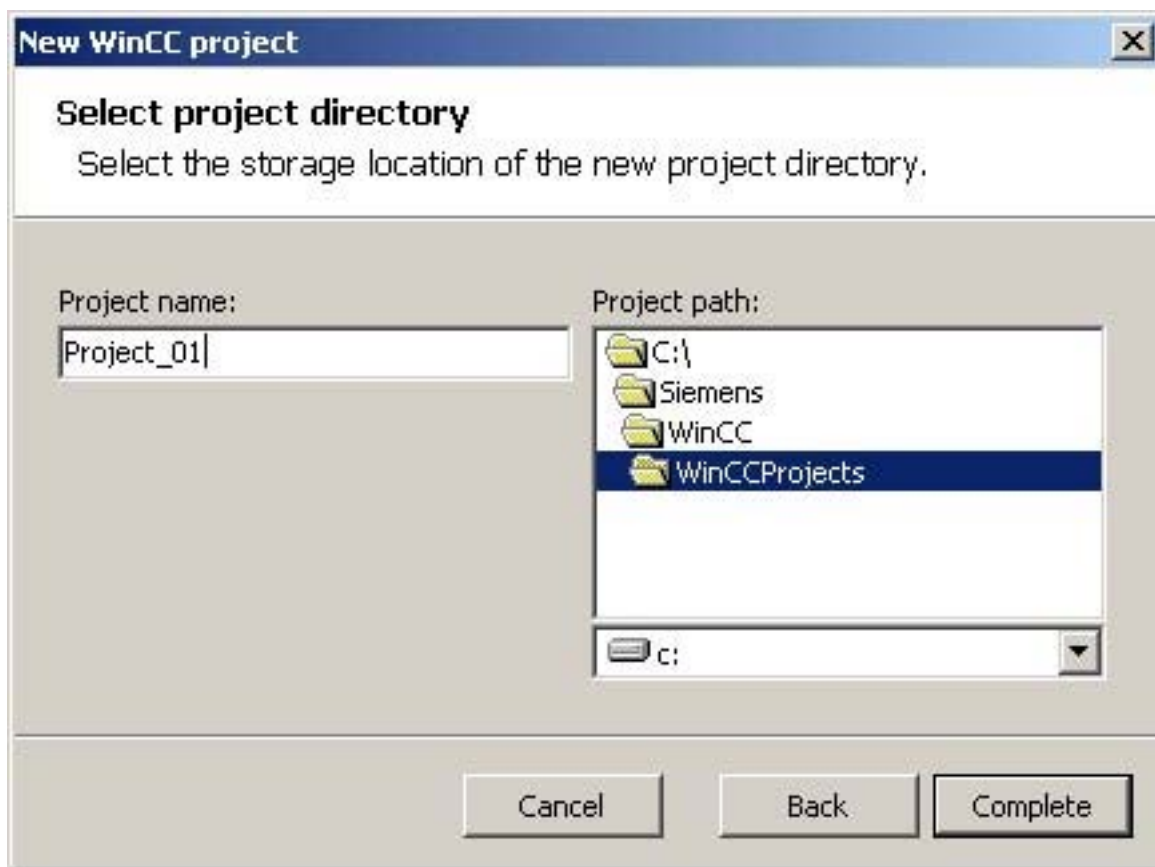
The configuration tool allows you to set up a new WinCC project for an existing project folder. This WinCC project is assigned to the project folder.

Procedure

1. Click the  button in the toolbar. The "New project folder" wizard is opened.
2. Select the type of the new WinCC project. Click "Continue" to open the second page of the wizard.



3. On the second page of the wizard, select the location in which the new WinCC project is to be saved. You must enter the name of the new WinCC project. Also specify the folder in which the project folder is to be set up. Click "Complete" to close the wizard.



1.8.5.3 Worksheets

Worksheets

Introduction

When a new project folder is created, at least one copy of each necessary sheet type is created. You also have the option to add additional sheets.

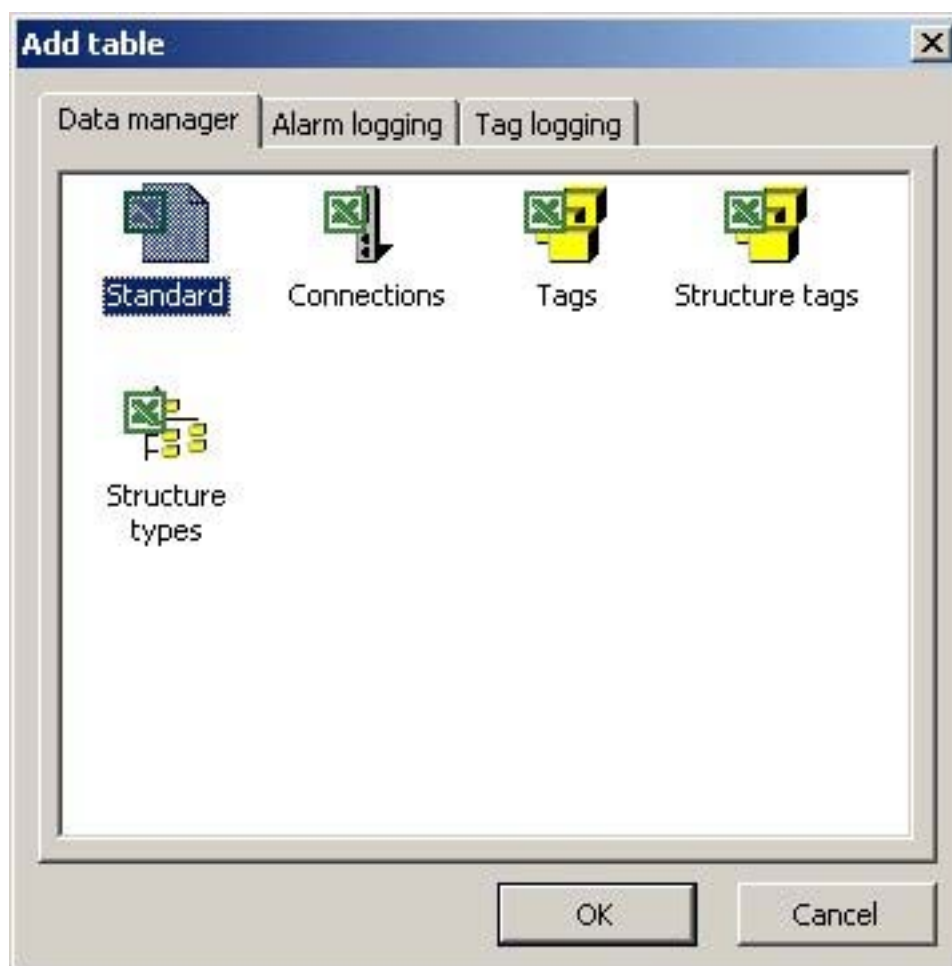
Structure of a worksheet

In the table, one row each is assigned to each object to be configured. The colored lines are headings, and the gray cells are the parameter headings. The data area consists of the cells below the parameter headings. The cells outside the data area are available for use as required.

Microsoft Excel - ConfigurationTool.xls				
	A	B	C	
1	Tags			
2	Name	Data type	Length	For
3	Tag_1	Binary tag	1	
4	Tag_2	8-bit value with sign	1	
5	Tag_3	8-bit value without sign	1	
6	Tag_4	Text tag, 8-bit font	0	
7	Tag_5	Floating point number 32-bit IEEE	4	
8	Tag_6	Raw data type	0	
9	Tag_7	Floating point number 64-bit IEEE	8	Dou
10	Tag_8	16-bit value with sign	2	Sho

Creating a new sheet

Open the "Add table" dialog by clicking the corresponding button in the toolbar or select it from the dropdown menu. Select the type of table required in the dialog. Click "OK" to add the table to the project folder.



Operation of the "Project Properties" sheet

Introduction

The "Project Properties" sheet contains information on the WinCC project. In addition, it is possible to define settings here that affect the entire project folder.

Microsoft Excel - ConfigurationTool.xls		
	A	B
1	Project properties	
2	WinCC Project	
3	Project name	C:\Siemens\WinCC\Win
4	Project type	Single-user project
5	Establish connection	Manual
6	Connection status	Connected
7		
8	Data input	
9	Use default values	Yes
10		
11	Add-in	
12	Max. number of lines	
13		
14	Create message	
15	Delete existing messages	Yes
16	Display dialog	Yes
17		
18	Create limit value monitoring	
19	Delete existing limit values	Yes
20	Display dialog	Yes
21		
22	Create archive tags	
23	Delete existing archive tags	Yes
24	Display dialog	Yes
25		
26	Alarm logging	
27	Check bits for use	Yes
28	Display request for modification of all identical status texts	No

WinCC project

Project name	Path and project file of the connected WinCC project.
Project type	Type of the project.
Establish connection	Choose between "Manual" and "On opening". If "Manual" is selected, use the toolbar button or dropdown menu to establish a connection to the associated WinCC project after the project folder is opened. If "On opening" is selected, the connection to the associated WinCC project is automatically established when the project folder is opened.
Connection status	Indicates whether the associated WinCC project is open or not. Data can only be written to WinCC, if a connection is established to the project.

Data input

Use default values	It is possible to define whether default values should be used or not. If this is the case, the default values from the default value table are used during configuration. If not, the default values are not used. The use of default values increases performance considerably.

Add-In

Maximum number of lines	This is used to define the number of lines after which a new sheet should be created when reading out data. Limiting the number of lines improves the performance as Excel requires a considerably longer time to create new data from a certain number of lines.

Create Messages

Deleting existing messages	Specify here whether the already existing messages are to be deleted when messages from the tag table are generated for the selected tags.
Display dialog	Define here whether you wish to use the dialog for the generation of messages from the tag table. If not, the settings defined in the "Alarm Logging Default Values" sheet are used.

Creating limit value monitoring

Delete existing limit values	Define here whether existing limit values for the selected tags are to be deleted when limit values are generated from the tag table.
Display dialog	Define here whether the dialog for the creation of limit values from the tag table is to be used. If not, the settings defined in the "Alarm Logging Default Values" sheet are used.

Creating archive tags

Delete existing archive tags	Define here whether existing archive tags for the selected tags are to be deleted when archive tags are created from the tag table.
Display dialog	Specify whether the dialog is to be used to create archive tags from the tag table. If you deactivate the dialog, the dialog will open one time. Enter a process value archive in the dialog as the default archive. Check the "Retain settings" option so that the dialog will not longer be opened in the future. The settings defined in the "Tag Logging Default Values" sheet will be used.

Alarm logging

Check bits for use	Define here whether the WinCC configuration tool is to check the tag bit combination in the alarm logging.
Prompt to display request for modification of all status texts used	This option is used to define whether, following modification of a status text in a message category, a prompt is to be displayed requesting the modification of all identical status texts.
Prompt to display request for modification of all message texts used	This option is used to define whether, following modification of a message text, a prompt is to be displayed requesting the modification of all identical message texts.
Delete unused texts	This option is used to define whether unused texts are to be automatically deleted from the text library when deleting objects from the alarm logging system.
Delete limit values when deleting your single message	This option is used to define whether limit values are also to be deleted when deleting the set single messages or whether a default message number is to be set.

Comments

Display comments	Define here whether the comments are to be displayed on the "Project Properties" sheet.

Data manager

Data manager

Introduction

Connections, tags, structure tags and structure types can be configured on the data manager sheets. It is also possible to define default values for tags in the "Data Manager Default Values" sheet. The procedure to configure data in the data manager is explained in the following chapters.

See also

Operation of the "Connections/Groups" sheet (Page 140)

Operation of the "Structure Tags" sheet (Page 145)

Operation of the "Structure Types" sheet (Page 147)

Operation of the "Tags" sheet (Page 142)

Operation of the "Data Manager Default Values" sheet (Page 137)

Operation of the "Data Manager Default Values" sheet**Introduction**

The configuration tool offers the option of pre-assigning default values to parameters of newly created objects. These default values can be defined on the "Data Manager Default Values" sheet.

The settings defined here are also used for the structure type elements.

Procedure

One row each is available for the data types that can be set for tags. Define the settings for the relevant data type in this row.

Microsoft Excel - ConfigurationTool.xls			
	A	B	
1	Default values	Data manager	
2	Tags		
3	Data type	Length	For
4	Binary tag		1
5	8-bit value with sign		1
6	8-bit value without sign		1
7	16-bit value with sign		2
8	16-bit value without sign		2
9	32-bit value with sign		4
10	32-bit value without sign		4
11	Floating point number 32-bit IEEE		4
12	Floating point number 64-bit IEEE		8
13	Text tag, 8-bit font		0
14	Text tag, 16-bit font		0
15	Text reference		4
16	Raw data type		0

Deactivating default values

The use of the default values can be deactivated on the "Project Properties" sheet. If the default values are deactivated, all parameters of a newly created object are checked for validity and corrected, if necessary (e.g. blank rows). This, however, impairs performance during the creation of objects. The use of default values is therefore activated by default.

Table structure

The following table lists all parameters that must be defined for tags in the "Data Manager Default Values" sheet. Parameters with dropdown list boxes are identified by a  icon.

Column	Short description
Data type	All data types available for a tag. This column is write-protected.
Length	Length of the tags in WinCC in bytes. For most data types, this value cannot be edited.
 Type Conversion	Type conversion of the tag. This can only be adjusted for external tags. Not all data types are assigned type conversion features.
 Connection	Connection of the tag.
 Group	Group of the tag. This parameter is optional.
Address	Address of the tag. This can only be adjusted for external tags. The structure of the address depends on the communication driver set for the connection. For the structure of the address, refer to the tags in the parameters column in the WinCC Explorer. The address is currently not checked for validity. All entries are thus considered valid.
 Update	Update of the tag. This option is only available for internal tags. You can choose between the options "for entire project" and "local computer related".
 Linear scaling yes/no	Specify here whether linear scaling is to be used. Linear scaling can only be chosen for external tags. Not all data types support linear scaling.
Linear scaling process from	Scaling range in the process (source scaling).
Linear scaling process to	Scaling range in the process (source scaling).
Linear scaling tag from	Scaling range of the tag (scaled representation).
Linear scaling tag to	Scaling range of the tag (scaled representation).
Upper limit	Upper limit value of the tag.
Lower limit	Lower limit value of the tag.
Start value	Start value of the tag.
Substitute value	Substitute value of the tag. Substitute values can only be configured for external tags.
 Upper limit substitute value	Substitute value to be used if the actual value is above the upper limit.
 Lower limit substitute value	Substitute value to be used if the actual value is below the lower limit.

Column	Short description
 Substitute start value	The start value is to be replaced with the substitute value.
 Error substitute value	The substitute value is to be used in the event of a connection error.
Tag synchronization	Internal tags are compared on partner computers as soon as one of the tags is modified on one of the redundant servers.

See also

Operation of the "Project Properties" sheet (Page 134)

Operation of the "Connections/Groups" sheet**Introduction**

The connection sheet is used for the configuration of two different types of WinCC objects. This concerns both the connection and the groups assigned to it. The logical assignment of connections and groups is determined by the position of the respective object in the table.

Microsoft Excel - ConfigurationTool.xls				
	A	B	C	
1	Connections	Groups		
2	Name		Communication driver	th
3		Name		
4	Internal tags		Internal tags	In
5		Group_1		
6		Group_2		
7	ext_Connection_1		SIMATIC S7 Protocol Suite	M
8		ext_Group_1		
9		ext_Group_2		
10				

Procedure

Connections

To create a new connection, you must assign a name to it. The connection thus becomes a valid object and can be written to WinCC. The connection parameters are assigned default values. The default values cannot be modified.

Groups

Groups can only be added to existing connections. In order to configure a group, you must assign only one name to the group.

Writing

Groups are always automatically written to WinCC together with the associated connection. In order to write groups, you must write the associated connection.

Delete

When a connection is deleted, all subordinate groups are deleted with it. In addition, all connection tags are deleted. The "Internal tags" connection cannot be deleted. When a group is deleted, all tags in the group are also deleted.

Note

The data are deleted permanently from the project folder and WinCC. The deletion of the data cannot be undone.

Table structure

The following table lists all parameters that must be set for connections in the "Connections/Groups" sheet. Parameters with dropdown list boxes are identified by a  icon.

Column	Short description
Name	Name of the connection. The name must be unique.
 Communication driver	Communication driver of the connection.
 Channel unit	Channel unit of the connection.

Column	Short description
Parameters	Parameter string of the connection. The structure of the parameter string depends on the chosen communication driver. The parameter string is currently not checked for validity. All entries are thus considered valid.
Error text	Short description of an error that occurred while writing to WinCC. Error-free objects are assigned the error text "OK".

The following table lists all parameters which must be set for groups in the "Connections/Groups" sheet.

Column	Short description
Name	Name of the group. The name must be unique.
Error text	Short description of an error that occurred while writing to WinCC. Error-free objects are assigned the error text "OK".

Operation of the "Tags" sheet

Introduction

The tags required can be configured in the "Tags" sheet. When configuring structure tags, their structure instance elements are automatically created in the tag table.

Microsoft Excel - ConfigurationTool.xls				
	A	B	C	
1	Tags			
2	Name	Data type	Length	For
3	Tag_1	Binary tag	1	
4	Tag_2	8-bit value with sign	1	
5	Tag_3	8-bit value without sign	1	
6	Tag_4	Text tag, 8-bit font	0	
7	Tag_5	Floating point number 32-bit IEEE	4	
8	Tag_6	Raw data type	0	
9	Tag_7	Floating point number 64-bit IEEE	8	Dou
10	Tag_8	16-bit value with sign	2	Sho

Procedure

In order to create a new tag, you must assign a name to it. However, the assignment of a name alone does not mean that the tag is a valid object or that it can be written to WinCC.

If no tag name has been specified, it is not possible to edit the other tag parameters. After specifying a tag name, only the data type can be edited. The tag becomes a valid object only after it has been assigned a data type.

If filling with default values has been activated, the other archive parameters of the tag are assigned the values defined in the default value table.

Writing

As soon as the tag becomes a valid object, it can be written to WinCC. Structure instance elements cannot be written from the tag table. They are automatically written together with the associated structure tags.

Delete

On deleting a tag, all limit value monitoring settings configured for this tag are also deleted. For other objects, the cross-references to the tag are deleted.

Note

The data are deleted permanently from the project folder and WinCC. The deletion of data cannot be undone.

Special functions

The configuration tool allows you to configure single messages, limit value monitoring and archive tags from the tag table. The row pop-up menu in the tag table provides a number of selection options. The procedure of creating objects via the row pop-up menu of the tag table is explained in the following chapters.

Table structure

The following table lists all parameters that must be set for tags in the "Tags" sheet. Parameters with dropdown list boxes are identified by a  icon.

Column	Short description
Name	Name of the tag. The name must be unique. The name may not correspond to that of an existing structure tag.
 Data type	All data types available for a tag.
Length	Length of the tags in WinCC in bytes. For most data types, this value cannot be edited.
 Type Conversion	Type conversion of the tag. This can only be adjusted for external tags. Not all data types are assigned format adaptation features.

Column	Short description
 Connection	Connection of the tag.
 Group	Group of the tag. This parameter is optional.
Address	Address of the tag. This can only be adjusted for external tags. The structure of the address depends on the communication driver set for the connection. For the structure of the address, refer to the tags in the parameters column in the WinCC Explorer. The address is currently not checked for validity. All entries are thus considered valid.
 Update	Update of the tag. This option is only available for internal tags. You can choose between the options "for entire project" and "local computer related".
 Linear scaling yes/no	Specify here whether linear scaling is to be used. Linear scaling can only be chosen for external tags. Not all data types support linear scaling.
Linear scaling process from	Scaling range in the process (source scaling).
Linear scaling process to	Scaling range in the process (source scaling).
Linear scaling tag from	Scaling range of the tag (scaled representation).
Linear scaling tag to	Scaling range of the tag (scaled representation).
Upper limit	Upper limit value of the tag.
Lower limit	Lower limit value of the tag.
Start value	Start value of the tag.
Substitute value	Substitute value of the tag. Substitute values can only be configured for external tags.
 Upper limit substitute value	Substitute value to be used if the actual value is above the upper limit.
 Lower limit substitute value	Substitute value to be used if the actual value is below the lower limit.
 Substitute start value	The start value is to be replaced with the substitute value.
 Error substitute value	The substitute value is to be used in the event of a connection error.
Tag synchronization	Internal tags are compared on partner computers as soon as one of the tags is modified on one of the redundant servers.
Error text	Short description of an error that occurred while writing to WinCC. Error-free objects are assigned the error text "OK".

Note

If external tags are created in the tag management of WinCC using the mapper from STEP7, the connection association must not also be changed using the WinCC Configuration Tool.

See also

Creating archive tags from the tag table (Page 213)

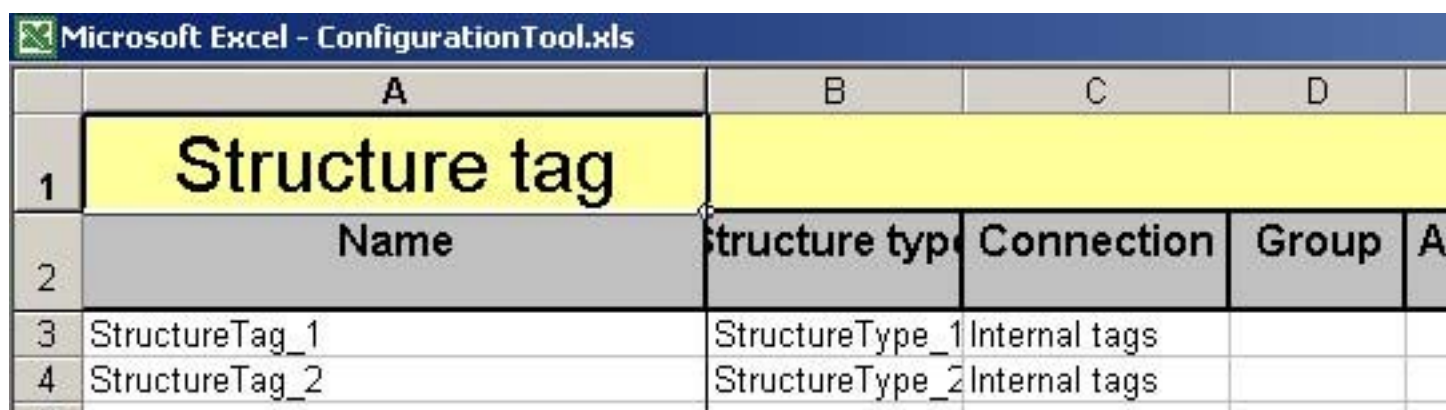
Creating limit value monitoring from the tag table (Page 222)

Creating single messages from the tag table (Page 217)

Address strings for Simatic S7 Protocol Suite (Page 240)

Operation of the "Structure Tags" sheet**Introduction**

The required structure tags can be configured on the "Structure Tags" sheet. To do so, at least one structure type must have already been configured in the project folder.



	A	B	C	D	A
1	Structure tag				
2	Name	structure type	Connection	Group	A
3	StructureTag_1	StructureType_1	Internal tags		
4	StructureTag_2	StructureType_2	Internal tags		

Procedure

In order to create a new structure tag, you must assign a name to it. However, the assignment of a name alone does not mean that the structure tag is a valid object or can be written to WinCC.

If no name has been assigned, it is not possible to edit the other structure tag parameters. After specifying a tag name, only the data type can be edited. The structure tag becomes a valid object after it has been assigned a data type.

After defining the structure type, the structure instance elements required for the structure tag are automatically created on a tag table. After modifications to the structure tag, its structure instance elements are automatically updated. This also applies to modifications to the structure type itself.

While adding new structure elements or changing data types of structure elements, you need to enter the address of all structure tags of this type.

Structure instance elements

When creating structure tags, their structure instance elements are automatically created in a tag table. Structure instance elements differ from normal tags in that most of their parameters are write-protected. Only the address and updating as well as upper limit, lower limit, start value and substitute value can be set.

Note

In the following SIMATIC channels, the addresses of the structure instance elements are generated automatically from the start address of the structure tags:

- SIMATIC S7 Protocol Suite
 - SIMATIC S5 Ethernet Layer 4
 - SIMATIC S5 PMC Ethernet
 - SIMATIC TI Ethernet Layer 4
 - SIMATIC S5 PMC Profibus
 - SIMATIC S5 Serial 3964R
 - SIMATIC S5 Programmers Port AS511
-

Write

If an attempt is made to write a structure tag to WinCC whose set structure type does not exist in WinCC or whose parameters in WinCC are not updated, a prompt appears requesting whether the structure type should also be written. If the prompt is answered with "No", the selected structure tags are not written to WinCC.

The structure instance elements are also automatically written to WinCC.

Delete

On deleting a structure tag, all associated structure instance elements are also deleted.

Note

The data are deleted permanently from the project folder and WinCC. The deletion of the data cannot be undone.

Table structure

The following table lists all parameters that must be set on the "Structure Tags" sheet.

Parameters with dropdown list boxes are identified by a  icon.

Column	Short description
Name	Name of the structure tag. The name must be unique. The name may not correspond to that of an existing tag.
 Structure type	Structure type of the structure tag.
 Connection	Connection of the structure tag.
 Group	Group of the structure tag. This parameter is optional.
Address	Address of the structure tag. The address can only be defined for external structure tags. The structure of the address depends on the communication driver set for the connection. For the structure of the address, refer to the tags in the parameters column in the WinCC Explorer. The address is currently not checked for validity. All entries are thus considered valid.
 Update	Update of the structure tag. This parameter can only be set for internal structure tags. You can choose between the options "for entire project" and "local computer related".
Error text	Short description of an error that occurred while writing to WinCC. Error-free objects are assigned the error text "OK".

See also

Address strings for Simatic S7 Protocol Suite (Page 240)

Operation of the "Structure Types" sheet

Introduction

The structure type table is used for the configuration of two different types of WinCC objects, namely the structure type itself and the structure type elements associated with it. The logical

assignment between structure type and structure type element is determined from the position of the respective object in the table.

Microsoft Excel - ConfigurationTool.xls			
	A	B	
1	Structure types	Structure type elements	
2	Name		C
3		Name	
4	StructureType_1		
5		Element_1	8-bit
6		Element_2	Float
7	StructureType_2		
8		Element_3	Binary
9		Element_4	32-bit

Procedure

Structure types

In order to create a new structure type, you must assign a name to it. In this way, the type becomes a valid object and could be written to WinCC. The other structure type parameters are optional.

Modifying Structure Types

While modifying structure types that have been used to insert one or more structure tags, then you need to enter the address again in the "Structure tags" spreadsheet.

Structure type elements

Structure type elements can only be configured if the structure type already exists. In order to create a new structure type element, you must assign a name to it. The name alone, however, does not make the structure type element a valid object. The structure type element becomes a valid object only after it has been assigned a data type.

As long as no structure type element name has been assigned, it is not possible to edit the other structure type element parameters. After specifying a structure type element name, only the data type can be edited. The structure type element becomes a valid object only after defining the data type.

If filling with default values has been activated, the other structure type element parameters are assigned the values defined for tags in the default value table.

Writing

It is not possible to write structure type elements individually to WinCC. You must always write the entire structure type to WinCC. It is irrelevant whether the structure type with all its structure type elements is selected or the structure type alone. The complete structure type with all its elements is always written to WinCC.

Deleting

If a structure type is deleted, all associated structure type elements, structure tags and structure instance elements are also deleted. If a structure type element is deleted, all associated structure instance elements are also deleted. The deletion of the structure instance element from WinCC is performed the next time the structure type is written.

Note

The data are deleted permanently from the project folder and WinCC. The deletion of data cannot be undone.

Table structure

The following table lists all parameters that must be defined for structure types on the "Structure Types/Structure Type Elements" sheet. Parameters with dropdown list boxes are identified by

a  icon.

Column	Short description
Name	Name of the structure type.
 Communication driver	Communication driver of the channel unit to be used for formatting. This parameter is optional.
 Channel unit	Channel unit whose formats are to be used. These parameters can only be set when a communication driver has been chosen.

The following table lists all parameters that must be defined for structure type elements on the "Structure Types/Structure Type Elements" sheet. Parameters with dropdown list boxes are

identified by a  icon.

Column	Short description
Name	Name of the structure type element. This name must be unique within a structure type.
 Data type	Data type of the tag.
Length	Length of the tag in the process in bytes. For most data types, the length depends on the format adaptation and cannot be modified.
Offset	Address offset of a structure instance element to the address set on the structure tag (start address).
Offset bit	Structure type element offset bit of the set structure tag start address.
 Format adaptation	Format adaptation of the tag. This can only be defined for external structure type elements. Not all data types are assigned format adaptation features.
 External tag	Specifies whether the structure type element is to be external or internal.
 Linear scaling yes/no	Specify here whether linear scaling is to be used. Linear scaling can only be chosen for external tags. Not all data types support linear scaling.
Linear scaling process from	Scaling range of the process (source scaling).
Linear scaling process to	Scaling range of the process (source scaling).
Linear scaling tag from	Scaling range of the tag (scaled representation).
Linear scaling tag to	Scaling range of the tag (scaled representation).
Error text	Short description of an error that occurred while writing to WinCC. Error-free objects are assigned the error text "OK".

Alarm logging

Alarm logging

Introduction

The alarm logging sheets are used to configure message blocks, message classes, group messages, single messages as well as limit value monitoring. Moreover, the "Alarm Logging Default Values" sheet can be used to define default values for individual messages and limit value monitoring. The procedure to configure data in alarm logging is explained in the following chapters.

See also

- Operation of the "Limit value monitoring" sheet (Page 166)
- Operation of the "Single Message" sheet (Page 164)
- Operation of the "Group Messages" sheet (Page 162)
- Operation of the "Message Classes/Message Types" sheet (Page 158)
- Operation of the "Message Blocks" sheet (Page 156)
- Operation of the "Alarm Logging Default Values" sheet (Page 151)

Operation of the "Alarm Logging Default Values" sheet**Introduction**

The configuration tool offers the option of pre-assigning default values to parameters of newly created objects. These default values can be defined on the "Alarm Logging Default Values" sheet.

Procedure

In alarm logging, you can set default values for single messages, the "Generate single messages" dialog, for limit value monitoring and for the "Generate limit values" dialog.

Microsoft Excel - Mappe2		
	A	B
1	Default values	Alarm logging
2	Single messages	
3	Number	Class
4	Number	Error
5		
6		
7	Generate messages	
8	Bit 0	Bit 1
9	Yes	No
10		
11		
12	Limit value monitoring	Limit value
13	Tag	
14		Limit value
15	Tag	
16		Limit value
17		
18	Generate limit values	
19	Limit value	Limit value indirect from
20		0

Deactivating default values

The use of the default values can be deactivated on the "Project Properties" sheet. If the default values are deactivated, all parameters of a newly created object are checked for validity and corrected, if necessary (e.g. blank rows). This, however, impairs performance during the creation of objects. The use of default values is therefore activated by default.

Table structure

Single messages

The following table lists all parameters that must be defined for single messages in the "Alarm Logging Default Values" sheet. Parameters with dropdown list boxes are identified by a



icon.

Column	Short description
Number	This parameter is write-protected.
 Class	Message class of the single message.
 Type	Message type of the single message
 Group	Group message of the single message. This parameter is optional.
Priority	Priority of the single message.
 Message tag	Message tag of the single message. Only certain data types can be used as a message tag. This parameter is optional.
Message bit	Message bit of the single message. If a message tag is set, a message bit must be specified.
 Status tag	Status tag of the single message. Only certain data types can be used as status tags. This parameter is optional.
Status bit	Status bit of the single message. If a status tag is set, a status bit must be specified.
 Acknowledgement tag	Acknowledgement tag of the single message. Only certain data types can be used as acknowledgement tags. This parameter is optional.
Acknowledgement bit	Acknowledgement bit of the single message. If an acknowledgement tag is set, an acknowledgement bit must be specified.
PLC number	PLC number of the single message
CPU number	CPU number of the single message

Column	Short description
Message text1 to message text10	Message texts of the single message. A message text can only be configured if the respective message block is activated.
Process value tag 1 to process value tag 10	Process value tags of the single message. A process value tag can only be configured, if the respective process value block must be activated.
Info text	Information text of the single message.
 Requires individual acknowledgement	Defines whether the single message must be individually acknowledged.
 Controls central signaling device	Defines whether the single message is to trigger the signaling device.
 Will be archived	Defines whether the single message is to be archived.
 Is created on a negative edge	Defines whether the single message is to be triggered at the negative edge.
 Triggers an action	Defines whether the standard GMsgFunction is to be triggered.
Loop in alarm function	Name of the function that is to be triggered by the single message.
Loop in alarm parameters	Transfer parameter of the loop in alarm function.
 FormatDll	Format DLL adapting incoming raw data telegrams to suit the WinCC alarm logging system.

Generate messages

The following table lists all parameters that must be defined in the "Alarm Logging Default Values" sheet for the "Generate messages" dialog. Parameters with dropdown list boxes are

identified by a  icon.

Column	Short description
 Bit 0 to bit 31	Indicates whether a message is to be generated for this bit.
Raw data quantity	Indicates the number of messages to be generated for one raw data tag.

Limit value monitoring/limit value

The following table lists all parameters that must be defined for limit value monitoring in the "Alarm Logging Default Values" sheet. Parameters with dropdown list boxes are identified by

a  icon.

Column	Short description
Tag	This parameter is write-protected.
One message for all limit values	Single message to be used in the case of all limit values.
Delay time	Defines the period of time the value is to be above or below the upper or lower limit value respectively, before a single message is triggered.
 Delay unit	Unit of delay time.

The following table lists all parameters that must be defined for limit values in the "Alarm Logging Default Values" sheet. Parameters with dropdown list boxes are identified by a

 icon.

Column	Short description
Limit value	This parameter is write-protected.
Limit value indirectly from tag	This parameter is write-protected.
 Limit	Defines whether the limit value is to be the upper or the lower limit value.
Message number	Single message to be triggered. If the parameter "One message for all limit values" is set for limit value monitoring, you cannot enter a message number here.
 Suppress message for quality codes other than "GOOD"	Defines whether the messages from tags whose quality code is not "GOOD" are to be suppressed.
 Hysteresis	Defines whether the hysteresis is absolute or relative (in percentages).
Hysteresis value	Value of hysteresis. 0 means that no hysteresis is used.
 Hysteresis effective from	Defines when the hysteresis is to take effect.

Generate limit values

The following table lists all parameters that must be defined in the "Alarm Logging Default Values" sheet for the "Generate limit values" dialog. Parameters with dropdown list boxes are

identified by a  icon.

Column	Short description
Limit value	Limit value of the tag.
 Limit value indirectly from tag	Tag from which the limit value is to be retrieved indirectly.
Limit	Defines whether the limit value is to be the upper or the lower limit value.

Operation of the "Message Blocks" sheet

Introduction

The "Message Blocks" sheet can be used to activate or deactivate system blocks, user text blocks and process value blocks.

Microsoft Excel - ConfigurationTool.xls						
	A	B	C	D	E	
1	Message blocks					
2	Apply	Name	Block type	Length	Flash	Al
3	Yes	Date	System bloc	0	No	Left
4	Yes	Time	System bloc	0	No	Left
5	No	Duration	System bloc	0	No	Left
6	No	Daylight Savin	System bloc	1	No	Left
7	No	Status	System bloc	1	No	Left
8	No	Acknowledged	System bloc	1	No	Left
9	No	Number	System bloc	3	No	Left
10	No	Class	System bloc	8	No	Left
11	No	Type	System bloc	2	No	Left
12	No	Controller/CP	System bloc	2	No	Left

Procedure

In order to activate a message block, set the parameter in the "Use" column to "Yes". Not all parameter settings can be modified in all message blocks. If a message block is deactivated, all parameter settings on the single message table defined for this message block are deleted.

If a user text block or process value block are activated, a new column is added to the single message table. The name of the message block is entered as the heading.

No further message blocks can be added.

Writing

Since all available message blocks already exist, the message blocks can be written to WinCC immediately.

Delete

Message blocks cannot be deleted in the configuration tool, only deactivated.

Table structure

The following table lists all parameters that must be set for message blocks. Parameters with dropdown list boxes are identified by a  icon.

Column	Short description
 Use	Defines whether a message block is to be used.
Name	Name of the message block.
Block type	Message block type. This parameter is provided for information purposes only and is write-protected.
Length	Message block length. This parameter cannot be set for all message blocks.
 Flashing	Defines whether a message displayed in the message window is to flash.
 Alignment	Defines the alignment of the text displayed in the message window.
 Format1	Representation in the message window. Format1 is determined by the message block and can only be chosen for some message blocks.
 Format2	Representation in the message window. Format2 is determined by the message block and can only be chosen for some message blocks.
Error text	Short description of an error that occurred while writing to WinCC. Error-free objects are assigned the error text "OK".

Operation of the "Message Classes/Message Types" sheet

Introduction

The "Message Classes/Message Types" sheet is used for the configuration of two different types of WinCC objects, namely message classes as well as the message types which are subordinate to the message classes. The logical assignment of message classes and message types is determined by the position of the respective object in the table.

Microsoft Excel - ConfigurationTool.xls			
	A	B	
1	Message categories	Message types	
2	Name		Ac
3		Name	Me
4	Error		Yes
5		Alarm	Arrived
6		Warning	Arrived
7		Failure	Arrived
8	System, requires acknowledgement		Yes
9		Process control system	Arrived
10		System messages	Arrived
11	System, without acknowledgement		No
12		Process control system1	Arrived
13		Operator input messages	Arrived
14			
15			

Procedure

Message classes

To create a new message class, you must assign a name to it. The remaining parameters are allocated default values. After entering a name, the message class is defined as a valid object and can be written to WinCC. The name of the message class must be unique and may not be identical to the name of a message type or group message.

It is possible to create a maximum of 16 user-defined message classes each with 16 message types. In addition, two system message classes each with two system message types are already provided and can only be configured in a limited respect.

Message types

In order to create a new message type, a message class must already be available. The new message type must be given a name. The remaining parameters are allocated default values. The coloration of the message types must be defined using the standard Excel function for cell formatting.

Note

The parameters of the group messages which are automatically generated in WinCC for each message class and message type are different in that they are already provided on the sheet.

Writing

Message types cannot be written to WinCC individually. The superordinated message class must be written to WinCC in order to write message types there. The subordinate message types are automatically written with the message classes.

Deleting

If a message class is deleted, all associated message types are deleted with it. The system message classes cannot be deleted. Message types can be deleted individually from the table. In order to delete message types from WinCC, the associated message class must be written to WinCC.

The single messages belonging to a message type or message class are also deleted during the deletion action.

Note

The data are deleted permanently from the project folder and WinCC. The deletion of data cannot be undone.

Table structure

The following table lists all parameters which must be set for message classes in the "Message Classes/Message Types" sheet. Parameters with dropdown list boxes are identified by a



icon.

Column	Short description
Name	Name of the message class. The name must be unique.
Acknowledgement came in	Defines whether incoming messages are to be acknowledged.
Acknowledgement went out	Defines whether outgoing messages are to be acknowledged.
Flashing	Defines whether queued messages are to flash.
Only first value flashing	Defines whether only the first queued message of this message class is to flash.
Message without status "went out"	Defines whether only the message "came in" status is to be archived.
Acknowledge central signaling device	Defines whether the signaling device is to be acknowledged by a separate acknowledgement button or via individual acknowledgement.
Central signaling tag	Tag through which the signaling device is triggered.
Status text arrived	Text for the "came in" status. The text can be maximum 63 characters long.
Status text departed	Text for the "went out" status. The text can be maximum 63 characters long.
Status text acknowledged	Text for the "acknowledged" status. The text can be maximum 63 characters long.
Status text arrived and departed	The text for the "arrived and departed" status. The text can be maximum 63 characters long.
Group message status tag	Group message status tag of the message class. This parameter is optional.
Group message status bit	Group message status bit of the message class. Two tag bits are used. If a status tag is specified, a status bit must also be specified.

Column	Short description
 Group message lock tag	Group message lock tag of the message class. This parameter is optional.
Group message blocking bit	Group message blocking bit of the message class. If a lock tag is specified, a lock bit must also be specified.
 Group message acknowledgement tag	Group message acknowledgement tag of the message class. This parameter is optional.
Group message acknowledgement bit	Group message acknowledgement bit of the message class. If an acknowledgement tag is specified, an acknowledgement bit must also be specified.
Error text	Short description of an error that occurred while writing to WinCC. Error-free objects are assigned the error text "OK".

The following table lists all parameters which must be set for message types in the "Message Classes/Message Types" sheet. Parameters with dropdown list boxes are identified by a



icon.

Column	Short description
Name	Name of the message type. The name must be unique.
Message arrived coloration	Text color and background color of the message with "message arrived" status.
Message departed coloration	Text color and background color of the message with "message departed" status.
Message acknowledged coloration	Text color and background color of the message with "message acknowledged" status.
 Group message status tag	Group message status tag of the message type. This parameter is optional.
Group message status bit	Group message status bit of the message type. Two tag bits are used. If a status tag is specified, a status bit must also be specified.
 Group message blocking tag	Group message lock tag of the message type. This parameter is optional.
Group message blocking bit	Group message blocking bit of the message type. If a lock tag is specified, a lock bit must also be specified.
 Group message acknowledgement tag	Group message acknowledgement tag of the message type. This parameter is optional.

Column	Short description
Group message acknowledgement bit	Group message acknowledgement bit of the message type. If an acknowledgement tag is specified, an acknowledgement bit must also be specified.
Error text	Short description of an error that occurred while writing to WinCC. Error-free objects are assigned the error text "OK".

Operation of the "Group Messages" sheet

Introduction

On the "Group Messages" sheet, you can configure group messages of all six possible levels.

Microsoft Excel - ConfigurationTool.xls			
	A	B	C
1	Group messages		
2	Name level 1	Name level 2	Name level 3
3	Group_1		
4		Group_1_1	
5		Group_1_2	
6			Group_1_2_1
7	Group_2		
8		Group_2_1	
9			

Procedure

To create a new group message, you must assign a name to it. After a name has been entered, the group message is defined as a valid object and can be written to WinCC. The name of the group message must be unique.

Group messages are structured hierarchically. This means that group messages related to the first level must be configured first, before configuring group messages for the second level. In order to configure third level group messages, second level group messages must have been configured beforehand, etc.

Writing

Only group messages from the first level can be written to WinCC. All subgroups which exist are automatically written to WinCC with them.

Delete

Group messages from all levels can be deleted. All subgroups are automatically deleted with them.

Single messages for which a deleted group message was specified are not deleted. Only the references to the group message are deleted.

Note

The data are deleted permanently from the project folder and WinCC. The deletion of the data cannot be undone.

Table structure

The following table lists all parameters that must be set for group messages. Parameters with dropdown list boxes are identified by a  icon.

Column	Short description
Name of layer 1 to layer 6	Name of the group message.
 Status tag	Status tag of the group message. This parameter is optional.
Status bit	Status bit of the group message. Two tag bits are used. If a status tag is specified, a status bit must also be specified.
 Lock tag	Lock tag of the group message. This parameter is optional.
Lock bit	Lock bit of the group message. If a lock tag is specified, a lock bit must also be specified.
 Acknowledgement tag	Acknowledgement tag of the group message. This parameter is optional.
Acknowledgement bit	Acknowledgement bit of the group message. If an acknowledgement tag is specified, an acknowledgement bit must also be specified.
Hide tag	The hide tag of a customized group message is used for automatically hiding single messages that belong to this group message.
Error text	Short description of an error that occurred while writing to WinCC. Error-free objects are assigned the error text "OK".

Operation of the "Single Message" sheet

Introduction

The required single messages can be configured on the "Single Message" sheet. The number of parameters to be set varies according to the number of message blocks that have been activated or deactivated. Single messages can also be generated from the tag table.

Microsoft Excel - Mappe2						
	A	B	C	D	E	
1	Single messages					
2	Number	Class	Type	Group	Priority	Mess
3		1 Error	Alarm		0	
4		2 Error	Alarm		0	Tag_9
5		3 Error	Alarm		0	Tag_9
6		4 Error	Alarm		0	Tag_9
7		5 Error	Alarm		0	Tag_9
8		6 Error	Alarm		0	Tag_10

Procedure

In order to create a single message, you must assign a name to it. If filling with default values has been activated, the other parameters of the single message are assigned the values defined in the default value table. Otherwise, the remaining parameters are assigned predefined values.

After a number has been entered, the single message is defined as a valid object and can be written to WinCC.

Single messages can also be generated from the tag table. The procedure for this is described in the chapter below.

Writing

As soon as the single message is defined as a valid object, it can be written to WinCC. The configured message texts are also written to WinCC. It is important to make sure the tags used in the single message table have been written to WinCC beforehand.

Delete

On deleting a single message, all message texts that are no longer necessary are also deleted. This is defined in the settings entered in the project properties table. Limit values are also

handled according to the settings in the project properties table (delete or set default message number).

Note

The data are deleted permanently from the project folder and WinCC. The deletion of the data cannot be undone.

Table structure

The following table lists all parameters that must be set for single messages. Parameters with dropdown list boxes are identified by a  icon.

Column	Short description
Number	Number of the single message
 Class	Message class of the single message.
 Type	Message type of the single message
 Group	Group message of the single message. This parameter is optional.
Priority	Priority of the single message.
 Message tag	Message tag of the single message. Only certain data types can be used as a message tag. This parameter is optional.
Message bit	Message bit of the single message. If a message tag is set, a message bit must be specified.
 Status tag	Status tag of the single message. Only certain data types can be used as status tags. This parameter is optional.
Status bit	Status bit of the single message. If a status tag is set, a status bit must be specified.
 Acknowledgement tag	Acknowledgement tag of the single message. Only certain data types can be used as acknowledgement tags. This parameter is optional.
Acknowledgement bit	Acknowledgement bit of the single message. If an acknowledgement tag is set, an acknowledgement bit must be specified.
PLC number	PLC number of the single message
CPU number	CPU number of the single message
Message text1 to message text10	Message texts of the single message. A message text can only be configured if the respective message block is activated.

Column	Short description
Process value tag 1 to process value tag 10	Process value tags of the single message. A process value tag can only be configured, if the respective process value block must be activated.
Info text	Information text of the single message.
 is single acknowledgment only	Defines whether the single message must be individually acknowledged.
 controls the central signaling device	Defines whether the single message is to trigger the signaling device.
 Will be archived	Defines whether the single message is to be archived.
 Is created on a negative edge	Defines whether the single message is to be triggered at the negative edge.
 Triggers an action	Defines whether the standard GMsgFunction is to be triggered.
Loop in alarm function	Name of the function to be triggered by the single message.
Loop in alarm parameters	The transfer parameter to the Loop in Alarm function.
 Format Dll	Format DLL adapting incoming raw data telegrams to suit the WinCC alarm logging system.
Hide mask	Specify the system statuses for hiding the single mask via the hide mask.
Error text	Short description of an error that occurred while writing to WinCC. Error-free objects are assigned the error text "OK".

See also

Creating single messages from the tag table (Page 217)

Operation of the "Limit value monitoring" sheet

Introduction

The "Limit value monitoring" sheet serves to configure two different types of WinCC objects, namely the limit value monitoring as well as the subordinate limit values. The logical

assignment between limit value monitoring and limit value is determined from the position of the respective object in the table.

Microsoft Excel - ConfigurationTool.xls			
	A	B	
1	Limit value monitoring	Limit value	
2	Tag		
3		Limit value	
4	Tag_2		
5		0	
6		100	
7	Tag_3		
8		0	
9		100	
10			

Procedure

Limit value monitoring

In order to create a limit value monitoring, you must set a tag. By setting a tag, the limit value monitoring is already defined as a valid object and can be written to WinCC. The limit value monitoring parameters are assigned default values. The default values can be defined on the "Alarm Logging Default Values" sheet.

Limit values

Limit values can only be added to existing limit value monitoring settings. In order to create a limit value, either a limit value must be entered or a tag from which the limit value can be indirectly transferred. In this way, the limit value is defined as a valid object and can be written to WinCC. The limit value parameters are assigned default values. The default values can be defined on the "Alarm Logging Default Values" sheet.

Writing

Limit values are automatically written together with the superordinated limit value monitoring. Limit values cannot be written to WinCC individually.

Note

In WinCC you have to activate the limit value monitoring function in the Alarm Logging editor by selecting "Tools - Add ins".

Deleting

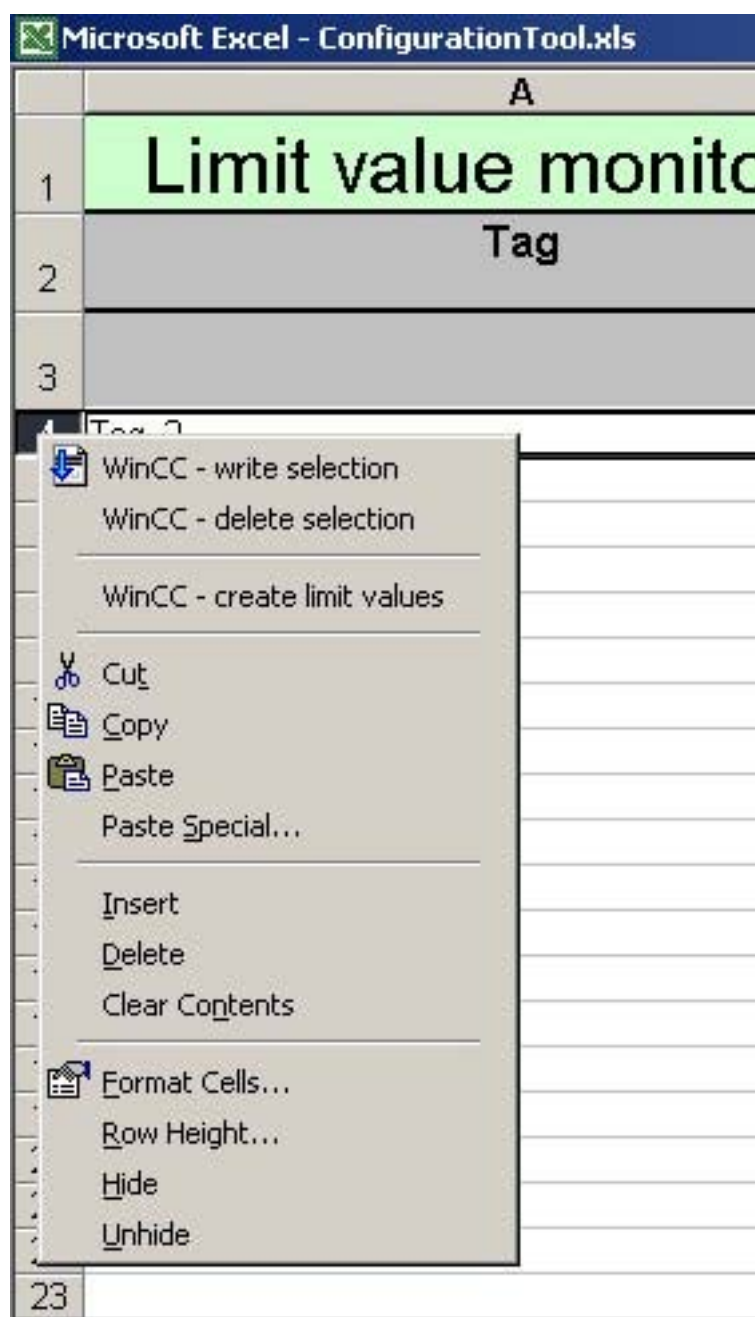
On deleting a limit value monitoring setting, all subordinate limit values are also deleted. Limit values can also be deleted individually.

Note

The data are deleted permanently from the project folder and WinCC. The deletion of the data cannot be undone.

Using the tag table pop-up menu, you can generate limit value monitoring together with limit values. The procedure for this is described in the chapter below.

Limit values for existing limit value monitoring can be generated by using the "Limit Value Monitoring/Limit Values" table sheet. To do this, select the limit value monitoring to which the limit values are to be added. Select the item "WinCC - create limit values" in the sheet pop-up menu.



The "Limit value monitoring" dialog is displayed. The subsequent procedure is described in the chapter below from item 3.

Table structure

The following table lists all parameters which must be set for limit value monitoring in the "Limit Value Monitoring/Limit Values" sheet. Parameters with dropdown list boxes are identified by



a icon.

Column	Short description
Tag	The tag to be monitored.
One message for all limit values	A single message to be used for all limit values. If a message number is entered, no other single messages can be entered for the individual limit values.
Delay time	Defines the period of time the value is to be above or below the upper or lower limit value respectively, before a single message is triggered. The value 0 signifies no delay. The total delay may be a maximum of 24 hours.
Delay unit	Unit of delay time.
Error text	Short description of an error that occurred while writing to WinCC. Error-free objects are assigned the error text "OK".

The following table lists all parameters which must be set for limit values in the "Limit Value Monitoring/Limit Values" sheet. Parameters with dropdown list boxes are identified by a



a icon.

Column	Short description
Limit value	Limit value of the tag. Can only be set if the limit value is not to be indirectly transferred from a tag.
Limit value indirectly from tag	The limit value of the tag whose limit value is being monitored is to be transferred indirectly from another tag.
Limit	Defines whether the limit value is to be the upper or the lower limit value.
Message number	Single message to be triggered.
Suppress message for quality codes other than "GOOD"	Defines whether the messages from tags whose quality code is not "GOOD" are to be suppressed.
Hysteresis	Defines whether the hysteresis is absolute or relative (in percentages).
Hysteresis value	Value of hysteresis. 0 means that no hysteresis is used.

Column	Short description
 Hysteresis effective from	Defines when the hysteresis is to take effect.
Error text	Short description of an error that occurred while writing to WinCC. Error-free objects are assigned the error text "OK".

See also

Creating limit value monitoring from the tag table (Page 222)

Tag logging

Tag logging

Introduction

The "Tag Logging" sheets can be used to configure times, process value archives and compressed archives. In addition, the "Tag Logging Default Value" sheet can be used to define default values for process value archives and compressed archives. The procedure to configure data in tag logging is explained in the following chapters.

See also

Operation of the "Compressed Archives" sheet (Page 181)

Operation of the "Process Value Archive" sheet (Page 178)

Operation of the "Times" sheet (Page 176)

Operation of the "Tag Logging Default Values" sheet (Page 171)

Operation of the "Tag Logging Default Values" sheet

Introduction

The configuration tool offers the option of pre-assigning default values to parameters of newly created objects. These default values can be defined on the "Tag Logging Default Values" sheet.

Procedure

The tag logging feature can be used to define default values for process value archives, process value archive tags, compressed archives and compressed archive tags.

Microsoft Excel - ConfigurationTool.xls			
	A	B	
1	Default values	Tag Logging	
2	Process value archives	Archive tag	
3	Archive name		
4		Tag	Arch
5	Process value archive		
6		Binary tag	Arch
7		Analog tag	Arch
8		Process controlled tag	Arch
9			
10			
11	Compressing archives	Archive tag	
12	Archive name		
13		Source archive	Sc
14	Archive name		
15		Source archive	Sc

Deactivating default values

The use of the default values can be deactivated in the "Project properties" sheet. If the default values are deactivated, all parameters of a newly created object are checked for validity and corrected, if necessary (e.g. blank rows). This, however, impairs performance during the creation of objects. The use of default values is therefore activated by default.

Table structure

The following tables list the parameters which must be set in the sheet. Parameters with dropdown list boxes are identified by a  icon.

Process value archives

Column	Short description
Archive name	This parameter is write-protected.
 Archive type	This parameter is write-protected.
Comment	Any comment.
 Archiving at system start	Defines whether the archiving is to begin when the system is started.
Start/enable action	Action called upon starting or enabling archiving.
Size	Size of the archive in data records.
 Memory location	The parameter can only be set for cyclic archives. Defines whether the archive is to be stored in the main memory or on the hard disk.

Process value archive tags

Column	Short description
Tag	This parameter is write-protected.
Archive tag name	This parameter is write-protected.
Archive tag type	This parameter is write-protected.
Comments	Any comment.
 Archiving at system start	Defines whether the archiving is to begin when the system is started.
 Supplying tags	Defines whether the tag data are to be applied automatically or manually.
 Acquisition type	Type of data acquisition.
 Acquisition cycle	Cycle to be used for data acquisition.

Column	Short description
 Archiving cycle	Time base to be used by the cycle for archiving and displaying data. The archiving cycle results from the product time base and the archiving cycle factor. This must be an integer multiple of the acquisition cycle.
Archiving cycle factor	Archiving cycle factor. The archiving cycle results from the product time base and the archiving cycle factor. This must be an integer multiple of the acquisition cycle.
Also write in tag	Tag into which the last archived value is to be transferred.
 Editing	Processing of the data acquired. All selection options are available in the case of cyclically selective and cyclically continuous archiving. Only the current value can be used in the case of acyclic archiving and archiving after every change.
Action for processing	Action with which the data is to be processed. This parameter can only be set if processing is set to action.
Unit	Unit of the time value entered.
Relevant long term	Defines whether the archive tag is to be written to the archive on the central archive server (WinCC CAS).
 Archiving on change	Defines whether the tag is only to be archived following a modification.
 Hysteresis	Defines whether the hysteresis is absolute or relative (in percentages).
Hysteresis value	Value of hysteresis. 0 means that no hysteresis is used.
 Archiving on	Trigger in the case of binary process value archive tags. This parameter is write-protected for analog and process controlled tags.
 Save on error	Defines whether the last value or replacement value is to be archived following an error.
Number of values leader	The number of values to be taken into account prior to archiving. This parameter can only be set in the case of cyclically selective archiving.
Number of values trailer	Number of values to be taken into account after archiving. This parameter can only be set in the case of cyclically selective archiving.
Representation limit, lower limit	The lower limit of the tag value range. 0 signifies no limit.
Representation limit, upper limit	The upper limit of the tag value range. 0 signifies no limit.
Start event	Name of the function that triggers archiving.

Column	Short description
Start tag	Binary tag.
Stop event	Name of the function that stops archiving.
Stop tag	Binary tag.
Relevant long term	You can define here whether the archive tag on the central archive server (WinCC CAS) is to be written to the archive.
Archive after segment change	It can be determined here whether the acyclic archived tags are archived during a segment change.
 Standardizing DLL	Adaptation program associated with the tag. It is determined by the AS manufacturer.

Compressed archives

Column	Short description
Archive name	This parameter is write-protected.
Comments	Any comment.
 Archiving at system start	Defines whether the archiving is to begin when the system is started.
Start/enable action	Action called upon starting or enabling archiving.
 Processing method	Method with which the source archive or compressed archive tag is processed.
 Compression time period	Time period for compressing in the archive set up. The compressing period must be at least one minute.

Compressed archive tags

Column	Short description
Source archive	This parameter is write-protected.
Source tag	This parameter is write-protected.
Archive tag name	This parameter is write-protected.
Comments	Any comment.
 Archiving at system start	Defines whether the archiving is to begin when the system is started.
 Editing	Processing method for the values compiled during the archiving cycle.

Column	Short description
Unit	Unit of the tag to be archived.
Relevant long term	You can define here whether the archive tag is to be written to the archive on the central archive server (WinCC CAS).

Operation of the "Times" sheet

Introduction

The time cycles required for the archives can be configured on the "Times" sheet. When a new project folder or new WinCC project is created, five times are provided by default. These can be modified or deleted.

Microsoft Excel - ConfigurationTool.xls					
	A	B	C	D	
1	Times				
2	Name	Basis	Factor	Trigger at system start	Trigger
3	500 ms	500 ms	1	No	No
4	1 second	1 second	1	No	No
5	1 minute	1 minute	1	No	No
6	1 hour	1 hour	1	No	No
7	1 day	1 day	1	No	No
8					

Procedure

In order to create a new time, it must be assigned a name. The remaining parameters are assigned predefined values. After entering a name, the time is defined as a valid object and it can be written to WinCC.

Writing

As soon as the time becomes a valid object, it can be written to WinCC. If a time is written to WinCC, all modified archive system objects are also written to WinCC.

Deleting

Only times that are not used by any archive can be deleted.

Note

The data are deleted permanently from the project folder and WinCC. The deletion of the data cannot be undone.

Table structure

The following table lists all parameters that must be set for times. Parameters with dropdown list boxes are identified by a  icon.

Column	Short description
Name	Name of the time.
 Base	Base of the time. The product of base x factor is the cycle time.
Factor	Factor with which the base is to be multiplied. The product of base x factor is the cycle time.
 Trigger at system start	Defines whether the cycle is to be triggered when the system is started.
 Trigger at start point	Defines whether the cycle is to be triggered at the start point.
Start point month	The start point month can only be set if the option "Trigger at start point" is set to "Yes" and the cycle time is at least one hour.
Start point day	The start point day can only be set if the option "Trigger at start point" is set to "Yes" and the cycle time is at least one hour.
Start point hour	The start point hour can only be set if the option "Trigger at start point" is set to "Yes" and the cycle time is at least one minute.
Start point minute	The start point minute can only be set if the option "Trigger at start point" is set to "Yes".
Start point second	The start point second can only be set if the option "Trigger at start point" is set to "Yes".
Error text	Short description of an error that occurred while writing to WinCC. Error-free objects are assigned the error text "OK".

Operation of the "Process Value Archive" sheet

Introduction

The process value archive table is used for the configuration of two different types of WinCC objects, namely the process value archive and the process value archive tags associated with it. The logical assignment between archive and tag is determined from the position of the respective object in the table.

	A	B
1	Process value archives	Archive tag
2	Archive name	
3		Tag
4	ProcessValueArchive_1	
5		Tag_1
6		Tag_2
7		Tag_3
8		

Procedure

Process value archives

In order to create a new process value archive, you must assign a name to it. As soon as a name has been assigned, the archive is defined as a valid object and can be written to WinCC. If filling with default values has been activated, the other archive parameters are assigned the values defined in the default value table.

Process value archive tags

In order to create a new process value archive tag, an existing tag must be entered in the "Tag" column. To enter a tag in the "Tag" column, use the tag selection dialog which can be opened by double-clicking the respective cell. As soon as a tag has been entered, the process value archive tag is defined as a valid object and can be written to WinCC.

If filling with default values has been activated, the other archive parameters of the process value archive tag are assigned the values defined in the default value table.

Note

Process value archives and process value archive tags can also be configured using the tag table. The procedure for the configuration of data in the tag table is explained in the following chapter.

Writing

If a process value archive or process value archive tag is written to WinCC, all modified objects of the tag logging function are automatically written to WinCC.

Delete

If a process value archive is deleted, all subordinate process value archive tags are deleted with it. In addition, all compressed archive tags set via the process value archive or process value archive tag are also deleted.

Note

The data are deleted permanently from the project folder and WinCC. The deletion of the data cannot be undone.

Table structure

The following tables list the parameters which must be set in the sheet. Parameters with dropdown list boxes are identified by a  icon.

Process value archives

Column	Short description
Archive name	Name of the process value archive. The name must be unique.
 Archive type	This parameter is write-protected.
Comment	Any comment.
 Archiving at system start	Defines whether the archiving is to begin when the system is started.
Start/enable action	Action called upon starting or enabling archiving.
Size	Size of the archive in data records.
 Memory location	The parameter can only be set for cyclic archives. Defines whether the archive is to be stored in the main memory or on the hard disk.

Process value archive tags

Column	Short description
Tag	Tag to be archived.
Archive tag name	Name of the process value archive tag. The name must be unique.
Archive tag type	This parameter is write-protected.
Comments	Any comment.
 Archiving at system start	Defines whether the archiving is to begin when the system is started.
 Supplying tags	Defines whether the tag data are to be applied automatically or manually.
 Acquisition type	Type of data acquisition.
 Acquisition cycle	Cycle to be used for data acquisition.
 Archiving cycle	Time base to be used by the cycle for archiving and displaying data. The archiving cycle results from the product time base and the archiving cycle factor. This must be an integer multiple of the acquisition cycle.
Archiving cycle factor	Archiving cycle factor. The archiving cycle results from the product time base and the archiving cycle factor. This must be an integer multiple of the acquisition cycle.
Also write in tag	Tag into which the last archived value is to be transferred.
 Editing	Processing of the data acquired. All selection options are available in the case of cyclically selective and cyclically continuous archiving. Only the current value can be used in the case of acyclic archiving and archiving after every change.
Action for processing	Action with which the data is to be processed. This parameter can only be set if processing is set to action.
Unit	Unit of the time value entered.
 Archiving on change	Defines whether the tag is only to be archived following a modification.
 Hysteresis	Defines whether the hysteresis is absolute or relative (in percentages).

Column	Short description
Hysteresis value	Value of hysteresis. 0 means that no hysteresis is used.
 Archiving on	Trigger in the case of binary process value archive tags. This parameter is write-protected for analog and process controlled tags.
 Save on error	Defines whether the last value or replacement value is to be archived following an error.
Number of values leader	The number of values to be taken into account prior to archiving. This parameter can only be set in the case of cyclically selective archiving.
Number of values trailer	Number of values to be taken into account after archiving. This parameter can only be set in the case of cyclically selective archiving.
Representation limit, lower limit	The lower limit of the tag value range. 0 signifies no limit.
Representation limit, upper limit	The upper limit of the tag value range. 0 signifies no limit.
Start event	Name of the function that triggers archiving.
Start tag	Binary tag.
Stop event	Name of the function that stops archiving.
Stop tag	Binary tag.
Relevant long term	You can define here whether the archive tag on the central archive server (WinCC CAS) is to be written to the archive.
Archive after segment change	It can be determined here whether the acyclic archived tags are archived during a segment change.
Error text	You can enter an error text here.
 Standardizing DLL	Adaptation program associated with the tag. It is determined by the AS manufacturer.

See also

Creating archive tags from the tag table (Page 213)

Operation of the "Compressed Archives" sheet

Introduction

The compressed archive table is used for the configuration of two different types of WinCC object, namely the compressed archive and the compressed archive tags associated with the

compressed archive. The logical assignment between archive and tag is determined from the position of the respective object in the table.

Microsoft Excel - ConfigurationTool.xls			
	A	B	
1	Compressing archives	Archive tag	
2	Archive name		
3		Source archive	Source
4	CompressingArchive_1		
5		ProcessValueArchive_1	Tag_1
6		ProcessValueArchive_1	Tag_2
7		ProcessValueArchive_1	Tag_3
8			

Procedure

Compressed archives

In order to create a new compressed archive, you must assign a name to it. As soon as a name has been assigned, the archive is defined as a valid object and can be written to WinCC.

If filling with default values has been activated, the other archive parameters are assigned the values defined in the default value table.

Compressed archive tags

In order to create a new compressed archive tag, a source archive must be specified. Open the "Archive tag selection" dialog by double-clicking the "Source archive" column. Use this dialog to select the process value archive tags to be compressed.

If filling with default values has been activated, the other archive parameters of the compressed archive tag are assigned the values defined in the default value table.

Writing

If a compressed archive or compressed archive tag is written to WinCC, all modified objects of the Tag Logging function are automatically written to WinCC, too.

Delete

If a compressed archive is deleted, all subordinate compressed archive tags are deleted with it.

Note

The data are deleted permanently from the project folder and WinCC. The deletion of the data cannot be undone.

Table structure

The following tables list the parameters which must be set in the sheet. Parameters with dropdown list boxes are identified by a  icon.

Compressed archives

Column	Short description
Archive name	Name of the compressed archive.
Comments	Any comment.
 Archiving at system start	Defines whether the archiving is to begin when the system is started.
Start/enable action	Action called upon starting or enabling archiving.
 Processing method	Method with which the source archive or compressed archive tag is processed.
 Compression time period	Time period for compressing in the archive set up. The compressing period must be at least one minute.

Compressed archive tags

Column	Short description
Source archive	Source archive containing the source tag.
Source tag	Process value archive tag to be compressed.
Archive tag name	Name of the compressed archive tag.
Comments	Any comment.
 Archiving at system start	Defines whether the archiving is to begin when the system is started.

Column	Short description
 Editing	Processing method for the values compiled during the archiving cycle.
Unit	Unit of the tag to be archived.
Relevant long term	You can define here whether the archive tag is to be written to the archive on the central archive server (WinCC CAS).
Error text	You can enter an error text here.

Text library

Text library

Introduction

The text library can be used to edit texts which are used by various modules in the Runtime system.

Operation of the "Texts" sheet

Introduction

The "Texts" sheet is used for the administration of the text library. By default, the languages German, English, French, Italian and Spanish are available. You have the option to add or remove texts and languages.

Microsoft Excel - Mappe2				
	A	B	C	D
1	Texts			
2	Text ID	German (Germany)	English (United States)	French (France)
3	1	Störung	Error	Incident
4	2	System, quittierpflichtig	System, requires acknowledgement	Système, acc
5	3	System, ohne Quittierung	System, without acknowledgement	Système, san
6	4	Alarm	Alarm	Alarme
7	5	Warnung	Warning	Avertissemen
8	6	Fehler	Failure	Erreur
9	7	Leittechnik	Process control system	Système de c
10	8	Systemmeldungen	System messages	Alarmes syst
11	9	Bedienmeldungen	Operator input messages	Alarmes de c
12	10	+	+	+

Procedure

In the "Texts" sheet, you can edit existing texts and add new ones.

New texts can be added in two different ways. You can enter a new text by assigning a new text ID. Alternatively, you can set up a new text by making an entry in any text column. In this case, the text ID is assigned automatically but can be changed.

Message texts can be configured directly in the respective sheets that are automatically created in the text library.

Writing

As soon as a text has been entered, it can be written to WinCC. If a sheet containing message texts is written to WinCC, the text library is also written to WinCC.

Deleting

Message block names, message class names and message type names cannot be deleted. If you delete a message text or a status text, it is also deleted at the point at which it is used. Texts not used in the message system can be deleted.

Note

The data are deleted permanently from the project folder and WinCC. The deletion of the data cannot be undone.

Special functions

To add a language to the text library, select "Add RT language" in the "WinCC" menu. The "Select language" dialog is displayed. It lists all available languages. Select the language you wish to add and click "OK" to close the dialog box. The selected language is added in a new column.



Table structure

Column	Short description
Text ID	ID of the text. The ID of the text can be changed provided that the text has not yet been written to WinCC.
Languages	List of all configured texts. The number of language columns is determined by the number of configured languages.
Error text	Short description of an error that occurred while writing to WinCC. Error-free objects are assigned the error text "OK".

Operation of the "Write Error" sheet

Introduction

This sheet lists all defective objects contained in the project folder.

Microsoft Excel - ConfigurationTool.xls

File Edit View Insert Format Tools Data Window WinCC Help

	A	B	C	D	E	F
1	Fault list					
2	Object name	Table type	Error text			
3		2 Single message	The set message tag does not yet exist!			
4		3 Single message	The set message tag does not yet exist!			
5		4 Single message	The set message tag does not yet exist!			
6		5 Single message	The set message tag does not yet exist!			
7		6 Single message	The set message tag does not yet exist!			
8	Tag_7	Tag table	The set connection does not yet exist!			

Procedure

The error list is added to the project folder if an error occurs while writing an object.

The first column in the error list contains the name of the object in which an error has occurred. The second column contains the table type, not the table name. The third column contains a short description of the error which has occurred.

Double click an entry in the error list to access the defective object directly.

Table structure

Column	Short description
Object name	Name of the object in which the error occurred.
Table type	Type of the table in which the error occurred.
Error text	Short description of the error.

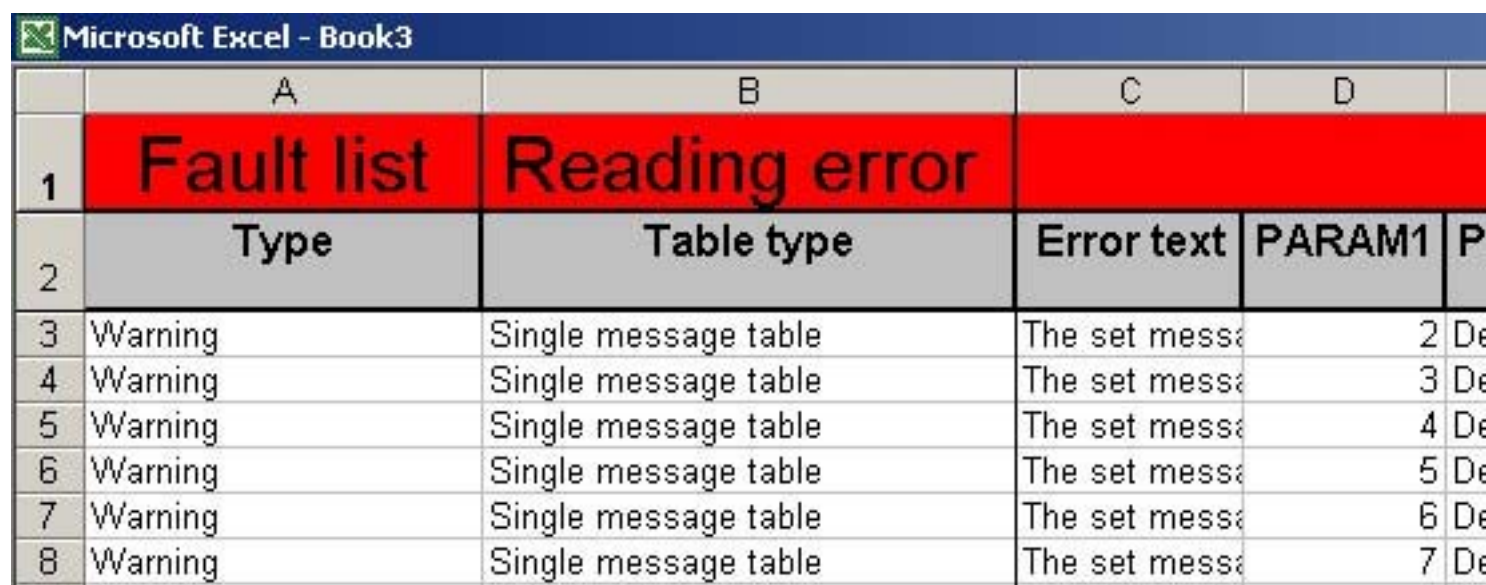
See also

Error column (Page 229)

Operation of the "Read Error" sheet

Introduction

This sheet lists all defective objects detected during the reading from WinCC.



Microsoft Excel - Book3					
	A	B	C	D	E
1	Fault list	Reading error			
2	Type	Table type	Error text	PARAM1	PARAM2
3	Warning	Single message table	The set message		2 De
4	Warning	Single message table	The set message		3 De
5	Warning	Single message table	The set message		4 De
6	Warning	Single message table	The set message		5 De
7	Warning	Single message table	The set message		6 De
8	Warning	Single message table	The set message		7 De

Procedure

The error list is only added to the project folder when an error has occurred while reading an object.

The first column in the error list indicates the type of the error. The system issues warnings and error messages. In the event of a warning, the object is entered in the list and the incorrect parameter is corrected. In the event of an error, the object is not entered in the table as the error could not be eliminated.

The second column contains the table type, not the table name. The third column contains a short description of the error which has occurred.

The other columns contain the object data in which the read error occurred.

Table structure

Column	Short description
Type	Type of the error.
Table type	Type of the table in which the error occurred.
Error text	Short description of the error.
Param 1 to Param 89	All parameters of the defective object.

1.8.5.4 Dialogs

Dialogs

Introduction

Overview of the configuration tool dialogs The dialogs of the configuration tool are described in detail in the following chapters.

See also

Operation of the "Archive tag" dialog (Page 193)

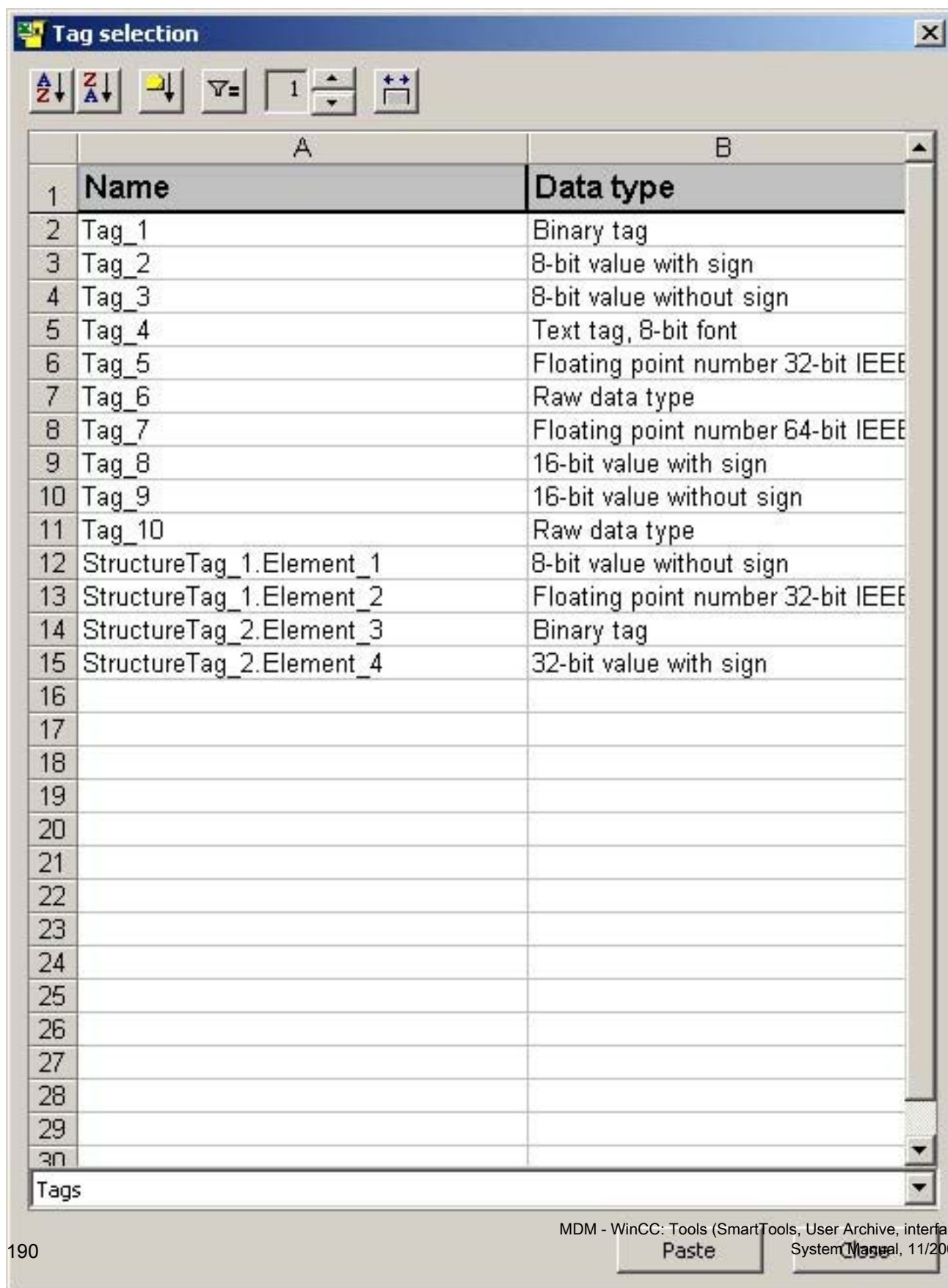
Operation of the tag dialog (Page 189)

Operation of the tag dialog

Introduction

The configuration tool provides a tag dialog for entering tags. We recommend that you always use the tag dialog. Firstly, this ensures that the tag already exists and, secondly, you can enter

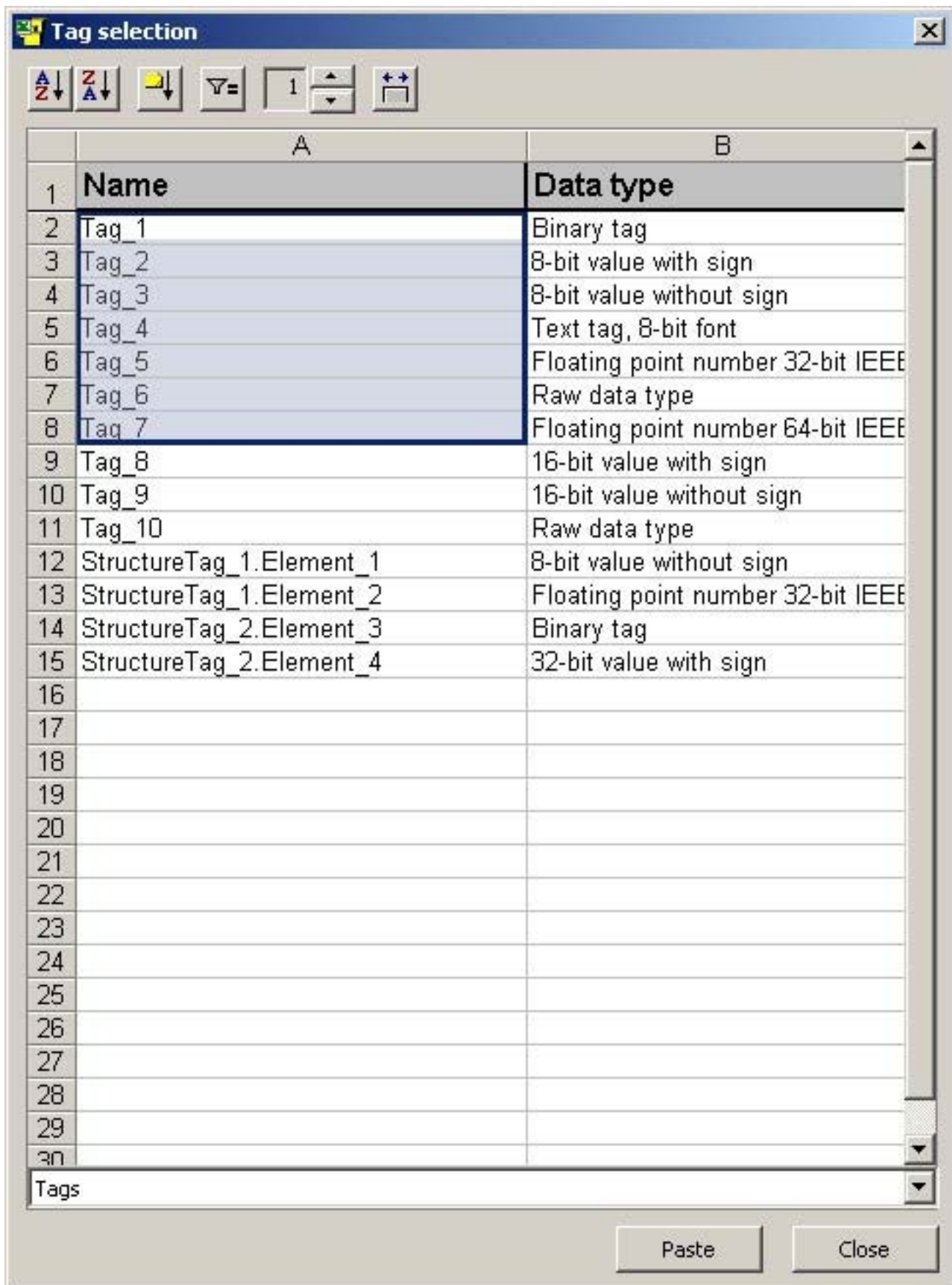
several tags in the tag dialog in one operation. At the bottom of the dialog there is a dropdown list box from which to select the tag table in which tags are to be inserted.



Operation of the archive tag dialog

Open the tag dialog by double-clicking a column in the configuration tool in which tags are to be entered. Select the tags to be inserted. Use the  spin box to define how often the tag is to be inserted. Click the "Paste" button to insert the tags.

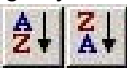
The tag dialog also allows for the insertion of various different tags in one operation. To do this, select the tags to be inserted.



Use the "Paste" button repeatedly to insert the selected tags the same number of times as defined with the  spin box.

Buttons of the archive tag dialog

- Sorting tags by name

Click the  buttons to sort the tags by name in alphabetical order (ascending or descending).

- Sorting tags by data type

Click the  button to sort the tags according to their data type in descending order.

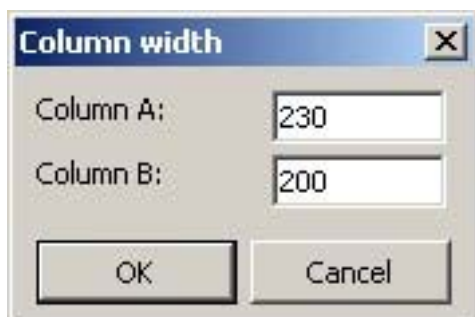
- Applying a filter

Click the  button to define a filter for the "Name" and "Data type" columns. Use the added dropdown list boxes to select a filter.

To reset the filters, click the  button again.

- Setting column widths in the dialog

Click the  button to access the "Column width" dialog. In this dialog, you can modify the width of the two columns.

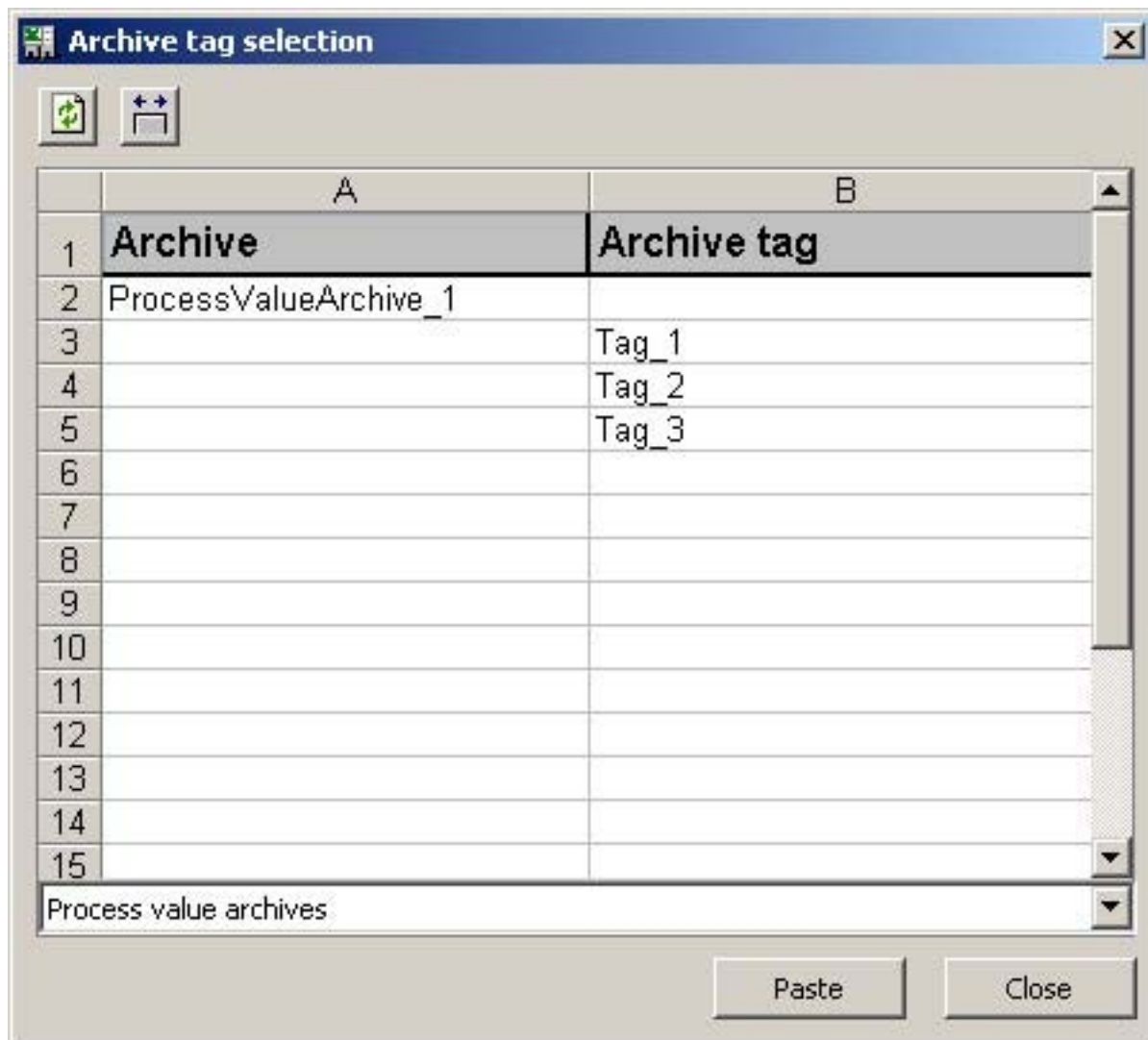


Operation of the "Archive tag" dialog

Introduction

The configuration tool provides an "Archive tag" dialog for entering process value archive tags in the compressed archive table. We recommend that you always use the "Archive tag" dialog. Firstly, this ensures that the archive tag already exists and, secondly, you can enter several

tags in the "Archive tag" dialog in one operation. At the bottom of the dialog there is a dropdown list box from which to select the process value table in which tags are to be inserted.



Operation of the tag dialog

Open the "Archive tag" dialog by double-clicking a column in the configuration tool in which tags are to be entered. Select the tags to be inserted. Click the "Paste" button to insert the tags.

The "Archive tag" dialog also allows for the insertion of various different tags in one operation. To do this, select the tags to be inserted.

Buttons of the archive tag dialog

- Updating display

Click the  button to update the view of the "Archive tag" dialog.

- Setting column widths in the dialog

Click the  button to access the "Column width" dialog. In this dialog, you can modify the width of the two columns.



1.8.5.5 Processing configured data

Processing configured data

Introduction

Generally, all functions offered by Excel can be used without restriction. However, when processing configured data in the configuration tool, certain points must be taken into account. These points are dealt with in detail in the following chapters.

See also

Sorting data (Page 207)

Setting a filter (Page 208)

Cutting data (Page 206)

Copying data (Page 206)

Deleting data from the project folder and WinCC (Page 201)

Writing data to WinCC (Page 196)

Writing data to WinCC

Introduction

In order to write data to WinCC, a connection must exist to the corresponding WinCC project. The following options are available for writing data to WinCC.

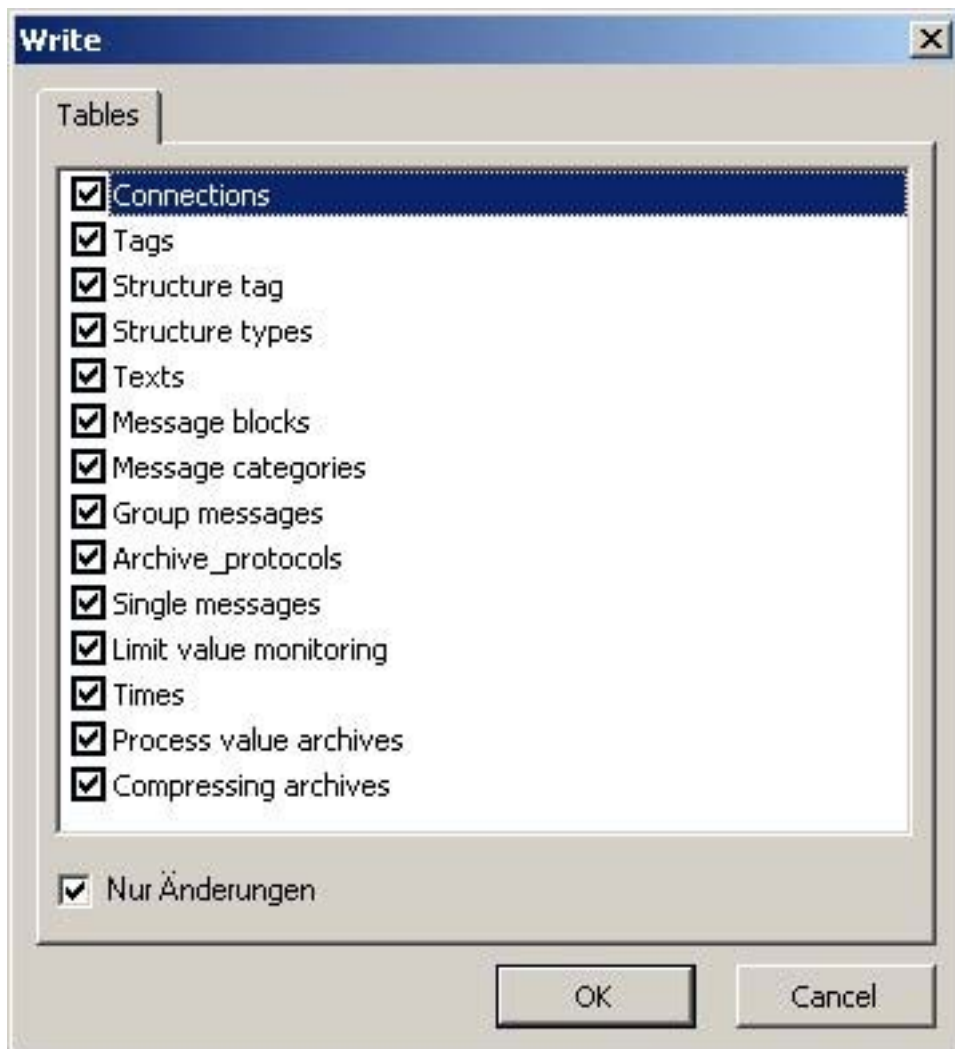
Note

If the data in the WinCC project and project folder do not match after writing the data, this may be due to an updating problem in the WinCC Explorer. In order to update the display, close the WinCC project and open it again.

Writing to WinCC using the toolbar

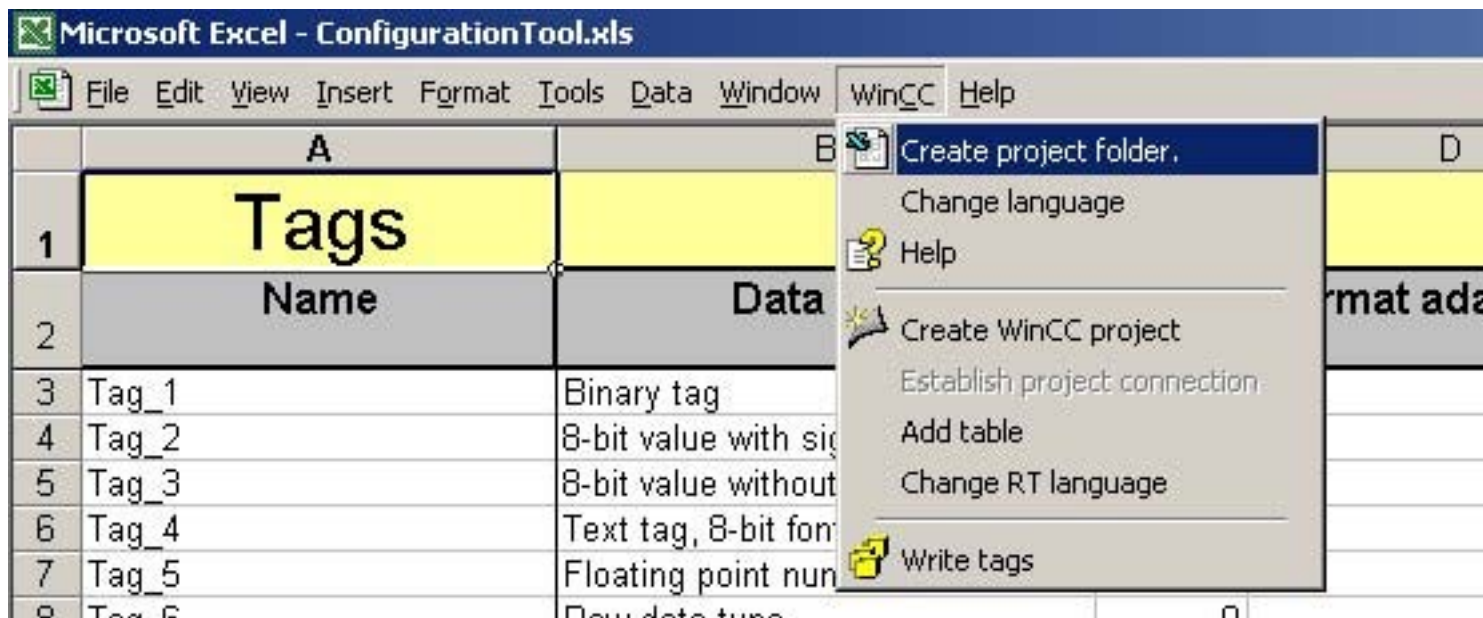
1. Click the  button in the toolbar. The "Write" dialog is opened.
2. Use the dialog to select the sheets whose data are to be written to WinCC.

3. Use the "Only changes" check box to specify that only changed objects are to be written to WinCC. If the check box is not selected, all objects are written to WinCC.
4. Click "OK" to write the data of the worksheets to WinCC.

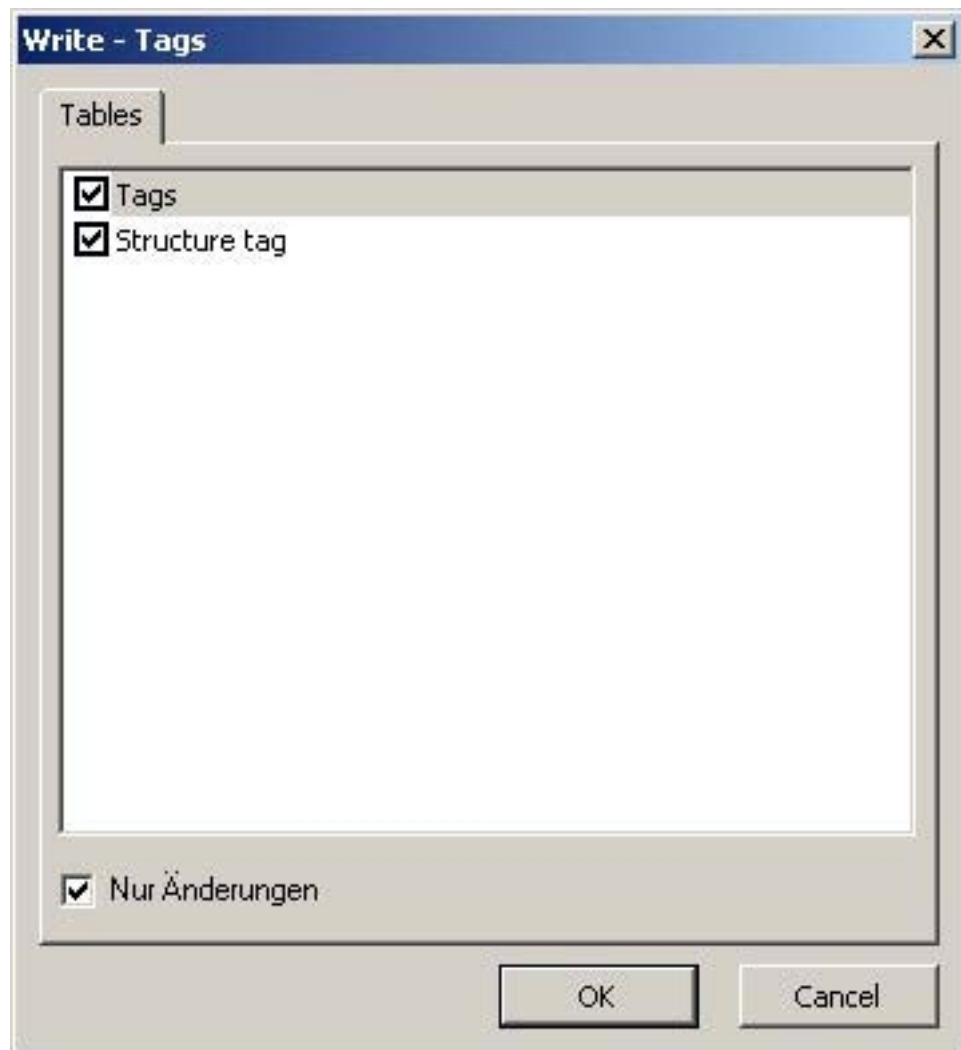


Writing to WinCC using the dropdown menu

1. Select the "WinCC" dropdown menu.
2. A menu item for writing objects is available in the dropdown menu, which is determined by the table that is active in the project folder. Select the "Write tags" option to open the "Write" dialog.



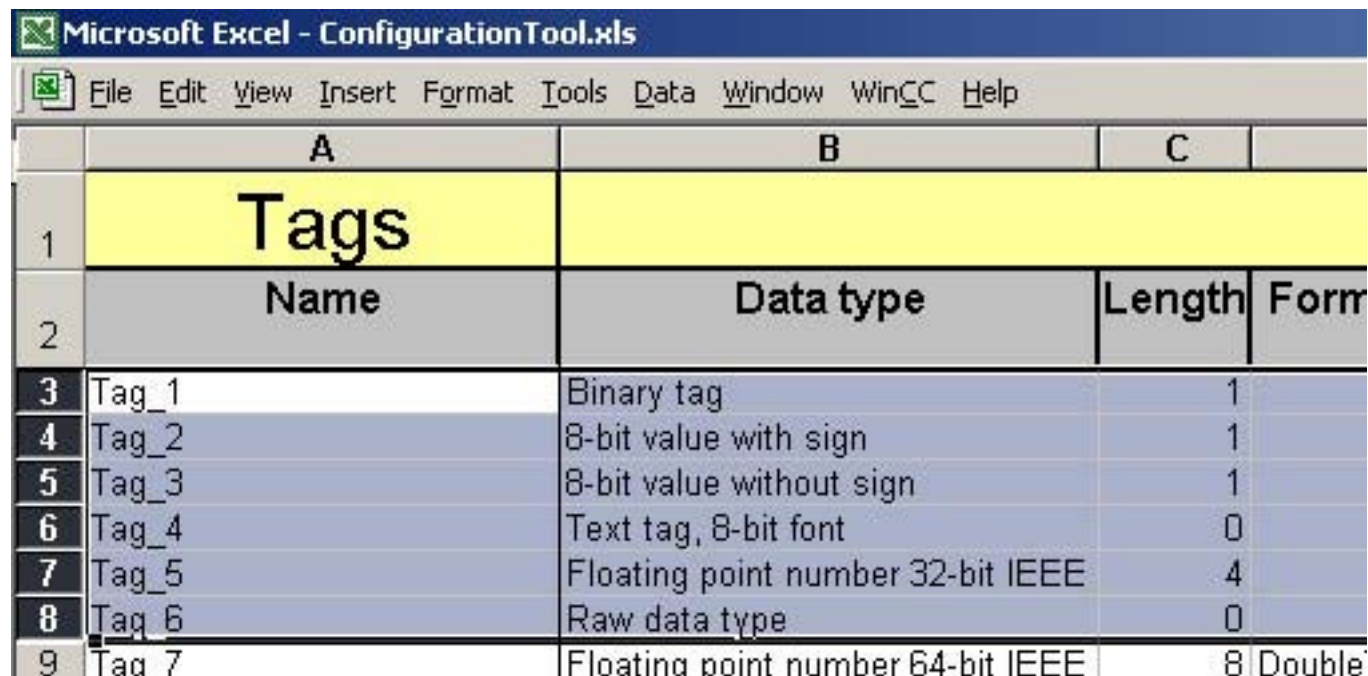
3. Use the dialog to select the sheets whose data are to be written to WinCC. Please note that only tables that are of the same type as the active table are available.



4. Continue as described in "Writing to WinCC using the toolbar".

Writing to WinCC using the pop-up menu

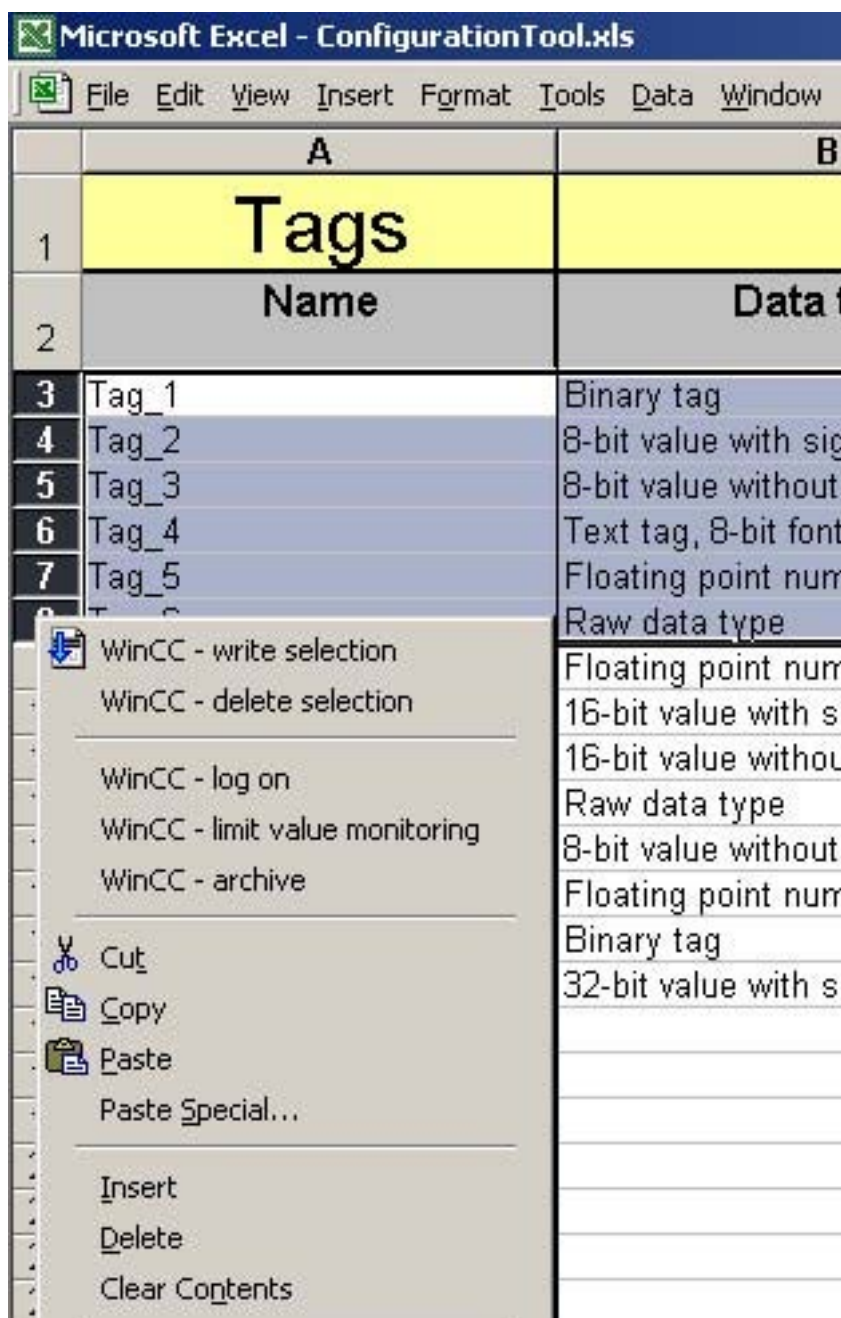
1. Select the objects to be written. You must always select a complete row.



The screenshot shows a Microsoft Excel window titled "Microsoft Excel - ConfigurationTool.xls". The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Window, WinCC, and Help. The spreadsheet has four columns: A (Name), B (Data type), C (Length), and D (Form). Row 1 is a header for "Tags". Rows 3 through 9 are selected, indicated by a dark blue background. The data in these rows is as follows:

	A	B	C	D
1	Tags			
2	Name	Data type	Length	Form
3	Tag_1	Binary tag	1	
4	Tag_2	8-bit value with sign	1	
5	Tag_3	8-bit value without sign	1	
6	Tag_4	Text tag, 8-bit font	0	
7	Tag_5	Floating point number 32-bit IEEE	4	
8	Tag_6	Raw data type	0	
9	Tag_7	Floating point number 64-bit IEEE	8	Double

2. Open the row pop-up menu using the right mouse button and select "WinCC - write selection". All selected objects are written to WinCC. By using the pop-up menu, only changed objects are written to WinCC.



Deleting data from the project folder and WinCC

Introduction

It is not possible to delete rows, columns or cells from the data areas of WinCC objects. To delete WinCC objects, use the pop-up menu item "WinCC - delete selection". As complete

sheets are deleted, all WinCC objects configured on it are deleted. Please note that not all sheets can be deleted.

Note

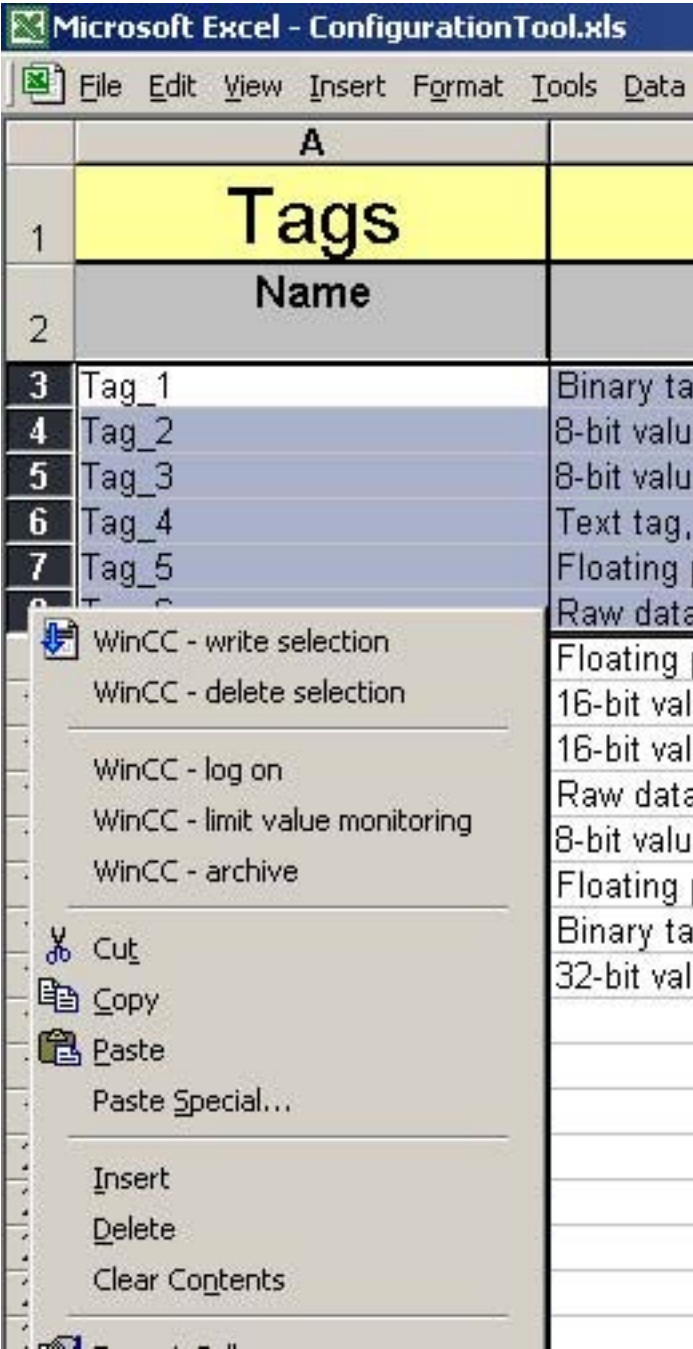
If the data in the WinCC project and project folder do not match after writing the data, this may be due to an updating problem in the WinCC Explorer. In order to update the display, close the WinCC project and open it again.

Deleting via the pop-menu

1. Select the objects to be deleted. You must always select a complete row.

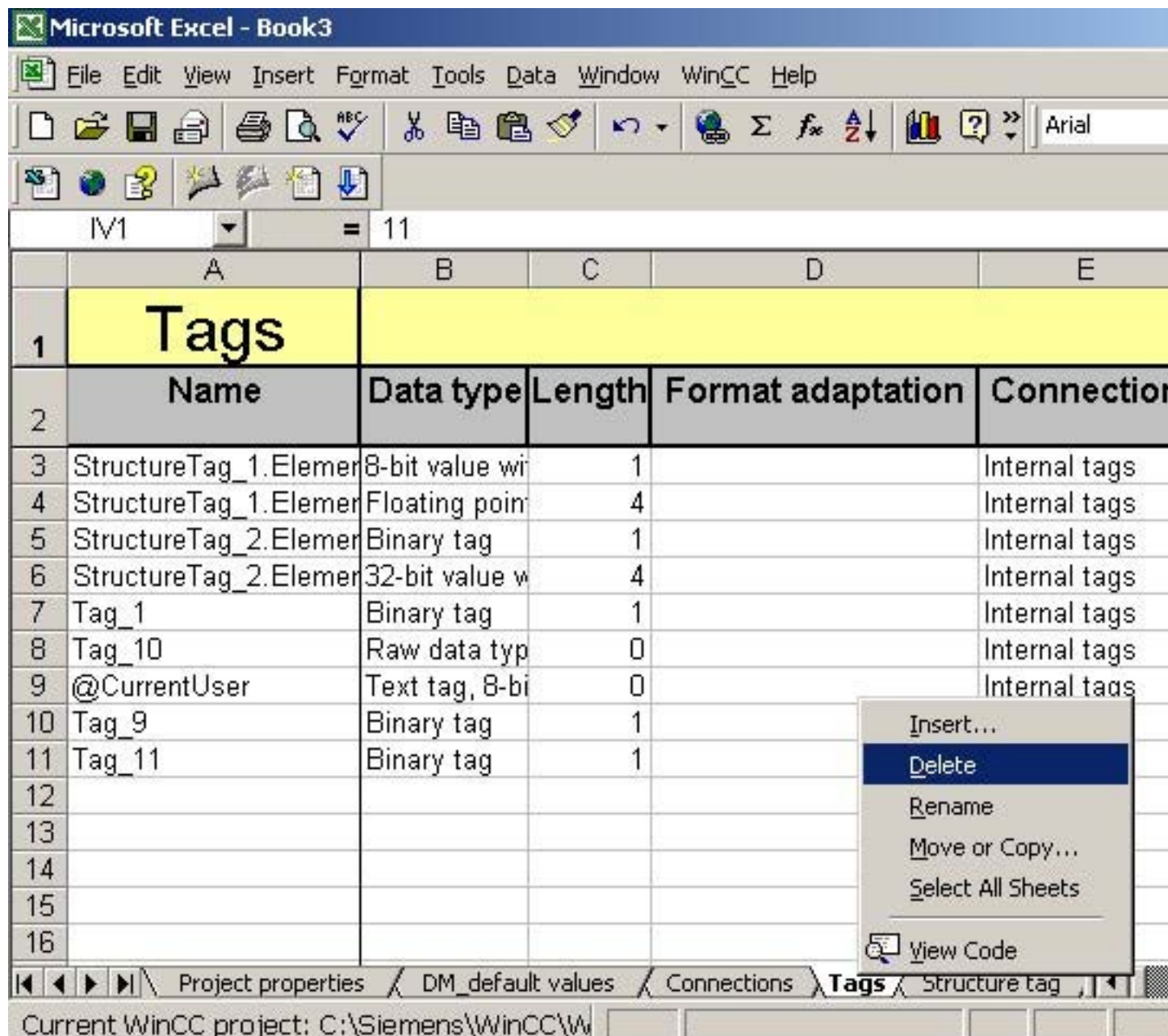
	A	B	C	
1	Tags			
2	Name	Data type	Length	Form
3	Tag_1	Binary tag	1	
4	Tag_2	8-bit value with sign	1	
5	Tag_3	8-bit value without sign	1	
6	Tag_4	Text tag, 8-bit font	0	
7	Tag_5	Floating point number 32-bit IEEE	4	
8	Tag_6	Raw data type	0	
9	Tag 7	Floating point number 64-bit IEEE	8	Double

2. Open the row pop-up menu using the right mouse button and select "WinCC - delete selection". All selected objects are deleted.



Deleting a Worksheet

1. Open the pop-up menu related to the sheet using the right mouse button and select "Delete". The sheet is deleted together with the data configured on it.



Deleting without connection to WinCC

In order to delete objects, it is not necessary that a connection exists to the assigned WinCC project. The next time a connection is established to the assigned WinCC project, the data is compared and all objects deleted from the project folder are also deleted from WinCC.

Deleting with connection to WinCC

Objects deleted from a project folder are also deleted from WinCC, provided they already exist in the WinCC project. Please note that deleted data cannot be restored.

Copying data

Introduction

In Excel you can copy data with pop-up menus, the dropdown menu, a toolbar command or a key combination. In the configuration tool, data can be copied and pasted without restriction. Data that is inserted is checked and corrected, if necessary. To copy entire WinCC objects, proceed as described below.

Copying WinCC objects

1. Access the "Project Properties" sheet and set the "Use Default Values" parameter to "No".
2. Copy the entire rows of the objects.
3. Paste the copied objects into a table of the same type. Tags can for example only be pasted into a tag table.
4. The configuration tool checks the pasted data. Since a unique name must be used, the Office Assistant prompts whether a unique name is to be generated. Confirm with "OK".

Note

If names have already been specified before copying the WinCC objects, only the object parameters can be copied.

Cutting data

Introduction

In Excel, any data can be cut and pasted at any location. This is however not possible in the configuration tool. In order to cut data in the WinCC configuration tool, you must always select the entire object. The cut data can only be cut and pasted between sheets of the same type. All options to cut data provided by Excel are available for use.

Cut

1. Select the objects to be cut. You must either select the entire row or all object parameters.
2. Cut the data.

Paste

1. Select either the area in which the cut data is to be pasted, or select the initial cell as starting point for the insertion of the data. If an area is selected, the cut size and paste size must be the same.
2. Paste the data.

Sorting data

Introduction

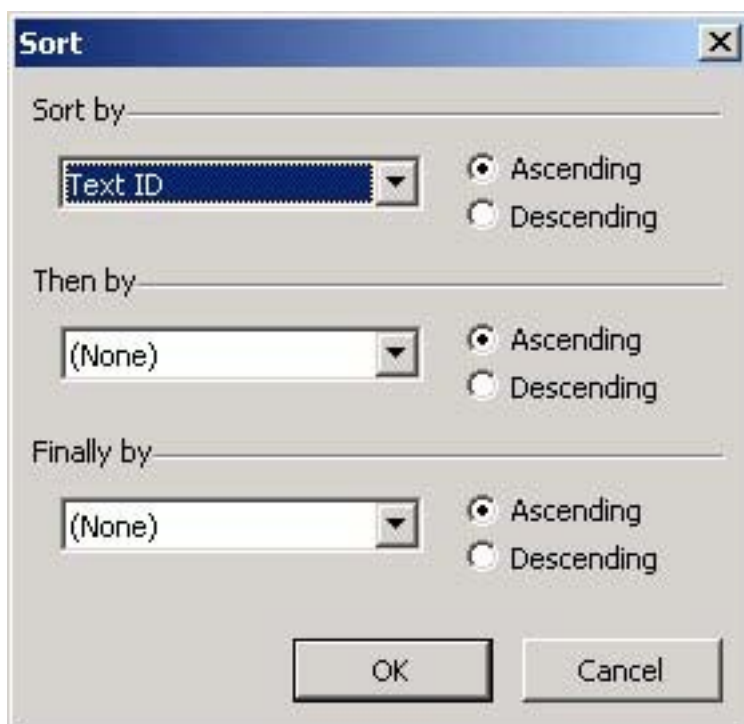
The WinCC configuration tool can be used to sort the data in the project folder by columns. This can be done in two ways.

Sorting by double-click

1. In order to sort data with the WinCC configuration tool in ascending or descending order, simply double-click the parameter heading. The data on the sheet is then sorted according to this column.

Sorting via the menu

1. Select the "Sort" item in the Control menu "Data". The "Sort" dialog is displayed.
2. In the "Sort" dialog, you can specify up to three columns to sort by ascending or descending order. Click "OK" to confirm the sort order.



Setting a filter

Introduction

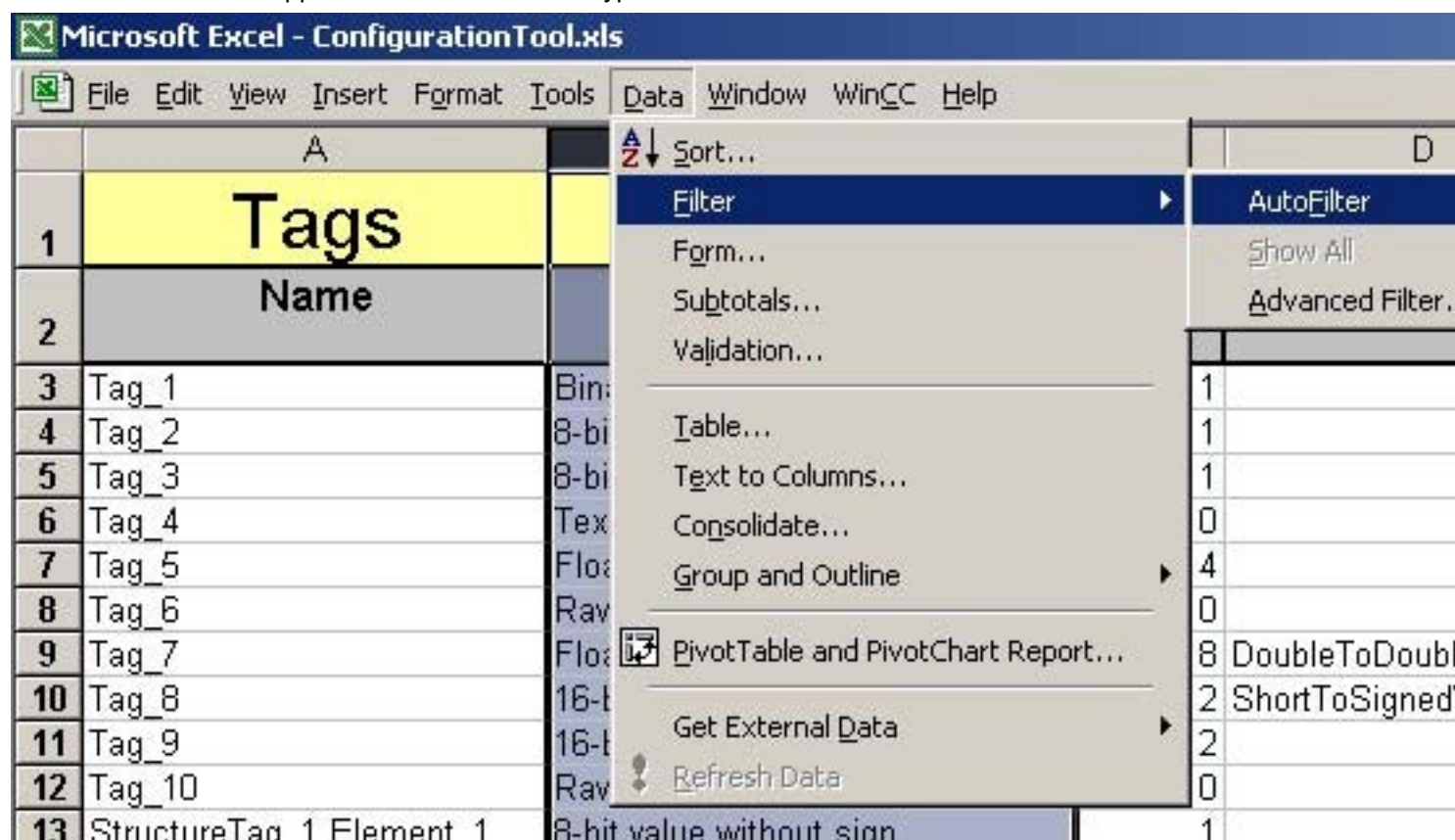
Filters are standard Excel functions. They can be used to organize data in the table in a more concise manner. When using the tag table, you can for example set a filter in the "Data type" column to view only the tags of one data type at a time.

Note

If a filter is applied, and you attempt to write data to WinCC or to delete data, using the pop-up menu, only the visible data are written or deleted.

Setting filters

1. Highlight the entire "Data type" column in the "Tags" sheet.
2. Select the option "Filter" in the "Data" menu. Select the suboption "Auto Filter". A filter is applied in the column "Data type".



3. The dropdown list box attached to the top cell in the "Data type" column indicates that a filter is set. In the dropdown list, you can now select the filter you wish to set, e.g. "Raw data type".

Microsoft Excel - ConfigurationTool.xls

File Edit View Insert Format Tools Data Window WinCC Help

	A	B	C	D
1	Tags			
2	Name	(All) (Top 10...) (Custom...)	Length	Format a
3	Tag_1	16-bit value with sign	1	
4	Tag_2	16-bit value without sign	1	
5	Tag_3	32-bit value with sign	1	
6	Tag_4	8-bit value with sign	0	
7	Tag_5	8-bit value without sign	4	
8	Tag_6	Binary tag	0	
9	Tag_7	Data type	8	DoubleToDon
10	Tag_8	Floating point number 32-bit IEEE 754	2	ShortToSign
11	Tag_9	Floating point number 64-bit IEEE 754	2	
12	Tag_10	Raw data type	0	
13	StructureTag_1 Element_1	Text tag, 8-bit font	1	

4. Only tags of the data type "Raw data type" are shown in the table.

Microsoft Excel - ConfigurationTool.xls

File Edit View Insert Format Tools Data Window WinCC Help

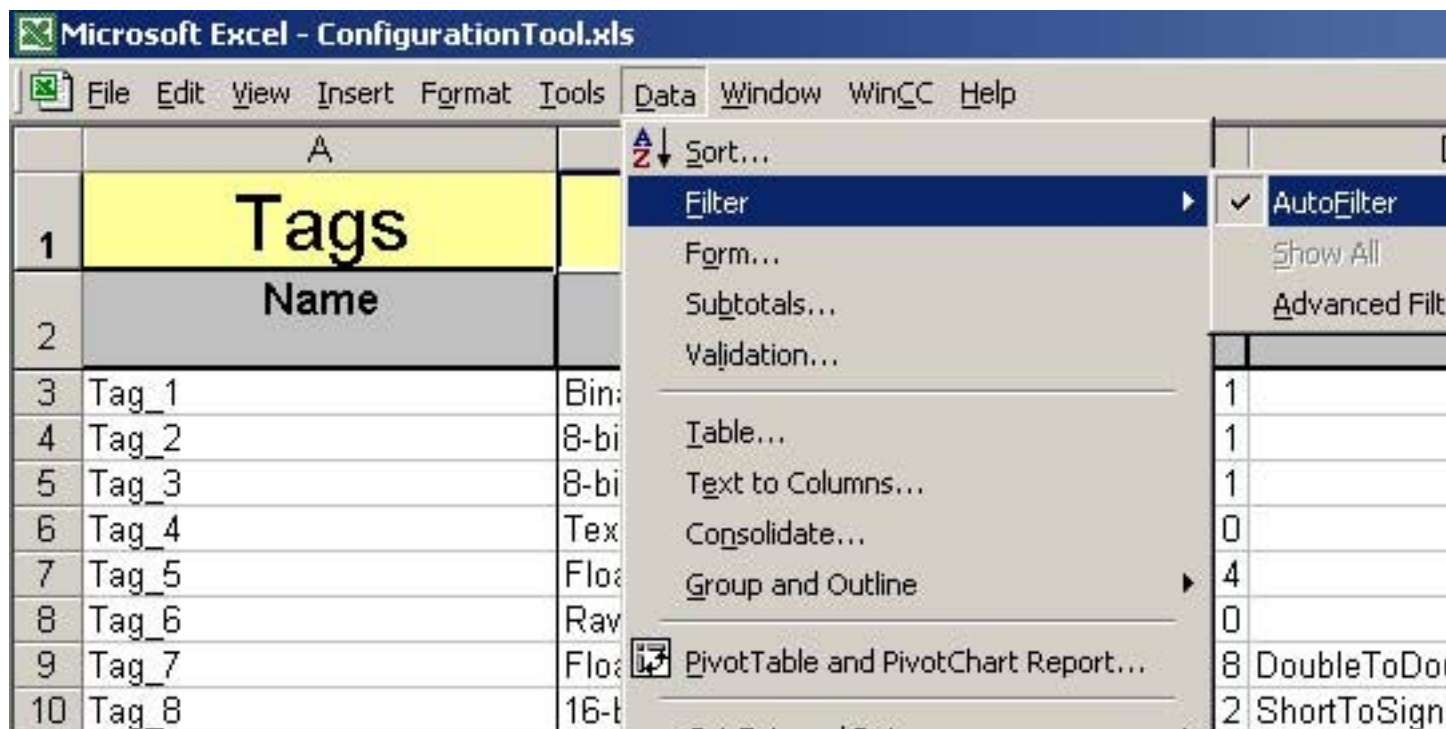
	A	B	C
1	Tags		
8	Tag_6	Raw data type	0
12	Tag_10	Raw data type	0
17			
18			

Resetting filters

1. In the dropdown list box of the filter, select item "(All)". All tags in the sheet are displayed.



2. In order to remove a filter for good, click the "Data" menu. Select the "Filter" option. Select the suboption "Auto Filter". The filter is removed.



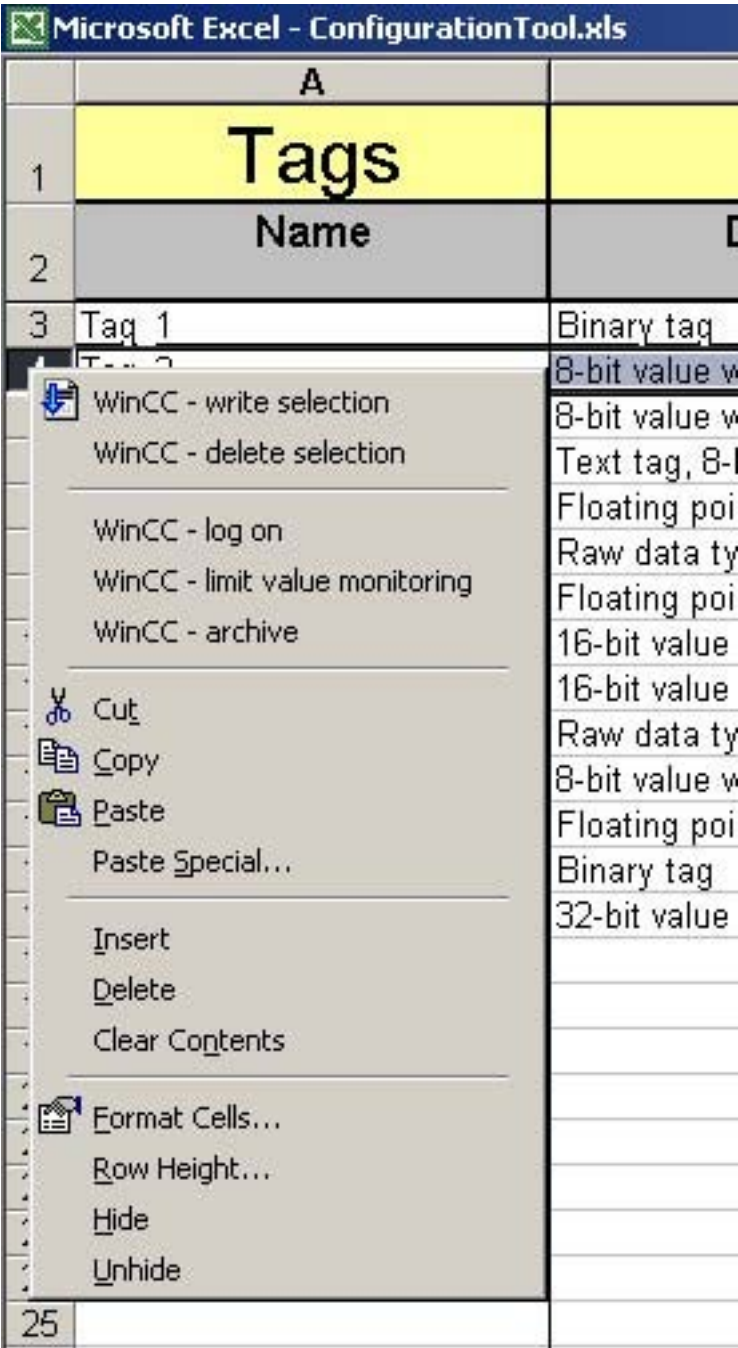
1.8.5.6 Creating objects by means of the pop-up menu of the tag table

Creating objects by means of the pop-up menu of the tag table

Introduction

The WinCC configuration tool provides the option of creating objects by using the tag table. This is done by means of the tag table row pop-up menu. This menu enables you to easily

create single messages, limit value monitoring processes and archive tags. The procedures for this are described in the chapters below.



See also

Creating archive tags from the tag table (Page 213)

Creating limit value monitoring from the tag table (Page 222)

Creating single messages from the tag table (Page 217)

Creating archive tags from the tag table**Introduction**

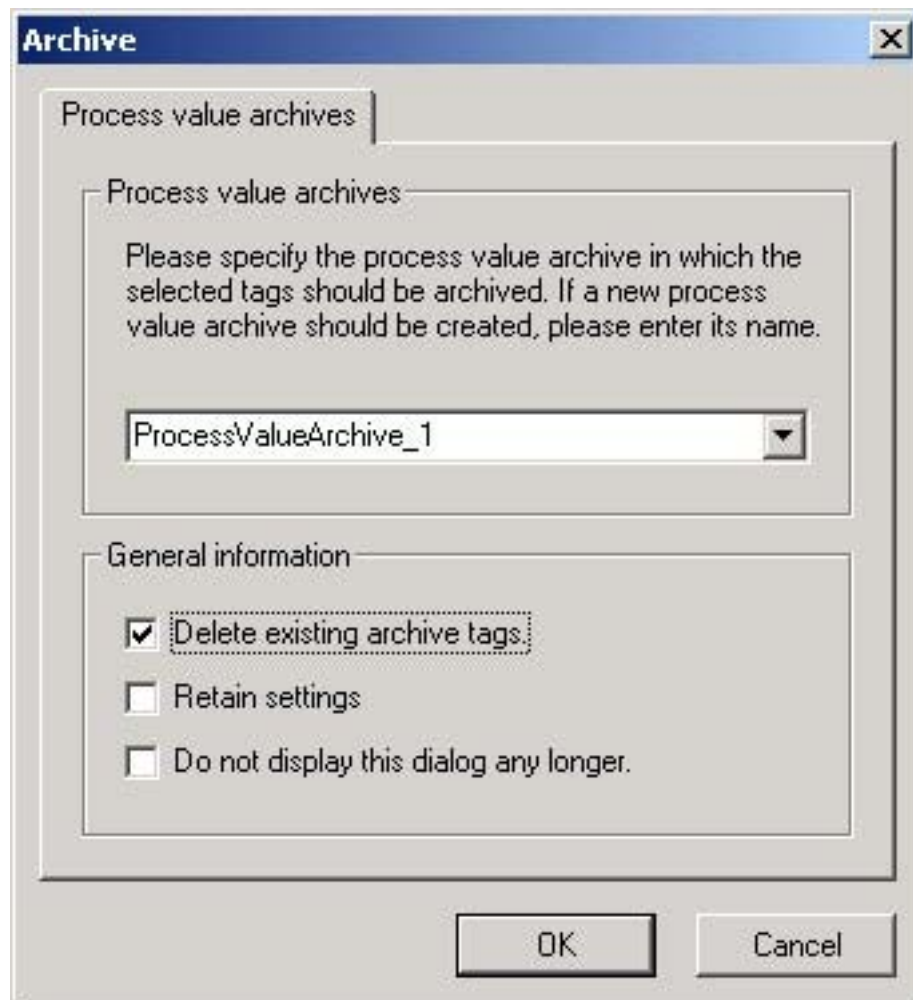
The WinCC configuration tool provides the option of archiving tags from the tag table. This is done by means of the tag table pop-up menu.

Procedure

1. Select all tags you wish to archive.
2. Open the row pop-up menu and select "WinCC - archive". The "Archive" dialog is displayed.

27

3. In the "Archive" dialog, you can select the process value archive to be used for archiving using the dropdown list box. If you wish to create a new process value archive, enter the name of the process value archive in the dropdown list box. Click "OK" to set up the archive tags.



4. The relevant process value archive tags are created on the process value archive table.

Microsoft Excel - ConfigurationTool.xls			
	A	B	
1	Process value archives	Archive tag	
2	Archive name		
3		Tag	Ar
4	ProcessValueArchive_1		
5		Tag_1	Tag
6		Tag_2	Tag
7		Tag_3	Tag
8			
9			

Creating single messages from the tag table

Introduction

The WinCC configuration tool provides the option of creating single messages from the tag table. This is done by means of the tag table pop-up menu.

Note

The settings in the dialogs can be adjusted by means of the "Alarm Logging Default Value" sheet.

Procedure

1. Select all tags for which you wish to generate messages.
2. Open the row pop-up menu and select "WinCC - log on". The "Log on" dialog is displayed.

WinCC - write selection
WinCC - delete selection

WinCC - log on

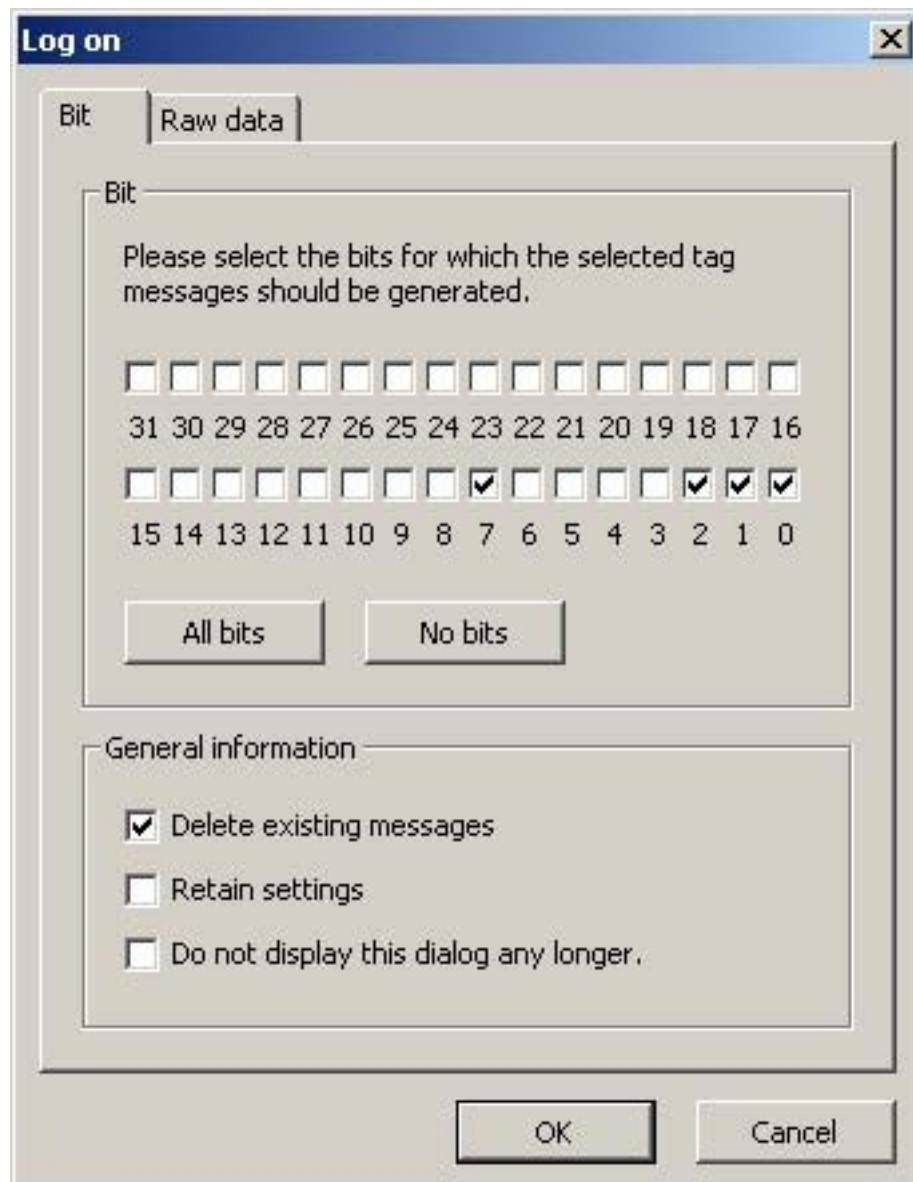
WinCC - limit value monitoring
WinCC - archive

Cut
Copy
Paste
Paste Special...

Insert
Delete
Clear Contents

Format Cells...
Row Height...
Hide
Unhide

3. Use the "Log on" dialog to select the tag bits for which single messages are to be generated. Select the "Delete existing messages" check box if you wish to delete any messages that might already exist for the tags and generate new ones. If this check box is not selected, existing single messages are retained and only those that do not exist generated.



4. In the case of raw data tags, enter the number of single messages to be generated.

The image shows a 'Log on' dialog box with a blue title bar and a close button (X) in the top right corner. Inside the dialog, there are two tabs: 'Bit' and 'Raw data'. The 'Raw data' tab is selected. Below the tabs, there is a section titled 'Raw data' containing the text: 'Please specify how many messages should be generated for a selected raw data tag.' Below this text is a text input field with the number '1' entered. At the bottom of the dialog, there is a section titled 'General information' containing three checkboxes: 'Delete existing messages' (checked), 'Retain settings' (unchecked), and 'Do not display this dialog any longer.' (unchecked). At the very bottom of the dialog are two buttons: 'OK' and 'Cancel'.

Click "OK" to generate the messages.

5. The relevant messages are created in the single message table.

Microsoft Excel - ConfigurationTool.xls					
	A	B	C	D	E
1	Single messages				
2	Number	Class	Type	Group	Message tag
3	1	Error	Alarm		
4	2	Error	Alarm		Tag_9
5	3	Error	Alarm		Tag_9
6	4	Error	Alarm		Tag_9
7	5	Error	Alarm		Tag_9
8	6	Error	Alarm		Tag_10

Creating limit value monitoring from the tag table

Introduction

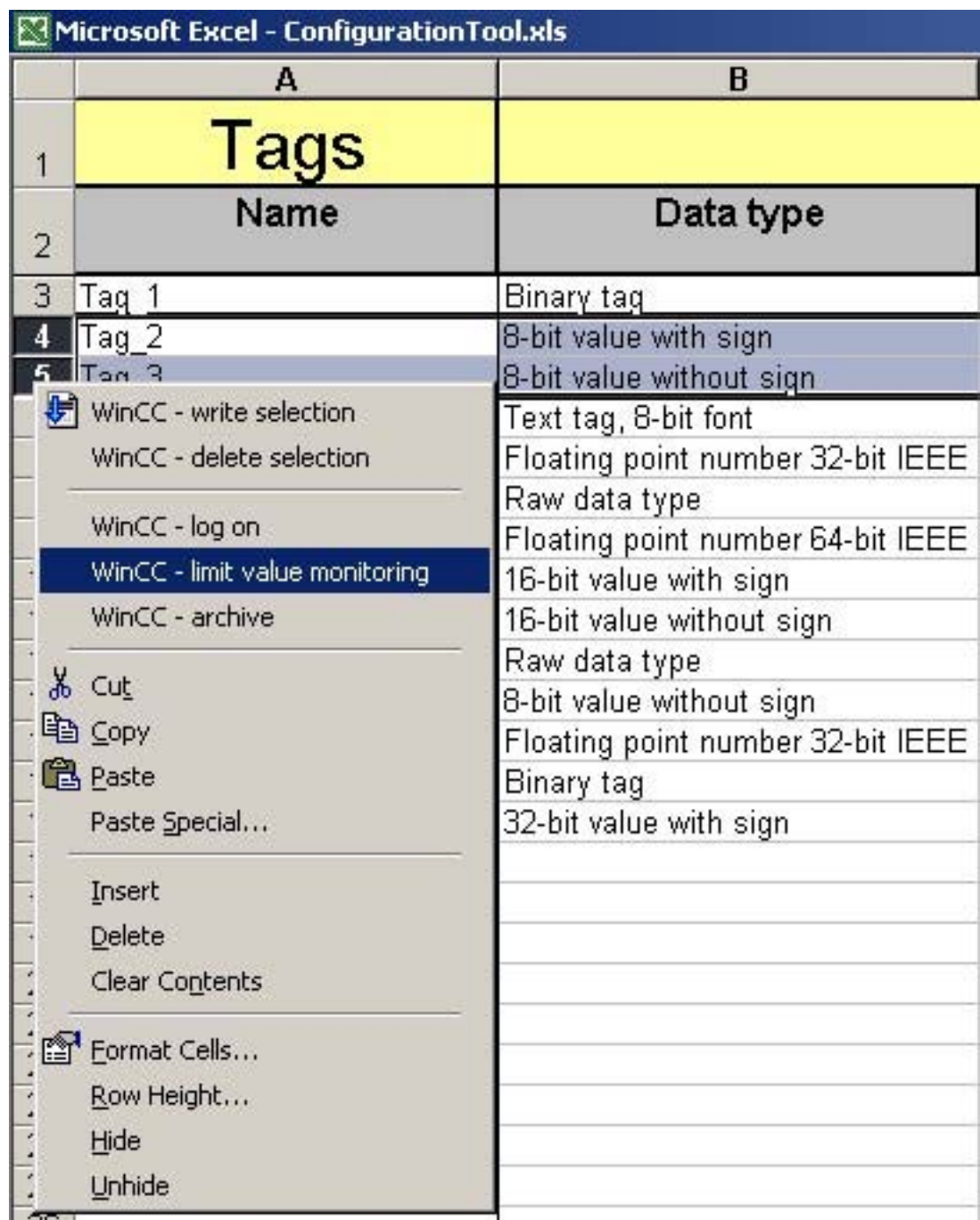
The configuration tool provides the option of creating limit value monitoring from the tag table. This is done by means of the tag table pop-up menu.

Note

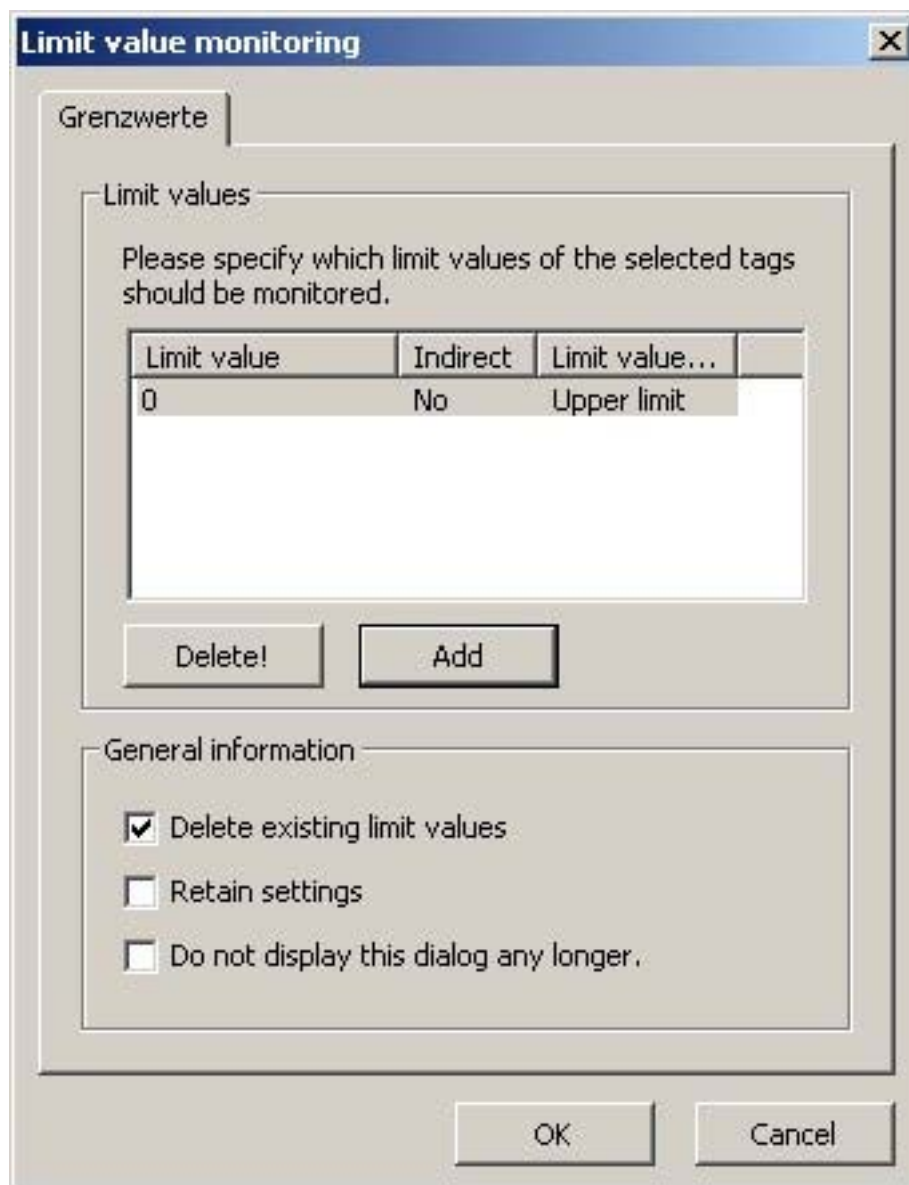
The settings in the dialogs can be adjusted by means of the "Alarm Logging Default Value" sheet.

Procedure

1. Select the tags for which you wish to set up limit value monitoring.
2. Open the row pop-up menu of the tag table and select "WinCC - limit value monitoring".
The "Limit value monitoring" dialog is displayed.



- Limit values that exist in the default values table are already entered in the "Limit value monitoring" dialog. Click the "Add" button to add further limit values for the tags. The "Add limit value" dialog is displayed.



4. You can enter a limit value in the "Add limit value" dialog. Ensure that the new limit value is within the actual limits determined by the tag. If you wish to read the limit value indirectly from a tag, select the "Indirect" check box and enter the name of an existing tag instead of the limit value. Use the "Limit" dropdown list to define whether the limit value is an upper or lower limit. Click "OK" to accept the limit value in the "Limit value monitoring" dialog.

Add limit value

Add limit value

Define the limit value which should be added to the limit value list.

Limit value / Limit value tag:

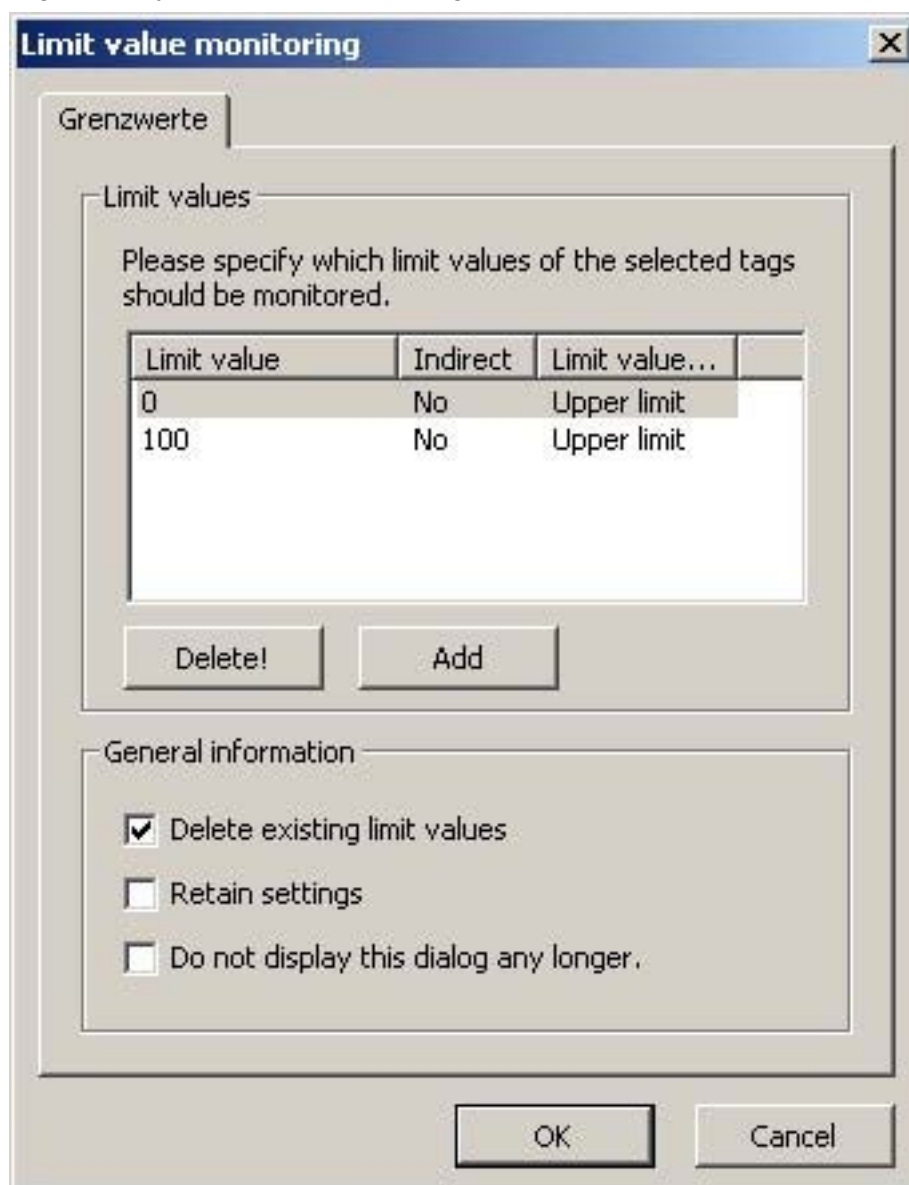
100 ☐ Indirect

Limit value type:

Upper limit

OK Cancel

5. Select the "Delete existing limit values" check box if you wish to delete any limit values that might already exist for the selected tag. Click "OK" to create the limit values.



The dialog box is titled "Limit value monitoring" and has a tab labeled "Grenzwerte". It contains two main sections: "Limit values" and "General information".

Limit values section:

Please specify which limit values of the selected tags should be monitored.

Limit value	Indirect	Limit value...
0	No	Upper limit
100	No	Upper limit

Buttons: Delete! Add

General information section:

- ☒ Delete existing limit values
- ☐ Retain settings
- ☐ Do not display this dialog any longer.

Buttons: OK Cancel

6. The limit values for the tags are generated on the "Limit value monitoring/Limit" sheet.

Microsoft Excel - ConfigurationTool.xls			
	A	B	
1	Limit value monitoring	Limit value	
2	Tag		A
3		Limit value	L
4	Tag_2		
5		0	
6		100	
7	Tag_3		
8		0	
9		100	
10			

1.8.5.7 Changing the data storage location

Changing the data storage location

Introduction

The configuration tool allows you to change the storage location of data. It is for example possible to transfer the project folder to a different computer or change the location at which the WinCC project is stored.

See also

Transferring a WinCC project (Page 227)

Transferring a project folder (Page 228)

Transferring a WinCC project

Introduction

You can transfer a WinCC project connected to a project folder to another storage location without problems. You just have to set up a new project connection.

Procedure

1. Close the WinCC project and transfer it to the desired target location. Open the project folder linked to the project. Use the dropdown menu or the toolbar to establish a new project connection.
2. The Excel Office Assistant for "Project connection" appears. Confirm with "Yes" to record the new path of the WinCC project in the project folder.
3. The "Open" dialog is displayed. Select the WinCC project and click on "Open" to close the dialog. The project connection is established.

Transferring a project folder

Introduction

The configuration tool allows you to transfer a project folder to another computer. It is thus possible to move a project folder to another PC for further processing.

Procedure

1. Save the existing project folder and transfer it to the new computer.
2. Open the project folder on the new computer. The Excel Office Assistant appears. Click "No" to answer the query.
3. The configuration tool asks you whether you wish to update the references. Click "Yes" to update the references of the project folder on the new computer.

1.8.6 Diagnostics

1.8.6.1 Diagnostics

Introduction

The configuration tool provides you with the Office Assistant and an error column for diagnostics and fault identification purposes. The Office Assistant supports you in the configuration of data. Each modification or correction executed by the configuration tool is initiated through the Office Assistant.

If errors occur during writing to WinCC, they are displayed in the error column. In addition, the "Error list" table is generated. This table lists all project folder errors.

See also

Error column (Page 229)

Operation of the "Read Error" sheet (Page 188)

1.8.6.2 Error column

Introduction

Each sheet on which data can be configured includes an error column. In the event of an error, the respective error text is shown in this column. The list below shows all error texts available in the configuration tool.

Microsoft Excel - ConfigurationTool.xls					
	A	AN	CK	CL	C
1	Single messages				
2	Number	Standard DI	Error text		
3	1		OK		
4	2		The set message tag does not		
5	3		The set message tag does not		
6	4		The set message tag does not		
7	5		The set message tag does not		
8	6		The set message tag does not		
9					

Error list

Error text	Cause	Corrective measure
OK	no error	no error
General error during creation!	Writing the entire data block to WinCC has failed.	Is the connection to WinCC still intact?
Error on creating the object!	Was the object created directly in WinCC?	
General error during modification!	Writing the entire data block to WinCC has failed.	Is the connection to WinCC still intact?
Error on modifying the object!	Was the object deleted directly from WinCC?	
The set connection does not yet exist!	The set connection has still not been written to WinCC.	Write the set connection to WinCC.
The message block name does not yet exist in the text library!	The text library has still not been written to WinCC.	Write the text library to WinCC.
The communication driver cannot be installed!	The set communication driver is not installed on the system.	Use a different communication driver or install the one set.
The set stroke tag does not yet exist!	The set stroke tag has not yet been written to WinCC.	Write the set stroke tag to WinCC.

Error text	Cause	Corrective measure
The set group does not yet exist!	The set group has not yet been written to WinCC.	Write the set group to WinCC.
The set structure type does not yet exist!	The set structure type has not yet been written to WinCC.	Write the set structure type to WinCC.
The message category name does not yet exist in the text library!	The text library has still not been written to WinCC.	Write the text library to WinCC.
One or more group message tags do not yet exist!	The set group message tag does not yet exist in WinCC.	Write the set group message tag to WinCC.
The message type name does not yet exist in the text library!	The text library has still not been written to WinCC.	Write the text library to WinCC.
The set message class does not yet exist!	The set message category does not yet exist in WinCC.	Write the set message category to WinCC.
The set message type does not yet exist!	The set message type does not yet exist in WinCC.	Write the set message type to WinCC.
One or more message texts do not yet exist in the text library!	The text library has not yet been written to WinCC.	Write the text library to WinCC.
One or more process value block tags do not yet exist!	The set process value block tags do not yet exist in WinCC.	Write the set process value block tags to WinCC.
Error on creating the information text!	Modifications have been carried out directly in WinCC.	Write all data which do not yet exist in WinCC to WinCC. Close the current project folder. Create a new project folder with a connection to your WinCC project.
Error on creating the Loop-In-Alarm!	Modifications have been carried out directly in WinCC.	Write all data which do not yet exist in WinCC to WinCC. Close the current project folder. Create a new project folder with a connection to your WinCC project.
Errors occurred on writing the higher-level object!	The higher-level object could not be written to WinCC.	Check the error text of the higher-level object. Attempt to correct the error and rewrite to WinCC.
Error on creating the filter!	Modifications have been carried out directly in WinCC.	Write all data which do not yet exist in WinCC to WinCC. Close the current project folder. Create a new project folder with a connection to your WinCC project.
The set tag does not yet exist!	The set tag does not yet exist in WinCC.	Write the set tag to WinCC.
The set message does not yet exist!	The set message does not yet exist in WinCC.	Write the set message to WinCC.
One or more status texts do not yet exist in the text library!	The text library has not yet been written to WinCC.	Write the text library to WinCC.
The set message tag does not yet exist!	The set message tag does not yet exist in WinCC.	Write the set message tag to WinCC.
The set status tag does not yet exist!	The set status tag does not yet exist in WinCC.	Write the set status tag to WinCC.

Error text	Cause	Corrective measure
The set acknowledgement tag does not yet exist!	The set acknowledgement tag does not yet exist in WinCC.	Write the set acknowledgement tag to WinCC.
The set group message does not yet exist!	The set group message does not yet exist in WinCC.	Write the set group message to WinCC.
The set blocking tag does not yet exist!	The set blocking tag does not yet exist in WinCC.	Write the set blocking tag to WinCC.
The set limit value tag does not yet exist!	The set limit value tag does not yet exist in WinCC.	Write the set limit value tag to WinCC.

1.8.7 Tips and tricks

1.8.7.1 Tips and tricks

Introduction

The following chapters contain some tips and tricks designed to assist you in working with the WinCC configuration tool.

See also

Packages (Page 242)
 Address strings for Simatic S7 Protocol Suite (Page 240)
 Special characters (Page 239)
 VBA macros (Page 239)
 Address generation (Page 235)
 Row limitation (Page 235)
 Changing the worksheet (Page 233)
 Example of a quantity structure in the configuration tool (Page 231)

1.8.7.2 Example of a quantity structure in the configuration tool

Introduction

Using the configuration tool, you can for example create a large number of tags, messages and archive tags in WinCC. Performance, however, is influenced by the hardware and Office version used.

The example below is based on a typical configuration with appropriate values.

Quantity structure - Example

Configuration

A WinCC project is read into Excel with the configuration tool.

A new WinCC project is created with the configuration tool. Data from the first project are written to the new WinCC project.

Quantity structure

- 20,000 internal tags
- 40,000 process tags
- 40,000 archive tags
- 10,000 messages
- 80,000 texts

Hardware used

- Pentium 4, 3.1 GHz
- Memory: 1024 MByte
- Virtual memory: 756 MByte

Software used

- Windows XP Professional SP1
- Internet Explorer 6.0 SP1
- Office 2003
- WinCC V6.0 SP3

Performance

- Reading to Excel: approx. 30 minutes
- Writing to WinCC: approx. 3.5 hours

Note

The values in this example depend upon additional factors, such as system configuration and configuration of the WinCC project.

See also

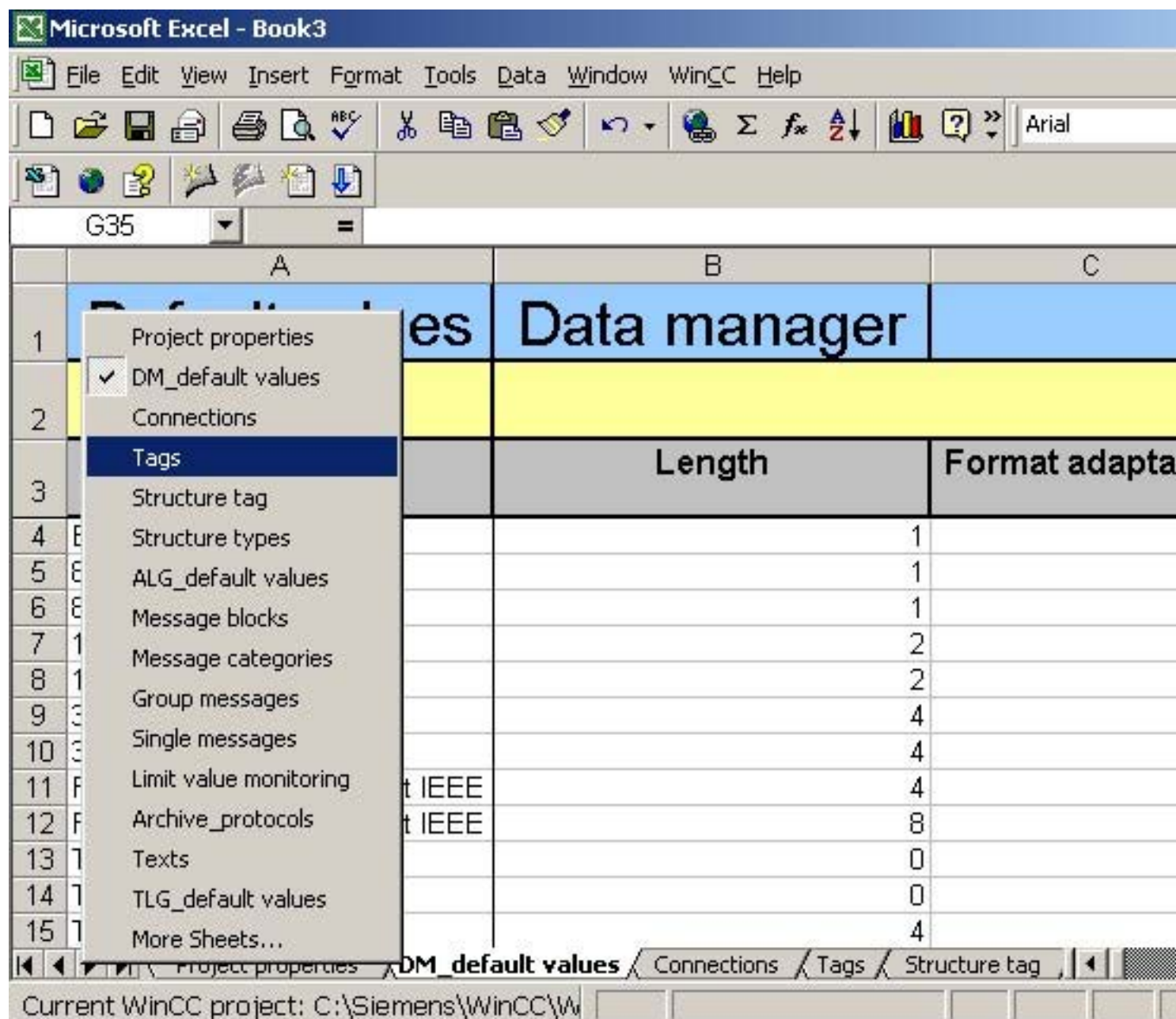
System requirements (Page 113)

1.8.7.3 Changing the worksheet

Introduction

From a certain number of sheets, it becomes necessary the change between sheets. Excel provides a useful tool for this.

Use the pop-up menu accessible at the left bottom corner of the window to switch between worksheets.



1.8.7.4 Row limitation

Introduction

During configuration in Excel, you should not use the maximum number of available rows, which is 65536. The number of rows used should thus be restricted, as the performance is greatly affected if Excel sheets are filled to the limit. The configuration tool provides you with the option to distribute the data to several worksheets.

The relevant default value for row limitation in the configuration tool is 16,000 rows per sheet. We recommend using this row limitation.

1.8.7.5 Address generation

Introduction

In the configuration tool, you can use standard functions provided by Excel to generate addresses for external tags.

The following example illustrates the generation of the address string of a SIMATIC S7 Protocol Suite with the "Signed 8-bit value" data type.

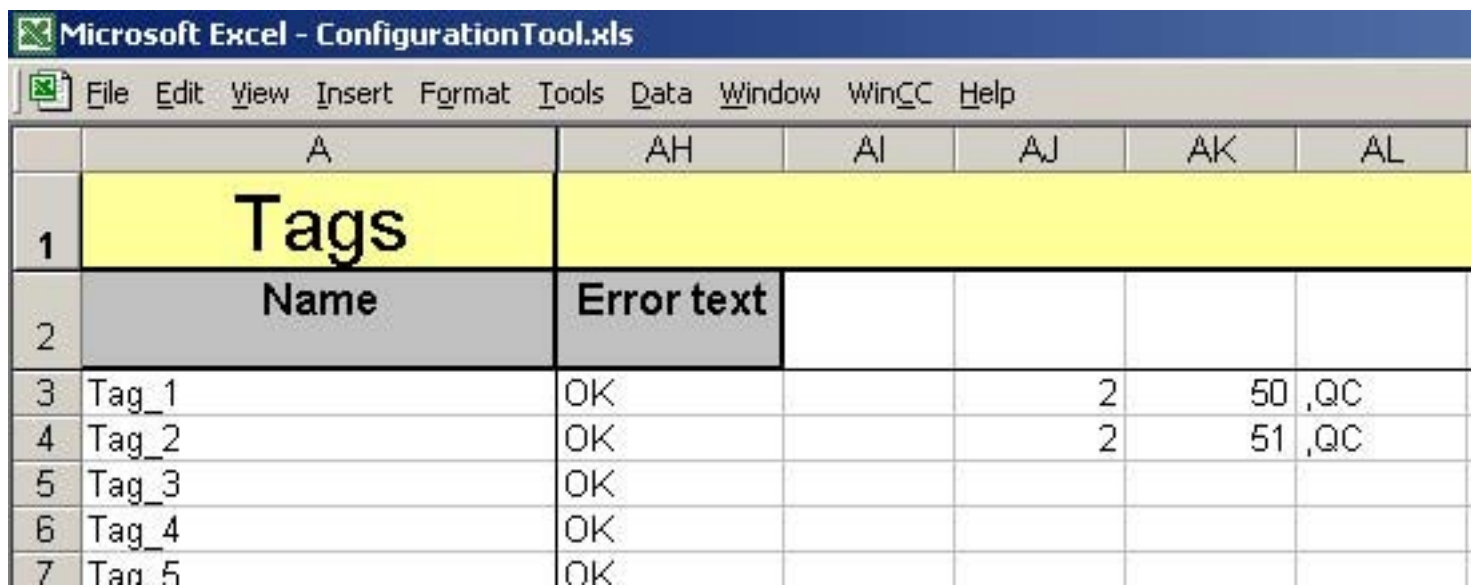
Procedure

The address string of the example is structured as follows:

DB50,DBB0,QC

The data block number 50 and byte number 0 are editable components of the address string. The QC string is optional and is used to define the quality code.

1. Define a column in the user-defined area of the tag table for each editable component of the address string. Complete the first and second row.



The screenshot shows a Microsoft Excel window titled "Microsoft Excel - ConfigurationTool.xls". The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Window, WinCC, and Help. The spreadsheet has columns labeled A, AH, AI, AJ, AK, and AL. Row 1 is highlighted in yellow and contains the text "Tags" in cell A1. Row 2 has a grey header for "Name" in column A and "Error text" in column AH. Rows 3 through 7 contain data for tags Tag_1 through Tag_5. The "Error text" column (AH) contains "OK" for all tags. The "AJ" column contains the value "2" for Tag_1 and Tag_2. The "AK" column contains the values "50" and "51" for Tag_1 and Tag_2 respectively. The "AL" column contains the text ",QC" for Tag_1 and Tag_2.

	A	AH	AI	AJ	AK	AL
1	Tags					
2	Name	Error text				
3	Tag_1	OK		2	50	,QC
4	Tag_2	OK		2	51	,QC
5	Tag_3	OK				
6	Tag_4	OK				
7	Tag_5	OK				

2. Highlight the first two addresses. Generate the remaining addresses automatically using the autofill function.

Microsoft Excel - ConfigurationTool.xls						
File Edit View Insert Format Tools Data Window WinCC Help						
	A	AH	AI	AJ	AK	AL
1	Tags					
2	Name	Error text				
3	Tag_1	OK		2	50	,QC
4	Tag_2	OK		2	51	,QC
5	Tag_3	OK		2	52	,QC
6	Tag_4	OK		2	53	,QC
7	Tag_5	OK		2	54	,QC
8	Tag_6	OK		2	55	,QC
9	Tag_7	The set connection does not yet exist!				
10	Tag_8	The set connection does not yet exist!				
11	Tag_9	OK				
12	Tag_10	OK				
13	StructureTag_1.Element_1	OK				
14	StructureTag_1.Element_2	OK				
15	StructureTag_2.Element_3	OK				

3. In the address column, enter the formula shown below. The column labels AJ3, AK3 and AL3 depend on where the address string components were set up in the user-defined area.

Microsoft Excel - ConfigurationTool.xls					
File Edit View Insert Format Tools Data Window WinCC Help					
	A	E	F	G	H
1	Tags				
2	Name	Connection	Group	Address	Update
3	Tag_1	Internal tags			For entire project
4	Tag_2	Internal tags			For entire project
5	Tag_3	Internal tags			For entire project
6	Tag_4	Internal tags			For entire project
7	Tag_5	Internal tags			For entire project
8	Tag_6	Internal tags			For entire project
9	Tag_7	ext_Connection_1		= "DB"&AJ3&" ,DBB"&AK3&AL3	
10	Tag_8	ext_Connection_1			
11	Tag_9	ext_Connection_1			
12	Tag_10	ext_Connection_1			

4. Click the first address. The remaining address strings are generated automatically using the autofill function.

Microsoft Excel - ConfigurationTool.xls					
File Edit View Insert Format Tools Data Window WinCC Help					
	A	E	F	G	H
1	Tags				
2	Name	Connection	Group	Address	Update
3	Tag_1	Internal tags			For entire projec
4	Tag_2	Internal tags			For entire projec
5	Tag_3	Internal tags			For entire projec
6	Tag_4	Internal tags			For entire projec
7	Tag_5	Internal tags			For entire projec
8	Tag_6	Internal tags			For entire projec
9	Tag_7	ext_Connection_1		DB2,DBB50	QC
10	Tag_8	ext_Connection_1		DB2,DBB51	QC
11	Tag_9	ext_Connection_1		DB2,DBB52	QC
12	Tag_10	ext_Connection_1		DB2,DBB53	QC

1.8.7.6 VBA macros

Introduction

Since the data in the configuration tool are available in an Excel folder, they can be addressed via VBA macros. This means, for example, that objects can be automatically created by means of a macro.

However, VBA macros should only be used by experienced users because the operational procedures for using the configuration tool differ from those for normal Excel folders.

1.8.7.7 Special characters

Introduction

Microsoft Excel interprets texts which begin with, or include, certain special characters as equations. This can cause problems when working with the configuration tool.

These special characters are "=", "+", and "-". Therefore, these characters should not be used as a part of object names (group names, tag names, texts, etc.).

1.8.7.8 Address strings for Simatic S7 Protocol Suite

Introduction

The address strings are not automatically generated in the configuration tool. You must enter them manually in the configuration tool. The table below lists all address strings for the Simatic S7 Protocol Suite.

Address strings

Table 1-1 Binary tag

Data area	Addressing	Address string
Bit	Bit	DB1,D0.0
Flag	Bit	M0.0
Input	Bit	E0.0
Output	Bit	A0.0

Table 1-2 Unsigned 8-bit value/signed 8-bit value

Data area	Addressing	Address string
DB	Byte	DB1,DBB0
Flag	Byte	MB0
Input	Byte	EB0
Output	Byte	AB0
DB	Word	DB1,DBW0
Flag	Word	MW0
Input	Word	EW0
Output	Word	AW0
Counter	Word	Z0
Timers	Word	T0
DB	Double word	DB1,DD0
Flag	Double word	MD0
Input	Double word	ED0
Output	Double word	AD0

Table 1-3 Unsigned 16-bit value/signed 16-bit value

Data area	Addressing	Address string
DB	Byte	DB1,DBB0
Flag	Byte	MB0
Input	Byte	EB0
Output	Byte	AB0
DB	Word	DB1,DBW0
Flag	Word	MW0

Data area	Addressing	Address string
Input	Word	EW0
Output	Word	AW0
Counter	Word	Z0
Timers	Word	T0
DB	Double word	DB1,DD0
Flag	Double word	MD0
Input	Double word	ED0
Output	Double word	AD0

Table 1-4 Unsigned 32-bit value/signed 32-bit value

Data area	Addressing	Address string
DB	Byte	DB1,DBB0
Flag	Byte	MB0
Input	Byte	EB0
Output	Byte	AB0
DB	Word	DB1,DBW0
Flag	Word	MW0
Input	Word	EW0
Output	Word	AW0
Counter	Word	Z0
Timers	Word	T0
DB	Double word	DB1,DD0
Flag	Double word	MD0
Input	Double word	ED0
Output	Double word	AD0

Table 1-5 Floating point number 32-bit IEEE 754/floating point number 64-bit IEEE 754

Data area	Addressing	Address string
DB	Byte	DB1,DBB0
Flag	Byte	MB0
Input	Byte	EB0
Output	Byte	AB0
DB	Word	DB1,DBW0
Flag	Word	MW0
Input	Word	EW0
Output	Word	AW0
Counter	Word	Z0
Timers	Word	T0
DB	Double word	DB1,DD0
Flag	Double word	MD0
Input	Double word	ED0
Output	Double word	AD0

Table 1-6 Text tag 8-bit character set/text tag 16-bit character set

Data area	Addressing	Address string
DB	Byte	DB1,DBB0
Flag	Byte	MB0
Input	Byte	EB0
Output	Byte	AB0
DB	Word	DB1,DBW0
Flag	Word	MW0
Input	Word	EW0
Output	Word	AW0
Counter	Word	Z0
Timers	Word	T0

Table 1-7 Raw data type

Data area	Addressing	Address string
DB	Byte	RAW_BSEND(DB1,DBB0)
Flag	Byte	RAW_BSEND(MB0)
Input	Byte	RAW_BSEND(EB0)
Output	Byte	RAW_BSEND(AB0)
DB	Word	RAW_BSEND(DB1,DBW0)
Flag	Word	RAW_BSEND(MW0)
Input	Word	RAW_BSEND(EW0)
Output	Word	RAW_BSEND(AW0)
DB	Double word	RAW_BSEND(DB1,DD0)
Flag	Double word	RAW_BSEND(MD0)
Input	Double word	RAW_BSEND(ED0)
Output	Double word	RAW_BSEND(AD0)

1.8.7.9 Packages

Introduction

The configuration tool can also be used to work with tags loaded from packages. To do this, create a new WinCC project in WinCC Explorer (multi-user project) and load the required packages. Read this WinCC project with the configuration tool. The tags provided in the package are then available for selection in the tag dialog.

Creating a New WinCC Project

A new WinCC project can be created from a project folder in which packages are used. The packages used, however, must be loaded into the new WinCC project (multi-user project) after it has been created.

1.9 WinCC Archive ConfigurationTool

1.9 Resources

1.9.1 Introduction

Introduction

The WinCC Archive ConfigurationTool is an Excel add-in. The tool enables simple high-performance configuration of bulk data for archiving WinCC process values.

It enables the handling of quantity structures for which the tag logging editor provides insufficient support.

Working with the WinCC Archive ConfigurationTool

The WinCC Archive ConfigurationTool offers the possibility of creating a new archive. In addition, existing archives can be read out and edited. In cases where extensive modifications must be made, it is recommended to delete existing and recreate a complete archive. In addition, it is possible to rewrite minimal modifications into the WinCC project selectively.

An archive is configured within an Excel sheet. The Excel sheet consists of an archive table and one or more archive tag tables. The archive table serves to configure the general properties of the archive. The archive tag tables serve to configure the archive tags and their properties.

WinCC Archive is not a full substitute for the tag logging editor. The configuration of the timers and the archive configuration (tag logging fast and slow) must still be completed in the tag logging editor.

The data is not checked during entry. However, a check of the entire archive is offered with regard to various aspects. These aspects include the general correctness of the parameters set, the uniqueness of the names used and the existence of the defined objects (tags, times, etc.).

Comparison

	Tag logging editor	WinCC Configuration Tool	WinCC Archive ConfigurationTool
Purpose	Standard configuration interface for the archive system.	Easy-to-use planning interface for complete WinCC projects.	High performance configuration interface for archives with exceptionally high quantity structures.
Configuring the archive system	The archive system is completely configurable.	The archive system is completely configurable except for the archive configuration (tag logging fast and slow).	One archive, respectively, can be configured.

	Tag logging editor	WinCC Configuration Tool	WinCC Archive ConfigurationTool
Quantity structure recommendation	Implementation of up to approx. 100 -200 archive tags.	Implementation of up to approx. 10,000 - 30,000 archive tags.	Implementation from approx. 10,000 archive tags.
Data input	Data input via property dialogs and table interfaces with limited use.	Data input in tabular form with comprehensive support (default values, combo boxes, dialogs, Office Assistant).	Data input in tabular form with restricted support.
Data check	Data are checked during entry.	Data are checked during entry.	Data are checked prior to writing into the WinCC project. Various checks can also be deactivated.
Expandability	None.	Can be extended by VBA macros.	Can be extended by VBA macros.

See also

Quick start (Page 245)

System requirements (Page 244)

1.9.2 System requirements**System requirements**

- Windows XP SP3 / Windows Server 2003 (R2) SP2 / Windows Vista SP1
- Microsoft Excel XP, Office 2003, Office 2007. Visual Basic for applications must be installed.
- Internet Explorer 6 or 7

Note

You will need the "Office Assistant" in Excel, so that the warning messages of the configuration tool will be output.

Working without WinCC

It is possible to use WinCC Archive without WinCC installed. In this case, however, data cannot be read from or written to WinCC. The full scope of configuration options using Excel remains available.

1.9.3 Installation of the Archive ConfigurationTool

The WinCC Archive Configuration Tool can be installed in two different ways.

Procedure

1. During WinCC setup, select "WinCC V7.0 complete" from the "Programs" dialog.
WinCC is installed together with SmartTools, the WinCC ConfigurationTool and WinCC Archive ConfigurationTool.

Start the WinCC Archive ConfigurationTool by selecting "SIMATIC" > "WinCC" > "Tools".

Alternative procedure

It is also possible to install the WinCC Archive ConfigurationTool from the WinCC DVD.

1. Switch to the WinCC DVD directory "InstData\WinCC\setup\Products\WinCCArchive".
2. Double-click setup.exe.
3. Follow the instructions in the dialog boxes.
The WinCC Archive ConfigurationTool is installed.

1.9.4 Quick start

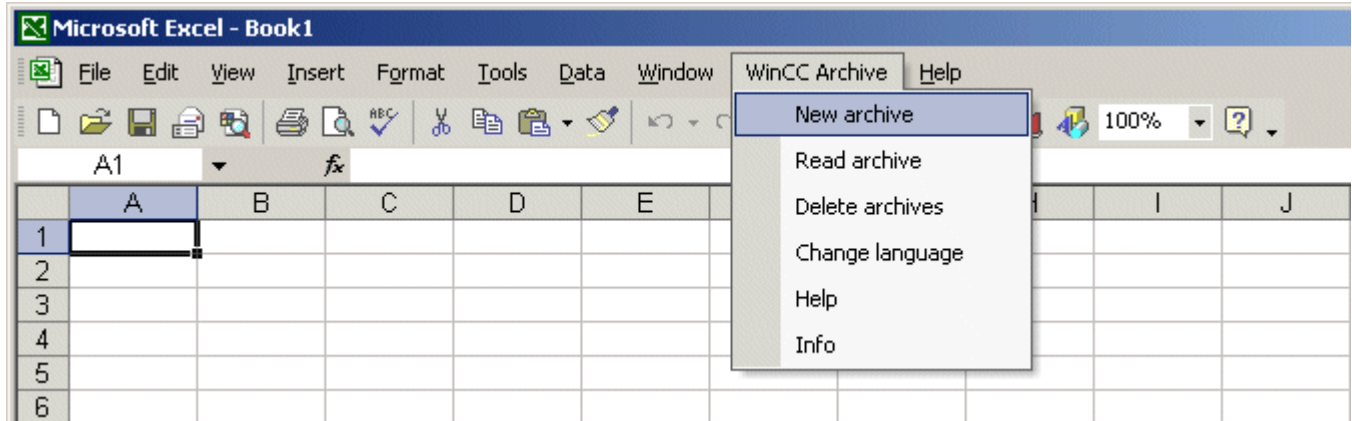
Introduction

This description enables a quick start for using the Excel add-in WinCC Archive. The most important available functions are introduced briefly.

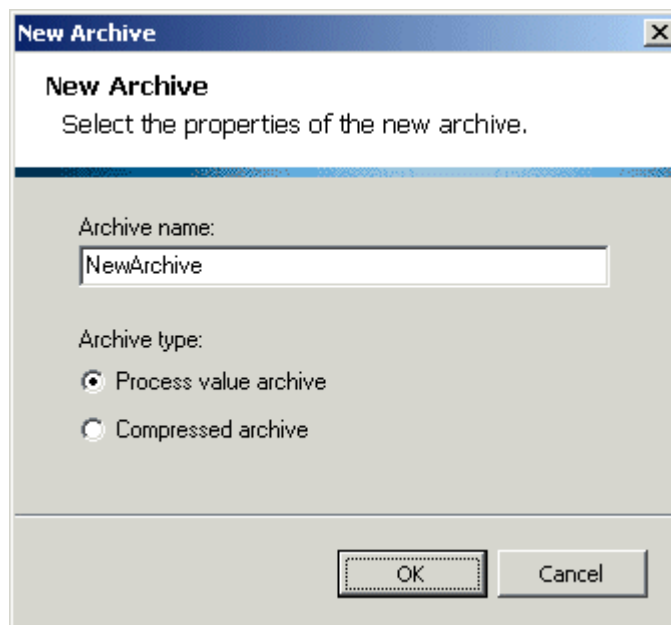
Creating new archives

Procedure

1. Open Excel. The Excel add-in WinCC Archive is automatically started at the same time. The Excel menu contains the "WinCC Archive" item. Select "New Archive" from this menu item.

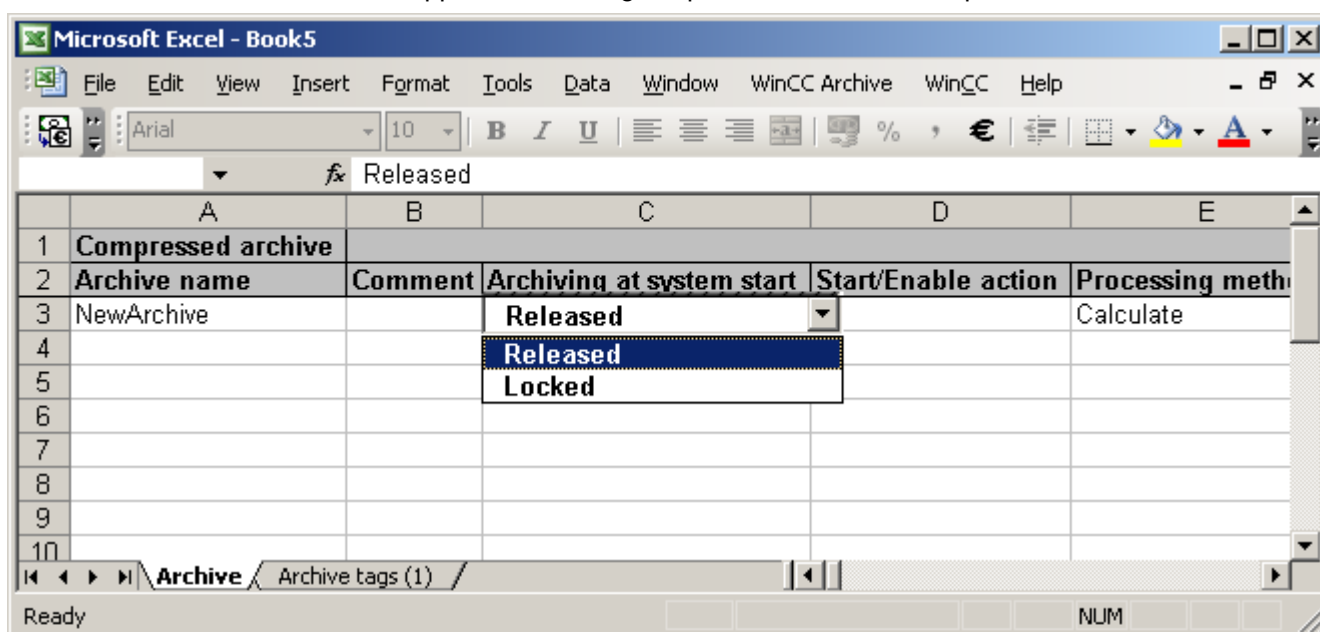


2. A dialog opens in which to define the basic properties of the archive to be created. Enter a name for the archive. Select the desired archive type. This description explains the creation of a process value archive. Click the "OK" button.

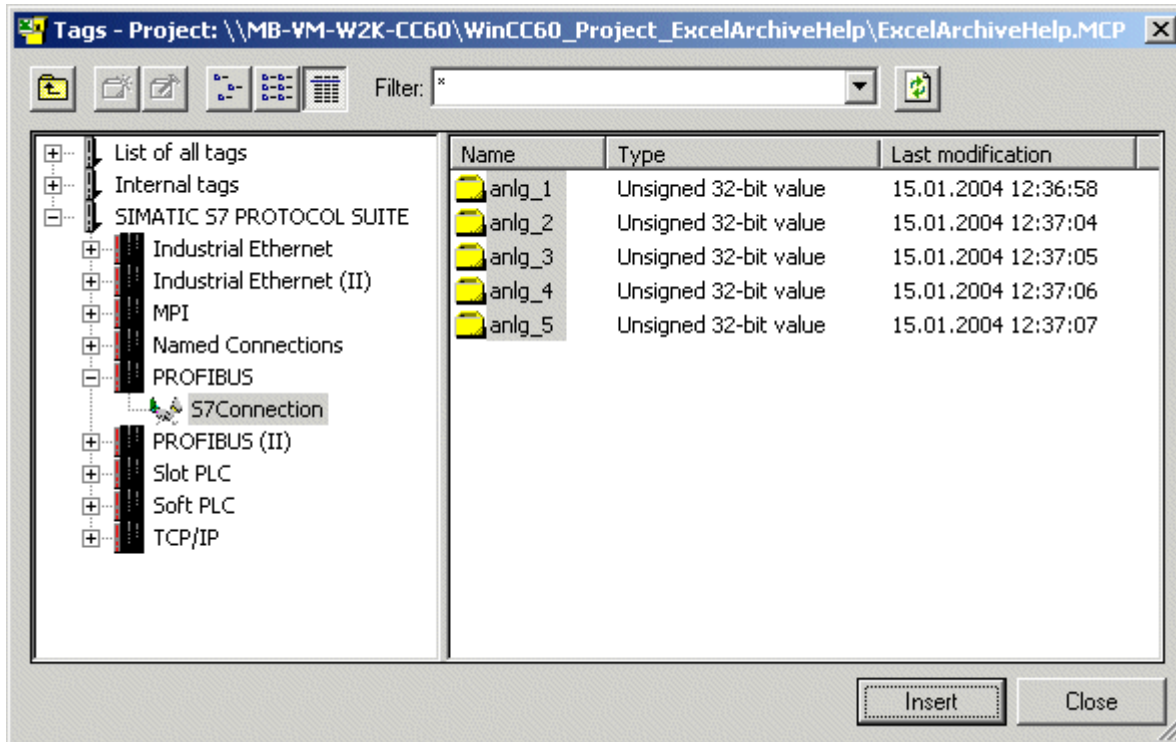


3. A new Excel sheet is created which serves to configure the new archive. The new Excel sheet contains the following list of tables:
 - "Archive": Table to define the properties of the archive.
 - "Archive tags (1)": Table to configure binary and analog archive tags.
 - "Archive tags (2)": Table to configure process controlled archive tags.

4. Define the archive properties in the "Archive" table. All the cells can be edited as required. In the case of several parameters, input is simplified by the availability of a combo box. Double-click in the cell, for example, to enter the "Archiving at system start" parameter. A combo box appears containing the possible values for this parameter.

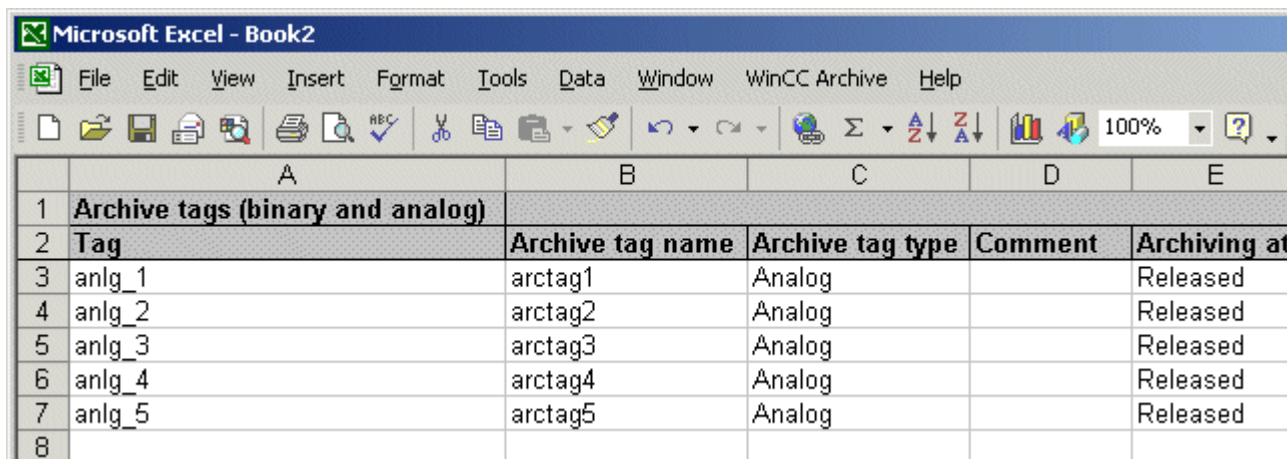


5. Change to the "Archive tags (1)" table. Up to 65634 binary and analog archive tags can be configured in this table. All the cells can be edited as required. In the case of several parameters, input is simplified by the availability of a combo box. If a WinCC project is already open, a further dialog is available in which to select the WinCC tags. Open this dialog by double-clicking in the data area of the "Tag" column.



6. The dialog for selecting WinCC tags includes the multiple selection function. When the dialog is open, it is still possible to work in the Excel table. Select the tags required from the dialog. Select the cells in the Excel table from which the name of the selected tags are to be inserted. Click the "Paste" button in the dialog. The dialog remains open. The names of the selected tags are inserted in the Excel table. Click "Close" to close the dialog.

7. Specify the names of the new archive tags in the "Archive tag name" column. These names must be unique within the archive. Define the correct type of the new archive tags in the "Archive tag type" column. The input for this parameter is supported by a combo box. Define the remaining properties of the new archive tags.

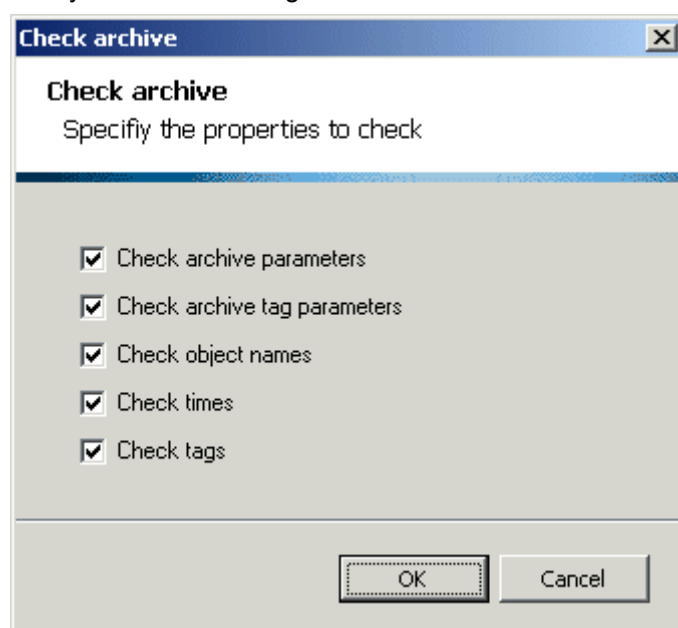


The screenshot shows a Microsoft Excel window titled "Microsoft Excel - Book2". The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Window, WinCC Archive, and Help. The toolbar contains various icons for file operations and data manipulation. The active worksheet displays a table with the following data:

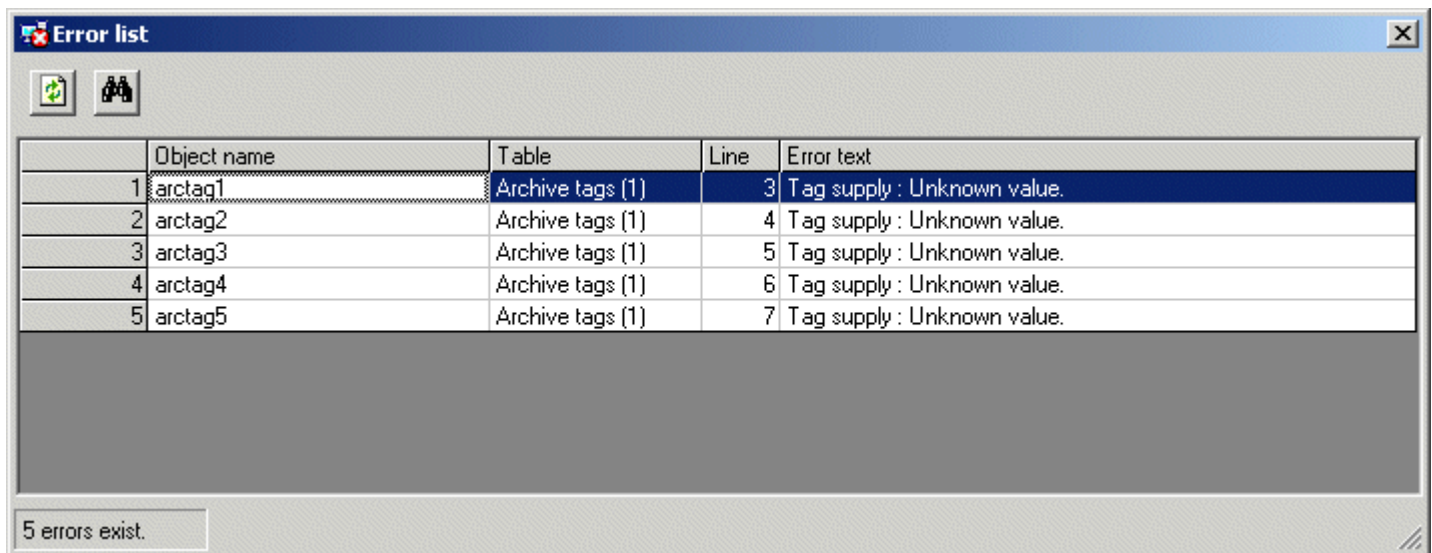
	A	B	C	D	E
1	Archive tags (binary and analog)				
2	Tag	Archive tag name	Archive tag type	Comment	Archiving at
3	anlg_1	arctag1	Analog		Released
4	anlg_2	arctag2	Analog		Released
5	anlg_3	arctag3	Analog		Released
6	anlg_4	arctag4	Analog		Released
7	anlg_5	arctag5	Analog		Released
8					

8. No checks are made when entering the data in the Excel table. Select the menu items "WinCC archive" > "Check archive". A dialog opens to check the entire archive. It is possible to restrict the archive properties to be checked. Activate all the relevant checkboxes and click the "OK" button.

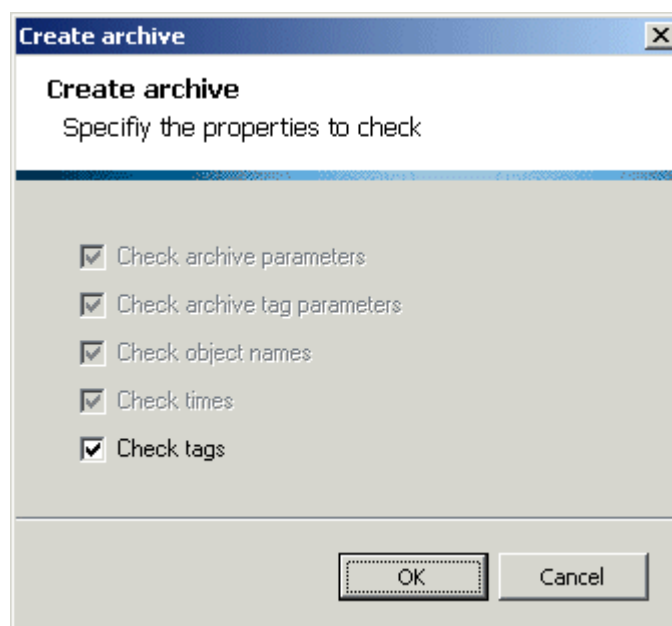
Note: If no configuration has yet been carried out for the WinCC project in the tag logging editor, the necessary cycle times are not available yet. They are displayed in the Tag Logging editor but are only stored in the database following a change in the configuration data. To trigger the tag logging editor to store the cycle times, simply rename the cycle time "500 ms" into "_500 ms" and save the project. Of course, you can then reset the name of the cycle time to the original value.



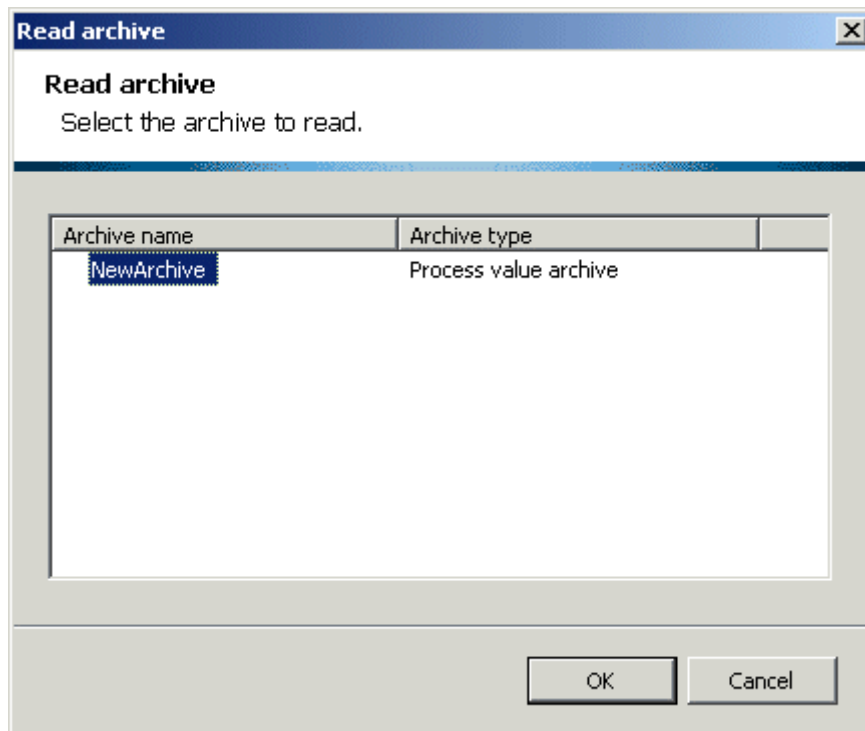
9. The test result is displayed. If faults are found, the user is prompted to state whether a list of the faults should be displayed. Confirm the prompt with "Yes". A dialog opens containing a list of the faulty objects together with the respective cause of the fault. By double-clicking an entry in the list, the system skips directly to the faulty object in the Excel table and the fault is cleared. The "Error text" column of the corresponding object also contains the cause of the fault. The cause of the fault has the following structure: "Column: Error". Clear all the errors found and run the test again.



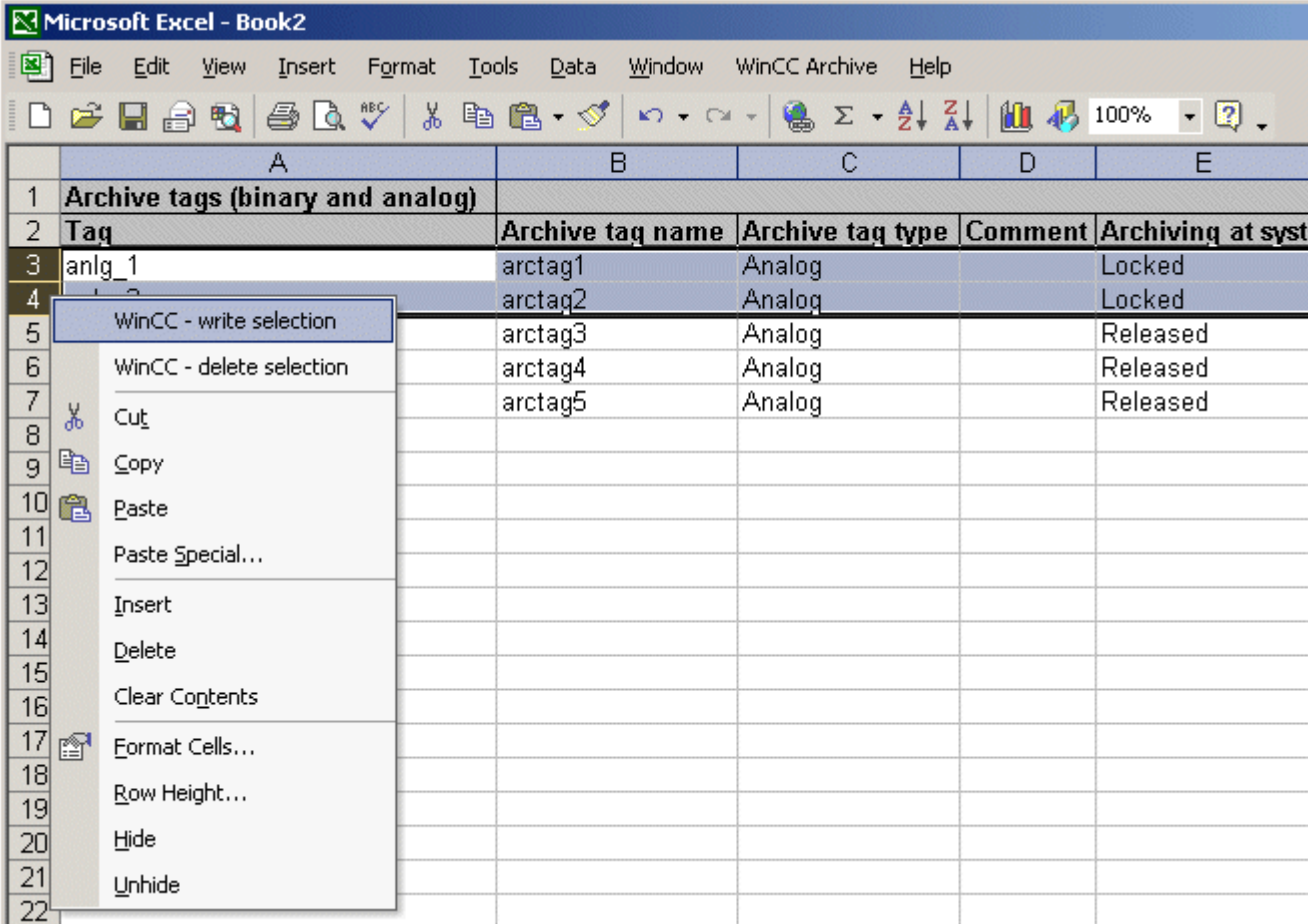
10. Select the menu items "WinCC archive -> Create archive". A dialog opens for creating the entire archive. Click the "OK" button. If an archive with the same name as the one to be created already exists, it must be deleted. For this reason, a corresponding prompt appears. Confirm the prompt by clicking "OK". The archive is created in the WinCC project. Close the Excel sheet.



11. The newly created archive cannot be opened in the tag logging editor. The only possibility to edit the archive is to read it back into an Excel table. Select the menu items "WinCC archive -> Read archive". A dialog opens containing a list of all the archives available in the WinCC project. Select the archive to be read and click the "OK" button. A new Excel sheet is created. The data is read into it from the selected archive.



12. The properties of the archive or properties of the individual archive tags can then be modified. The modifications carried out can be transferred to the WinCC project selectively. Select the relevant objects which have been modified. Open the context menu by clicking the right mouse button and select the menu item "WinCC - write selection". Before the archive tags are written, a dialog opens in which to define the test criteria. Confirm the dialog by clicking "OK".



	A	B	C	D	E
1	Archive tags (binary and analog)				
2	Tag	Archive tag name	Archive tag type	Comment	Archiving at syst
3	anlg_1	arctag1	Analog		Locked
4		arctag2	Analog		Locked
5		arctag3	Analog		Released
6		arctag4	Analog		Released
7		arctag5	Analog		Released
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					

1.9.5 Operating WinCC Archive

1.9.5.1 Operating WinCC Archive

Introduction

This section provides details of all the aspects of operating WinCC Archive.

See also

Creating, modifying and deleting (Page 289)
Checking the archive data (Page 284)
Configuring a compressed archive (Page 276)
Configuring a process value archive (Page 259)
Creating an archive folder (Page 253)

1.9.5.2 Creating an archive folder**Creating an archive folder****Introduction**

This section describes general procedures for creating and operating archive folders. The archive folder relates to an Excel sheet which is created by WinCC Archive. An archive folder serves to configure the properties of an archive and the associated archive tags.

See also

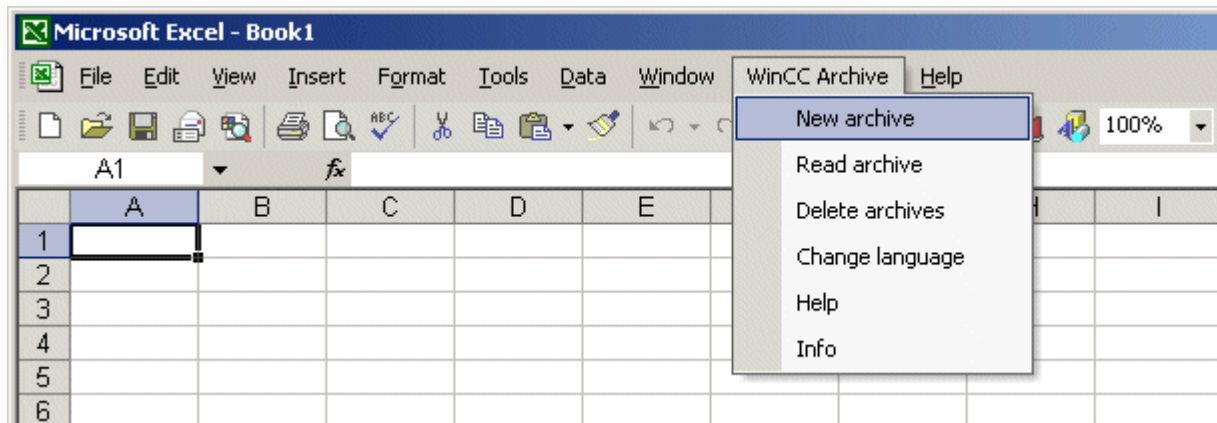
How to add a new table (Page 257)
Reading an existing archive (Page 255)
Creating a new archive folder (Page 253)

Creating a new archive folder**Introduction**

This description explains the procedure for creating a new archive folder.

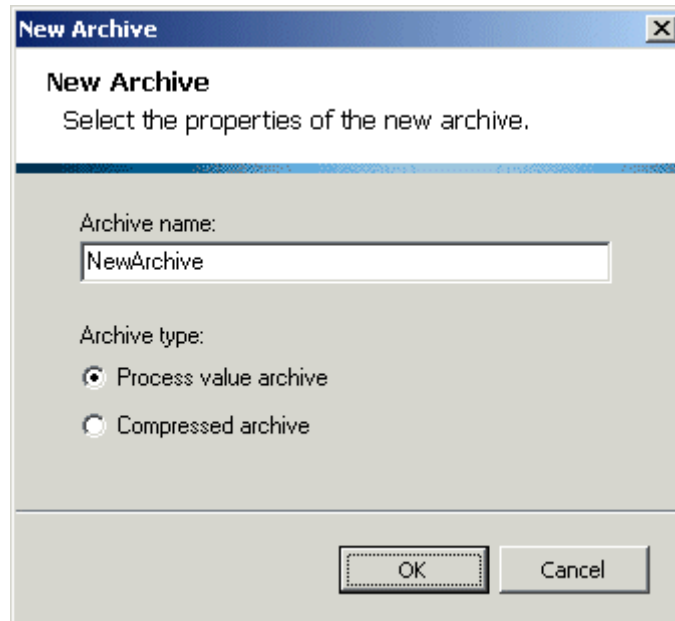
Procedure

1. Open Excel. The Excel add-in WinCC Archive is automatically started at the same time. The Excel menu contains the "WinCC Archive" item. Select "New archive" from this menu item.



2. A dialog opens in which to define the basic properties of the archive to be created. Enter a name for the archive. Select the desired archive type. Two different types of archives can be configured using WinCC Archive. These are process value archives and compressed archives. The archive folder created is dependent on the archive type selected. Archive folders for process value archives and for compressed archives differ as to the structure of the tables contained.

3. Click the "OK" button.



4. A new Excel sheet is created which serves to configure the desired archive type. An archive folder, by default, contains a table with the name "Archive". The archive properties are defined in it. Definition of the archive properties occurs in the first line of the table data area (Line 3).
An archive folder contains at least one table named "Archive tags" and a consecutive number. These tables serve to configure the archive tags associated to the archive. The entire data area (from the third line) of the table can be used for configuring the archive tags. It is possible to add further archive tag tables to the archive folder.

Note

Microsoft Excel interprets texts which begin with, or include, certain special characters as equations. This can cause problems when working with the configuration tool.

These special characters are "=", "+", and "-". Therefore, these characters should not be used as a part of object names (group names, tag names, texts, etc.).

See also

How to add a new table (Page 257)

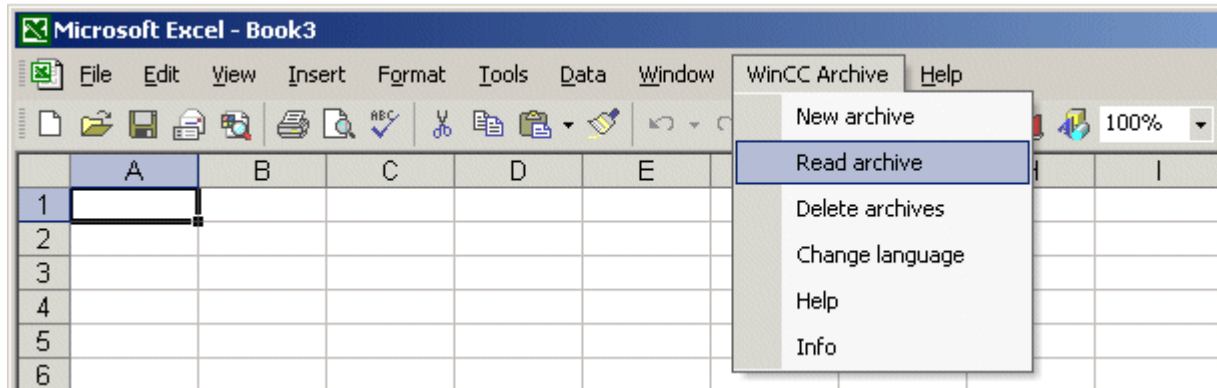
Reading an existing archive (Page 255)

Reading an existing archive**Introduction**

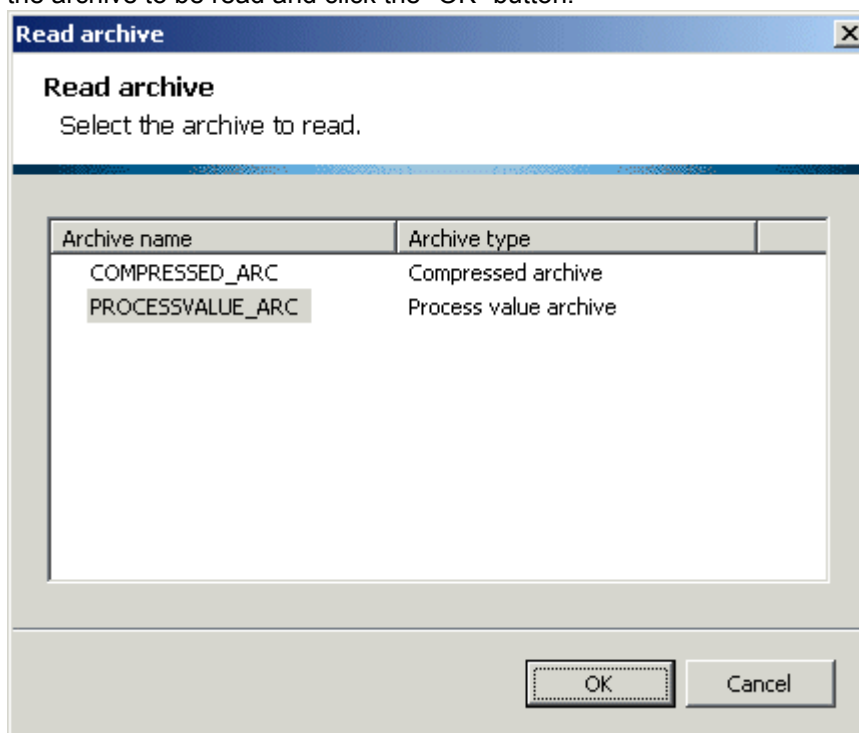
This description explains the procedures for reading an archive from the WinCC project.

Procedure

1. Open Excel. The Excel add-in WinCC Archive is automatically started at the same time. The Excel menu contains the "WinCC Archive" item. Select "Read archive" from this menu item.



2. A dialog opens containing a list of all the archives available in the WinCC project. Select the archive to be read and click the "OK" button.



3. A new archive folder is created. The data is read into it from the selected archive. The type of the archive folder to be created is dependent on the type of the archive being read. WinCC Archive can be used to edit process value archives and compressed archives. An archive folder contains a table with the name "Archive". This contains the archive's properties. The archive folder also contains at least one table named "Archive tags" and a consecutive number. These tables contain the archive tags associated to the archive.

See also

How to add a new table (Page 257)

Creating a new archive folder (Page 253)

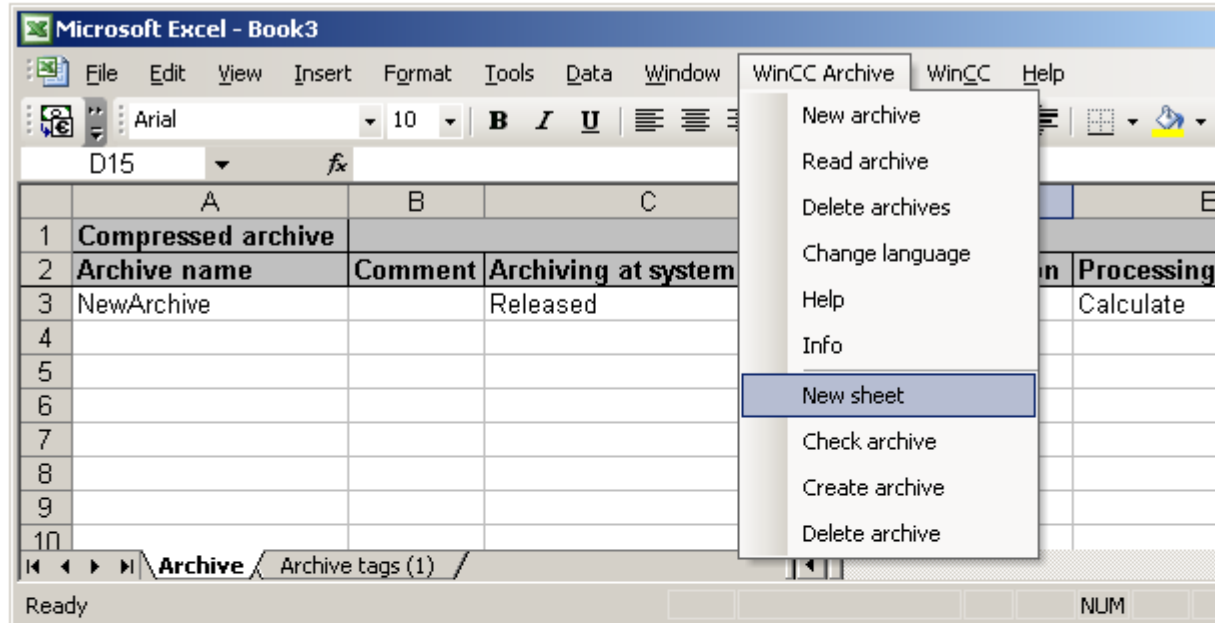
How to add a new table

Introduction

This description explains the procedure for adding a new table to an archive folder.

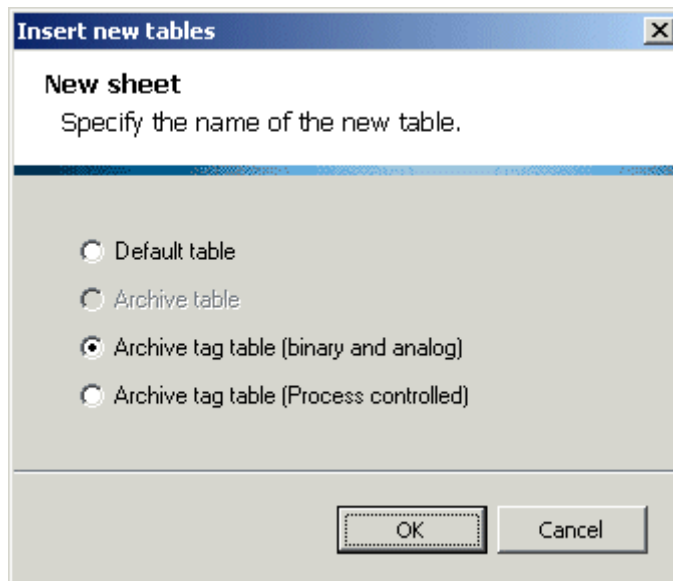
Procedure

1. Open the archive folder to which a new table is to be added. Select the menu items "WinCC Archive -> New table".



2. A dialog opens in which the type of the new table can be selected. The following table types are available.
 - Standard table: Normal Excel table.
 - Archive table: Table to define the properties of the archive. This type is only available for selection when the existing archive table has been deleted.
 - Archive Tag table: Table to configure archive tags. In the case of process value archives, there are two different types of archive tag tables available for selection. In the case of compressing archives, only one type is available for selection.

Select the desired type and click the "OK" button.



3. A new table has been added to the archive folder. This can be renamed, if so required.

See also

- Reading an existing archive (Page 255)
- Creating a new archive folder (Page 253)

1.9.5.3 Configuring a process value archive

Configuring a process value archive

Introduction

This section describes the general procedure for configuring a process value archive using WinCC Archive.

See also

- Using the dialog for tag selection (Page 274)
- Configuring archive tag parameters (process-controlled) (Page 270)
- Configuring the archive tag parameters (binary and analog) (Page 263)
- Configuring the archive parameters (Page 260)

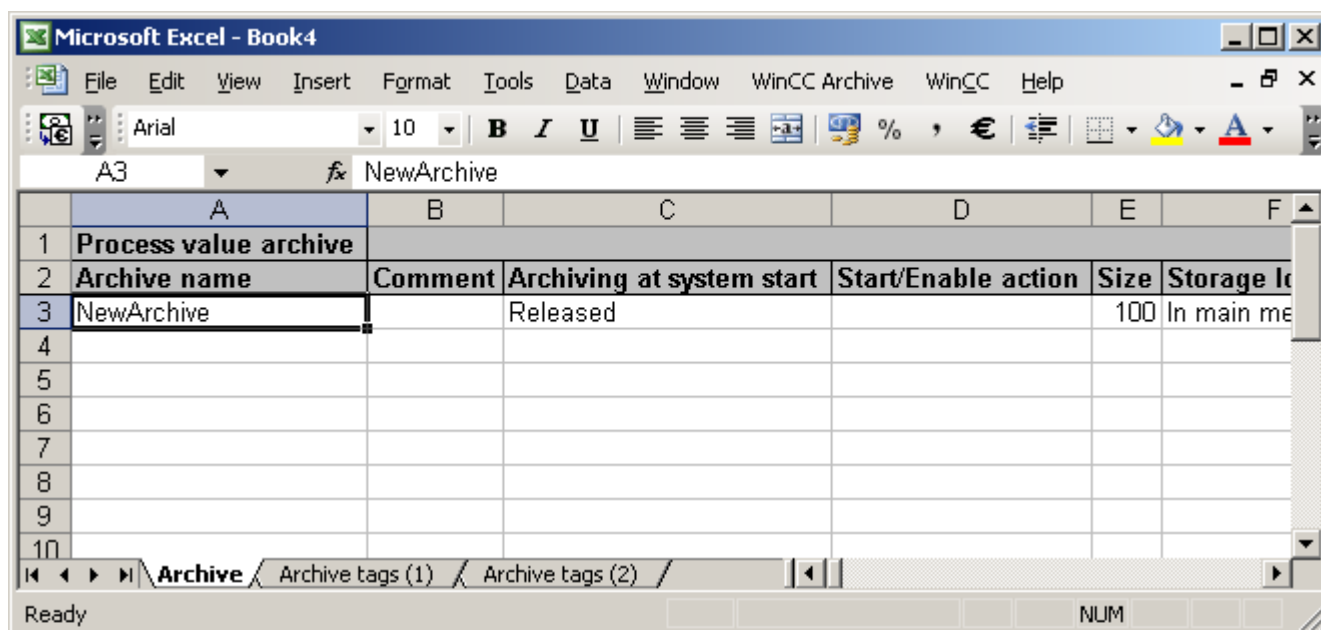
Configuring the archive parameters

Introduction

This section describes the configuration of the archive parameters. The archive parameters are explained individually.

Procedure

1. Activate the archive table. It normally has the name "Archive". The name of the table can be changed.



2. The archive parameters must be defined in the first line of the archive table's data area. Only the content from this line is evaluated by WinCC Archive.

Archive parameter

The following table lists all the parameters which must be entered for the process value archive.

	Column	List	Parameters	Description
1			Archive name	Specify the name of the archive here.
2			Comments	Enter a comment on the archive here.
3			Archiving at system start	Define whether archiving after the system is started should be "enabled" or "disabled".
4			Start/enable action	Specify an action which must be executed when archiving is started.
5			Size	Specify the number of data records to be stored in the archive. This parameter is only relevant for archives in the main memory.
6			Memory location	Define the memory location of the archive. You can choose between "Hard disk" or "In the main memory".

The following section explains the significance of the symbols used in the "List" column.

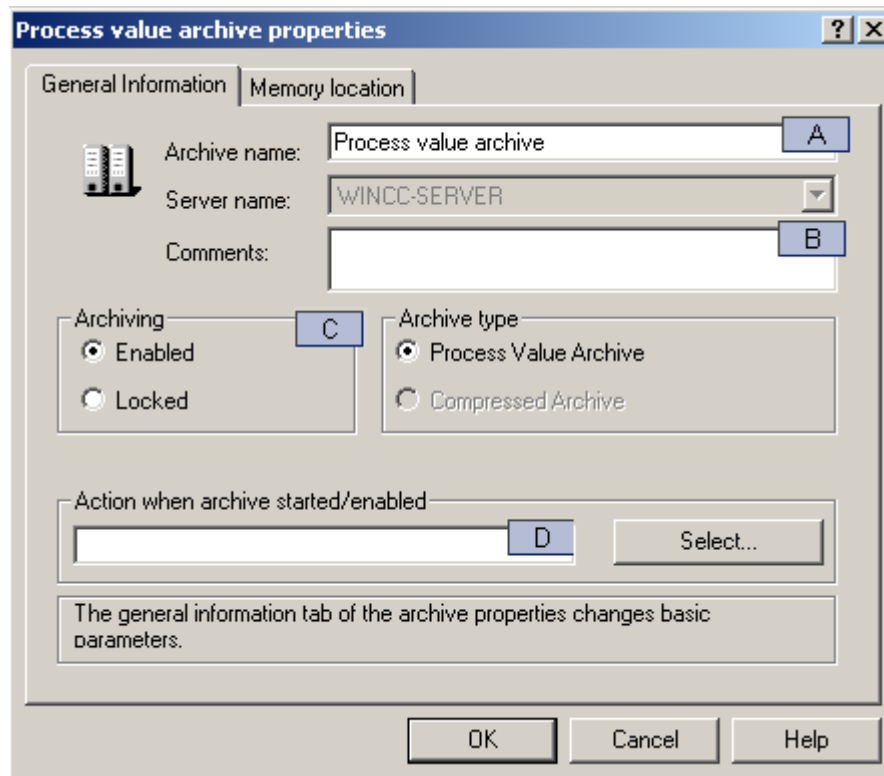


Double-click to open a combo box.

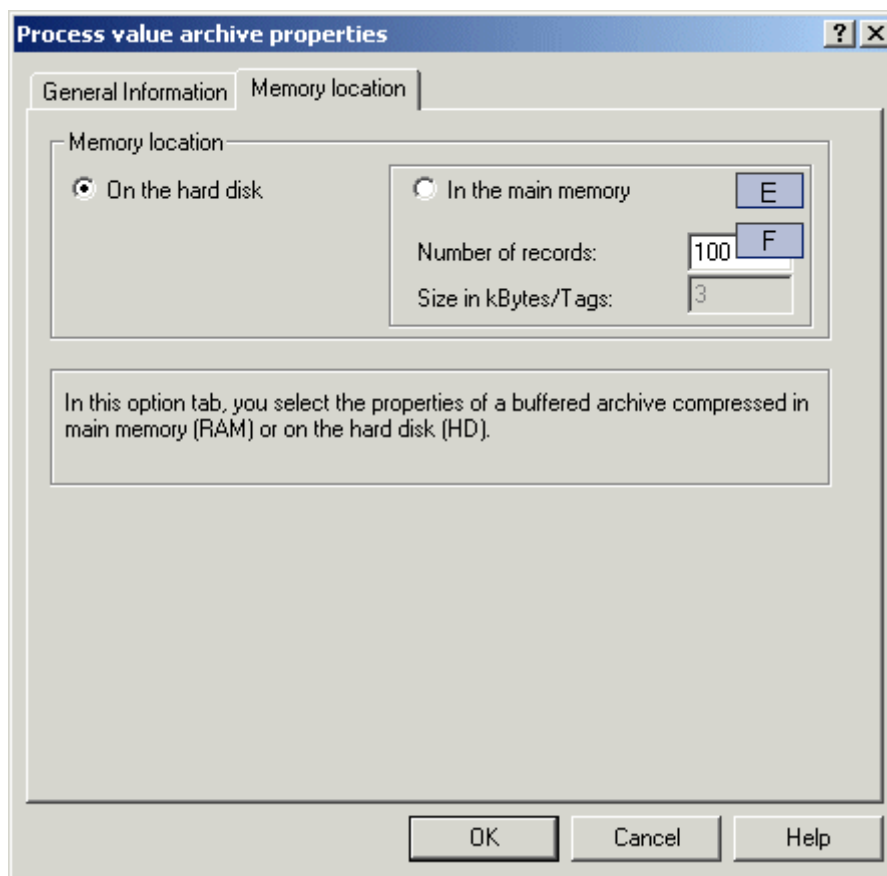
Tag logging editor

The following diagrams depict the relationship between the configuration using the Tag Logging editor and the configuration using WinCC Archive. In addition to the parameters to be set in the Tag Logging editor, the respective columns in WinCC Archive are specified.

1. The following diagram illustrates the "Process value archive properties" dialog. The "General" tab has been selected. The corresponding column in the table structure of WinCC Archive is displayed for each parameter.



2. The following diagram illustrates the "Process value archive properties" dialog. The "Memory location" tab has been selected. The corresponding column in the table structure of WinCC Archive is displayed for each parameter.



See also

Configuring archive tag parameters (process-controlled) (Page 270)

Configuring the archive tag parameters (binary and analog) (Page 263)

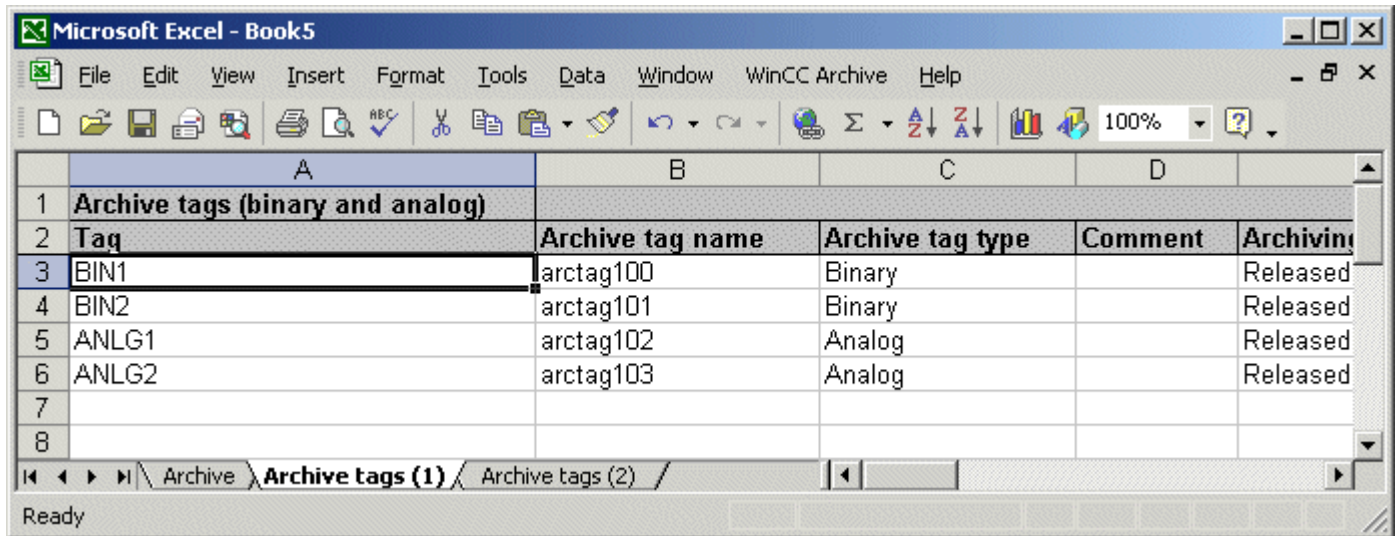
Configuring the archive tag parameters (binary and analog)

Introduction

This description explains the configuration of binary and analog archive tags. The parameters which must be defined for binary and analog archive tags are explained individually.

Procedure

1. Activate the table for configuring the binary and analog archive tags. This normally has the name "Archive Tags (1)". The name of the table can be changed. Since the parameters of binary and analog archive tags are very similar, both are configured within the same table structure.



2. The entire data area of the table is available for configuring the archive tags. As a result, up to 65534 archive tags can be configured on the table. If this is insufficient, further tables can be added.

Archive tag parameters

The following table lists all the parameters which must be defined for binary and analog archive tags.

	Column	List	Parameters	Description
1	A		Tag	This is used to enter the name of the tag to be archived.
2	B		Archive tag name	This is used to specify the name of the archive tag here. This must be unique within the archive.
3	C		Archive tag type	This is used to specify the type of the archive tag. It must be defined according to the type of tag to be saved, "analog" or "binary".
4	D		Comment	This is used to enter a comment on the archive tag.
5	E		Archiving at system start	This is used to define whether archiving after the system is started should be "enabled" or "disabled".
6	F		Supplying tags	Define the type of supplying tags. Select "System" or "Manual input".

	Column	List	Parameters	Description
7			Acquisition type	This is used to define the type of acquisition of the archive tags. Select between "acyclic", "cyclic-continuous", "cyclic-selective" or "Upon change".
8			Acquisition cycle	This is used to specify the cycle at which the value of the tag is to be requested.
9			Archiving cycle	This is used to specify the cycle at which the value of the tag should be archived.
10			Archiving cycle factor	This can be used to specify a factor for the archiving cycle. The cycle at which the tag value should be archived results from the product of "Archiving cycle" and "Archiving cycle factor".
11			Also write in tag	The name of a tag can be specified in which the archived value should also be written.
12			Editing	This is used to specify how the compiled tag values are to be processed. The "Actual value", "Average value", "Sum", "Minimum value", "Maximum value" and "Action" options are available for selection.
13			Action for processing	This can be used to specify an action for process the tag values. To do this, enable the "Action" option in the "Processing" column.
14			Unit	This is used to specify a unit for the archived tag values.
15			Archiving on change	This is used to specify that archiving should only occur when the tag value changes.
16			Hysteresis	This is used to define the type of hysteresis when "Archiving on change" has been activated. You can select between "Absolute" and "In percent".
17			Hysteresis value	This is used to define the number value of the hysteresis when "Archiving on change" has been activated.
18			Archiving on	This is used to specify when to archive binary archive tags. The settings "every signal change", "always", "on signal change 0 ->1" and "on signal change 1->0" are available.
19			Save on error	This is used to define which value to save in the event of a fault. You can select between "Last value" and "Replacement value".
20			Number of values leader	This can be used to specify the number of values which must be taken into account before the moment of archiving.

	Column	List	Parameters	Description
21			Number of values trailer	This can be used to specify the number of values which must be taken into account after the moment of archiving.
22			Representation limit, lower limit	This can be used to specify the lower limit of the value range of the archive tags.
23			Representation limit, upper limit	This can be used to specify the upper limit of the value range of the archive tags.
24			Start event	This can be used to specify a function as an action which starts archiving.
25			Start tag	This can be used to specify as an action a binary tag which starts archiving.
26			Stop event	This can be used to specify a function as an action which stops archiving.
27			Stop tag	This can be used to specify as an action a binary tag which stops archiving.
28			Relevant long term	You can define here whether the archive tag is to be written to the archive on the central archive server (WinCC CAS).
29			Archive after segment change	It can be determined here whether the acyclic archived tags are archived during a segment change.
30	AD		Status	The status of an archive can be displayed here.
31	AE		Error text	You can define an error text here.

The following section explains the significance of the symbols used in the "List" column.



Double-click to open a combo box.



Double-click to open a combo box. A WinCC project must be open.



Double-click to open the tag dialog. A WinCC project must be open.

Tag logging editor

The following diagrams depict the relationship between the configuration using the Tag Logging editor and the configuration using WinCC Archive. In addition to the parameters to be set in the Tag Logging editor, the respective columns in WinCC Archive are specified.

1. The following diagram illustrates the "Process tag properties" dialog. The "Archive tag" tab has been selected. The corresponding column in the table structure of WinCC Archive is displayed for each parameter.

Process tag properties

Archive Tag | Archiving | Parameters | Display

Name of the archive tag: **B** Tag Type: **C**

Name of the process tag: **A**

Comments: **D**

Supplying tags: ☒ System ☐ Manual input **F**

Archiving: ☒ Enabled ☐ Disabled **E**

☒ Relevant long term **AB**

Also put archived value in tag: **K**

The general tab of the tag properties changes basic parameters.

2. The following figure shows the parameters of the "Archiving" tab. The corresponding column in the table structure of WinCC Archive is displayed for each parameter.

Process tag properties ? x

Archive Tag Archiving Parameters Display

 tank_Arch
Analog

Archiving type: **cyclic** **G**

Acquisition: 500 ms **O** **H**

Archiving

Factor: 1 **J** x Cycle: 500 ms **I**

Hysteresis: ☒ abs. **P** ☐ in % **Q** 0

☐ Archive after segment change **AC**

Display

Factor: 1 x Cycle: 500 ms

Events

Tag: ...

C script: ...

Actions

Start:

Tag: **Y** ...

C script: **X** ...

Stop:

Tag: **AA** ...

C script: **Z** ...

OK Cancel Help

3. The organization of the "Parameters" tab differs for binary as compared to analog archive tags. The structure with respect to analog archive tags is depicted. In the case of binary archive tags, the "Processing", "Unit" and "Archiving on change" areas are not available. The corresponding columns in the WinCC Archive table structure must be left empty for binary archive tags. In the case of binary archive tags, on the other hand, an area named "Archiving on" exists. This area is assigned to column "R" in the WinCC Archive table structure. This column is irrelevant in the case of analog archive tags and must remain empty then.

Process tag properties

Archive Tag | Archiving | **Parameters** | Display

tank_Arch Analog

Processing

☒ Actual value ☐ Sum ☐ Max. value
☐ Mean value ☐ Action ☐ Min. value

M

Number of values

Leader **T**
Trailer **U**

Unit

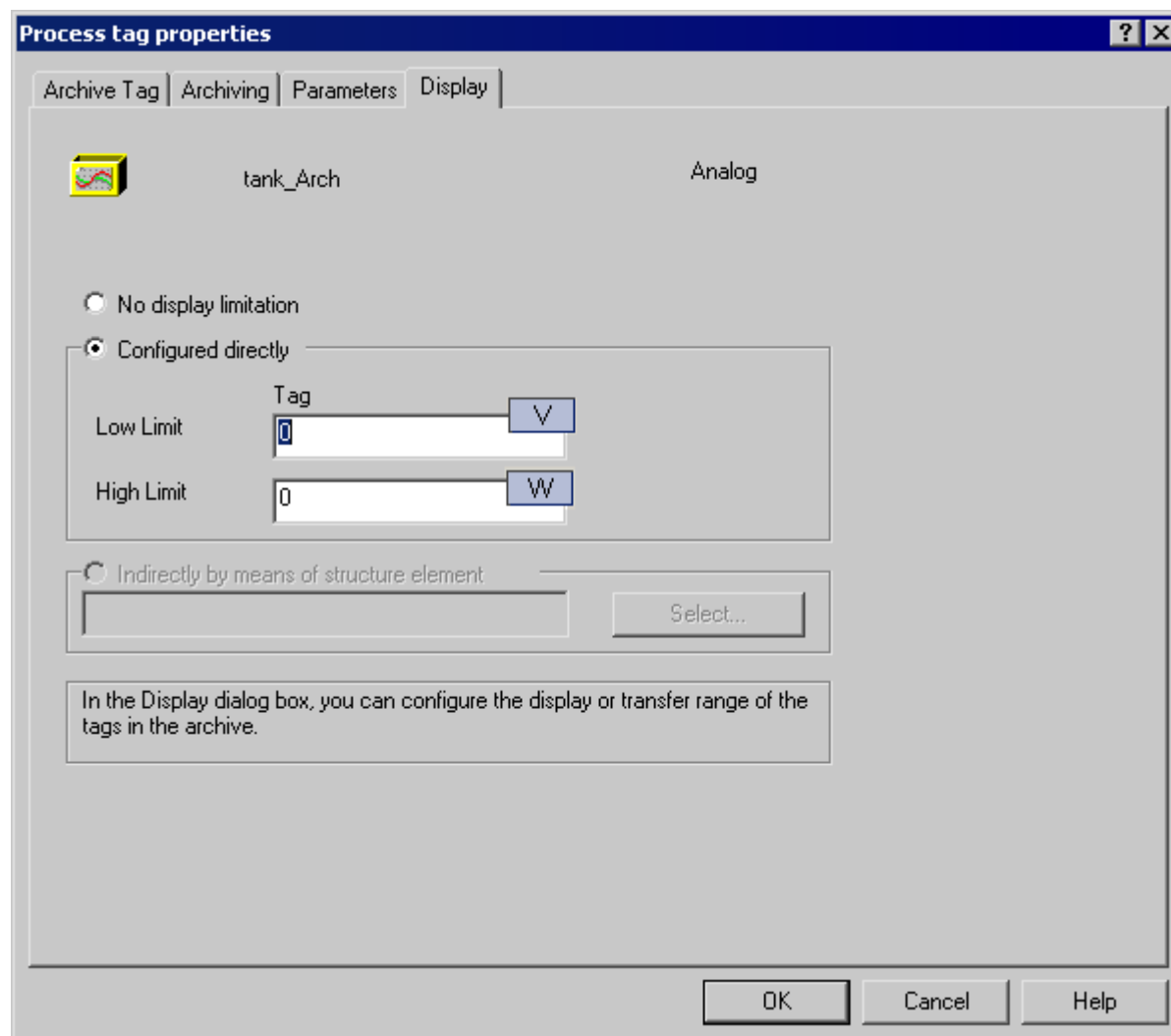
☒ Direct
☐ Structure elem.

Save on error

☒ Last value **S**
☐ Subst. value

Settings for cyclic analog measurement points

4. The following figure shows the parameters of the "Display" tab. The corresponding column in the table structure of WinCC Archive is displayed for each parameter. The "Display" tab is not available for binary archive tags. The corresponding columns in the WinCC Archive table structure must be left empty for binary archive tags.



See also

Configuring the archive parameters (Page 260)

Using the dialog for tag selection (Page 274)

Configuring archive tag parameters (process-controlled) (Page 270)

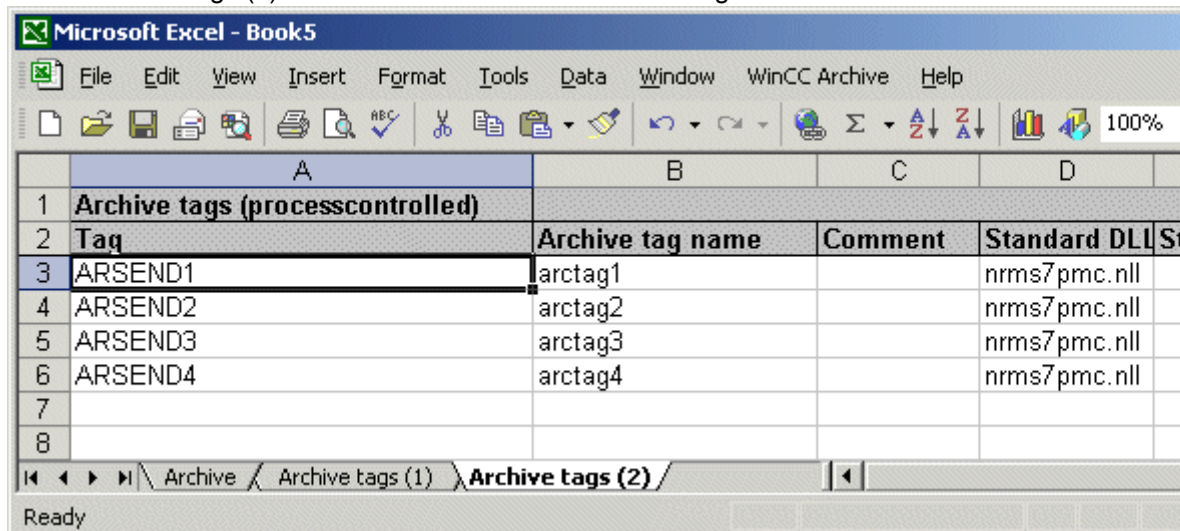
Configuring archive tag parameters (process-controlled)

Introduction

This description explains the configuration of process controlled archive tags. The parameters which must be defined for process controlled archive tags are explained individually.

Procedure

1. Activate the table for configuring the process controlled archive tags. By default, it has the name "Archive tags (2)". The name of the table can be changed.



2. The entire data area of the table is available for configuring the archive tags. As a result, up to 65534 archive tags can be configured on the table. If this is insufficient, further tables can be added.

When configuring process controlled archive tags, the format DLL to be used must be specified. WinCC Archive only supports the configuration of process controlled archive tags for the format DLL "nrms7pmc.nll".

Archive tag parameters

The following table lists all parameters which must be defined for process controlled archive tags.

	Column	List	Parameters	Description1
1	A		Tag	This is used to specify the name of the raw data tag by means of which the archiving is performed.
2	B		Archive tag name	Specify the name of the archive tag here. This must be unique within the archive.
3	C		Comment	This is used to enter a comment on the archive tag.
4	D		Format DLL	Set here the format DLL to be used. Only the format DLL "nrms7pmc.nll" is supported.
5	E		Format DLL parameter 1	With "nrms7pmc.nll": AR_ID
6	F		Format DLL parameter 2	With "nrms7pmc.nll": AR_ID Subnumber

The following section explains the significance of the symbols used in the "List" column.



Double-click to open a combo box.

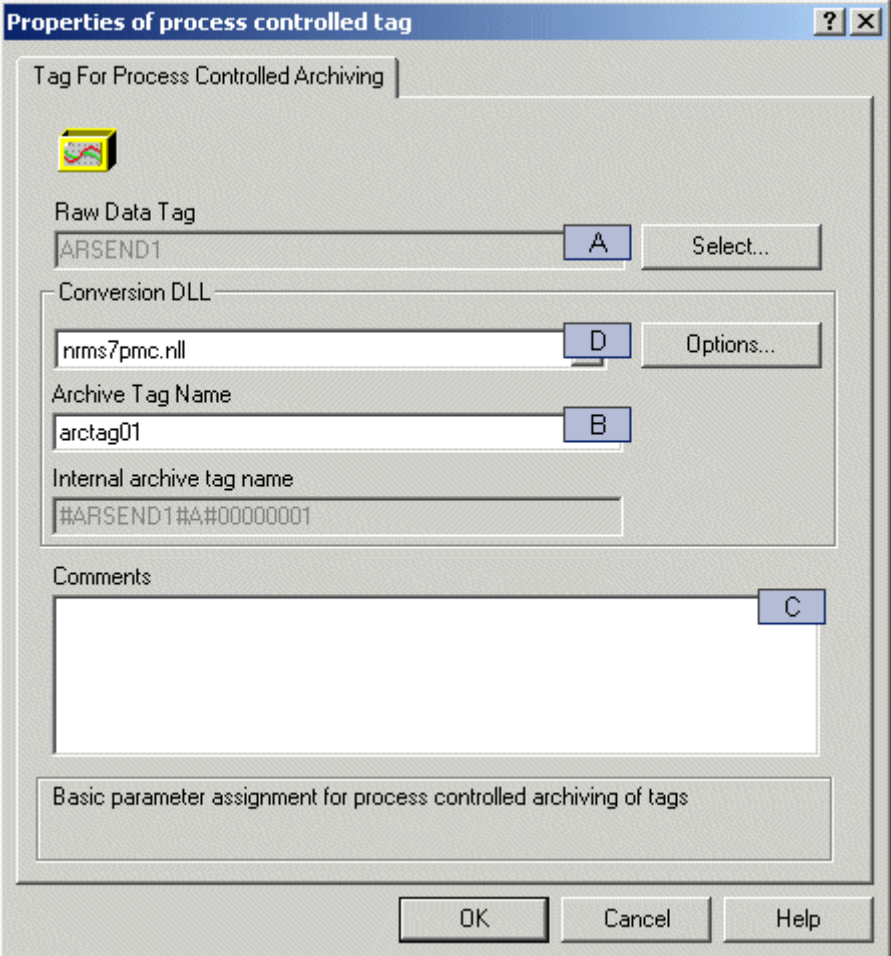
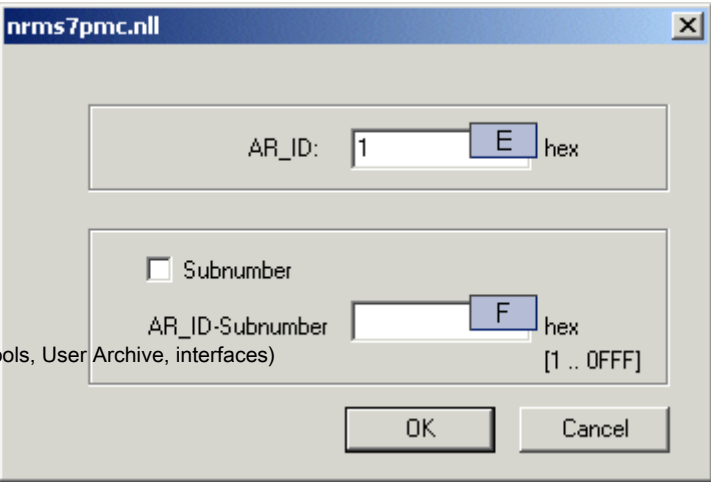


Double-click to open the tag dialog. A WinCC project must be open.

Process controlled archive tags are provided with a parameter "Archive tag name". This is available in the WinCC Archive table structure. In addition, process controlled archive tags have an internal archive tag name. This cannot be processed by the user directly and, therefore, has not been included in the WinCC Archive table structure. The internal archive tag name is automatically assigned during transfer to the WinCC project.

Tag logging editor

The following diagrams depict the relationship between the configuration using the Tag Logging editor and the configuration using WinCC Archive. In addition to the parameters to be set in the Tag Logging editor, the respective columns in WinCC Archive are specified.

Tab	
1	The following diagram illustrates the "Properties of process controlled tag" dialog. The corresponding column in the table structure of WinCC Archive is displayed for each parameter. 
2	The following diagram depicts the "nrms7pmc.nll" dialog. It can be opened by clicking the "Options" button in the "Properties of process controlled tag" dialog. A condition for this, however, is that "nrms7pmc.nll" is defined as the "Format DLL" (refer to Step 1). The corresponding column in the table structure of WinCC Archive is displayed for each parameter. 

See also

Using the dialog for tag selection (Page 274)

Configuring the archive tag parameters (binary and analog) (Page 263)

Configuring the archive parameters (Page 260)

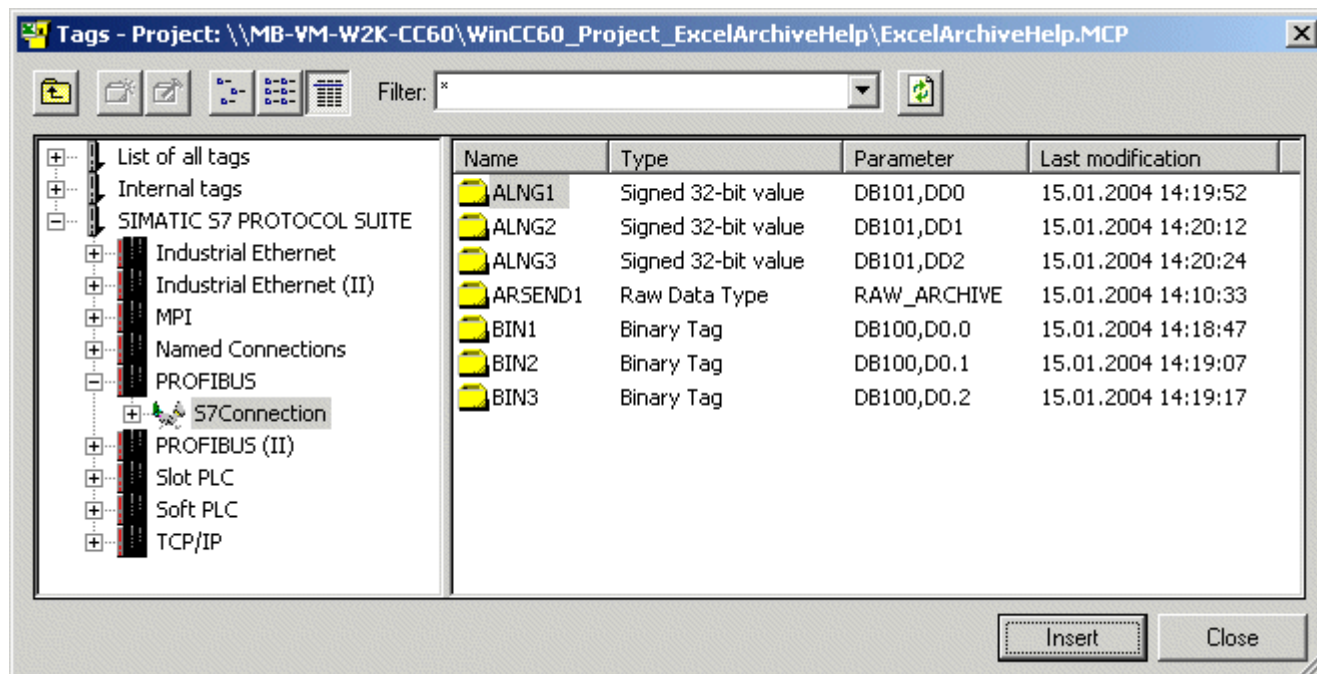
Using the dialog for tag selection

Introduction

The first parameter to enter when configuring an archive tag is the name of the WinCC tag to be archived. This parameter can be entered directly in the corresponding cell, as can all the other parameters. In addition, WinCC Excel provides the option of using a special dialog to select tags. It offers all the tags in the WinCC project currently open for selection. If no WinCC project is open, the dialog is not available.

Procedure

1. Ensure that a WinCC project is open. Open or create an archive folder to configure a process value archive. Activate any archive tag table. Double-click the data area in a cell of the "Tag" column. The Tag selection dialog opens.



2. The Tag selection dialog does not inhibit the operation of Excel. This means that work in the archive folder can continue even when the Tag selection dialog is open.
Select the cell in the archive tag table in which to insert the name of a WinCC tag. Select the desired WinCC tag in the Tag selection dialog. Click the "Paste" button in the Tag selection dialog. The name of the WinCC tag selected is inserted in the archive tag table. The Tag selection dialog remains open.
Pasting individual WinCC tags can also be performed by double-clicking the desired tag in the selection list.
3. The Tag selection dialog supports multiple selection. This means several WinCC tags can be selected at a time by simultaneously holding the Ctrl or Shift keys.
Select the cell in the archive tag table from which to insert the names of the WinCC tags. Select the desired WinCC tag in the Tag selection dialog. Click the "Paste" button in the Tag selection dialog. The names of the WinCC tags selected are pasted in the archive tag table. The Tag selection dialog remains open.
4. The number of WinCC tags displayed in the Tag selection dialog can be restricted by entering filter criteria.
Enter the required filter criterion in the "Filter" input field. For example, when entering "B*" as a filter criterion, only those tags are displayed whose name begins with the letter B. Click the  button to apply the filter criteria. Only WinCC tags, the names of which match the selected filter criteria, are shown.
Enter "*" as the filter criterion to display all available WinCC tags again.
5. Click the "Close" button to terminate work using the Tag selection dialog.

See also

Configuring archive tag parameters (process-controlled) (Page 270)

Configuring the archive tag parameters (binary and analog) (Page 263)

1.9.5.4 Configuring a compressed archive

Configuring a compressed archive

Introduction

This section describes the general procedures for configuring a compressed archive using WinCC Archive.

See also

Configuring the archive tag parameters (Page 280)

Configuring the archive parameters (Page 276)

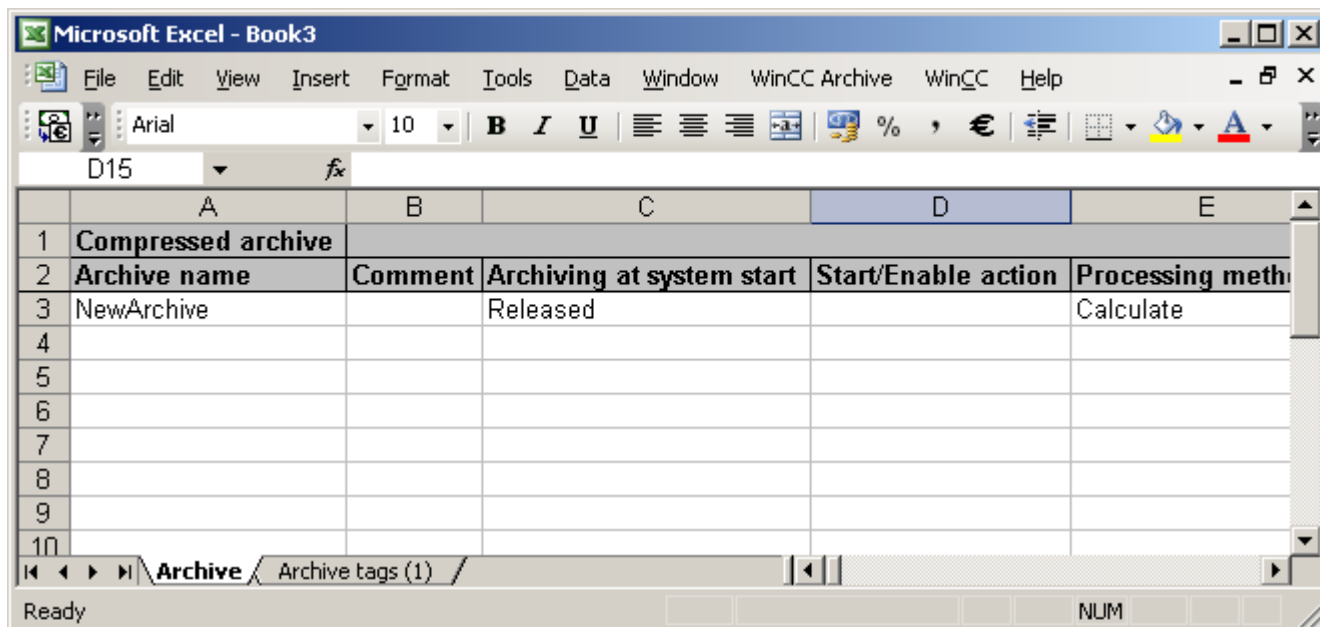
Configuring the archive parameters

Introduction

This section describes the configuration of the archive parameters. The archive parameters are explained individually.

Procedure

1. Activate the archive table. It normally has the name "Archive". The name of the table can be changed.



2. The archive parameters must be defined in the first line of the archive table's data area. Only the content from this line is evaluated by WinCC Archive.

Archive parameters

The following table lists all the parameters which must be entered for the process value archive.

	Column	List	Parameters	Description
1	A		Archive name	Specify the name of the archive here.
2	B		Comments	Enter a comment on the archive here.
3	C		Archiving at system start	Define whether archiving after the system is started should be "enabled" or "disabled".

	Column	List	Parameters	Description
4			Start/enable action	Specify an action which must be executed when archiving is started.
5			Processing method	This is used to define how the compressed archive should be processed. The "Calculate", "Calculate and copy", "Calculate and delete" and "Calculate, copy, and delete" options are available for selection.
6			Compression time period	This is used to specify the cycle at which the values in the archive tags are to be compressed to an entry in the compressed archive.

The following section explains the significance of the symbols used in the "List" column.



Double-click to open a combo box.

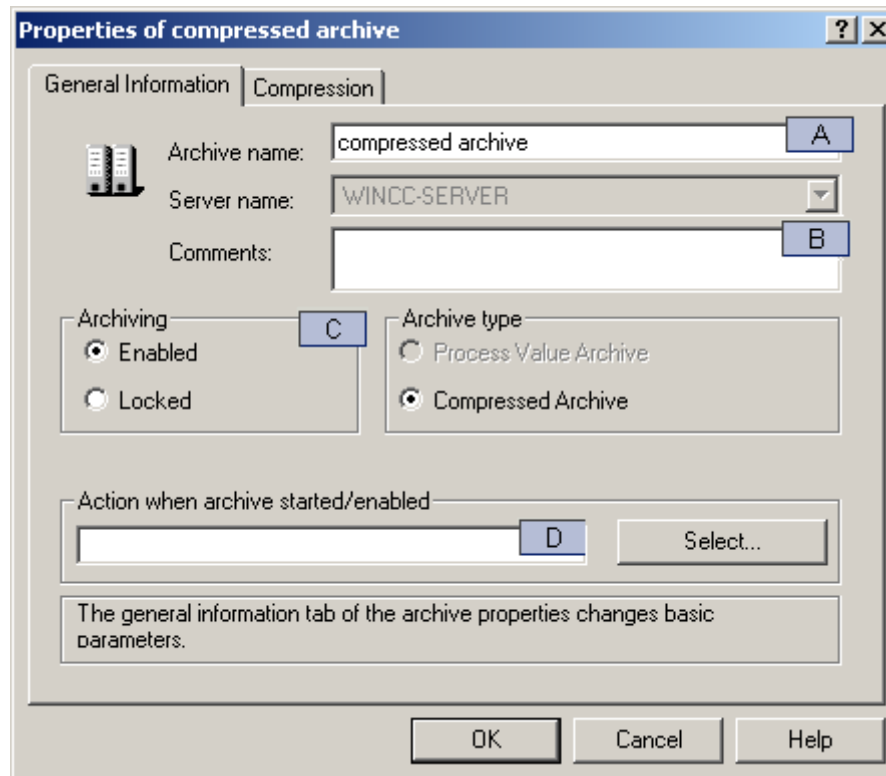


Double-click to open a combo box. A WinCC project must be open.

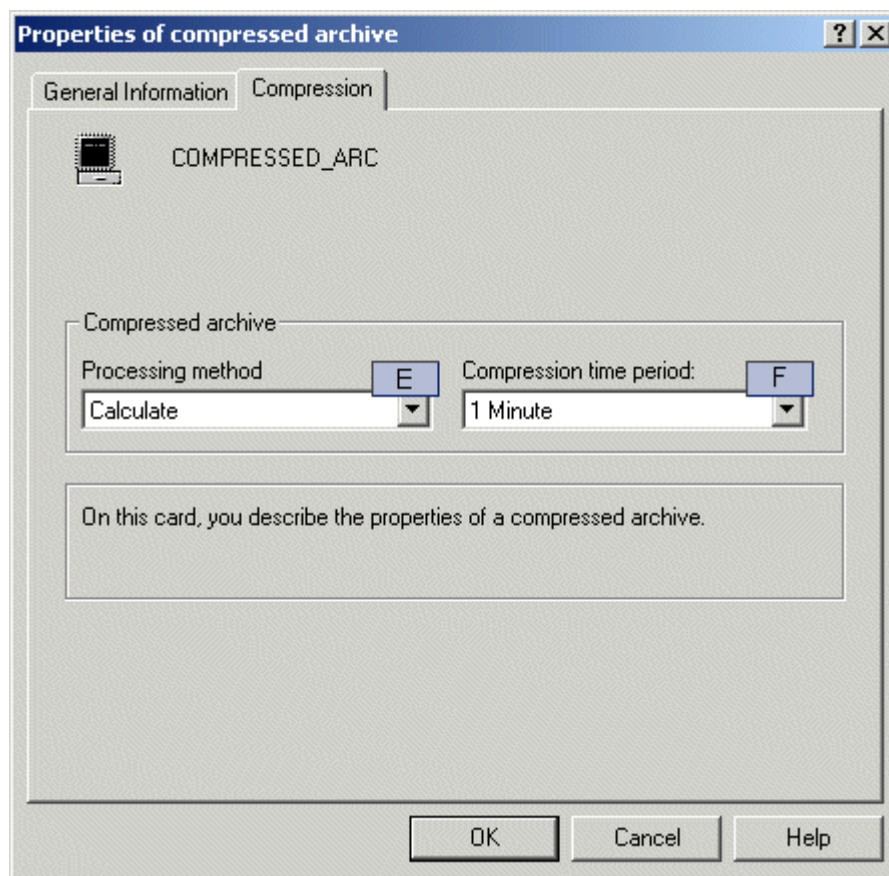
Tag logging editor

The following diagrams depict the relationship between the configuration using the Tag Logging editor and the configuration using WinCC Archive. In addition to the parameters to be set in the Tag logging editor, the respective columns in WinCC Archive are specified.

1. The following diagram illustrates the "Properties of compressed archive" dialog. The "General" tab has been selected. The corresponding column in the table structure of WinCC Archive is displayed for each parameter.



2. The following diagram illustrates the "Properties of compressed archive" dialog. The "Compression" tab has been selected. The corresponding column in the table structure of WinCC Archive is displayed for each parameter.

**See also**

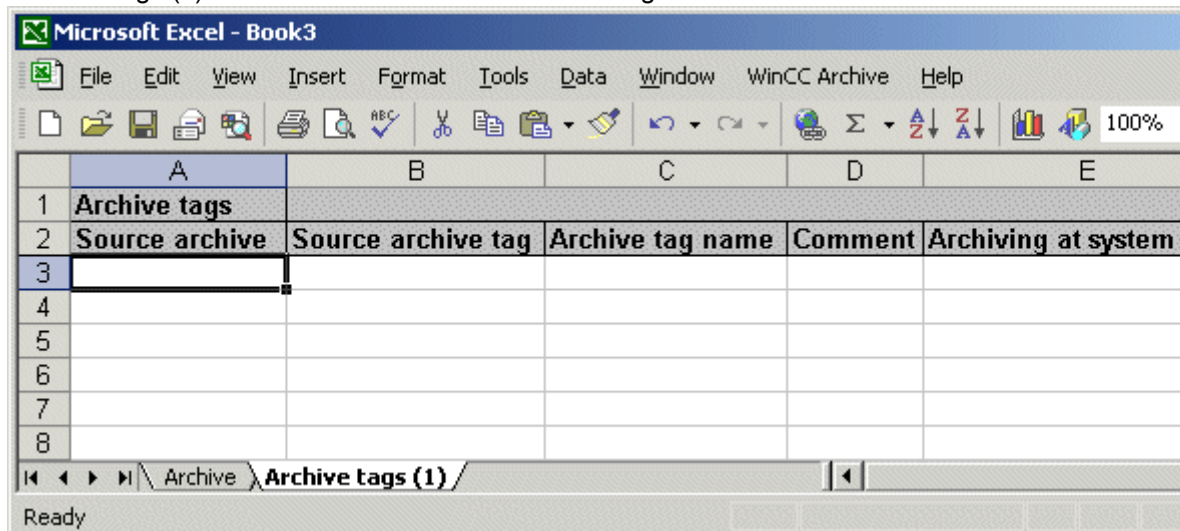
Configuring the archive tag parameters (Page 280)

Configuring the archive tag parameters**Introduction**

This description explains the configuration of compressed archive tags. The parameters which must be defined for compressed archive tags are explained individually.

Procedure

1. Activate the table for configuring the compressed archive tags. By default, it has the name "Archive tags (1)". The name of the table can be changed.



2. The entire data area of the table is available for configuring the archive tags. As a result, up to 65534 archive tags can be configured on the table. If this is insufficient, further tables can be added.

Archive tag parameters

The following table lists all the parameters which must be defined for compressed archive tags.

	Column	List	Parameters	Description
1	A		Source archive	This is used to specify the name of the archive in which the archive tags to be compressed are located.
2	B		Source archive tag	This is used to specify the name of the archive tag to be compressed.
3	C		Archive tag name	Specify the name of the archive tag here. This must be unique within the archive.
4	D		Comments	This is used to enter a comment on the archive tag.
5	E		Archiving at system start	This is used to define whether archiving after the system is started should be "enabled" or "disabled".
6	F		Editing	This is used to define how the compressed archive tag values should be processed. The "Mean value", "Sum", "Minimum value" and "Maximum value" options are available for selection.
7	G		Unit	This is used to enter a unit for compressing archive tag values.

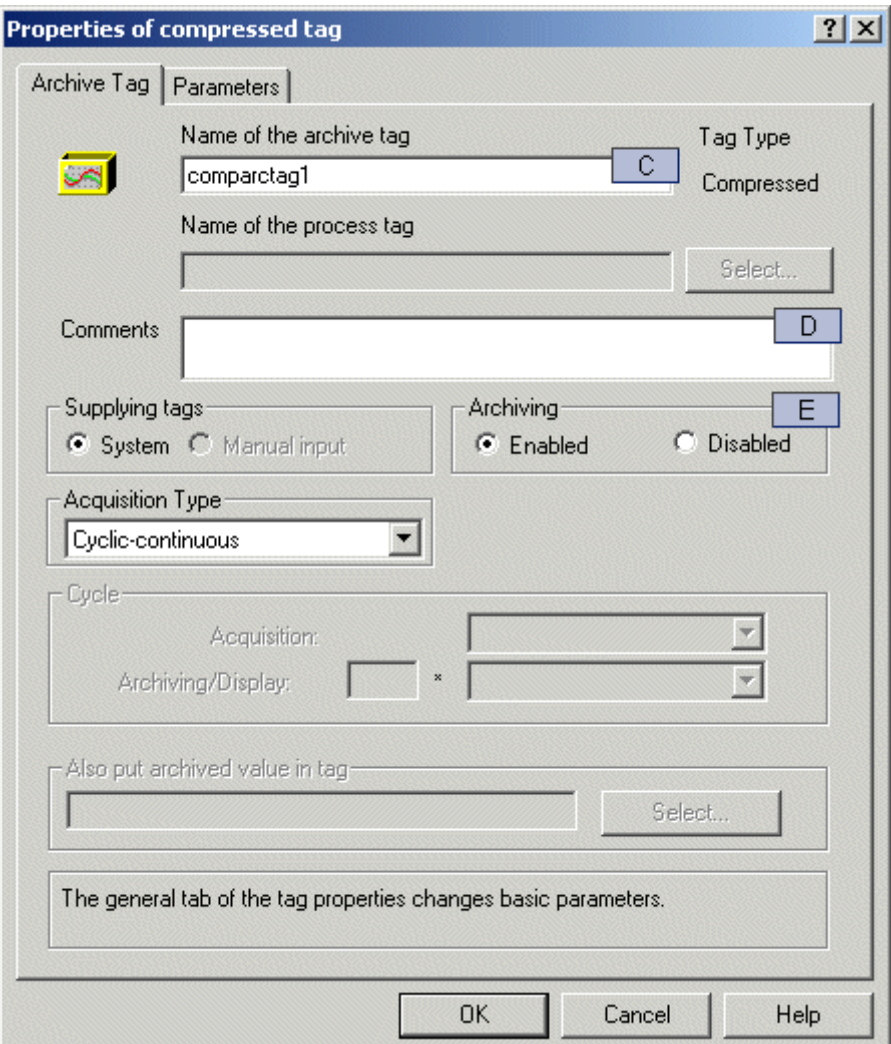
The following section explains the significance of the symbols used in the "List" column.



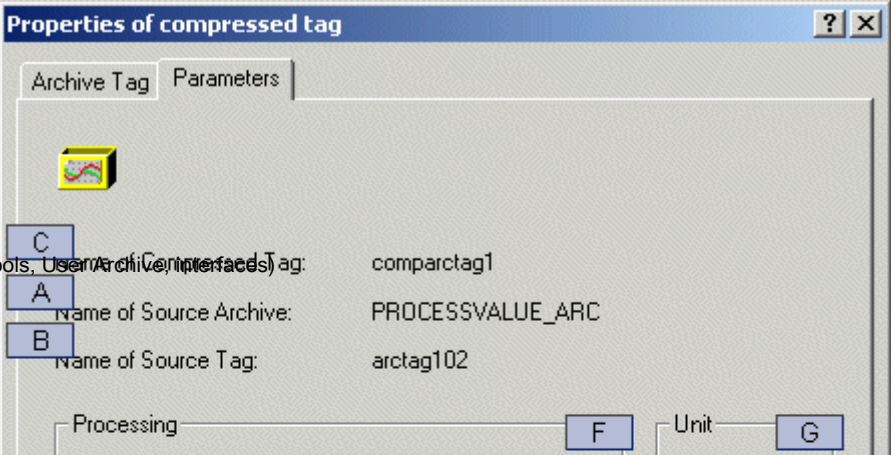
Double-click to open a combo box.

Tag logging editor

The following diagrams depict the relationship between the configuration using the Tag Logging editor and the configuration using WinCC Archive. In addition to the parameters to be set in the Tag logging editor, the respective columns in WinCC Archive are specified.

Tab	Tab
1	The following diagram illustrates the "Properties of compressed tag" dialog. The "Archive tag" tab has been selected. The corresponding column in the table structure of WinCC Archive is displayed for each parameter. 

2	The following diagram illustrates the "Properties of compressed tag" dialog. The "Parameters" tab has been selected. The corresponding column in the table structure of WinCC Archive is displayed for each parameter.



See also

Configuring the archive parameters (Page 276)

1.9.5.5 Checking the archive data

Checking the archive data

Introduction

This section describes the procedures for checking the configured data.

During the configuration, WinCC Archive does not carry out any tests with respect to the data. As a result, the user can use all the options to enter data. Before transferring an archive to the WinCC project, it is essential to ensure that the configuration is correct. For this purpose, WinCC Archive offers special options to check the archive data.

See also

Error list (Page 288)

Using the error dialog (Page 286)

Checking the archive data (Page 284)

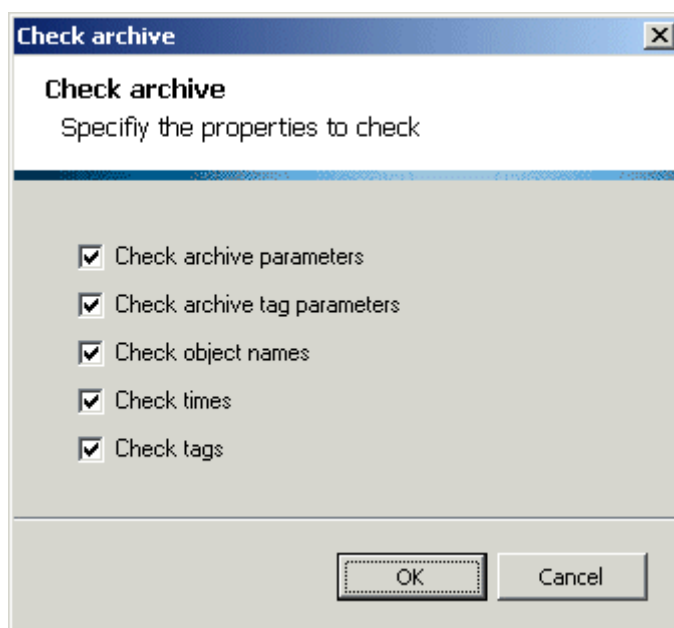
Checking the archive data

Introduction

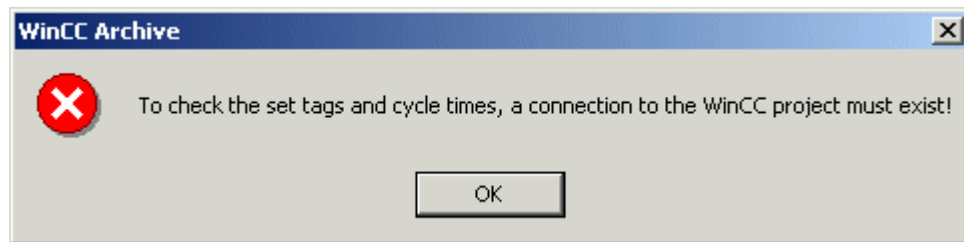
This description explains the options provided by WinCC Archive to check the archive data.

Procedure

1. Open the archive folder to be checked. Select the menu items "WinCC archive" > "Check archive". The "Check archive" dialog is opened.
The "Check archive" dialog provides options for subjecting the archive data to various checks. The checks available are explained in brief in the following section.
 - Check archive parameters: The settings defined in the archive table are checked. The check does not involve the archive tag tables.
 - Check archive tag parameters: The settings defined in the archive tag tables are checked. A check is made whether all necessary parameters have been set and have a value which can be interpreted properly.
 - Check object names: The names of the archive tags are checked for uniqueness within the archive. In addition, a check is made that the names comply to the archive tag name conventions.
 - Check cycle times: A check is made whether the set cycle times are available in the WinCC project. Another check is made as to whether the values set are valid for the required use.
 - Check tags: A check is made whether the set tags are available in the WinCC project. Another check is made as to whether their data types are valid for the required use.
 - Check archive tags: This check is offered for compressed archives instead of the "Check tags" test. A check is made whether all the set archive tags are available in the WinCC project.

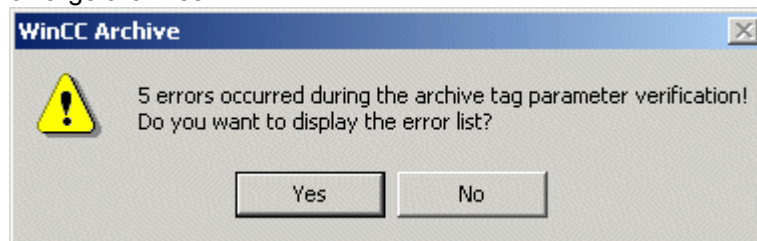


2. Select the required checks by activating the corresponding check boxes. By default, all the checks are selected. Click the "OK" button.
The checks "Check cycle times" and "Check tags" (or "Check archive tags") can only be performed if a WinCC project has been opened. A corresponding error message appears indicating this following incorrect input.

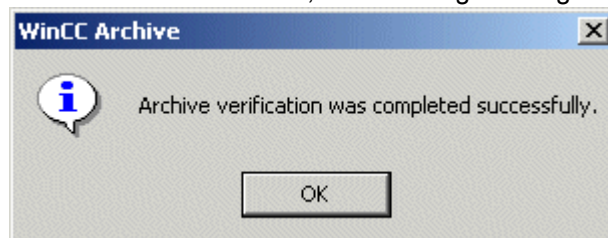


3. The selected checks are executed in succession. A check is only performed when the previous one has been completed without error. Every object which is checked and found to be free of errors is identified in the "Error text" column by "OK". Each erroneous object is identified in the "Status" column by the value "1". More detailed descriptions of the errors found appear in the "Error text" column.

If errors are detected during a check, this is indicated by a corresponding message. An option is provided to open a list of all faults found. This is a particular advantage in the case of large archives.



4. Clear all the errors found and run the test again. Repeat the procedure until no more errors are detected. In this case, the following message appears.



See also

Error list (Page 288)

Using the error dialog (Page 286)

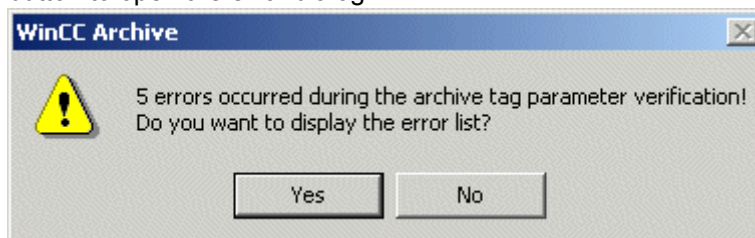
Using the error dialog

Introduction

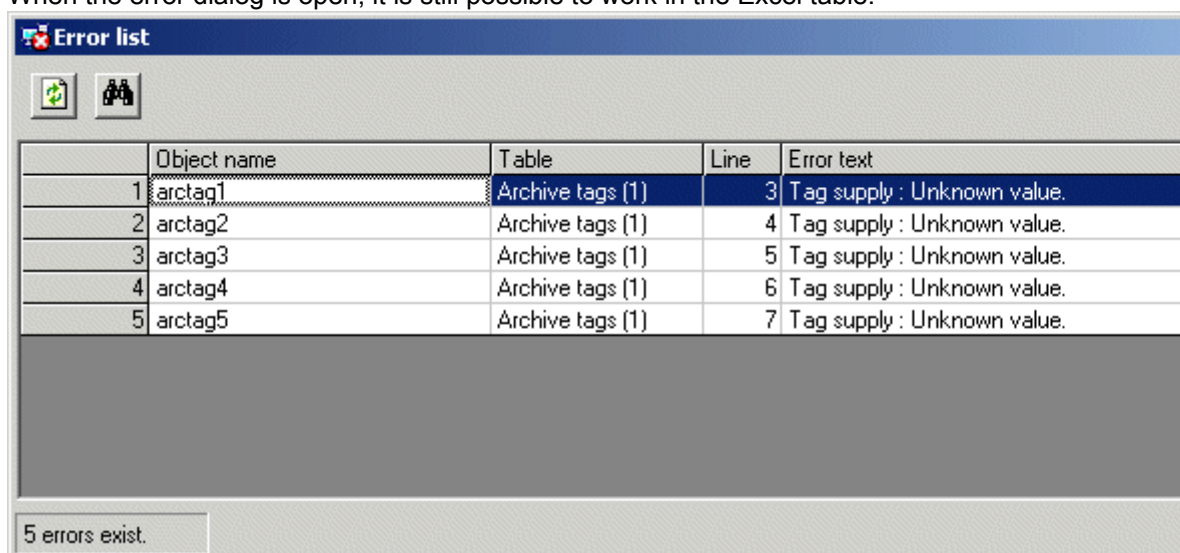
WinCC Archive provides users with a help function in the form of an overview of all the errors found in the archive data during the tests. The help is provided in the error dialog. This section describes the error dialog and how it is used.

Procedure

1. A message appears after the archive data has been checked. It provides information on the result of the test. If errors were found, the message below appears. Click the "Yes" button to open the error dialog.



2. The error dialog is opened. The error dialog contains a list of all faulty objects. This list displays each object with its name, the name of the associated table and the associated row number as well as with the error text.
3. Double-click an object in the list to skip directly to the erroneous object. The object is marked in the Excel table, the error can be cleared. The error dialog need not be closed to do this. When the error dialog is open, it is still possible to work in the Excel table.



4. If the error dialog has been closed, it can be reopened by double-clicking in the "Status" or "Error text" columns. This only applies, however, when the archive folder contains errors.

See also

Error list (Page 288)

Checking the archive data (Page 284)

Error list

Introduction

This section contains a list of all errors which could possibly be detected when the archive data are checked. There is a description of each error and instructions on clearing it.

Status column

The "Status" column may contain the contents listed below.

	Status	Description
1		The cell is empty. The object in this row has not been checked yet or the checks were completed successfully.
3	1	The cell contains the value 1. The object in this row was checked. An error was detected.

The Status column serves to locate erroneous objects quicker. Select any cell, for example, in the Status column. Press the "End" button. Then press one of the cursor keys, up or down. The selection skips to the next cell containing the value 1, i.e. to the next erroneous object.

Error text column

The Error Text column may contain the contents listed below.

	Error text	Description
1		The cell is empty. The object in this row has not been checked yet. Archive tags are only checked if a tag is specified to be archived (or an archive tag in a compressed archive).
2	OK	The cell contains the text "OK". The object in this row was checked. The checks carried out were completed successfully.
3	Processing: Unknown value.	The cell contains an error text. The object in this row was checked. An error was detected. The error text contains, on the one hand, the name of the column containing the source of the error and, on the other hand, a more detailed description of the error which occurred.

Error list

	Error	Description
1	Unknown error	An error has occurred which cannot be located more accurately.
2	Wrong data type	A different data type is required for this parameter. A text has been entered, for example, but a numeric value is required.
3	The text is too long	The length of the specified text exceeds the permissible value for this parameter.
4	The value is outside the permitted limits	The specified value is outside the permissible value range for this parameter.

	Error	Description
5	Invalid ratio of cycle times	The ratio between acquisition cycle and archiving cycle is not correct. The archiving cycle must be a whole number multiple of the acquisition cycle.
6	No cycle time specified	A cycle time must be specified for this parameter.
7	Unknown value	The specified text does not comply to any selectable options for this parameter. Parameter entry is supported by a combo box.
8	Tag was not found	The tag specified was not found in the WinCC project.
9	Cycle time was not found	The cycle time specified was not found in the WinCC project.
10	Tag is not analog	An analog tag must be specified for this parameter. This is necessary for the tag parameter, for example, when the archive tag type is defined as analog.
11	Tag is not binary	A binary tag must be specified for this parameter. This is necessary for the tag parameter, for example, when the archive tag type is defined as binary.
12	Invalid name	The name of the object does not comply with the required conventions. It may contain e.g. invalid special characters.
13	Name is not unique	The name of the object is not unique. It is used several times within the archive.
14	No name specified	No name has been assigned to the object.
15	No raw data tag	A raw data tag must be specified for this parameter. This is the case for the tag parameter concerning process controlled archive tags.
16	Configuration already in use	The configuration of this process controlled archive tag is already in use. It must be unique, however, within the archive. An internal archive tag name is generated from the configuration.
17	Invalid source	The source of the specified compressed archive tag is not correct. It is missing completely or partly, for example.
18	Source does not exist	The source specified for the compressed archive tag was not found in the WinCC project. The inherent archive must not be specified as the source archive.
19	Time value under one minute	A time of greater than or equal to one minute must be specified as the compressing period.

See also

Using the error dialog (Page 286)

Checking the archive data (Page 284)

1.9.5.6 Creating, modifying and deleting

Creating, modifying and deleting

Introduction

This section explains the general procedures for creating, modifying and deleting entire archives as well as individual archive tags.

See also

Deleting archives (Page 292)

Deleting individual archive tags (Page 296)

Creating and modifying individual archive tags (Page 294)

Creating a complete archive (Page 290)

Creating a complete archive

Introduction

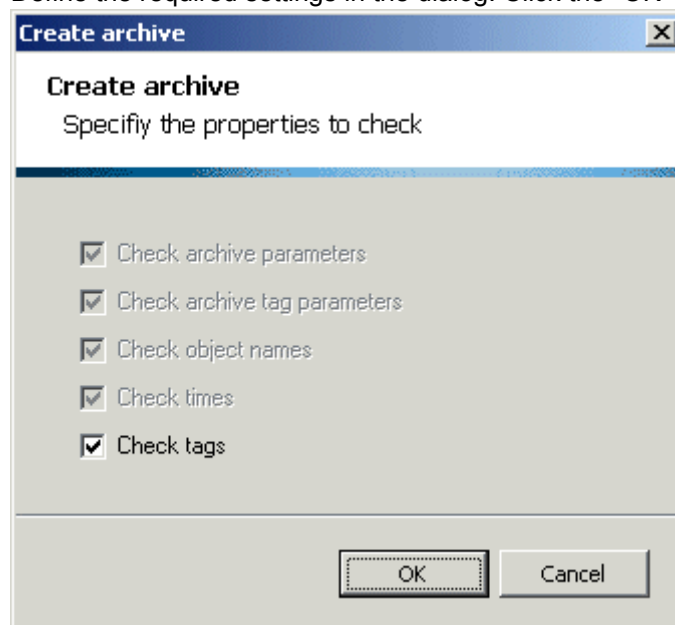
This description explains the procedures to create the archive configured in an archive folder. Only complete archives can be created using the options described. If the corresponding archive already exists, it is deleted.

Checking the archive data

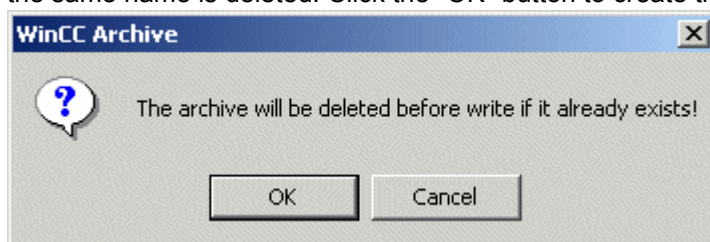
The configured data should be subjected to all available tests before an archive is created. In principle, it is also possible to execute the tests while the archive is being created. In this case, all the error-free objects are created, the erroneous objects, however, are not. After clearing the errors found, the complete archive can be created again (all the previously created objects are deleted beforehand). It is also possible to create the erroneous objects selectively. As opposed to checking the data before creating the archive, the two other methods described are much more time-consuming.

Procedure

1. Open the archive folder which contains the archive to be created. The configured data should be checked beforehand. Select the menu items "WinCC archive -> Create archive". The "Create archive" dialog is opened.
The "Create archive" dialog displays all checks performed during archive creation. A major part of the tests is always executed and cannot be deactivated. This concerns all the checks requiring a comparatively small amount of time.
Checking the tags, or in the case of compressed archives checking the archive tags, can be deactivated. These checks may be very time-consuming in the case of larger WinCC projects. If the check was made prior to creating the archive, it is not longer necessary during writing. It can be deactivated in this case.
2. Define the required settings in the dialog. Click the "OK" button.



3. Before the archive is created, a message appears. It points out that an existing archive of the same name is deleted. Click the "OK" button to create the archive.



4. The archive is created. The archive tags are written to the WinCC project in blocks of up to 256 objects. When the creating process is completed, a message appears. It provides information as to whether errors have occurred or not. If errors have occurred, use the error dialog to locate the erroneous objects.



See also

Deleting archives (Page 292)

Deleting individual archive tags (Page 296)

Creating and modifying individual archive tags (Page 294)

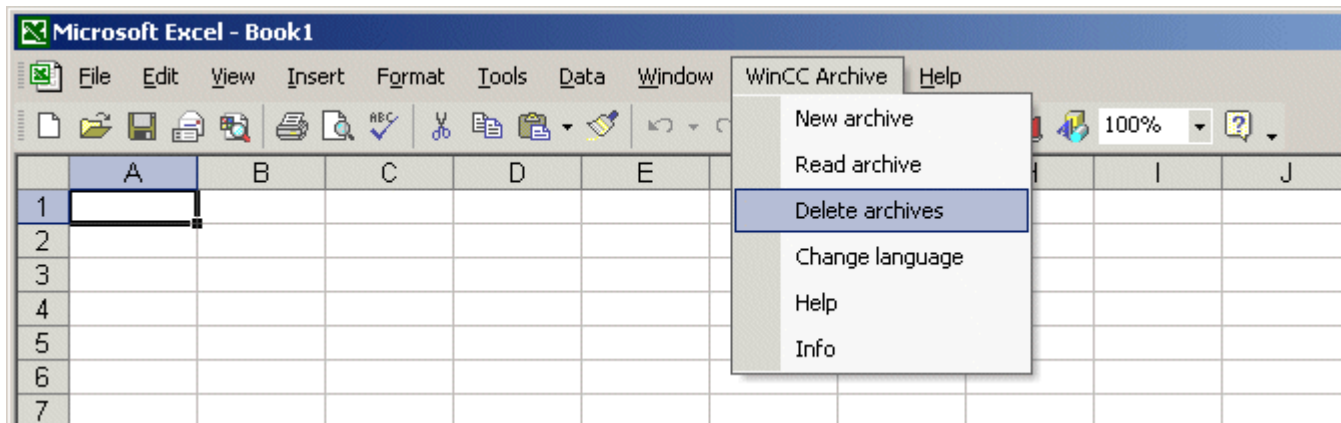
Deleting archives

Introduction

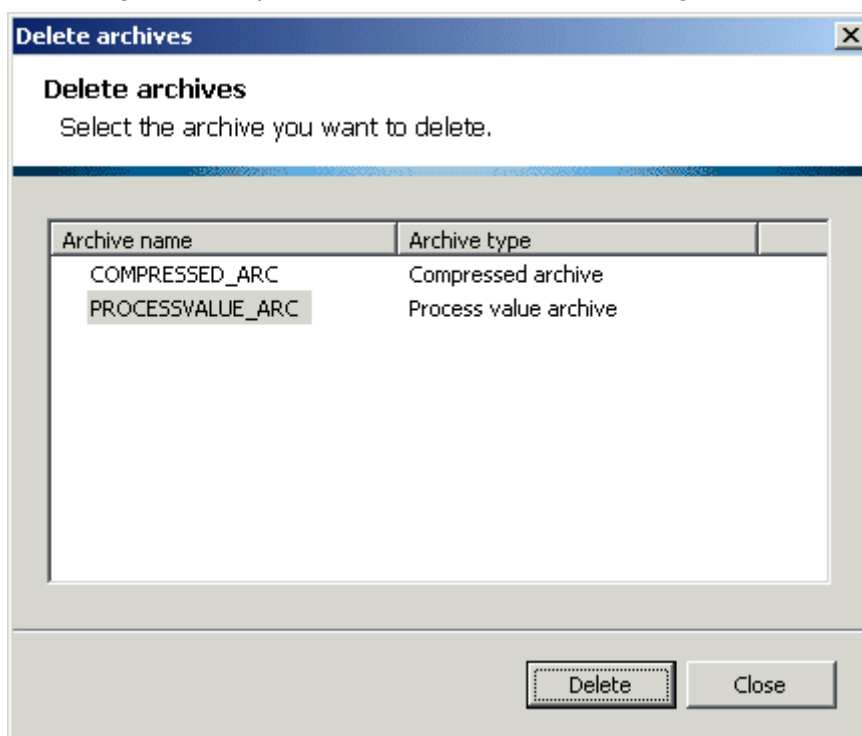
This description explains the procedures for deleting any archive from the WinCC project.

Procedure

1. Open Excel. The Excel add-in WinCC Archive is automatically started at the same time. The Excel menu contains the "WinCC Archive" item. Select "Delete archive" from this menu item.



2. A dialog opens containing a list of all the archives available in the WinCC project. Select the archive to be deleted and click "Delete". The archive is deleted from the WinCC project. The dialog remains open. Click "Close" to close the dialog.



See also

- Deleting individual archive tags (Page 296)
- Creating and modifying individual archive tags (Page 294)
- Creating a complete archive (Page 290)

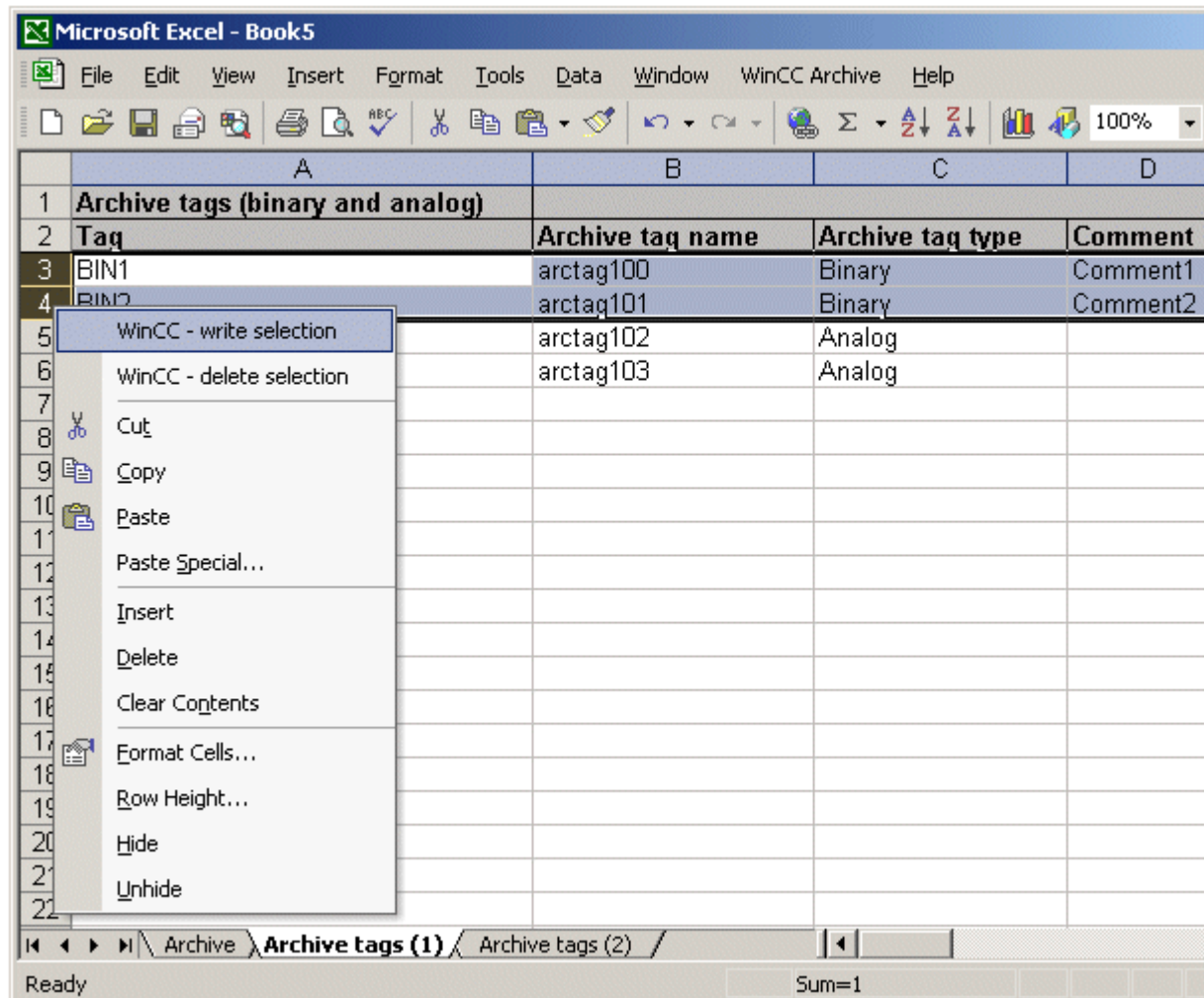
Creating and modifying individual archive tags

Introduction

This description explains the procedures for adding individual archive tags to an existing archive and the procedures for modifying existing archive tags.

Procedure

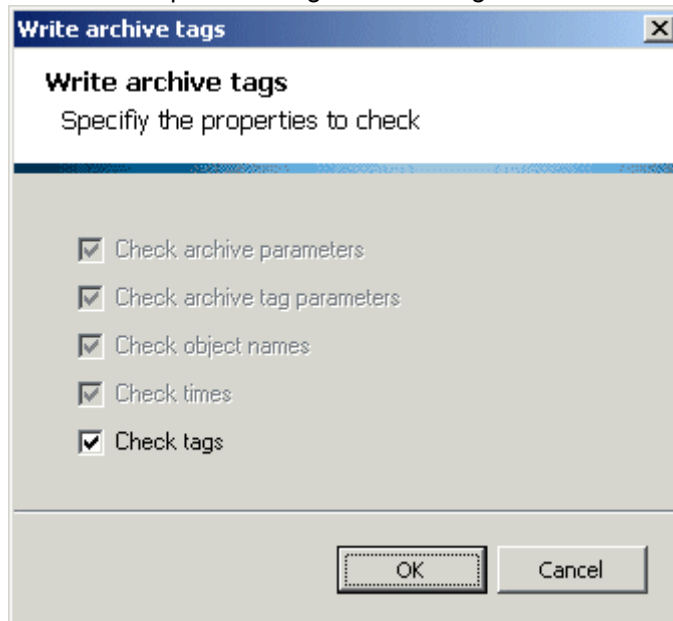
1. Open the archive folder which contains the archive to be edited. The archive to be edited can also be read in from the WinCC project using the menu items "WinCC archive -> Read archive".
2. Carry out the necessary modifications to the archive. Add new archive tags or modify existing archive tags. Select the newly added or modified objects. Open the pop-up menu and select "WinCC - write selection".



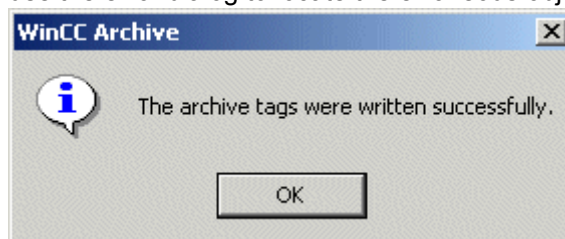
3. The "Write archive tags" dialog is opened. The dialog displays all the checks executed while the archive tags are being written. A major part of the tests is always executed and cannot be deactivated. This concerns all the checks requiring a comparatively small amount of time.

Checking the tags, or in the case of compressed archives checking the archive tags, can be deactivated. These checks may be very time-consuming in the case of larger WinCC projects. If the check was executed beforehand, it need not be executed again during the write process. It can be deactivated in this case.

4. Define the required settings in the dialog. Click the "OK" button.



5. The archive tags are written. When the writing process is completed, a message appears. It provides information as to whether errors have occurred or not. If errors have occurred, use the error dialog to locate the erroneous objects.



See also

- Deleting archives (Page 292)
- Deleting individual archive tags (Page 296)
- Creating a complete archive (Page 290)

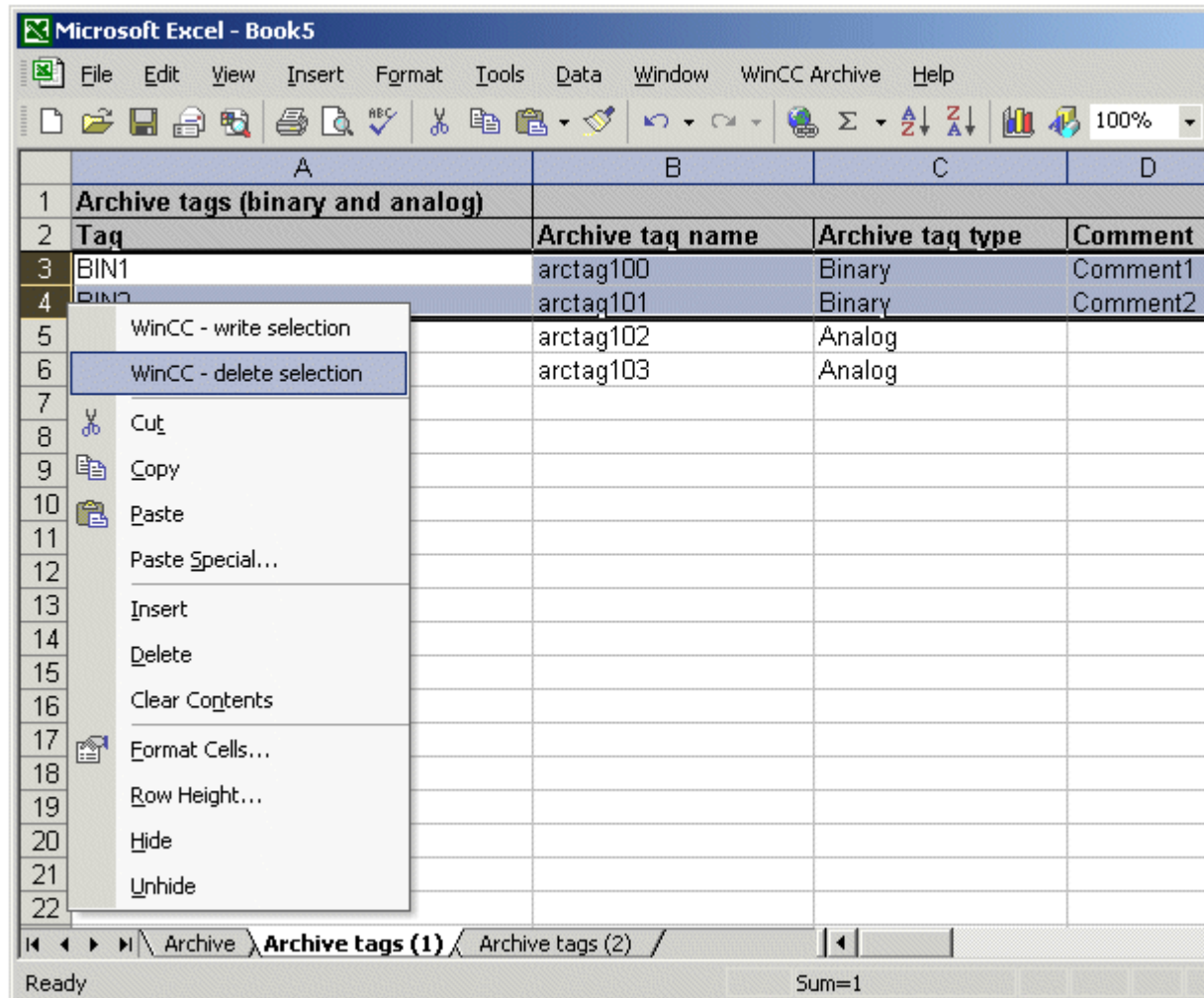
Deleting individual archive tags

Introduction

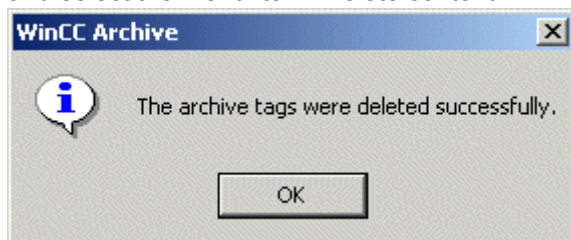
This description explains the procedures for deleting individual archive tags from an existing archive.

Procedure

1. Open the archive folder which contains the archive to be edited. The archive to be edited can also be read in from the WinCC project using the menu items "WinCC archive -> Read archive".
2. Select the archive tags to be deleted. Open the pop-up menu and select "WinCC - delete selection".



3. The archive tags are deleted. When the deleting process is completed, a message appears.
4. The archive tags deleted from the WinCC project are not deleted from the Excel table. To delete archive tags from the Excel table, press the "Del" button or open the pop-up menu and select the menu item "Delete content".



See also

- Creating and modifying individual archive tags (Page 294)
- Deleting archives (Page 292)
- Creating a complete archive (Page 290)

User archive

2 Resources

2.1 Introduction to User Archive

In the introduction to the User Archives you will receive the following information:

- Application Areas of the User Archives
- The components of the Editor User Archives
- Configuration and Runtime
- Functional Scope of User Archive

WinCC's "User Archives" editor can be used to continually save data from technical processes on a Server PC. In Graphics Designer, a WinCC UserArchiveControl can be configured that will permit a tabular display of the online data from the user archive in runtime.

User Archives are also used in order to prepare data for automation systems such as e.g. S5, S7. When necessary, the data can be read from the controller in the form of recipes or setpoint values.

The Editor User Archive offers two type of database tables:

- User archives: User Archives are database tables in which the user can create his own data fields. User Archives are used for the storage of data and offer a standardized access to these data according to the SQL database conventions.
- Views: Views receive the data from the user archives and are used for the summarization of data, for example, in order to receive overviews about product groups.

For the creation and editing of the user archives there are two possibilities:

- The Editor User Archive for the comfortable interactive configuration of the user archives.
- The functions for the editing of user archives in the WinCC script language.

With the functions of the WinCC script language you can achieve also multiple actions for the runtime mode. In the Runtime screen you can configure a table, which is connected directly with the process image of the automation system.

See also

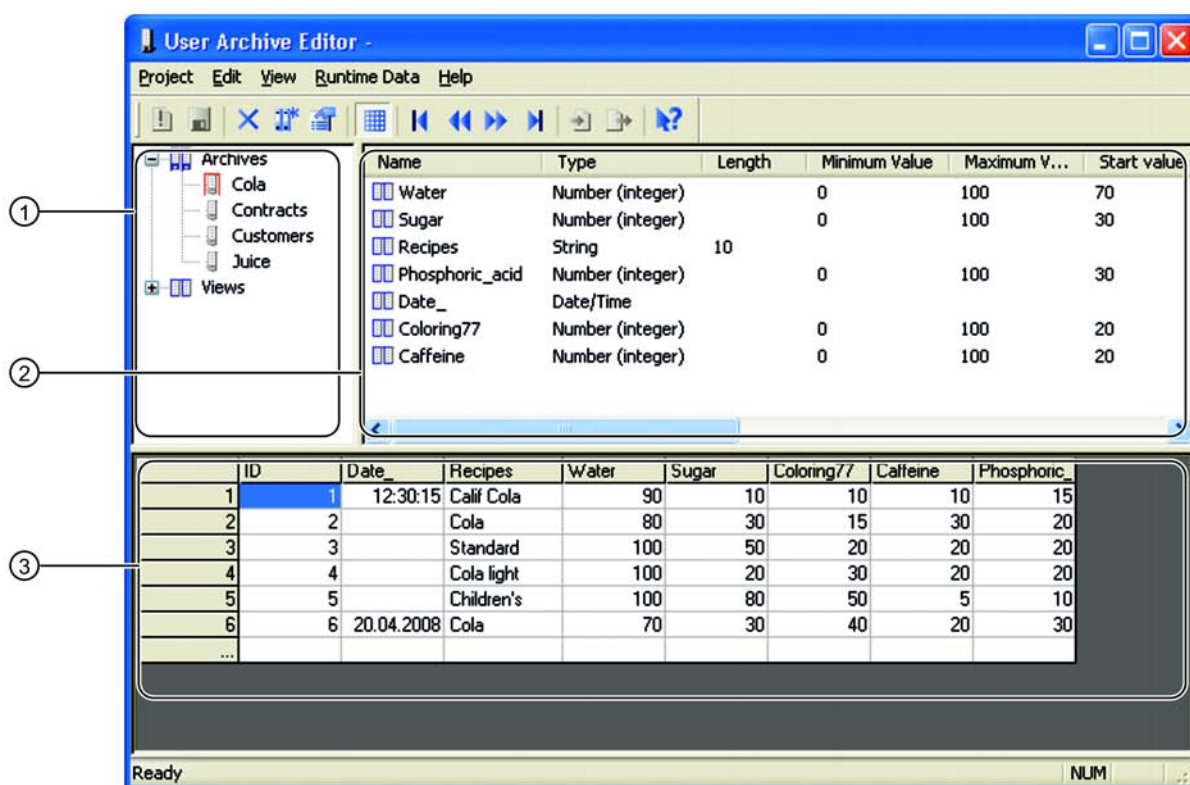
Application areas for the User Archive (Page 302)

2.2 The Editor "User Archive"

2.2.1 Introduction to Editor "User Archive"

2.2.1.1 The Editor "User Archive"

The Editor "User Archive" offers with its Windows user interface a comfortable way to create and maintain User Archives. The workspace of the Editors "User Archive" is divided into three areas:



- ① Navigation window
- ② Data Window
- ③ Table window

- the **Navigation window** for selecting user archives and views.
- the **Data window** for displaying and changing fields. In the data window the fields of the user archives and views are displayed, which have been selected in the navigation window.
- the **Table window** for displaying and modifying online data of selected user archives and views. In the table window of the Editor an online connection to the process images of the PLC's is possible.

The navigation and data window of the editor offers a fast access to all the elements of the user archives with a user interface similar to Explorer. The creation and changing of user archives is done user-friendly with dialog boxes and wizards.

See also

Application areas for the User Archive (Page 302)

2.2.1.2 The WinCC UserArchiveControl

A WinCC UserArchiveControl can be configured with the Graphics Designer. The UserArchiveControl is for displaying and changing the user archive data in runtime.

The basic properties of the WinCC UserArchiveControl:

- The UserArchiveControl is operated with icons and key combinations.
- You can create, change and delete the contents of fields in the UserArchiveControl.
- Browsing eases the access to large user archives as well.
- You can import and export user archives.
- Selection criteria and sorting conditions can be used for structuring the displayed contents.
- By a direct connection to the automation system you can read and write data online.

During configuration, a UserArchiveControl is connected to a selected user archive or form and can then only access that user archive or form. For access, the user archive / form must be enabled (access protection). Specific authorizations can be assigned to the UserArchiveControl in the User Administrator.

If this access protection is canceled, the UserArchiveControl must be reconnected in the Graphics Designer to the user archive so that the UserArchiveControl detects the canceled access protection.

Access protection for an archive or field is queried on opening a screen of a UserArchiveControl. Access protection for the control tags of a protected archive must be implemented separately via the object properties, e.g. of the picture, I/O field or button.

Note

Previous to WinCC V7, the display of the user archives was configured in a User Archives Table Element.

2.2.1.3 The standard function of the WinCC script language

The functions of the WinCC script language are divided to:

- **Configuration functions** for configuring the user archives
- **Runtime functions** for configuring actions in runtime operations

The runtime functions are activated by actions in the runtime screen, for example a mouse click on a specific button. The WinCC script language is based on the high language C, and the database functions are based on the SQL standard.

See also

Introduction to User Archive (Page 299)

2.2.1.4 Application areas for the User Archive

Functional scope

For the configuration of User Archives you can create your own database tables with the "User Archive" editor or with the functions of the WinCC script language.

The "User Archive" editor allows also the creation of new data records and the editing of data in existing data record even during the configuration.

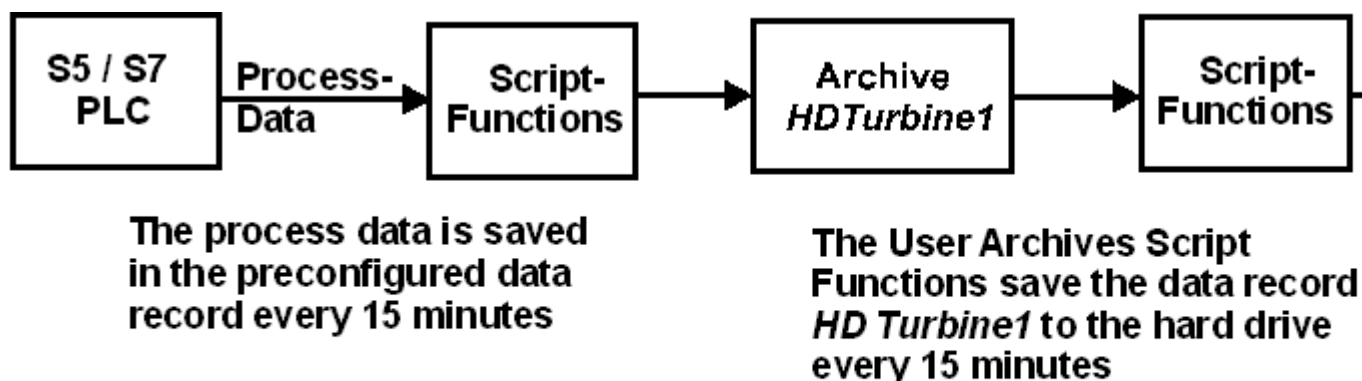
In runtime user archives (synonym with database tables) in the picture windows of the WinCC UserArchiveControl can be shown as tables. By raw data tags or WinCC tags a continuous data exchange can be done with the AS.

Example with operational data recording of Turbine

An energy production operator sets the user archive "HDTurbine1". This user archive is used for the operation status monitoring of a high pressure turbine. The User Archive "HDTurbine1" has the following data fields:

HDTurbine1
Index
Speed
Inlet pressure
Exit pressure
Steam temperature1
Steam temperature2
Oscillation frequency
Oscillation amplitude
Storage temperature1
Storage temperature2

In runtime you can then save in set intervals the operation data of the turbine as data records of the user archives on the PC mass storage (hard disk):



You can use the functions of the WinCC script language to analyze later data of the user archives, or visualized with the WinCC UserArchiveControl.

Example with recipes of a drink producer

An example for a data flow to the PLC are recipes. A drink producer who is producing in our example a Cola drink and an orange juice, uses the User Archives to prepare the recipes for the contents of his drinks for the AS.

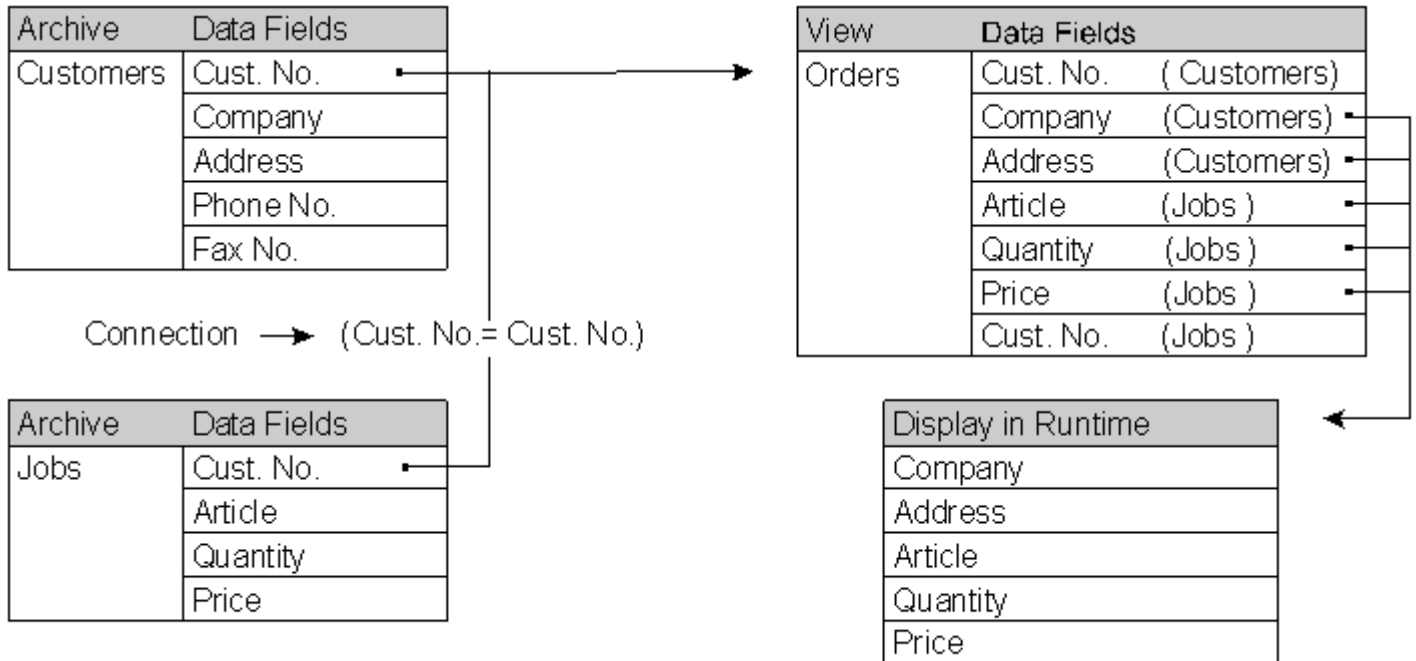
User archive	Data fields
Cola	Water
	Sugar
	Dyestuff7
	Phosphoric acid
	Caffeine

The user archives use the data interface that is provided by WinCC to the AS, either through the raw data of the WinCC data managers or via WinCC Tags. For the data transfer from / to the AS, WinCC provides a suitable set of C actions.

Application methods of views

WinCC offers as an additional performance sign "Views" to multiple user archives of a server. Views enable the summarizing of data fields from different user archives. So, for example, you can create logical operations in the SQL language through data fields of different user archives,

to display the desired relationships in runtime as forms. The used user archives must have at least one common factor.



In our example the user creates a form for orders. He gets the required information for the orders from the archives "Customers" and "Jobs". The customer number is the common factor of the two user archives and is used as the connection marker for the form. The user wants only the required fields of the connected user archives to be displayed in runtime.

Note

Available software, which has done direct ODBC database accesses to User Archives versions older than 4.02, will not be able to access the user archives of version 4.02 or later.

2.2.1.5 Functional Scope of User Archives

The performance markers of the User Archives are introduced below in short headings:

Configuration

- User archives and views can be created in table form, this will provide a simple, direct assignment of the data to the fields of the user archives or forms (division in columns and rows).
- Online display in runtime in table format
- Input / output of data through I/O fields (assignment of the user archive fields through C actions/ control tags)

Logging

- of the configuration data and the runtime data in table form through WinCC reports
- Export of the data in CSV Format (processing through external programs e.g. Excel)

Transfer from / to the AS (S5, S7 etc.)

- whole data records of a user archive (through raw data tags)
- individual data fields of a data record (through WinCC tags)
- Communication through all the interfaces provided by WinCC.

Editing possibilities

- in the table views
- through I/O fields (with C actions/ control tags)

Operation

- in tables through standardized icons
- with C actions

Delete or create new data records

- in tables create data records through buttons
- with C actions create and delete data records

Control tags

- Wizard supported creation of WinCC tags as control tags
- Fast access possibilities to User Archives for scripts and the AS
- Indirect addressing for C actions

Quantity framework

In the "User Archive" editor you can configure maximum 500 archives and 500 views of archives. Per archive maximum 500 fields can be created.

Archives

The maximum number of data records in an archive is limited and depends on the number of the configured columns and the data records contained in the archive. The product of columns and data record cannot be more than 320000. For the number of columns you must also include the column "ID" that was created by the system and if selected the columns "Last User" and "Last Access".

Example:

There are 15 individual columns configured in the archive and the column "Last Access" is selected. So, including the column "ID", a total of 17 columns is configured, which gives a maximum amount of $320000 / 17 = 18823$ data records.

See also

Application areas for the User Archive (Page 302)

2.2.2 The Editor "User Archive"

2.2.2.1 Structure of the "User Archive" editor

Configuration

The "User Archive" editor can be operated through its menus, its toolbar, by hotkeys or by direct mouse clicks.

Below you will receive information about the following topics:

- The menus of the "User Archive" editor
- The toolbar of the "User Archive" editor

2.2.2.2 The menus of the "User Archive" editor

The menus of the "User Archive" editor

Menu operation

The menu operation is described in this section. Functions that correspond to the Windows standard will not be described here.

The "User Archive" editor provides the following menus:

Menu	Menu command	Shortcut
Project	Restore	Ctrl + N
	Save	Ctrl + S
	Export...	
	Import...	
	Check...	
	Exit	
Edit	Cut	Ctrl + X

	Copy	Ctrl + C
	Paste	Ctrl + V
	Runtime Data	Ctrl + R
	Options	Ctrl + O
View	Toolbar	
	Status Bar	
	Split	
	Update	F 5
Runtime data	Import	
	Export	
Help	Help Topics	
	Log-File...	
	About...	

Note

The functions "Cut, Paste and Copy" can be used only in the data window. Only one user archive, field or form can be cut, copied or pasted at a time. It can only be saved when at this time no user archive is referred to (User Archive is referred to e.g. during the display of the table window in the configuration system (CS) or in runtime).

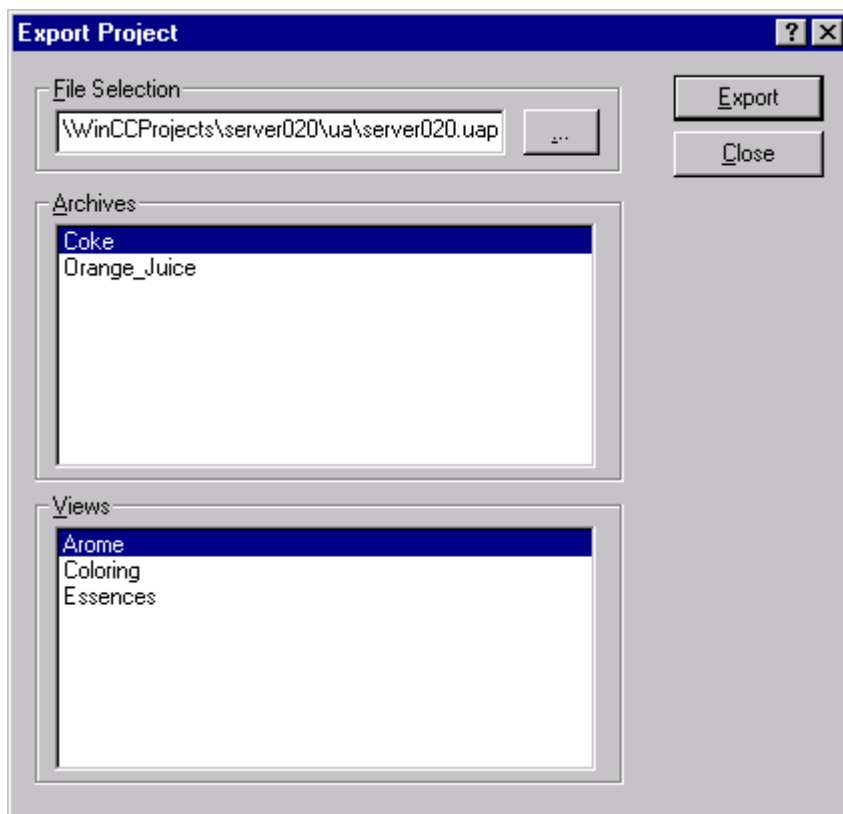
Menu command "Restore"**Restore**

By "Restore" the last changes are cancelled and the last saved state is restored without closing the editor. In addition, you can use this function to accept changes, which have been carried out and saved since the opening of the editor of scripts or external programs. Such external changes are not done to the editor automatically.

Menu command "Export" (Menu Project)

Export (menu project)

Use this menu command to export user archive and view structures (CS data) of the open WinCC project.



When you click on the Export command of the menu "Project", a dialog opens in which you can select between the user archives and views that were created in the open WinCC project. Simple and multiple selection are possible. In the area file selection, the project path of the open project and a file name of the project name and the file extension "uap" is set automatically. After you have clicked on the button for the file selection, a selection dialog is opened where you can set a free selectable memory location. After you have set the memory location, the export is carried out after clicking on the button "Export". Close the dialog after the export of the selected user archives and views.

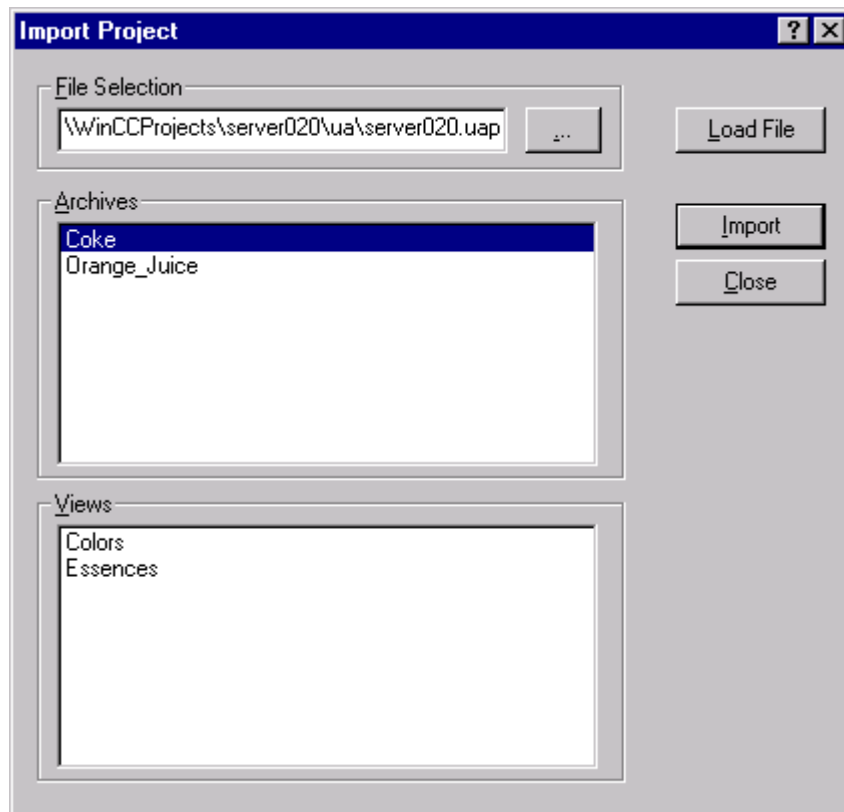
Note

In order to export runtime data, you must use the menu command "Export" in the menu "Runtime data".

Menu command "Import" (Menu Project)

Import (Menu project)

Use this menu command to import user archive and view structures (CS data) of the open WinCC project.



In order to be able to import user archives and views, first the respective user archives and views in the project to be imported must be exported in order to create a uap file. In order to start the import, click in the menu "Project" on the command "Import". A dialog opens for the selection of the user archives and views you want to import. Simple and multiple selection are possible. In the area file selection, the project path of the open project and a file name, consisting of the name of the project and the file extension "uap" is set automatically. When you click on the button for the file selection, a selection dialog opens where you can select the file you want to import. After you have selected the file, the import will be carried out after you have clicked on the button "Import". Close the dialog after the import of the selected user archives and views.

In order to maintain the consistency of the structure data, a view is always exported together with the related archives and imported during the import together with this view. Existing archives with the same name must not be overwritten during an import. If you want to keep the archive name, you must delete the archives with the same names that exist in the project before the import. Existing runtime data must first be saved, since they are also deleted during the deleting of the archives.

Note

In order to export runtime data, you must use the menu command "Import" in the menu "Runtime data".

Menu command "Check"

Checking

This menu command allows you to check whether the tags referred to in the Editor User Archives exist in the WinCC Explorer. If an error is detected, you will get the following message: "No errors were found during the check". If an error occurs, you will get the following error message.



The user archive, the related field and the missing tag in the tag management is displayed.

Note

Structure tags are not checked with this function.

Menu command "Runtime Data"

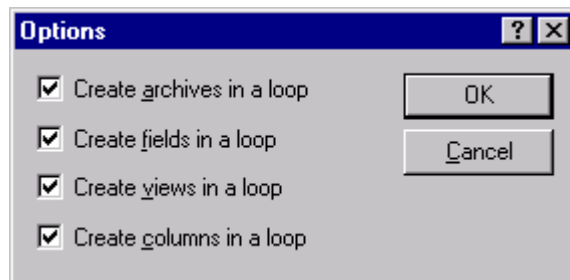
Runtime Data

This menu command allows you to edit the Online data in the table window. A check in the menu shows that the "Runtime Data" status is active.

Menu command "Options"

Options

Use this menu command to set how the creation of user archives and views shall be done. The following dialog box appears after clicking on this menu:



Create user archives in a loop:

If this option is active, the dialog box for entering further User Archives appears after the input of a user archive and its fields automatically.

Create fields in a loop:

If this option is active, the dialog box for entering further data fields appears after the input of a user archive data field automatically.

Create views in a loop:

If this option is active, the dialog box for entering further views appears after the input of a view and its columns automatically.

Create columns of a view in loop:

If this option is active, the dialog box for entering further columns appears after the input of columns automatically.

Menu command "Split"

Split

This menu command is used for changing the size of the three dividing windows of the Editor User Archives.

Menu command "Import" (Menu "Runtime Data")

Import (Menu-Runtime Data)

This menu command is used to import the data records (Runtime Data) into the selected user archive.

In the import file there are no information about the data type and number of the columns. Therefore the structure of the import data and the target archive must be equal or the import is done into the user archive from which the data were exported before.

During the export, data record IDs are entered into the export data, in order to facilitate a unique assignment of the imported data during the import. If WinCC realises during the import, that one of the IDs that will be imported already exists in the user archive, an error message is given and an entry is made into the log file "UALogFile.txt", specifying the related ID.

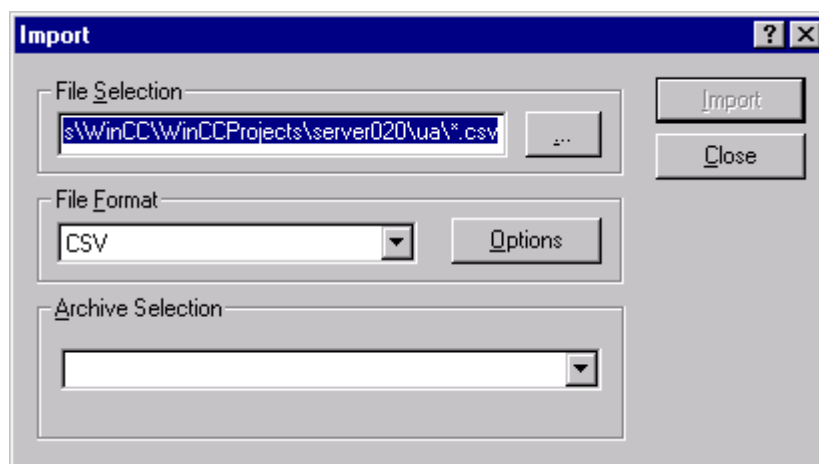
The data with a new data record ID are added as new data records into the user archive.

Note

If data will be imported, which come from the current user archive, were edited outside of WinCC and will now overwrite the existing archive data, first all data records of the archive must be deleted. Otherwise there are error messages during the import because of equal data record IDs.

In order to import structures from the user archive and view structures, you must use the menu command "Import" in the menu "Project".

The menu item is disabled if the function "Runtime Data" is enabled (Menu "Edit").



In the field "File Selection" enter the path and file information of the user archive to be imported. The button "..." supports you during the file selection. The file path is set automatically on the folder "ua" in the project path of the active user archive.

In the field "File format" you can select the file format of the user archive that will be read from. Use the button "Options" to specify the desired separator sign. The default separator is the semicolon ";".

In the field "Archive Selection" select one of the User Archives of the current project as the target archive. After the selection the button "Import" is enabled.

After the button "Import" is used the import will be carried out.

Note

With a Client-Server Project the following has to be considered: If there is a user archive on the server, e.g. at "c:\Projects\Test\UA", it is enabled with this specified path. The client maps the enablement via a network drive e.g. "I:\Test\UA". Thereafter, the standard path of the User Archive is on the client "I:\Test\UA". However, this directory does not exist on the server with this description. If you want to import/export user archive to the client, you have to change the standard path on the client, in our example to "C:\Projects\Test\UA".

Menu command "Export" (Menu "Runtime Data")**Export (Menu Runtime Data)**

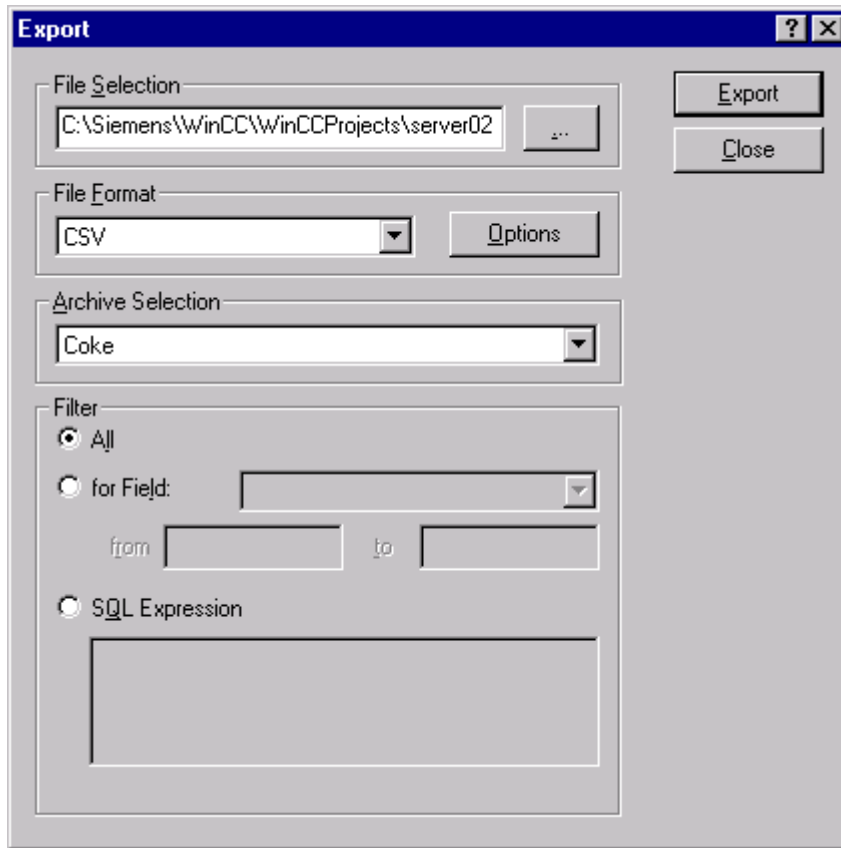
Use this menu command to export the data records (Runtime Data) of the selected user archives. The exported data can be edited in another user program e.g. MS-Excel and then be imported back into the user archive.

Note

The menu item is disabled if the function "Runtime Data" is enabled (Menu "Edit").

In order to export structures from the user archive and view structures, you must use the menu command "Export" in the menu "Project".

After WinCC V5.1 the column headings are exported as well and must not be changed.



In the field "File Selection" enter the path and file information of the user archive to be exported. The button "..." supports you during the file selection. The file path is set automatically on the folder "ua" in the project path of the active user archive.

In the field "File format" you can select in which file format the user archive will be exported. Use the button "Options" to specify the desired separator sign. The default separator is the semicolon ";".

In the field "Archive Selection" select one of the User Archives of the current project as the target archive. After the selection the button "Export" is enabled.

In the area "Filter" you can specify a filter for the export of the user archive, if necessary. In the field "Filter on field" specify the field to which the filter will refer. In the fields "from... to" give the value range which shall pass through the filter.

After enabling the option button "SQL Statement" you can enter in the input field under it a filter statement in the SQL language. For more about SQL statements refer to the appendix.

After the button "Export" is used the export will be carried out.

During the export, data record IDs are entered into the export file, in order to facilitate a unique assignment of the imported data during the import.

Note

With a Client-Server Project the following has to be considered: If there is a user archive on the server, e.g. at "c:\Projects\Test\UA", it is enabled with this specified path. The client maps the enablement via a network drive e.g. "I:\Test\UA". Thereafter, the standard path of the User Archive is on the client "I:\Test\UA". However, this directory does not exist on the server with this description. If you want to export user archive data, you have to change the standard path on the client, in our example to "C:\Projects\Test\UA".

2.2.2.3 The toolbar of the "User Archive" editor

Toolbars

The "User Archive" editor can be operated by direct mouse clicks on its toolbar. The individual symbols on the toolbar are described below in alphabetical order.

Symbol	Description
	Browse
	Properties
	Export
	Help
	Import
	Delete
	New
	Runtime Data
	Save
	Restore

Browse

The "Browse" symbols allow you a simple browsing in the user archive in runtime.

Properties

Use the "Property" symbol to edit the properties of the User Archives or data fields. A click with the right mouse button onto a user archive or data field enables also the editing of the properties of a user archive or data field.

In runtime you can move only at activated "Properties" symbol during the editing of the data fields with the cursor control keys through the table, the fields can here be edited immediately. The button "Properties" can be enabled only after a data field has been selected.

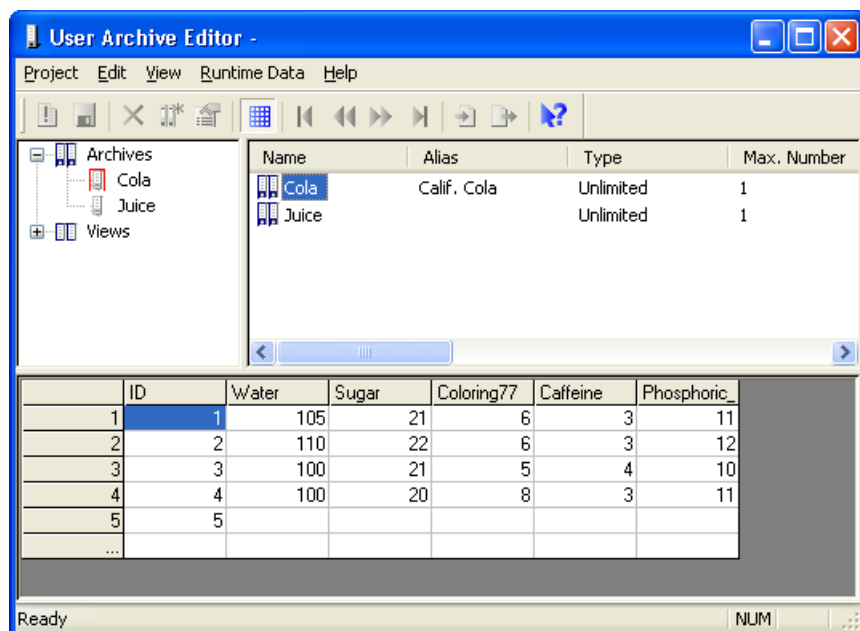
Delete

Use the "Delete" symbol to delete user archives or data fields. A click with the right mouse button onto a user archive or data field enables also the deleting of an archive or the data field. In addition, it is also possible to click on a user archive or data field and then press the "Delete" key.

New

Use the "New" symbol to create new user archives or data fields. A click with the right mouse button in one of the top windows allows also the new creation of a User Archive.

2.2.2.4 The table window of the "User Archive" editor



You can use the menu item "Edit -Runtime Data" or with the corresponding button to switch on or off the table field. You can use a double click on one of the fields to enable the data input. This is marked by the text cursor. In runtime you can move only at activated "Properties" symbol during the editing of the data fields with the cursor control keys through the table, the fields can here be edited immediately. The button "Properties" can be enabled only after a data field has been selected. The editing functions can also be used in the table field through a pop-up menu. In order to copy data records into an external program, select the desired table lines and copy them with the key combination "Ctrl" + "c" into the clipboard. The pasting into the external program is done with the hot key "Ctrl" + "v". The pasting of external data into the table window of the "User Archive" editor cannot be done by this method.

Note

If one or more values are changed in the table field of the "User Archive" editor or in a UserArchiveControl table, you must exit the data record (by clicking on another table cell or row) after making the entry in order for the value to be accepted into the database and be updated in all displays.

2.2.3 Configuration

2.2.3.1 Configuration

The first step is the configuration of a new user archive. For the configuration you can use wizards, which offer a comfortable, user-controlled method. The following configuration steps are necessary:

Configuration of User Archives

- Create User Archives
- Setting of the user archive fields

Configuration of views

- Create view
- Setting the data fields
- Setting the relations

Configuration of WinCC UserArchiveControl

- Creating a UserArchiveControl

2.2.3.2 Configuration of a User Archive

Create new user archive

Create new user archive

Start the Editor User Archives from the WinCC Explorer.

- In order to do so, click on User Archives and select in the pop-up menu the menu item "Open". You will see the command surface of the Editor User Archives on the screen.

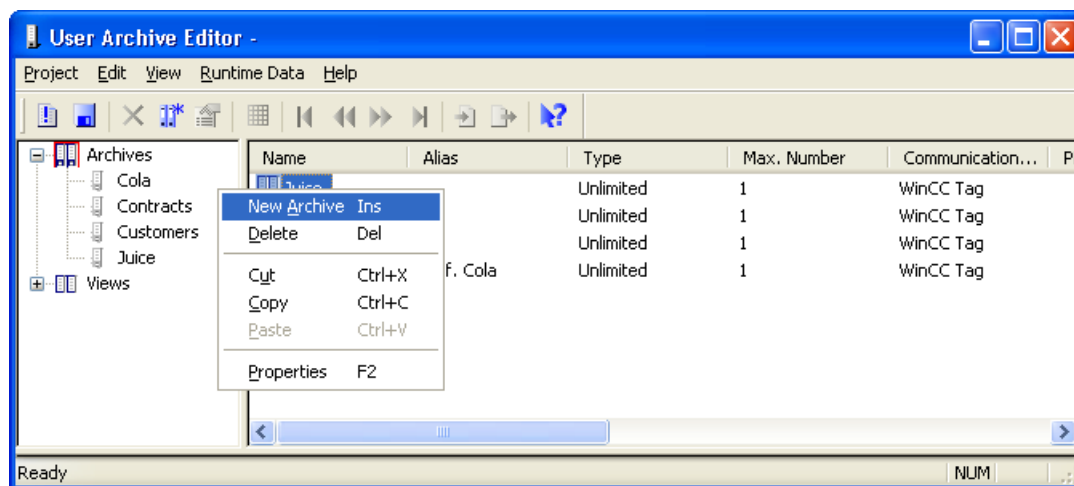
Note

Per User Archive 500 fields can be created.

The default option "Put archive in cycle" allows the creation of multiple user archives one after another. If you want to create only one User Archive, you can disable this option in the menu "Edit -Options".

To create a User Archive follow the steps.

1. Click on the navigation window with the left mouse button on "Archives".
2. Then click in the navigation or data window with the right mouse button. The pop-up menu that is displayed in the picture below appears.



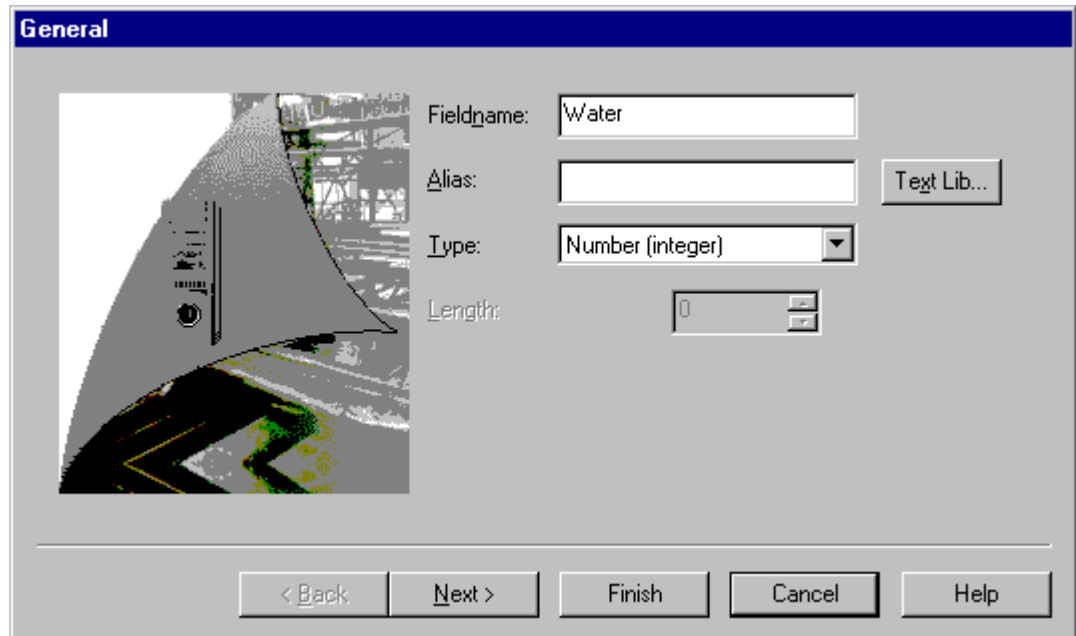
3. Select the option "New Archive"

The wizard for the configuration of User Archive appears.

General properties of user archive fields

Specifying general properties of user archive fields

In the dialog "General" you will determine the user archive field, which you wish to create as well as the field type. With the field type "String", an additional field length can be specified.



- In the input field "Field name" you can enter the name for the first user archive field; in our example, the first recipe ingredient "Water".
- In the field Alias you can give the field a second name in order to comment the field or to realize a language change through the text library for the display in Runtime. This input is optional.
 - For more information on switching the language, refer to the Online Help.

The entered names serve for the future assignment of the fields for the table view.

In the input field "Type" you can specify one of the following tag types:

- Integer Signed 32-bit value
- Float Floating-point number 32-bit IEEE 754
- Double Floating-point number 64-bit IEEE 754
- string Text tag, 8-bit character set
- Date/time no special data type available

Note

For the tag type "Date/Time" the input format for the date and the time depends on the settings in the operating system.

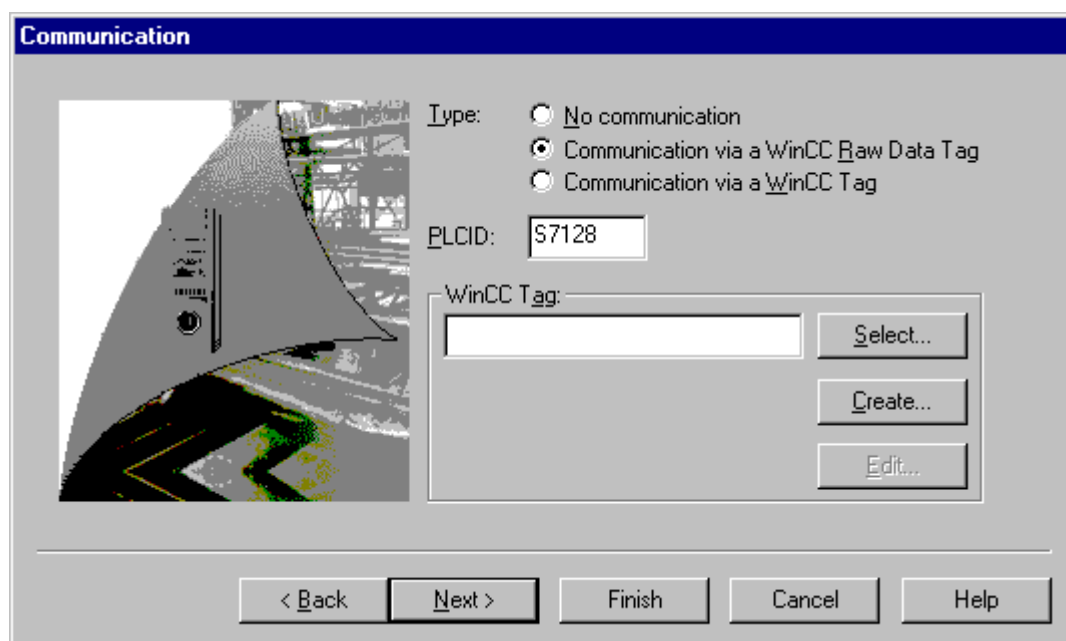
See also

Changing language (Page 351)

Settings for Communication

Settings for Communication

In the dialog box "Communication" you can set the connection type between the PLC and the user archive:



Under the item "Type" you can specify the communication type:

- **none:** no communication possible
- **via raw data tag:** Access to AS via raw data tag.
- **via WinCC tag:** Access to AS via WinCC tags

In order to obtain a link through raw data tags, click on "Through Raw Data Tags". Enter "PLCID" as archive identification. The "PLCID" exists of exactly 8 ASCII characters and is unique within the WinCC Project. It identifies the respective user archive and is the prerequisite for the AS to return the process image data to the correct user archive.

If you have selected "through raw data tag", you can click on "Select" and then select a raw data tag.

If you have selected communication through the WinCC Tag, the assignment of the tags is done in the properties dialog of the user archive fields.

Note

For the communication through raw data tags a complete data record is connected to a raw data tag. If the WinCC Tag is used, a tag is connected to a user archive field.

For the communication of User Archive through raw data tags the PLCID is used as the unique name of the archive. The R_ID that was configured in the used raw data tag cannot be used, since it is relevant only for the communication with the AS. In addition, multiple user archives can be supplied by the same raw data tag.

Setting Control Tags**Setting Control Tags**

On the tab "Control Tags" the Control Tags are set in the form of WinCC tags, which can be used to access the user archive field.

In the four input fields of the tab the WinCC tags will be set, which can be used to access the data record IDs, job codes, archive fields and archive field values.

Next to each input field you can see a button "Select" which can be used to open the tag selection dialog. Here you can display and select all existing WinCC tags.

Use the button "Create..." to create the tags automatically. You can use this to create a new tag group "@UA[Archive name]" and save the created tags there in the form @UA[Archive name]ID, @UA[Archive name]Job etc.

Note

In order to check the function, always all four Control Tags of a User Archive must be given or created with the Wizard.

The data type of the Control Tags must not be changed.

When a User Archive is created, we recommend you to use the wizards for the creation and saving of the Control Tags.

With the help of these four Control Tags you can trigger a User Archive. For the triggering you must supply either the tags "ID" and "Job" or the tags "Job", "Field" and "Value" with the respective values.

More information about the PLC of the User Archive can be found by the Control Tags in the chapter "Properties of Control Tags".

For instance, if you do not want to use the Control Tags, you can quit the dialog without any input. You can find an example for the use of Control Tags in the chapter "Example for the use of the Control Tags".

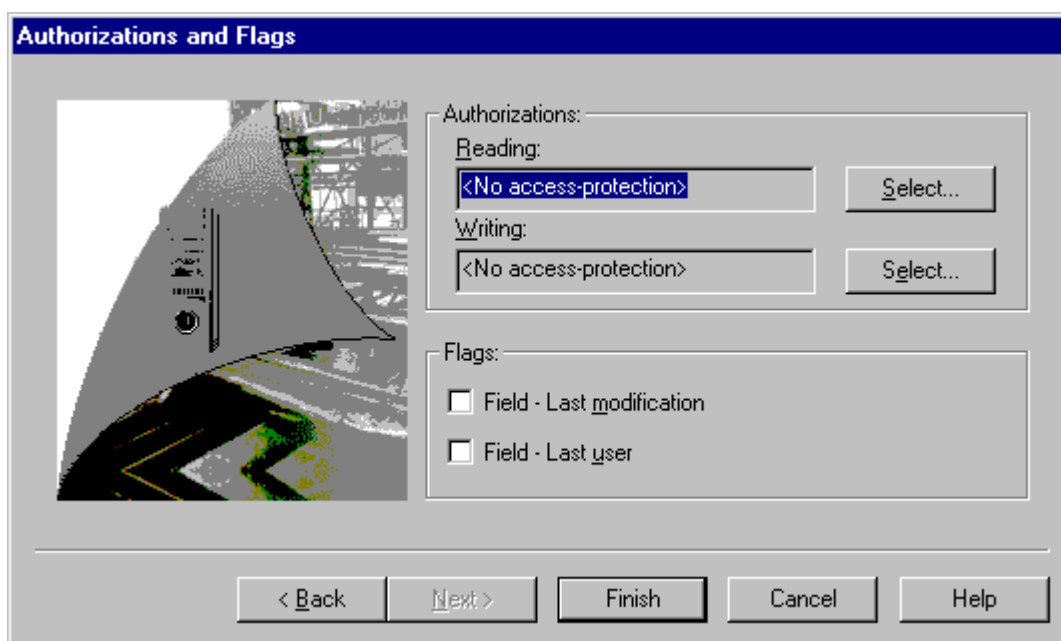
See also

Example for the use of control tags (Page 334)

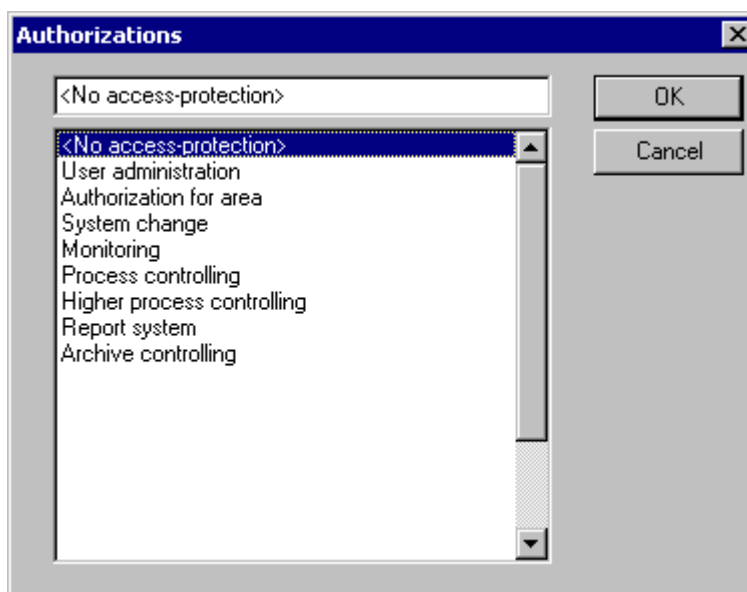
Setting Authorizations and Flags

Setting Authorizations and Flags

The dialog "Authorizations and Flags" is used for the setting of the access authorizations to the User Archives and the settings for the output of the last access / user in separate columns.



All the currently set authorizations are displayed for reading and writing accesses. In order to change these settings you can click on one of the "Select" buttons. Then you will get the dialog box "Authorizations" where you can select between the authorizations that were set in the User Administrator:



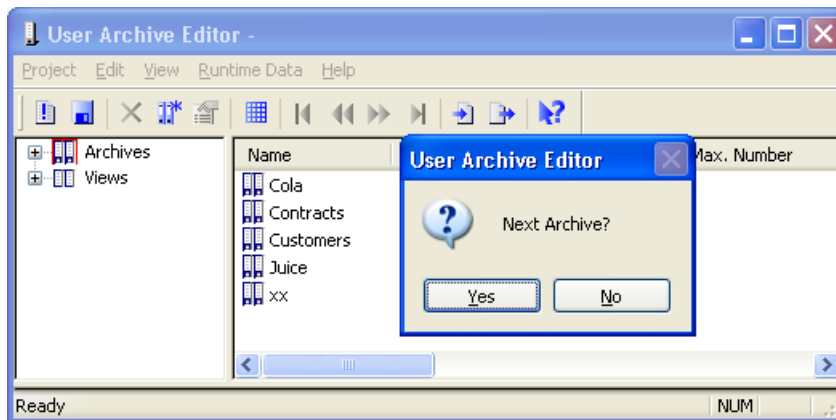
By activating the option "Field - Last modification" a column is set with date and time of the last access. With the Option "Field - Last user" a column will be set, where the user is listed who has accessed the user archive last.

2.2 The Editor "User Archive"

1. Selecting one of the authorizations
2. Select for instance the field "Last User".
3. Exit the user archive creation with the key "Finish".

After the User Archive is created, the query "Add fields?" appears. If you confirm with "Yes", the dialog "General" is opened for the creation of user archive fields. For information on how to create user archive fields, refer to the chapter "Create user archive fields".

If you have activated the option "Create archives in cycle" in the menu "Edit - Options", the query "Next Archive ?" appears after the fields have been created. If you click on "Yes" the initial dialog box "General" appears for the input of the next User Archive.



Save the new User Archive at the end by a mouse click on the Save symbol or by activating the menu "Project - Save".

Note

Changes on the User Archives are adopted into the database only by "Save". If a User Archive will be aligned through the option "Redundancy", the flag "Last Access" must be activated.

The properties of the user archive "Cola" in our example are:

User Archives	Properties
Cola	Name: Cola
	Alias: Calif Cola
	Type: Unlimited
	Max.Recs: 1
	Com.Type: Raw
	PLCID: S7112
	Var.Name: CalifVarGroup
	Right read: 0
	Right write: 0
	Flags: U

	Pos.: 3
	Last access: 03/05/98 12:54

See also

Create new user archive fields (Page 325)

Create new user archive fields

In this section you will learn how the data fields of a user archive are created.

The default option "Put archive in cycle" allows the creation of multiple fields one after another. If you want to create only one field, you can disable this option in the menu "Edit -Options".

1. Expand "Archive" in the Navigation Window (click on the "+" character). The new User Archive "Cola" is then displayed in the navigation Window.
2. Click on the navigation window with the left mouse button on the user archive "Cola". The following pop-up menu is displayed:

<u>N</u> ew Field	Ins
<u>D</u> elete	Del
<u>R</u> estore RT Data	Ctrl + Y
<u>P</u> roperties	F2

1. Click on the menu item "New Field".

The dialog "General" will appear.

Note

In case of changes of the user archive fields data can get lost under the following circumstances:

If for existing data a new consistency condition cannot be fulfilled anymore, such as for instance for Unique, Not Null etc.

If a field name has been renamed.

When a new data type cannot convert the data from the source anymore.

More information about the changing of the user archives can be found under the chapter "Change configuration of User Archives".

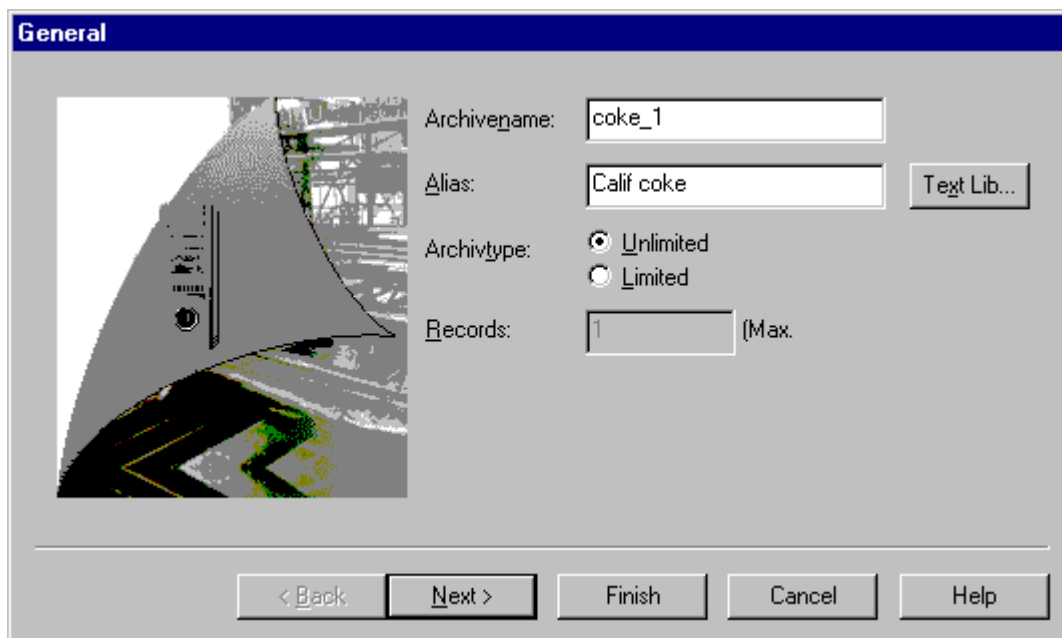
See also

Modification Configuration of User Archives (Page 339)

General Properties of User Archives

Setting general properties for User Archives

You can see the dialog "General", which you can use to create a new user archive.



Enter as the user archive name for example "Cola". In the field "Alias" you can give the user archive a second name, for instance "Calif Cola", in order to comment the user archive or to realise a language change through the text library in runtime. This input is optional.

For more information on switching the language, refer to the Online Help.

If you specify the archive type "Limited", you can set the maximum number of data records in the field "Number". The type "Unlimited" creates user archives with an unlimited number of data record.

Note

Keywords (or reserved words) of the database language SQL must not be used as archive or field names. For more details refer to the chapter "Alphabetical List of SQL Keywords".

When creating data records it will not be checked whether these are complete or correct.

See also

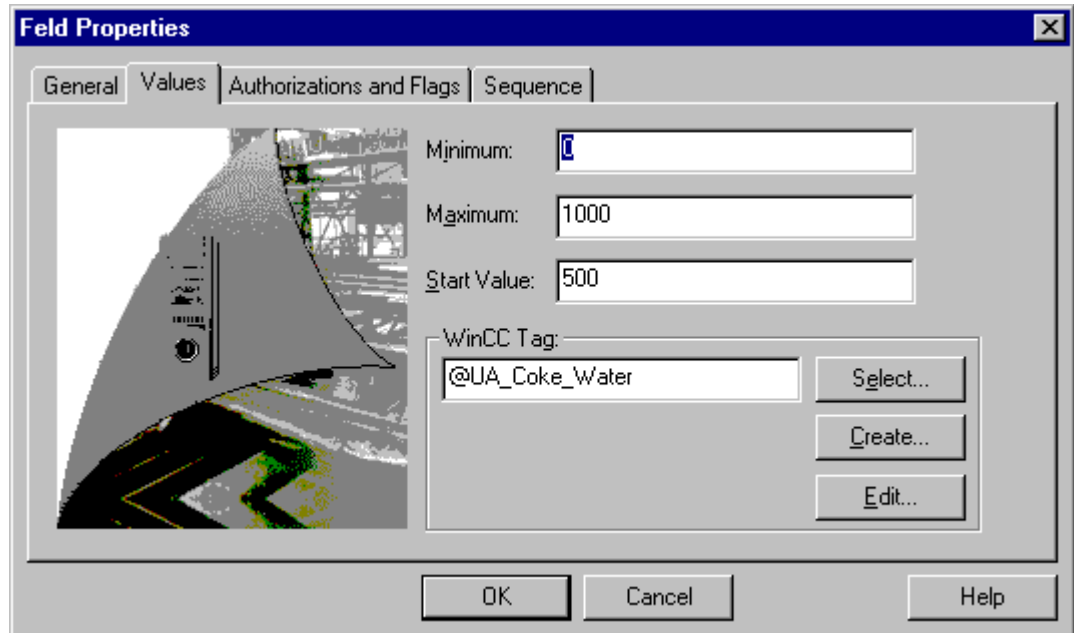
Changing language (Page 351)

Alphabetical list of SQL keywords (Page 362)

Setting of the Values of User Archive Fields

Setting of the values

In the dialog box "Values" you can enter the minimum, maximum and start values.



Minimum, Maximum and Start Value

Here you can set the respective values depending on the selected data type.

WinCC Tag

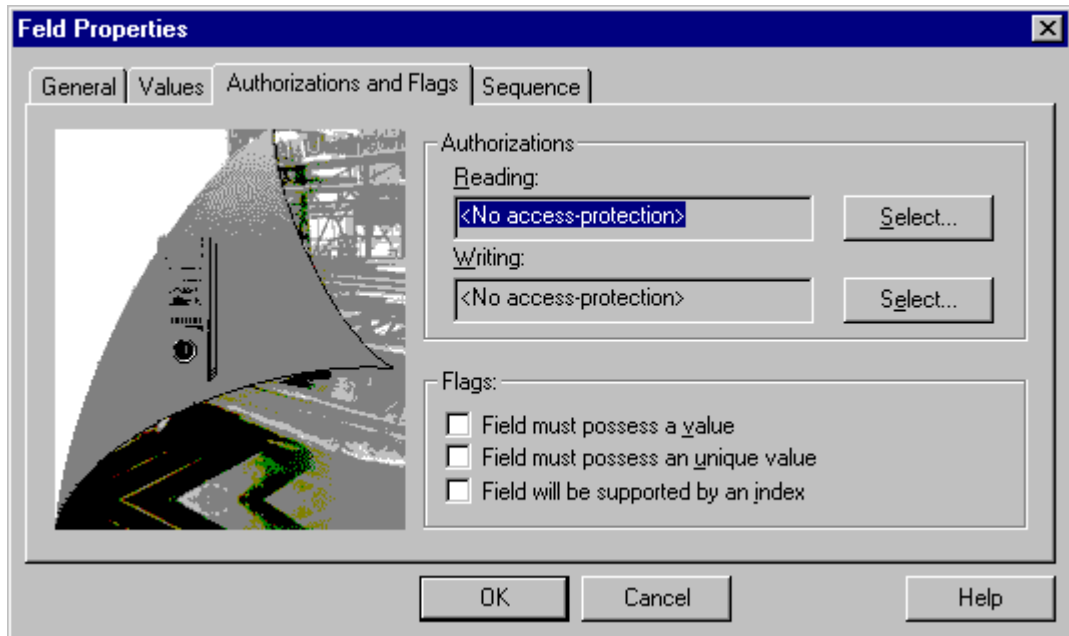
Here you can set a WinCC tag, which will save the value of the user archive field. You can...

1. enter the tag directly into the input field
2. select or create a tag interactively with the "Select" button
3. create automatically a new tag with the "Create" button
4. change the properties of the tags with the "Edit" button

Settings for the "Authorizations and Flags"

Setting Authorizations and Flags

In the dialog "Authorizations and Flags" you can set access authorization and attributes for the user archive fields.



Rights

With the help of the "Select" button you can set here the authorizations for reading and writing access. The possible authorizations are set in the User Administrator. The setting of the access authorization is done as in the section "Create a User Archive".

Flags

In the area "Flags" you can set the following attributes for the selected data field:

1. "Field must possess a value":
The field has a value, which must be different from zero.
2. "Field must possess an unique value":
The field must possess an unique value, that is, the values in this column must be different from each other.
3. "Field should be supported by an index":
The field supports an index value, if this is possible. This index can increase the performance for search commands.
4. Exit the data field input by the button on "Finish".
This will create now a new data field in the User Archive "Cola".
If you have activated the option "Create fields in cycle" in the menu "Edit - Options", the query "Next Archive ?" appears after the fields have been created. If you click on "Yes" the first dialog box "General" appears for the input of the next field.
5. Save the new User Archive

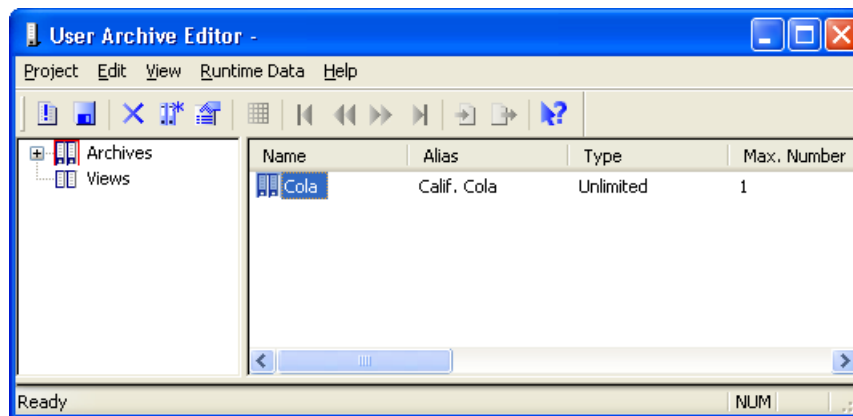
Note

Changes on the User Archives are adopted into the database only by "Save".

Properties of user archives

To edit the properties of a form...

- Click the navigation window with the right mouse button on the user archive "Cola" (first expand the archive).
- Select "Properties" in the pop-up menu.



You get the "Archive Properties" dialog where you can change the properties. The "General", "Communication", "Flags" and "Select Authorization" tabs have been described in the "Create User Archive" chapter. The additional tab "Order" is used for the setting of the order of the User Archives.

User archives: "Sequence" Tab

"Sequence" Tab

The tab "Order" is used for the setting of the order of the User Archives.



Select one or more of the user archives and move its position with the "Up" and "Down" keys. Confirm your entries with "OK." Save the User Archive at the end by a mouse click on the Save symbol or by activating the menu "Project - Save". The order of the user archives is then displayed in the Editor User Archives in the column "Pos."

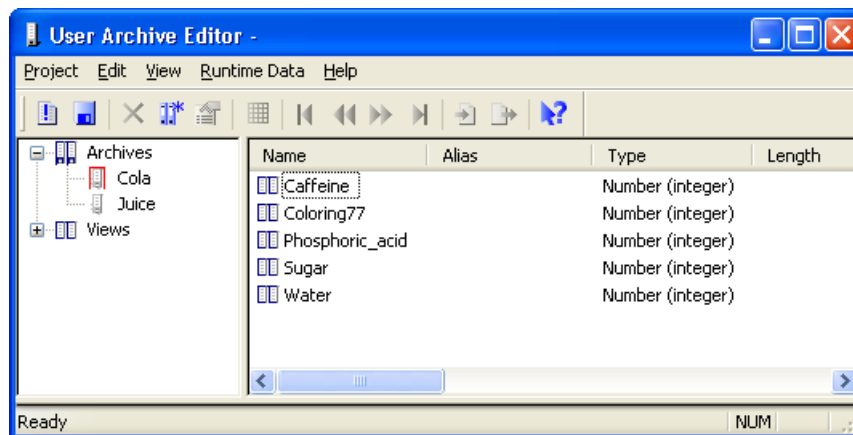
Note

Changes to the User Archives are adopted into the database only by "Save".

Properties of User Archive Fields

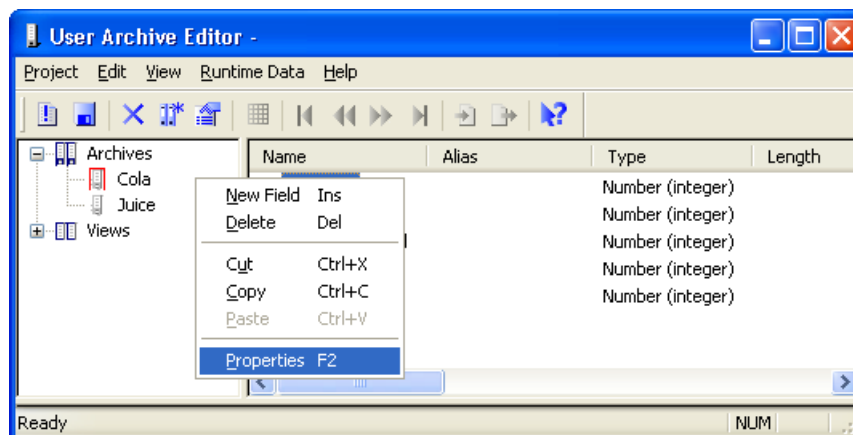
To edit the properties of data fields...

- Click the navigation window of the user archive "Cola" (first expand the archive).
- In the data window of the Editor User Archives, you should now see the data fields of the user archive "Cola":



To edit the data fields of a user archive...

- Click the field name "Water" in the data window of the Editor User Archives
- Click the "Properties" button in the pop-up menu



You get the "Field Properties" dialog where you can change the properties of the data field.

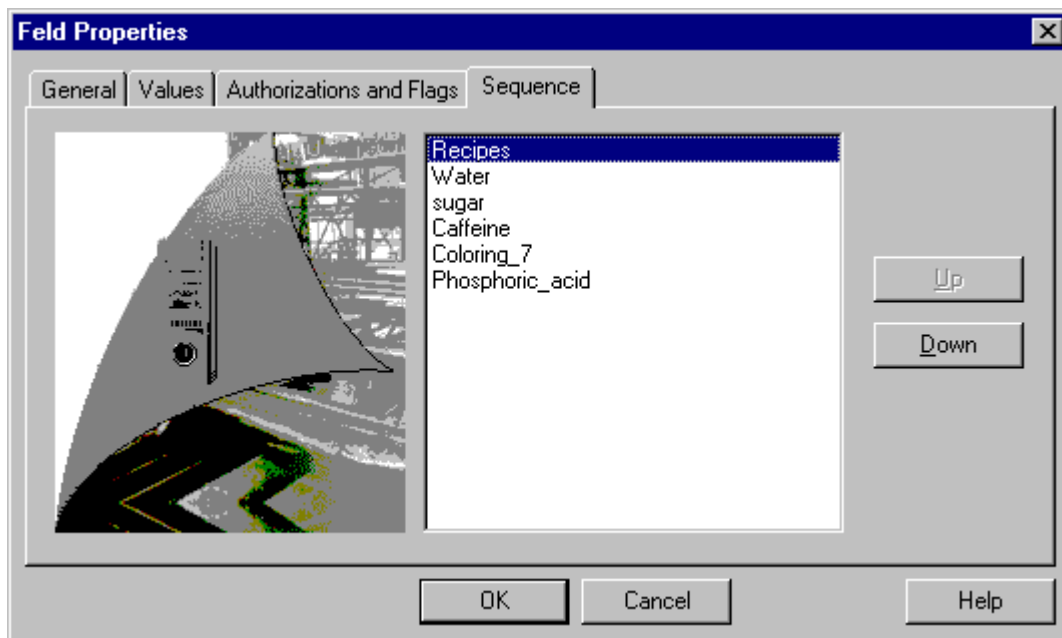
The "General", "Values", "Flags" and "Select Authorization" tabs have been described in the "Define User Archive Fields" chapter. The additional tab "Order" is used for the setting of the order of the User Archive fields.

User archive fields: "Sequence" Tab

"Sequence" Tab

The tab "Order" is used for the setting of the order of the data fields. The order that is set here has an effect for the display of the data in the table window of the Editor User Archives, in the

control of the runtime screen and for the assignment of the indexes for the access through the functions of the WinCC script language.



Select one or more of the fields and move its position with the "Up" and "Down" keys. Confirm your entries with "OK." Save the User Archive at the end by a mouse click on the Save symbol or by activating the menu "Project - Save". The order of the user archive fields is then displayed in the Editor User Archives in the column "Pos.".

In our example the User Archive "Cola" contains the following properties:

User Archives	Data fields	Properties
Cola	Water	Name: Water
		Alias: Wasser_aus_Brunnen_4
		Type: Integer
		Length:
		Precision:
		Min Value: 1000
		Max.Value: 1200
		Start Value: 1100
		Tag n...:
		Right (read): 0
		Right (write): 0
		Flags: NN
		P...: 3
		Last access: 03/05/98 12:54
	Sugar	Name
		Alias

		
	Dye 7	Name
		Alias
		
	Caffeine	Name
		Alias
		
	Phosphoric acid	Name
		Alias
		

Save the User Archive at the end.

Note

Changes on the User Archives are adopted into the database only by "Save".

Properties of Control Tags

"Tag Properties" dialog

You can use the properties dialog of the User Archives and user archive field to edit the properties of the Control Tags. Click the "Edit" button in the respective tab in the toolbar. The dialog "Tag properties" appears, where you can control and change the properties of tags, if necessary.

Data Types of the control tags	
Data type for @UA_Cola_ID	Signed 32-bit value
Data type for @UA_Cola_Job	Signed 32-bit value
Data type for @UA_Cola_Field	Text tag, 8-bit
Data type for @UA_Cola_Value	Text tag, 8-bit

Note

The data type of the Control Tags must not be changed.

With the help of these four Control Tags you can trigger a User Archive. For the triggering you must supply either the tags "ID" and "Job" or the tags "Job", "Field" and "Value" with the respective values.

Function of the control tags	
ID	The ID (corresponds to the record number) of the user archive

Job	Three jobs are possible: Read, write and delete: Read = 6 Write = 7 Deleting = 8 After the job has been carried out, an error ID can be seen in this control tag: No error = 0 Faults = -1
Array	The archive field
Value	The archive field value

More value combinations of the Control Tags "ID" and "Job"

ID	Job = 6	Job = 7	Job = 8
-1	Add data record	-	Delete data record with the lowest ID
-6	Read data record with the lowest ID	Write data record with the lowest ID	Delete data record with the lowest ID
-9	Read data record with the highest ID	Write data record with the highest ID	Delete data record with the highest ID

The Control Tags provide two methods to access the user archives:

1. By entering the control tags "ID" and "Job" you can write or read or delete aimed values in a data record.
2. Instead of the Control Tag "ID" you can use the Control Tags "Field" and "Value" to search for a data record. With the Control Tags "Job" you can write or read or delete the data record which you have selected by this method. This type of the data selection can be used if, for instance, data records must be deleted from the table and appended at the end of the table again. The field "Value" must be unique, otherwise the first data record that is used for which the condition value in the field applies.

Note

In order to check the function, always all four Control Tags of a User Archive must be given or created with the Wizard.

When a User Archive is created, we recommend you to use the wizards for the creation and saving of the Control Tags.

For more information on how the Control Tags can be supplied please refer to the example about the use of the control tags.

See also

Example for the use of control tags (Page 334)

Example for the use of control tags

Example for the use of control tags:

In order to be able to work in the example with the Control Tags you must carry out the following steps:

In the "User Archive" editor

1. Create a User Archive (in our example the User Archive "Cola"). Please enter the settings that were executed below during the creation of the User Archive with the help of the wizard. If the project "Cola" was already created, you can check and, if necessary, change the settings through the properties of the User Archive.

Properties of the User Archive "Cola"	
User archive type	"Unlimited"
Communication	via WinCC tags
Control tags	Creating
Tag group	"@UA_Cola"
Data type for @UA_Cola_ID	Signed 32-bit value
Data type for @UA_Cola_Job	Signed 32-bit value
Data type for @UA_Cola_Field	Text tag, 8-bits
Data type for @UA_Cola_Value	Text tag, 8-bits

1. Create in the User Archive the data fields "Water", "Sugar", "Dye stuff 7", "Caffeine" and "Phosphoric acid" (Type Integer).
2. Create a data field "Recipes" (Type String).

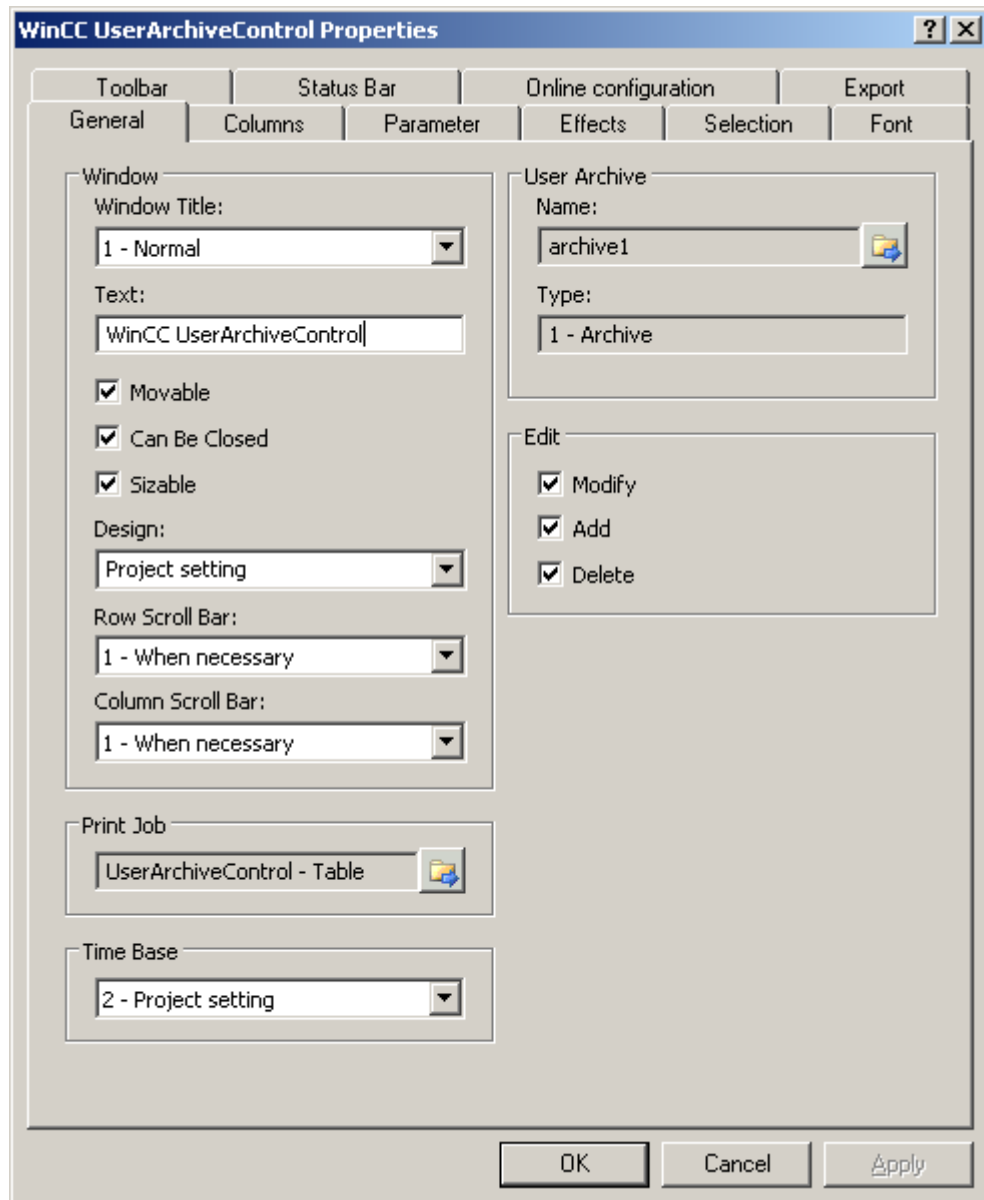
In Graphics Designer

1. Open a new picture and create a WinCC UserArchiveControl.

Note

Previous to WinCC V7, the display of the user archives was configured in a User Archives Table Element.

2. With a double click of the left mouse button you can open the dialog "Properties of WinCC UserArchiveControl".



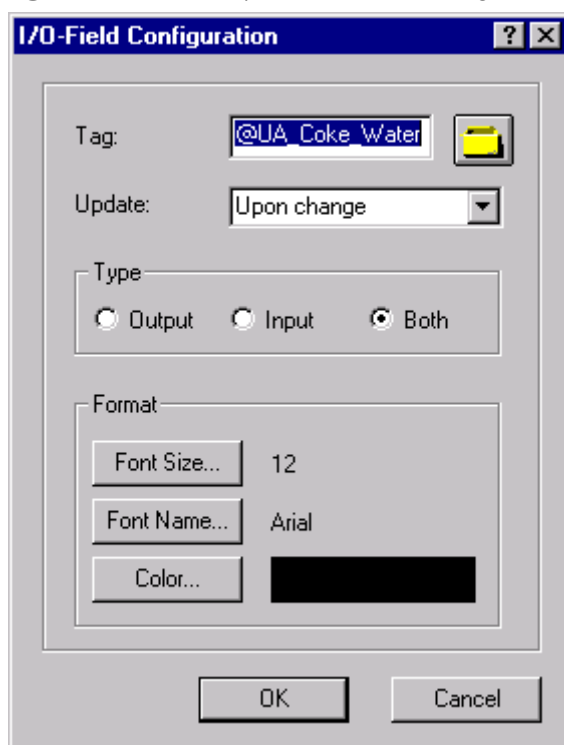
3. Change the following settings:

- Select the user archive "Cola" from the "General" tab in the "User Archive" area.
- Activate access types "Change", "Add" and "Delete" in the "Edit" field.
- The other settings can be adopted without any changes. If you have already created a UserArchiveControl, you can check and in certain cases change the settings with the properties dialog.

4. Create one I/O Field for each of the four Control Tags and select the following settings:

Control Tags	Data Format	Output format
@UA_Cola_ID	Decimal	0999
@UA_Cola_Job	Decimal	s9
@UA_Cola_Field	String	*
@UA_Cola_Value	String	*

1. Select for each tag the object property "Update for changes".
2. Create an I/O Field for each configured data field (Water, sugar etc.) and connect them with the related tags (e.g. the I/O Field for "Water" with the process tag "@UA_Cola_Water"). Select for each tag the object property "Update for changes".

**Note**

You will find additional information about configuring I/O fields in the documentation of the Graphics Designer.

Create a text field for each configured I/O Field for the labeling, so that you can assign the individual fields in runtime. Save the made entries and activate WinCC Runtime. Now enter five Data records in the table window. Enter in the data record with the ID 2 in the column recipe "Cola" and in the fourth data record "Cola Light". The following figure shows the example with the WinCC User Archive Table Element (previous to WinCC V7):

The interface is divided into two main sections: **process tags** and **control tags**.

process tags section:

- Five input fields with labels: **water**, **sugar**, **caffeine**, **coloring 7**, and **acid**.
- Five corresponding value fields containing: **100**, **50**, **20**, **20**, and **20**.

control tags section:

- Four input fields with labels: **ID**, **JOB**, **FIELD**, and **VALUE**.
- Four corresponding value fields containing: **0003**, **+0**, and two empty fields.

Table Window:

ID	Recipes	Water	sugar	Caffeine	Color_7	Phosphoric_aci
1	Calif Coke	90	10	10	10	15
2	Coke	80	30	15	20	20
3	Standard	100	50	20	20	20
4	CokeLight	100	20	30	20	20
5	CherryCoke	100	80	50	5	10

Below the table, status indicators show: **Finished**, **Rec 1/5**, **Row 1**, and **Col 1**.

Procedure for the execution of the possible individual actions.

1. Select a data record with the ID and write the values of the data record:
 - Enter the ID "3" into the I/O Field "ID" and write into the I/O Field "Job" a 7 (write).
 - Now the values of the data record "3" are displayed in the I/O Fields of the process tags.

- If the action was successful, the error number "0" is displayed in the I/O Field "Job". In the case of an error, the error number "-1" is displayed.
- The Control Tags "Field" and "Value" are not required.

Note

By entering the ID "-1" and the job "6" the current contents of the process tags are read into the table. The new values are attached at the end of the table; the IDs of the records are incremented. You will find other combinations of values of the control tags "ID" and "Job" in the "Properties of Control Tags" chapter.

1. Select a data record with the ID and read the values of the data record:
 - Change the values in the I/O Fields of the process tag and enter a "5" in the field "ID". In the I/O Field "Job" enter a "6" (read).
 - The changed values of the process tag are written into the data record "5". The values that were earlier contained in this data record are overwritten.
 - The Control Tags "Field" and "Value" are not required.
1. Select a data record with the Control Tags "Field" and "Value":
 - Enter in the I/O Field "Field" the word "Recipe" and in the I/O Field "Value" write "Cola Light". In the I/O Field "Job" enter a "7" (write).
 - Now the data record "Cola Light" is written, the values of the data record are displayed in the I/O Fields of the process tags.
 - The Control Tag "ID" is not needed and must therefore be set to 0.

Note

The field which is referred to with the Control Tags "Value" must be assigned to the flag "Field must possess an unique value" in the dialog "Authorizations and Flags". A unique assignment of the data record to the value in the field is not possible otherwise.

See also

Create an IO field in the Graphics Designer

2.2.3.3 Modification Configuration of User Archives

If you want to change or expand an existing User Archive, existing data in the database table could be lost. Especially if the structure of a database table is changed or the properties of a field are changed, the consistency conditions of the database cannot be fulfilled anymore.

In order to avoid the loss of data, we recommend the following procedure:

1. Disable the runtime, open the User Archive in the User Archives Editor and carry out the desired changes. Save the archive only after all the changes have been done completely (no cache saving).
2. After saving press on the button "Edit Runtime Data". On the opened table you can see whether the existing data in the User Archive are still available.

3. If the data are still present, you can use the archive or carry out further changes. In the case of changes, check after each save process whether the data are still available.
4. If after a change the data are not present anymore, first undo all the done changes in the archive without saving. Now select the archive in the User Archives Editor and select in the pop-up menu the command "Restore RT data". This will write the previous runtime data back into the table. Then save the archive and check whether the data has been written back into the archive.

<u>N</u> ew Field	I <u>n</u> s
<u>D</u> elete	D <u>e</u> l
<u>R</u> estore RT Data	C <u>t</u> rl + <u>Y</u>
<u>P</u> roperties	F <u>2</u>

Note

Check after each save process whether the runtime data are still available. If you have saved a first change step and already lost the data in this process and then save a second change on it, the data are lost.

If the runtime was started after a change and then the data loss detected, the data can be recovered nevertheless as described above by the command "Restore Rt data" into the table. The data can be restored even if the Editor User Archives is closed or WinCC quit. The important point in this relationship is only the single saving of a change.

In the case of comprehensive changes on a User Archive we recommend to export the existing data first to forego a data loss. The data can be written back into the changed archive by customizing the exported data table.

If you change the configuration of user archives on the server, the client without its own project requires an update of the server package in order to correctly display the data in Runtime. You may generate the update manually on the server, or by using the implicit update package. For the implicit update package, configure the setting "Generate WinCC CS Server data immediately when changes are made" in the "Server Data" editor of the server, using the "Implicit Update" pop-up menu.

See also

Menu command "Export" (Menu "Runtime Data") (Page 313)

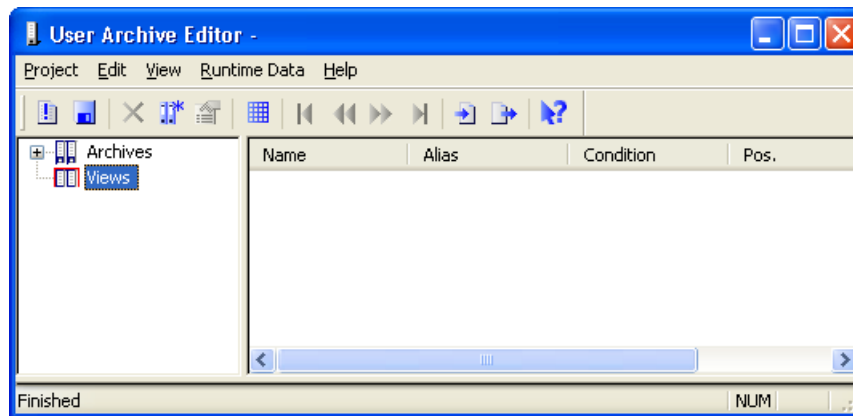
2.2.3.4 Configuration of views

Create new views

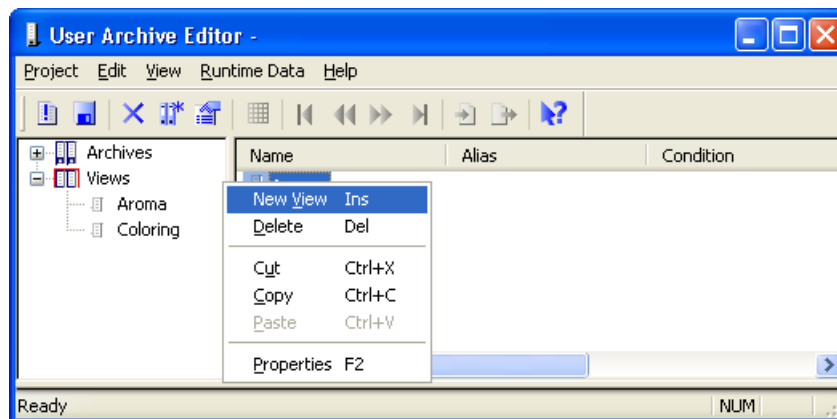
The default option "Put views in cycle" allows the creation of multiple views one after another. If you want to create only one view, you can disable this option in the menu "Edit -Options".

In order to create a new view you can follow the following steps:

1. Click on the navigation window with the left mouse button on "views".

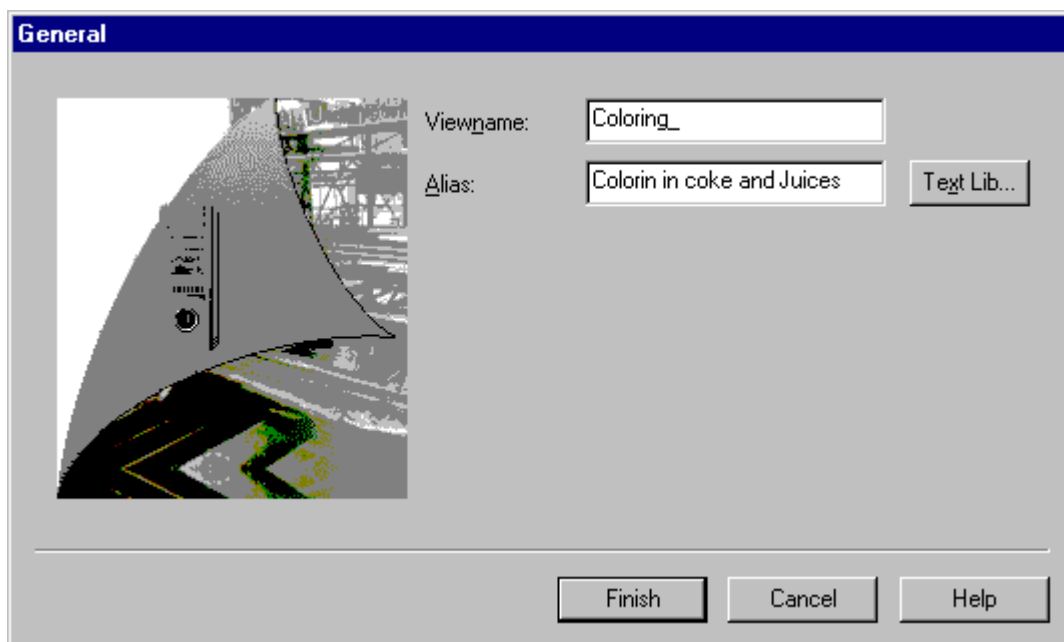


1. Then click in the navigation or data window with the right mouse button. The pop-up menu that is displayed in the picture below appears.



1. Select the option "New View"

Now the wizard is started for the configuration of the views . You can see the dialog "General", which you can use to create a new form.



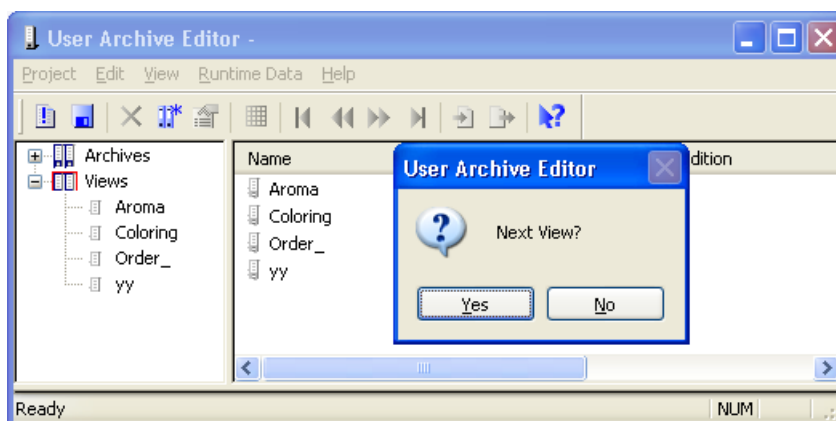
Enter as the form name for example "Dyes". In the field "Alias" you can give the form a second name, for instance "Dye stuffs in Cola and Orange Juice", in order to comment the user archive or to realise a language change through the text library in runtime. This input is optional.

For more information on switching the language, refer to the Online Help.

Exit the creation of the form by the button "Finish".

After the Form is created, the query "Add columns?" appears. If you confirm with "Yes", the dialog "General" is opened for the creation of a form. For information on how to create columns in a form, refer to the chapter "Create columns of a form".

If you have activated the option "Create forms in cycle" in the menu "Edit - Options", the "Next form?" query appears after the columns of a form have been created. If you click on "Yes" the initial dialog box "General" appears for the input of the next form.



Save the forms after finishing.

Note

Changes on the forms are adopted into the database only by "Save".

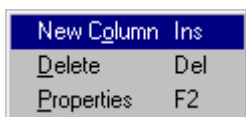
See also

Changing language (Page 351)

Create New Columns of a view

The default option "Put columns of a view in cycle" allows the creation of multiple columns one after another. If you want to create only one column of a view, you can disable this option in the menu "Edit -Options".

1. Expand "Views" in the navigation window. The new view, for instance, "Dyes" is then displayed in the Navigation Window.
2. Click on the navigation window with the left mouse button on the view "Dyes". The following pop-up menu is displayed:



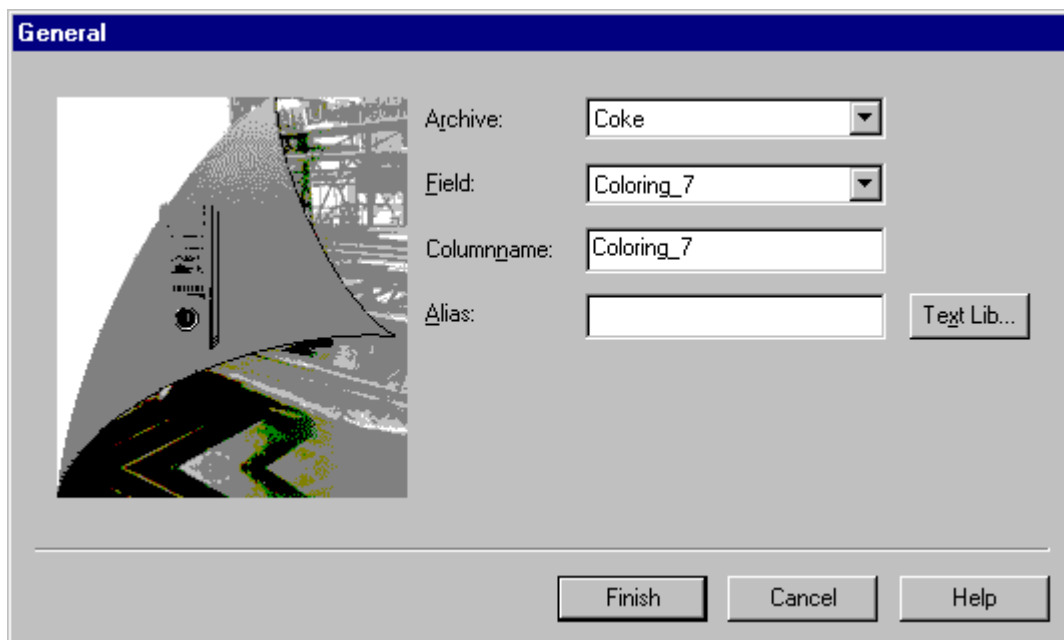
1. Click on the "New Column" button.

The dialog box "General" will appear.

General Properties of Columns of a view

General Properties of Columns of a view

In the dialog box "General" you can select fields from the User Archive, declare these as columns of your created form and give them their own names.

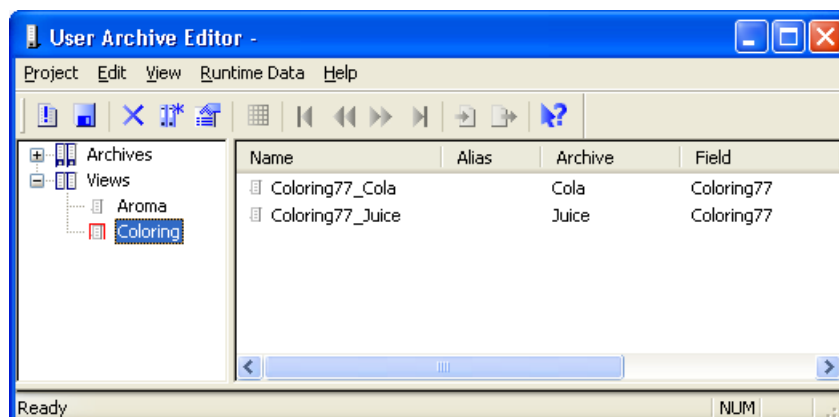


- In the selection dialog "Archive" you can select one of the set User Archives. For instance, leave the setting "Cola" without any changes.
- In the selection dialog "Field" select one of the fields from the User Archive Cola. For instance, leave the setting "Dye 7" without any changes. Toggle into the next field with the TAB key or with a mouse click for further entries.
- If you click on the field "Column Name", the entry on the field "Field" is accepted. You can also select the Column Name freely, however, this name must be unique inside the view. For instance, accept the name "Dyes".

In the field Alias you can give the column a second name in order to comment the column or to realise a language change through the text library in runtime. This input is optional.

For more information on switching the language, refer to the Online Help.

After you have clicked on "Finish", the configured data field is set:



The drink producer in our example for example creates a form "Dyes" by combining the data fields "Dye 7" and "Dye 16" from the User Archive "Cola" and "Juice":

If you have activated the option "Create columns of a view in cycle" in the menu "Edit - Options", the "Next column?" query appears. If you click on "Yes" the initial dialog box "General" appears for the input of the next column.

Save the field of the view at the end.

Note

Changes on the forms are adopted into the database only by "Save".

See also

Changing language (Page 351)

Properties of Views

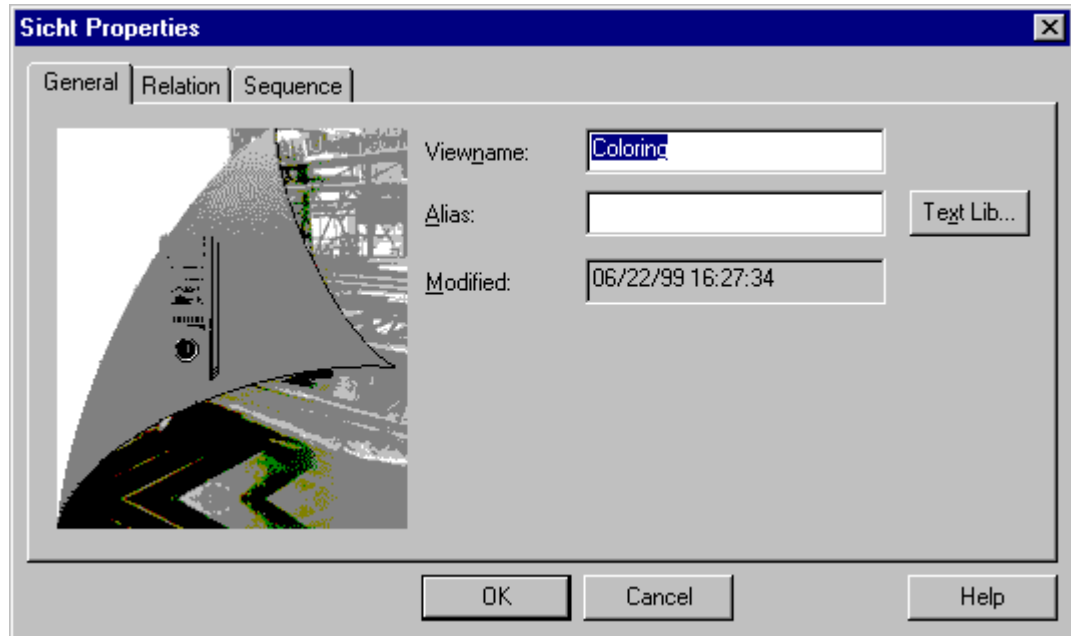
To edit the properties of a view

1. 1. Click with the right mouse button on one of the views in the Editor User Archives
 2. 2. Select "Properties" in the pop-up menu.
- then the dialog "Properties of the view" will appear.

General Properties of views

General Properties of views

In the tab "General" the properties of the selected view will be displayed.



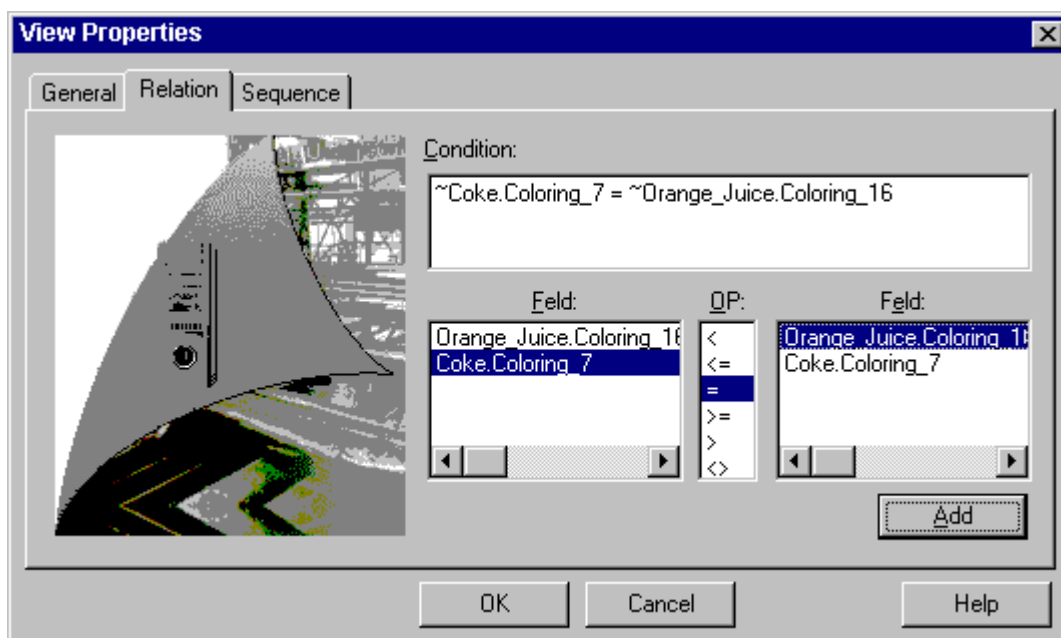
In the field "View name" you can change the name of the view and in the field "Alias" the alias name. In this dialog you can also see the date and time of the last change.

Settings of the Relations of Views

Settings of the Relations of Views

In the tab "Relation" you can create a relation between multiple User Archives for the output of a view. You can either formulate the logical operations directly in the SQL language or set

them interactively with the given relation operators. Make sure that the user archive fields for which you want to set a relationship are of the same tag type.



Relation

In the field "Relation" you can enter the direct SQL Statements. IN the appendix you can find more on the .

Conditions

In the selection fields you can enter the conditions interactively. In order to do so, click on the fields in the left and the right "Field" list and set the relation by clicking on the respective operation in the list "OP". After clicking on "Add" the condition will be accepted and this will then appear in the field "Relation".

Principle

All fields that show a relationship are connected to each other in the selected User Archive. By using the set relation, the field contents are filtered and the result is displayed as a form in runtime. The data of a view can also be edited in runtime, the modified data are inserted into the original archive.

Note

The connected user archives must show at least one common factor or a relationship.

See also

Changing language (Page 351)

Setting of the Order of Views

Setting of the Order of Views

In the "Sequence" tab, set the order of the forms.

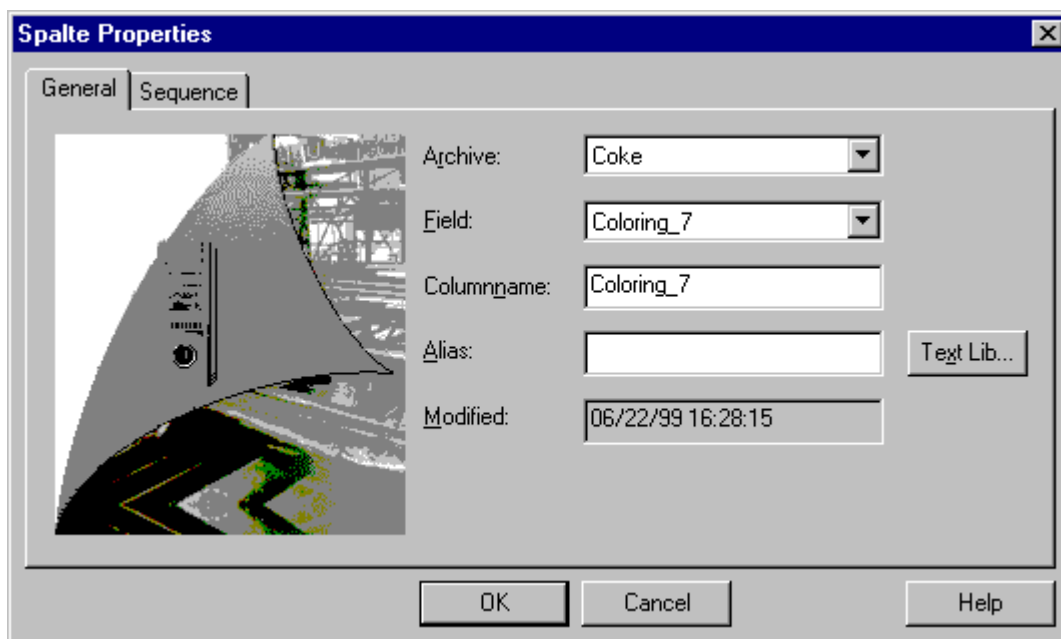


Select one or more of the views and move its position with the "Up" and "Down" keys. Confirm your entries with "OK." Save the views at the end by a mouse click on the Save symbol or by activating the menu "Project - Save". The order of the views is then displayed in the Editor User Archives in the column "Pos.".

Properties of Columns of a View

To edit the properties of the columns of a view, follow the following steps:

1. Click with the right mouse button on one of the columns of a view in the Editor User Archives
2. Select "Properties" in the pop-up menu. Then the dialog box "Properties of column" appears:

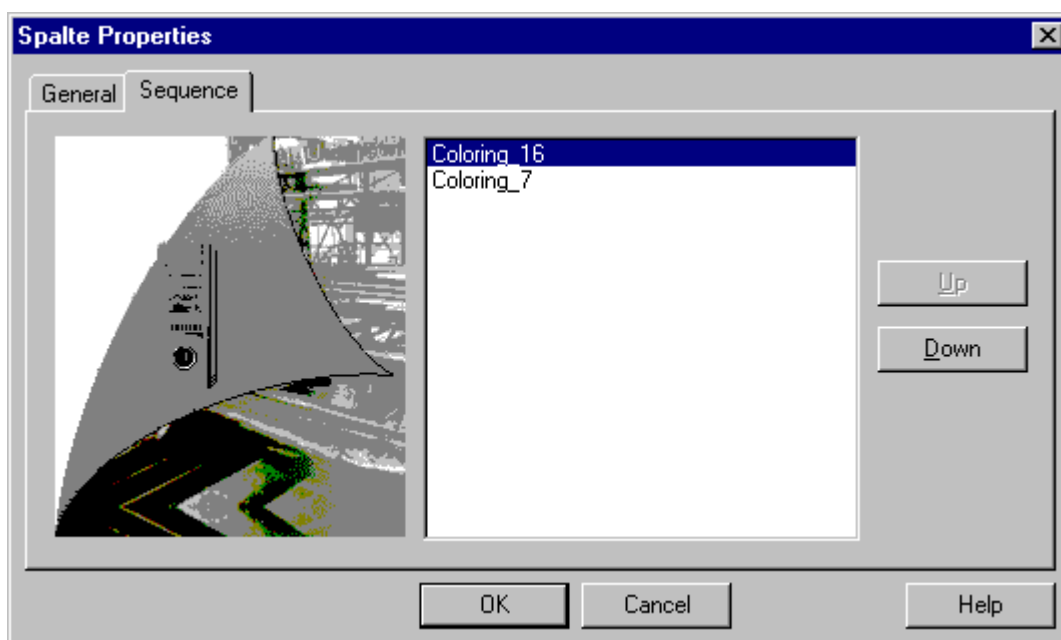


The "General" tab contains the same fields as in the setup of a new view column. The date and time of the last change are shown in the field "Changed".

Setting of the Order of Columns of a View

Setting of the Order of Columns of a View

In the "Sequence" tab, you can set the order of the columns of a view.



Select one or more of the columns and move its position with the "Up" and "Down" keys. Confirm your entries with "OK." Save the views at the end by a mouse click on the Save symbol or by activating the menu "Project - Save". The order set here has an effect during the display of the columns in the table window of the Editor User Archives and in the control of the Runtime screen

Tips for configuration

- The communication between the AS and the User Archive is limited to one connection per User Archive.
- While the communication with the automation system is established, the PLCID cannot contain more than 8 characters.
- Terms that contain special characters or reserved words are not acceptable as field and table names. For more details refer also to the chapter "Alphabetical List of SQL Keywords".
- The saving of configuration changes in runtime is possible only if there is no redundancy alignment at any of the user archives at the time.
- If WinCC Redundancy is used, the same structure must be used at both the servers with the User Archives that will be aligned. Therefore the configuration of these user archives must be identical in terms of their properties and field/record structure.
- For the communication of the User Archives through raw data tags the name of the used raw data tags on both servers must be identical as well.
- If after the start of the "User Archive" editor all fields of the toolbar are grayed out (except restore) the file "UAEditor.loc" in the project path must be deleted. This also applies for a UserArchiveControl that is no longer operable.
- If you get the error message "Error while connecting the data!" while starting runtime or while switching the UserArchiveControl to the runtime view, it means that the UserArchiveControl is not connected to a user archive or a view. Please check whether the connection was entered correctly, whether the configuration was changed or whether the selected User Archive or the view still exist.

Note

During configuration, a UserArchiveControl is connected to a selected user archive or form and can then only access that user archive or form. For access, the user archive / form must be enabled (access protection). Specific authorizations can be assigned to the control in the User Administrator.

If this access protection is canceled, the UserArchiveControl must be reconnected in the Graphics Designer to the user archive so that the UserArchiveControl detects the canceled access protection.

Access protection for an archive or field is queried on opening a screen of a UserArchiveControl. Access protection for the control tags of a protected archive must be implemented separately via the object properties, e.g. of the picture, I/O field or button.

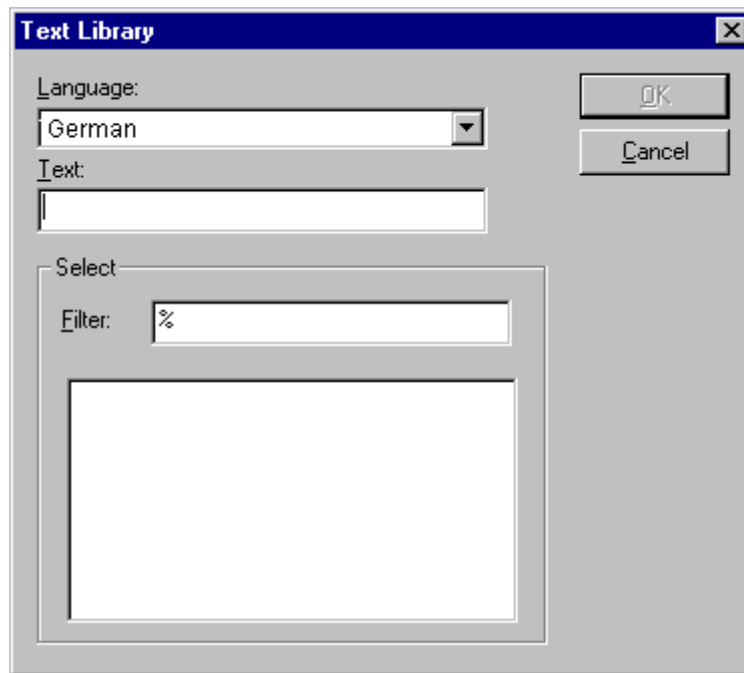
See also

Redundant User Archives

Alphabetical list of SQL keywords (Page 362)

Changing language

For User Archives, user archive fields, views or view fields you can realise a language change by the text library. Click the button "TextLib..." in the respective dialog box for this. The dialog to the text library is opened



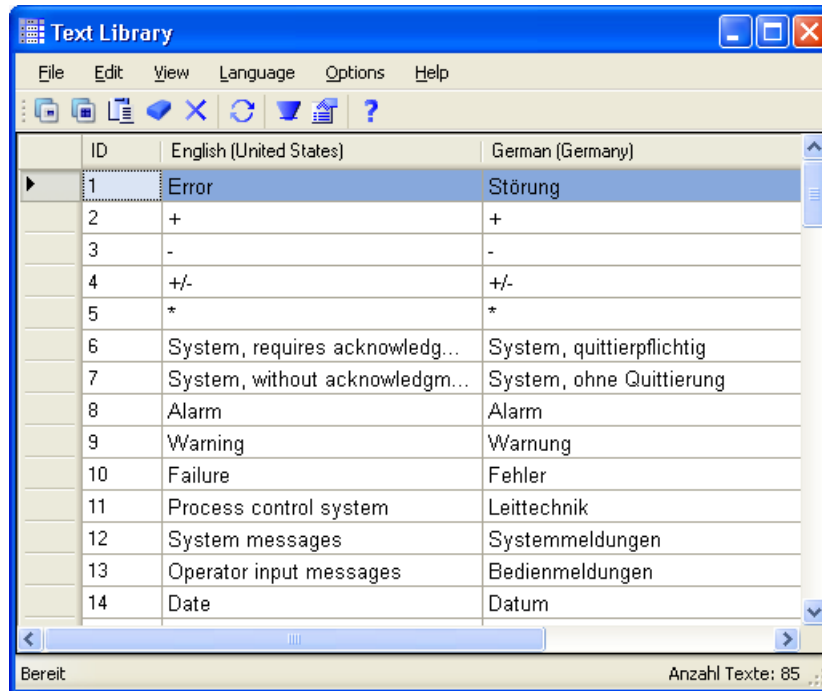
Procedure for texts from the text library

1. First set in the selection dialog "Language" the language in which you are configuring.
2. If the planned languages have already been created in the Text Library, you can load all existing texts by a mouse click into the selection field. These are displayed in the selection window and you can select the desired term.
3. After the selection the selected term will appear in the field "Text".
4. Acknowledge the dialog with "OK".
5. In the field "Alias" now the position number of this word is given from the Text Library.

If the language is changed in the runtime the term for the selected language that was selected in the Text Library will appear.

Procedure for new texts

1. First set in the selection dialog "Language" the language in which you are configuring.
2. Enter the text or term for which you want to change the language in the field "Text".
3. Acknowledge the dialog with "OK".
4. In the field "Alias" now the position number of this word is given to the Text Library.
5. Open the editor for "Text library". Here you can now enter the translation of the text that was entered into the Editor User Archives into the columns of the desired language.



6. Close the Text Library after the translation is finished.

If the language is changed in the runtime the term for the selected language that was selected in the Text Library will appear.

Note

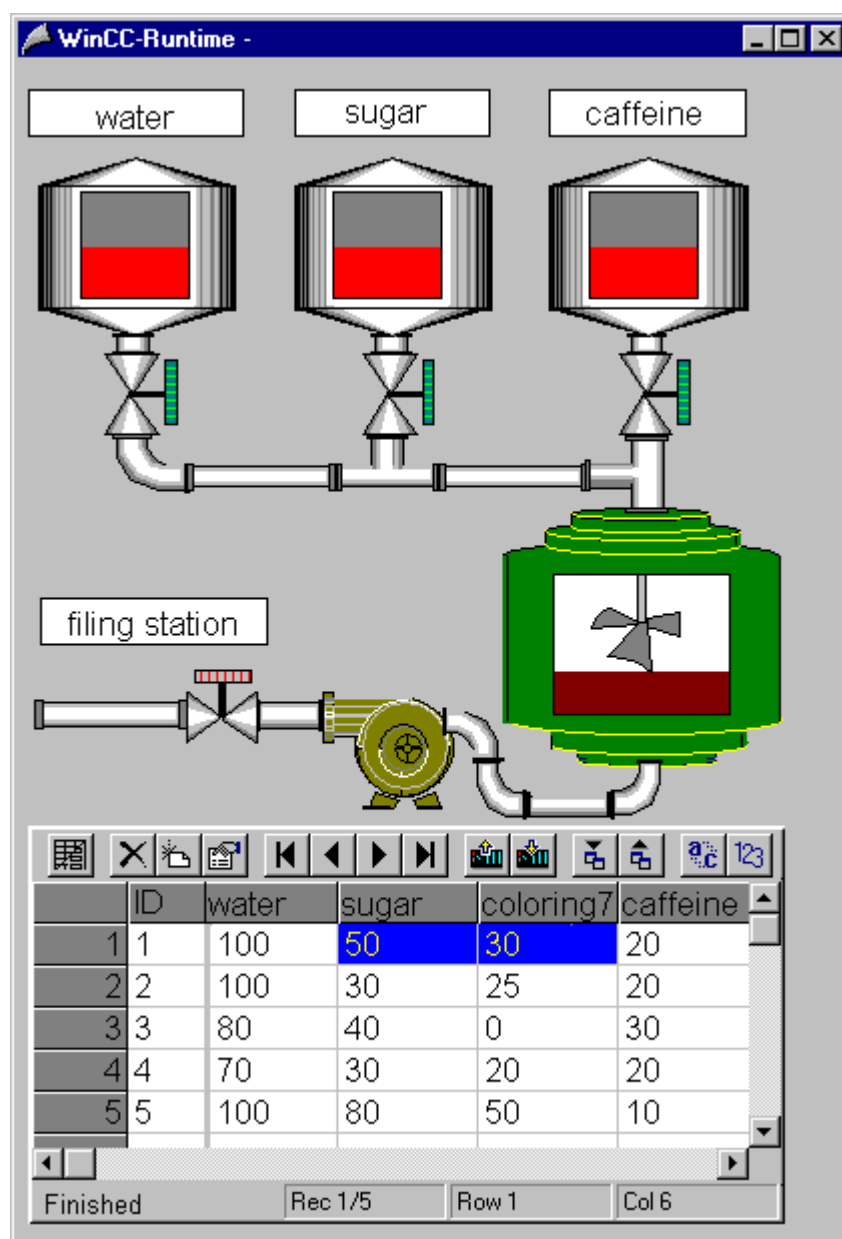
In the case of client projects, the same text IDs must be used in the text library of the server and client for the texts of the user archives, otherwise the text is displayed wrong in the runtime on the client.

In the field "Filter" you can specify filter properties for the texts from the text library, e.g. "a%" lists all terms from the text library that start with the letter a. If a new filter criterion is selected, you must click on the selection window again to update the text selection.

2.2.3.5 Example of a User Archive

User archive for beverage manufacturing

The drink producer "Sun Drink", produces in our example the Cola drink "Calif Cola" and the orange juice "Sunny Juice". In order to save the recipes for the contents of his drinks he uses the User Archive of WinCC. If a storage container of the filling system is empty, the recipe data are sent per communication channels from WinCC to the AS. The AS can then fill the storage container with the help of the recipe data again. The following figure shows the example with the WinCC User Archive Table Element (previous to WinCC V7):



The User Archives are used as follows:

- **User archives:** Here there is one user archive for the Cola drink and one user archive for the orange juice.
- **Views:** Use the views to combine data fields of two archives; in this example: the dyestuff product group.

Each user archive consists of data fields with editable properties. In the Cola drink in our example the contents are in the data fields. Each data field has properties such as name, alias name, type, lengths, value etc. The representation of the data fields and properties in the Editor User Archives is done in lines and columns. Therefore, we are talking of rows instead of data fields and of columns instead of properties. The structure of the user archive "Cola" looks for instance as follows:

User Archive Cola	Properties (Columns)						
Data fields (Lines)	Name	Alias	Type	Length	Min. value	Max. value	Start value
Water	Water	Brun.5	Int	2	1000	1500	1000
Sugar	Sugar	Zmela	Int	2	120	140	130
Dye stuff 7	Dyestuff7	FS1007	Int	2	6	8	6
Caffeine	Caffeine	Caffeine	Int	2	2	3	2
Phosphoric acid	Phosphoric acid	PhosAc	Int	2	170	190	170

2.2.4 Data exchange using SIMATIC S5 / S7

2.2.4.1 SIMATIC message frame interface

Data exchange between the User Archives and S5 and S7 PLCs can be carried out through raw data tags or through WinCC tags. You can use all SIMATIC interfaces, except the AS511 programming interface.

Data exchange with WinCC using raw data tags can be done with the following commands:

- S7-400
- S5-PLC-115U or later

The following topic will be explained:

- Data Exchange with S5 and S7 via WinCC Tags
- Data exchange with S5 and S7 via raw data tags
- Data format differences between WinCC and S5/S7

2.2.4.2 Data Exchange with S5 and S7 via WinCC Tags

The data exchange with S5 and S7 via WinCC Tags is very simple. However you must take care that for the User Archives Data Types only specific Data Types of the Tag Management can be used.

When using the data types Integer, Double and String in Editor User Archives, the following data types must be used in the tag management of the data manager. For the User Archives data type Date/Time there is no suitable data type in the tag management.

Selection in the Editor User Archives	Tag Management / WinCC Tag
Number (integer)	Signed 32-Bit Value
number (float)	Floating-point number 32-bit IEEE 754
number (double)	Floating-point number 64-bit IEEE 754
Character String (String)	Text tag, 8-bit font
Date/time	No suitable data type

2.2.4.3 Data exchange with S5 and S7 via raw data tags

Data exchange with S5 and S7 via raw data tags

Below we will describe the data exchange between the User Archive and the automation system via WinCC raw data tags. In order to do so, the function BSEND/BRCV is used in AS. The raw data tags are sent from the AS actively. The message frames contain one or several requests to the User Archive of WinCC. These requests can be write as well as read requests. As an answer to these request, WinCC sends the requested data and an processing acknowledgement back.

Note

Since the AS is the active partner for a data exchange, a function of the user archive that is desired by the WinCC user must be started directly in the AS, e.g. write/read from archive values. This triggering can be done by e.g. the value of external WinCC tags in the AS being used for triggering a corresponding function of the User Archive.

The parameter "Job type" that was used during the data exchange in the job or acknowledgement header cannot be used for the triggering of functions of the AS, since it has only functionality inrelationship with the User Archives.

You will find information about the following topics:

- Send jobs / data to WinCC
- Send processing acknowledgement / data to SIMATIC S5 and S7
- Structure of Kommunikation Header

Send jobs / data to WinCC**Raw data**

Structure of the raw data tag to send jobs and data from the SIMATIC S5 and S7 PLCs to WinCC:

Telegram to S5 / S7
Telegram HeaderTelegram Header
Job Header 1Job Header1
Data of Job1Data of Job 1
Possible job header 2
Possible data of the job 2
Job n

Send processing acknowledgement / data to SIMATIC S5 / S7**Raw Data Tag**

Structure of the raw data tag to send processing acknowledgements and data from WinCC to the SIMATIC S5 and S7 PLCs:

Raw data tag to send to S5 and S7
Processing acknowledgement
Acknowledgement header
Acknowledgement data

Structure of Kommunikation Header**Structure of Telegram Blocks**

Structure of the separate message frame blocks (distribution in bytes):

Function of the field	Comment
message frame length in byte LSB *)	Length of the field 4 bytes
.	max. length 4091 byte
.	.(bec. S5/ S7 transport)
message frame length in bytes MSB **)	.
Transfer type	1 from WinCC, 2 from the PLC
Reserved	
Number of the jobs in the message frame LSB *)	Length of the field 2 Byte
Number of the jobs in the message frame MSB**)	.
PLCID 1.character	Specifying the name

.	is done in ASCII
.	Length of the field 8
.	Byte
.	.
.	.
.	.
PLCID 8.character	.

*) LSB = Least Significant Byte (lowest value byte)

**) MSB = Most Significant Byte (highest value byte)

Job Header

Structure of the Job Header

Structure of the separate job header (distribution in bytes):

Function of the field	Comment
Job length in bytes LSB	Length of the field 2 Byte
Job length in bytes MSB	.
Job TypeJob Type	See description
Reserved	
Field number LSB	Length of the field 2 Byte
Field number MSB	.
Record number LSB	Length of the field 4 Byte
.	.
.	.
Data record number MSB	.
Selection Criterion LSB	Field number, according to which the selection will be made
Selection criterion MSB	(not for 0) Length of the field 2 bytes

Data of the job

Data of the job

The data of the job corresponds to the contents of a data record (or the addressed field).

Note

1. Text boxes are not \0-terminated !!!
2. Numbers must be transferred in Intel Format (LSB first, MSB last).
3. An Integer field has a length of 4 Byte, a float field 4 Byte, a double field 8 Byte.
4. The data move at the length of the field, which has been selected as the selection criterion, when the selection criterion has a value not equal to 0.
5. If the selection criterion will be used, the start of the data area is used as a selection value in the field size of the selection criterion.

The acknowledgement header**Acknowledgement header**

Structure of the separate acknowledgement header (distribution in bytes):

Function of the field	Comment
Message frame length in byte LSB	Length of the field 4 Byte
.	.
.	.
message frame length in bytes	.
Transfer type	1 from WinCC, 2 from the PLC
Reserved	
Job TypeJob Type	See description
ErrorCodeErrorCode	See description
Reserved	
Reserved	
Field number LSB	Length of the field 2 Byte
Field number MSB	.
Record number LSB	Length of the field 4 Byte
.	.
.	.
Data record number MSB	.
PLCID 1.character	Specifying the name is done
.	using ASCII
.	Length of the field 8 Byte
.	.
.	.
.	.
.	.
PLCID 8.character	.

Data of the Acknowledgement

The acknowledgement contains either the data record or the addressed field (during a read request) or it is empty (write job, archive job).

Description of the job types and the error codes

Description of the job types

Type	Description
4	Check if user archive exists
5	Delete all of the records in the User Archive
6	Read data set
7	Write data record
8	Delete record
9	Read data record field
10	Write data record field

Description of the error codes

Group	No.	Description	Possible fault causes
General	0	The function has been executed	--
archive	2	Data not available	- no archive configured with this PLCID
Data record	101	Data not allowed	- Record layout does not match, e.g. number or type of fields - adding or updating records fails because e.g. archive of the type "limited" is configured or a minimum or maximum value is configured for a field - filter criterion incorrect
Data record	102	Data not available	(only for job type 6) - no data available - filter criterion incorrect
Array	201	Data not allowed	(only for job type 10) - filter criterion incorrect because e.g. field not available or a minimum or maximum value is configured for a field
Array	202	Data not available	(only for job type 9) - filter criterion incorrect or no field found because which meets the filter criterion
General	254	Function not available	--
General	255	Undefined error	--

2.2.4.4 Data format differences between WinCC and S5/S7

The data formats in WinCC are different principally from the data formats in the SIMATIC-S5/S7 PLCs. This must be considered in order to avoid unwanted errors.

In WinCC the data formats of Intel and Microsoft are kept, where principally least significant bytes are stored first and most significant bytes last. This data format is very common and is known in general as the "Intel Format". An example should clarify the "Intel Format":

Intel Format

In the Intel format the decimal number 300 is stored as follows:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0

Hex	0	1	2	C

The decimal number 300 corresponds according to the Intel Format to the hex-number 12C ($1 \cdot 256 + 2 \cdot 16 + 12$).

SIMATIC Format

In the SIMATIC Format the least significant bytes are stored on most significant places. In the SIMATIC format the decimal number 300 is stored as follows:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1

Hex	2	C	0	1

The decimal number 300 corresponds according to the SIMATIC-Format to the hex-number 2C01. If 2C01 would be erroneously interpreted according to the Intel-Format, you would get 11265 decimal, that is, a significant deviation.

For the SIMATIC-PLCs there are function block available, which can carry out corresponding data conversions. These should be started always before and after the data transfer between S5/ S7 and WinCC. The function blocks can be downloaded from Siemens Customer Support on the Internet:

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

Enter as search expression: "Funktionsbausteine ANSI_S5"

The compressed file ANSI_S5.EXE is offered for downloading. ANSI_S5.EXE contains function block "IEEE:GP".

Active sending is described in the reference manuals of the PLCs or the CPs (Communication Processors).

2.2.5 Appendix

2.2.5.1 Appendix

Topics in the Appendix

You will find the following topics on this appendix:

- SQL instructions for specifying sorting and filter condition of user archives
- Alphabetical list of SQL keywords, these must not be used as archive or field names in the user archives
- Typical application, especially the performance when writing and reading of tags and the performance of the screen opening times
- What you must pay attention to

2.2.5.2 The SQL language

SQL (Structured Query Language) is a strong and common database language. In the functions of the WinCC script language the SQL language is used for database jobs. For further information please apply to the professional references.

For some standard functions as well as for some functions in the Editor User Archives you must specify the conditions in the database language SQL for the specification of the data records to be edited. Below you will see some examples of how an SQL instruction must be provided:

- FieldA > '1992-12-31 23:45:12.12': The statement selects all data records where the value in the column "FieldA" is greater than the one entered. FieldA is of the data type DB_TYP_TIME.
- FieldB like 'Cauldron%': This is used for example to select the data records which have in the column "FieldB" the value "Cauldron1", "Cauldron4", "Cauldron12". FieldB is of the data type DB_TYP_CHAR.
- FieldC > 100: By a condition of this form all data records are selected, which have in the column "FieldC" a value greater than 100. FieldC is of the data type DB_TYP_INTEGER
- BETWEEN FieldC = 20 AND FieldC = 200: The statement selects all data records the value of which in the column "FieldC" is between 20 and 200. FieldC is of the data type DB_TYP_INTEGER
- FieldD: This sorts according to the column "FieldD".
- FieldE desc: This sorts according to the column "FieldE" in reverse alphabetical order (descending order).

2.2.5.3 Alphabetical list of SQL keywords

SQL keywords

Archive, view and field names can be made only of letters, numbers and underscores "_" and can be made of maximum 25 characters. The first character must always be an alphabet.

The following terms must not be used as archive, view and field names:

- "Archives"
- "View"
- "Field"
- "ViewCol"
- All SQL keywords

Keywords (or reserved words) of the database language SQL must not be used as archive, view or field name in User Archives. Below you can see the SQL keywords:

Keywords used in SQL language			
add	all	alter	and
any	as	asc	begin
between	binary	break	by
call	cascade	cast	char
char_convert	character	check	checkpoint
close	comment	commit	connect
constraint	continue	convert	create
cross	current	cursor	date
dba	dbspace	deallocate	dec
decimal	declare	default	delete
desc	distinct	do	double
drop	else	elseif	encrypted
end	endif	escape	exception
exec	execute	exists	fetch
first	float	for	foreign
from	full	goto	grant
group	having	holdlock	identified
if	in	index	inner
inout	insert	instead	int
integer	into	is	isolation
join	key	left	like
lock	long	match	membership
message	mode	modify	named
natural	noholdlock	not	null
numeric	of	Off	on
open	option	Options	or

Keywords used in SQL language			
order	others	out	outer
passthrough	precision	prepare	primary
Print	privileges	proc	procedure
raiserror	readtext	Real	reference
references	release	remote	rename
resource	restrict	return	revoke
right	rollback	save	savepoint
schedule	select	set	share
smallint	some	sqlcode	sqlstate
start	stop	subtrans	subtransaction
synchronize	syntax_error	table	temporary
then	time	tinyint	to
tran	trigger	truncate	tsequal
union	unique	unknown	update
user	using	validate	values
varbinary	varchar	variable	varying
view	when	where	while
with	work	writetext	

2.2.5.4 Quantity framework

Test environment

The measurements that are described below were made in the following test environment:

- Hardware: Pentium III 600 / 256MB
- Connection: S7 Protocol Suite, Channel Unit MPI
- Project environment:
 - Data volume: Three User Archives, Archive 1 with 100 fields 3000 data records, Archive 2 with 200 fields 1500 data records and Archive 3 with 500 fields 500 data records.
 - Measurements in runtime, first and last line are read or written, triggering of the write/read commands in table control above toolbar buttons.
 - Editor closed, no C functions executed, no picture exchange executed.

WinCC Tags and raw data tags

During these measurements the WinCC tags were used. As the measurements show, the access times grow with greater user archives.

For greater user archives we recommend the use of raw data tags. Raw data tags transfer data in packs and offer a faster access even for greater archives.

Note

Per User Archive maximum 500 fields can be created.

2.2.5.5 Performance to read and write tags

Performance

In the performance measurements that were described here the behavior of the User Archives was measured during writing and reading of tags.

Please remember that the performance depends on the applied typical application.

Number of fields	Number of data records	Time for writing into the tags in sec	Time for reading from the tags in sec
100	1	1	2
100	10	1	2-3
100	50	1	3-4
100	100	1	3-9
100	1000	1-2	>3 (depending on connection)
200	1	2	3
200	10	2	4
200	50	1-2	>4
200	100	1-2	>4
200	1000	2-3	>4
500	1	3	4
500	10	3	7
500	50	3-4	Approx. 15
500	100	4	>15
500	500	4	>15

The calculated times depend on the size of the respective table.

2.2.5.6 Performance of the screen opening times

Screen Opening Times

In the following table you will see the data from one of the performance measurements of the screen opening times. We assume that in the Editor User Archives the table window is inactive.

Please remember that the screen opening times depend on the applied typical application.

Number of fields	Number of data records	Screen opening time in sec
100	1	1
100	10	2

100	100	3
100	500	3
100	1000	3
100	2000	3
100	3000	3
200	1	1
200	10	2
200	100	4
200	500	4
200	1000	4
200	1500	>4
500	1	3
500	10	4
500	100	>4
500	500	>4

2.3 User Archive Scripts

2.3.1 Standard Functions of the WinCC Script Language

The description of the user archive standard functions is divided into the following sections:

- General Information on Script Programming
- Functions for Editing and Displaying User Archives
- Handles of the Standard Functions
- Practical Example of a Script
- Reference of the Standard Functions for User Archives
- A detailed description of the user archives functions can be found in the WinCC User Archives Online Help.

WinCC provides a number of standard functions enabling the user to implement user archives in a flexible manner.

These standard functions are identified by a uniform naming convention. All standard functions for the user archives begin with "ua", for example "uaConnect", "uaArchiveOpen", "uaArchiveGetFields" etc. User archive runtime functions always begin with "uaArchive"

The functions are divided into configuration and runtime functions. Handles are required for the configuration and runtime functions, which are returned by the previously called functions "uaQueryConfiguration", "uaConnect" and "uaOpen".

Note

"ua" in the user archive functions is always written in the lower case.

Within a script, it must be ensured that the data is up-to-date.

If a script has opened a user archive and a record is added or deleted via a Control or User Archive Editor, the script is not informed of the change. The script is only notified of the changes after a requery.

See also

Example of a Script (Page 371)

Functions for Editing and Displaying User Archives (Page 366)

Handles for the Configuration of User Archives (Page 368)

Description of the Standard Functions of the WinCC Script Language (Page 379)

2.3.2 Configuration of Actions

Carry out the following steps in order to configure an action:

1. Start the Graphic Editor and create a picture of the system
2. Right-click on the object, for which an action (e.g. button) should be added.
3. Select Properties
4. Select the element from the "Properties" or "Events" tab and double click on the desired action (e.g. to configure an action for the "Press Left" mouse action, select "Events/Mouse/ Press left"). In the following dialog box, the C-code can be entered directly and then be compiled.
5. Click "OK" to conclude configuration of the action.

2.3.3 Functions for Editing and Displaying User Archives

Configuration with the Standard Functions

The "uaQueryConfiguration" function provides a handle (UAHCONFIG) for the configuration functions. The "uaSetArchive", "uaAddArchive", "uaSetField", "uaAddField", etc. configuration functions can be called with this handle. The "uaReleaseConfiguration" function terminates the configuration.

Establishing a Connection to the User Archives.

In order to access in runtime, the uaConnect standard function must be called to establish a connection to the user archive components. uaConnect generates the UAHCONNECT handle

with which the user archives and views can be opened. The "uaDisconnect" function terminates the connection to the User Archives.

Opening the Runtime Functions

A configured user archive is required for the runtime operation. The "uaQueryArchive" and "uaQueryArchiveByName" functions provide a handle for the runtime functions. After opening a user archive with the "uaArchiveOpen" function, the user archive runtime functions can be used.

Functions for the Runtime Operation

The "uaArchiveNext", "uaArchivePrevious", "uaArchiveFirst" and "uaArchiveLast" functions move the pointer. A unique assignment to a data record of the user archive is generated via the "hArchive" handle. This assignment also allows indirect addressing, as required by the screen dialog boxes.

The "uaArchiveUpdate" function stores the temporary data record in the user archive and overwrites the data record to which the pointer is currently pointing. This data record must previously be read by the "uaArchiveNext", "uaArchivePrevious", "uaArchiveFirst" or "uaArchiveLast" functions.

Terminating the Connection to the User Archives

The "uaArchiveClose" function closes a user archive. The "uaReleaseArchive" function terminates the connection to the current user archive and the "uaDisconnect" function terminates the connection to the user archive component.

Note

The connections to the user archives established in a script must also be terminated again in that script.

Function for Establishing a Connection	Function for Terminating a Connection
UaQueryConfiguration	uaReleaseConfiguration
uaConnect	uaDisconnect
uaQueryArchive	uaReleaseArchive
uaQueryArchiveByName	uaReleaseArchive
uaArchiveOpen	uaArchiveClose

For the user archives, there are two forms of API calls:

1. With prefix "ua" (lower case letters) for scripts (Global Script and action programming).
2. With prefix "UA" (upper case letters) for programs that run outside of WinCC.

If the calls for the User Archives are used in a Dynamic Wizard, they must be preceded by the prefix "UA" (upper case letters).

2.3.4 Handles for the Configuration of User Archives

"UAHCONFIG" handle

The "uaQueryConfiguration" function generates the "UAHCONFIG" handle which is a prerequisite for configuration of the user archives. This means that the "uaQueryConfiguration" function must be called first in order to receive the "UAHCONFIG" handle. This handle then allows you to call the configuration functions listed below. Finally, to complete the configuration, "uaReleaseConfiguration" must be called.

Handles for the Configuration of User Archives	
UaQueryConfiguration	---> UAHCONFIG handle
	required by:
	uaAddArchive
	uaAddField
	uaGetArchive
	uaGetField
	uaGetNumArchives
	uaGetNumFields
	uaReleaseConfiguration
	uaRemoveAllArchives
	uaRemoveAllFields
	uaRemoveArchive
	uaRemoveField
	uaSetArchive
	uaSetField

2.3.5 Handles for the Runtime Functions

"UAHCONNECT" handle

The "uaConnect" User Archives function generates the "UAHCONNECT" handle, which is required for opening and closing user archives and views. This means that the "uaConnect" function must be called first in order to receive the "UAHCONNECT" handle. This handle then allows you to call the functions listed below for opening and closing archives and views. Finally, to complete the configuration "uaDisconnect" must be called.

The "uaQueryArchive" and "uaQueryArchiveByName" functions generate the "UAHARCHIVE" handle. This handle is a prerequisite for the "uaArchiveOpen" function which opens the user archive for runtime operation. To terminate the connection, the "uaRelease" and "uaArchiveClose" functions must be called.

Note**Sort and filter user archives**

You can use the "uaArchiveSetSort" and "uaArchiveSetFilter" functions on a user archive without opening the user archive using "uaArchiveOpen".

Handles for the Runtime Functions		
uaConnect	-> Handle UAHCONNECT	
	required by...	
	uaDisconnect	
	uaQueryArchive	--> UAHARCHIVE handle
	uaQueryArchiveByName	--> UAHARCHIVE handle
		required by...
		uaArchiveOpen
		Required for:
		uaArchiveClose
		uaArchiveDelete
		uaArchiveExport
		uaArchiveGetCount
		uaArchiveGetFieldLength
		uaArchiveGetFields
		uaArchiveGetFieldType
		uaArchiveGetFieldValueDate
		uaArchiveGetFieldValueDouble
		uaArchiveGetFieldValueFloat
		uaArchiveGetFieldValueLong
		uaArchiveGetFieldValueString
		uaArchiveGetFieldName
		uaArchiveGetFilter
		uaArchiveGetID
		uaArchiveGetName
		uaArchiveGetSort
		uaArchiveImport
		uaArchiveInsert
		uaArchiveMoveFirst
		uaArchiveMoveLast
		uaArchiveMoveNext
		uaArchiveMovePrevious
		uaArchiveReadTagValues
		uaArchiveReadTagValuesByName
		uaArchiveRequery

Handles for the Runtime Functions		
		uaArchiveSetFieldValueDate
		uaArchiveSetFieldValueDouble
		uaArchiveSetFieldValueFloat
		uaArchiveSetFieldValueLong
		uaArchiveSetFieldValueString
		uaArchiveSetFilter*
		uaArchiveSetSort*
		uaArchiveUpdate
		uaArchiveWriteTagValues
		uaArchiveWriteTagValuesByName
		uaReleaseArchive
*: Functions can only be called up before "uaArchiveOpen".		

2.3.6 Example of a Script

The screenshot displays a WinCC Graphics Designer interface. At the top, there is a toolbar with various icons for editing and navigation. Below the toolbar is a data table with the following columns: ID, Date, Recipes, Water, Sugar, Coloring 7, Caffeine, and Ph. The table contains six rows of data:

	ID	Date	Recipes	Water	Sugar	Coloring 7	Caffeine	Ph
1	1	12:30:15	Calif Coke	90	10	10	10	15
2	2		Coke	80	30	15	20	20
3	3		Standard	100	50	20	20	20
4	4		CokeLight	100	20	30	20	20
5	5		CherryCo	100	80	50	5	10
6	6	30.04.99	KidsCoke	70	30	40	20	30

Below the table, there is a status bar showing "Finished", "Rec 1/6", "Row 1", and "Col 2".

Below the status bar, there is a "Global Script - Diagnose" window. This window has a toolbar with icons for opening, saving, and printing. The main area of the window displays the following information:

```

Data of Field 5:
Field Type = Integer
Field Value = 0
Field Length = 4
Data of Field 6:
Field Type = Integer
Field Value = 0
Field Length = 4
Data of Field 7:
Field Type = Integer
Field Value = 0
Field Length = 4
Data of Field 8:
Field Type = String
Field Length = 10
Data of Field 9:
Field Type = Date & Time
Jahr 1999  Monat 04  Tag 07
Field Length = 16
  
```

At the bottom of the "Global Script - Diagnose" window, there is a timestamp: "07.05.99 15:04:03".

To the right of the "Global Script - Diagnose" window, there are four buttons: "Go to WM 2", "Go to Start", "Write Archive", and "Read Archive". Below these buttons is a "Conf." button.

The following section contains an example with two standard functions to read and write a user archive in runtime. The "UAREadFromArchive" function reads the "Cola" user archive and displays the data read in the Global Script Diagnostics Control window. The "UAWriteToArchive" function writes the user archive and displays the states and messages. The diagnostics window is created by placing an OLE Control (from the Object Palette -> Smart Objects) in the Graphics Designer and then selecting the "WinCC Global Script - Diagnostics Control" from the "Insert OLE Control (OCX)" dialog box.

In the Graphics Designer, create a new screen for your project. In the screen, create the "Read Archive" and "Write Archive" buttons and assign them the standard functions described below. The procedure is as follows:

2.3 User Archive Scripts

1. Open the "Object Palette" selection window and select the "Button" field under "Windows Objects".
2. Place the button in the working area of Graphics Designer and drag it to size while keeping the mouse button pressed.
3. Right-click on this newly created button and select "Properties" from the pop-up menu. In the "Properties" tab, the button label (text) and color can be defined. The labels, for example, could read "Read Archive" and "Write Archive".
4. In the "Events" tab, assign an action to the mouse by selecting "Mouse" and then double-clicking on "Mouse Action". The Script Editor is displayed. Enter the "UAREadFromArchive" standard script function listed below:

```

#include "apdefap.h"

void UAREadFromArchive()
{
    UAHCONNECT    hConnect = 0;
    UAHARCHIVE    hArchive = 0;
    LONG          IndexArchive;
    LONG          FieldLength;
    LONG          FieldType;
    LONG          NumberOfFields;
    LONG          Index;
    long          IntValue;
    double        DoubleValue;
    char          ArchiveName[255], StringField[255];
    SYSTEMTIME    SysDate;

    //*****  Connect to Component User Archives *****
    if (uaConnect( &hConnect ) == FALSE )
    {
        printf("uaConnect error: %d\n", uaGetLastError() );
        return;
    }
    if (hConnect == NULL)
    {
        printf("Handle UAHCONNECT equals 0\n" );
        return;
    }

    //*****  Connect to Archive via Archive Name *****
    if (uaQueryArchiveByName( hConnect, "Cola", &hArchive ) == FALSE )
    {
        printf("uaQueryArchive Error: %d\n", uaGetLastError() );
    }

```

goto finish;
}
//***** Opens Archive *****
if (uaArchiveOpen(hArchive) == FALSE)
{
printf("uaArchive Open Error\n");
goto finish;
}
//***** Move to first record set *****
if (uaArchiveMoveFirst(hArchive) == FALSE)
{
printf("uaArchiveMoveFirst Error = %u\n");
goto finish;
}
//***** Get Number of Fields *****
NumberOfFields = uaArchiveGetFields(hArchive);
printf("Number of Fields = %u\n", NumberOfFields);
//***** Read and show Data Fields *****
for (Index = 1; Index < NumberOfFields; Index++)
{
printf("Data of Field %u: \n", Index);
FieldType = uaArchiveGetFieldType(hArchive, Index);
switch (FieldType)
{
case UA_FIELDTYPE_INTEGER :
printf("Field Type = Integer\n");
if (uaArchiveGetFieldValueLong (hArchive, Index, &IntValue) == TRUE)
printf("Field Value = %u\n", IntValue);
else
printf("Error callinguaArchiveGetFieldValueLong: %d\n", uaGetLastError());
break;
case UA_FIELDTYPE_DOUBLE :
printf("Field Type = Double\n");
if (uaArchiveGetFieldValueDouble (
hArchive, Index, &DoubleValue) == TRUE)
printf("Field Value = %g\n", DoubleValue);
else

2.3 User Archive Scripts

printf("Error calling uaArchiveGetFieldValueDouble: %d\n", uaGetLastError());
break;
case UA_FIELDTYPE_STRING :
printf("Field Type = String\n");
if (uaArchiveGetFieldValueString (hArchive, Index, StringField, 20) == TRUE)
printf("Field Value = %s\n", StringField);
else
printf("Error calling uaArchiveGetFieldValueString: %d\n", uaGetLastError());
break;
case UA_FIELDTYPE_DATETIME :
printf("Field Type = Date & Time\n");
if (uaArchiveGetFieldValueDate (hArchive, Index, &SysDate) == TRUE)
printf("%d.%d.%d\n ", SysDate.wDay, SysDate.wMonth, SysDate.wYear);
else
printf("Error calling uaArchiveGetFieldValueLong: %d\n", uaGetLastError());
break;
case -1 :
default:
printf("Error executing uaArchiveGetFieldType\n");
}
//***** Read and show Field Length *****
FieldLength = uaArchiveGetFieldLength(hArchive, Index);
if (FieldLength != -1)
printf("Field Length = %u\n", FieldLength);
else
printf("Error executing uaArchiveGetFieldLength\n");
}
//***** Close all handles and connections *****
finish.;
//***** Close Archive *****
if(NULL != hArchive)
{
if (uaArchiveClose (hArchive) == FALSE)
{
printf("error on closing archive\n");
}

```

    }
//***** Release Connection to Archive *****
if( NULL != hArchive )
{
    if (uaReleaseArchive ( hArchive ) == FALSE )
    {
        printf("error on releasing archive\n" );
    }
    hArchive = 0;
}
//***** Disconnect to Component User Archives *****
if( NULL != hConnect )
{
    if (uaDisconnect ( hConnect ) == FALSE )
    {
        printf("error on disconnection\n" );
    }
    hConnect = 0;
}
}

```

Create a second button for describing the user archive. Follow the procedure of the first button. Select the standard function "UAWriteToArchive" and enter the following script:

```

void UAWriteToArchive()
{
    UAHCONNECT    hConnect = 0;
    UAHARCHIVE    hArchive = 0;
    LONG          IndexArchive;
    LONG          FieldLength;
    LONG          FieldType;
    LONG          NumberOfFields;
    LONG          Index;
    long          IntValue;
    double        DoubleValue;
    char          StringField[255];
    SYSTEMTIME    SysDate;

//***** Connect to Componente User Archives *****
if (uaConnect( &hConnect ) == FALSE )
{
    printf("uaConnect error: %d\n", uaGetLastError() );
}

```

return;
}
if (hConnect == NULL)
{
printf("Handle UAHCONNECT equals NULL\n");
return;
}
//***** Connect to Archive via Name *****
if (uaQueryArchiveByName(hConnect, "Cola", &hArchive) == FALSE)
{
printf("uaQueryArchive Error: %d\n", uaGetLastError());
goto finish;
}
//***** Opens Archives *****
if (uaArchiveOpen(hArchive) == FALSE)
{
printf("uaArchive Open Error\n");
goto finish;
}
//***** Get Number of Fields *****
NumberOfFields = uaArchiveGetFields(hArchive);
printf("Number of Fields = %u\n", NumberOfFields);
//***** Read Last Data Set *****
if (uaArchiveMoveLast(hArchive) == TRUE)
printf("Number of Fields = %u\n", NumberOfFields);
else
{
printf("uaArchiveMoveLast Error: %d\n", uaGetLastError());
goto finish;
}
//***** Write into Data Fields *****
IntValue = 32;
DoubleValue = 64;
strcpy(StringField, "Text12");
GetSystemTime(&SysDate);
for (Index = 1; Index < NumberOfFields; Index++)
{

printf("Data of Field %u: \n", Index);
FieldType = uaArchiveGetFieldType(hArchive, Index);
switch (FieldType)
{
case UA_FIELDTYPE_INTEGER :
printf("Field Type = Integer\n");
if (uaArchiveSetFieldValueLong (hArchive, Index, IntValue) == TRUE)
printf("Field Value = %u\n", IntValue);
else
printf("Error calling uaArchiveSetFieldValueLong: %d\n", uaGetLastError());
break;
case UA_FIELDTYPE_DOUBLE :
printf("Field Type = Double\n");if (uaArchiveSetFieldValueDouble (hArchive, Index, DoubleValue)
== TRUE)
printf("Field Value = %g\n", DoubleValue);
else
printf("Error calling uaArchiveSetFieldValueDouble: %d\n", uaGetLastError());
break;
case UA_FIELDTYPE_STRING :
printf("Field Type = String\n");
if (uaArchiveSetFieldValueString (hArchive, Index, StringField) == TRUE)
printf("Field Value = %s\n", StringField);
else
printf("Error calling uaArchiveSetFieldValueString: %d\n", uaGetLastError());
break;
case UA_FIELDTYPE_DATETIME :
printf("Field Type = Date & Time\n");
if (uaArchiveSetFieldValueDate (hArchive, Index, &SysDate) == TRUE)
printf("%d.%d.%d\n ",SysDate.wDay, SysDate.wMonth, SysDate.wYear);
else
printf("Error calling uaArchiveGetFieldValueLong: %d\n", uaGetLastError());
break;
case -1 :
default:

```

printf("Error executing uaArchiveGetFieldType\n");
}

FieldLength = uaArchiveGetFieldLength( hArchive, Index );
if ( FieldLength != -1 )
printf("Field Length = %u\n", FieldLength );
else
printf("Error executing uaArchiveGetFieldLength\n");
}

// ***** Update Archive *****
if (uaArchiveUpdate(hArchive) == FALSE )
{
printf("uaArchiveUpdate Error:\n" );
}

//***** Close all handles and connections *****
finish;;

//***** Close Archive *****
if( NULL != hArchive )
{
if (uaArchiveClose ( hArchive ) == FALSE )
{
printf("error on closing archive\n" );
}
}

//***** Release Connection to Archive *****
if( NULL != hArchive )
{
if (uaReleaseArchive ( hArchive ) == FALSE )
{
printf("error on releasing archive\n" );
}
hArchive = 0;
}

//***** Disconnect to Component User Archives *****
if( NULL != hConnect )
{
if (uaDisconnect ( hConnect ) == FALSE )
{
printf("error on disconnection\n" );
}
hConnect = 0;
}
}

```

The dialog can then be closed and runtime started. The effects of the scripts can then be monitored in the "Global Script Diagnostics Window".

2.3.7 Appendix

2.3.7.1 Description of the Standard Functions of the WinCC Script Language

WinCC Script Language

The following section contains a detailed description of the functions of the WinCC script language.

The functions are divided into four groups:

- Standard functions for user archives
- Functions for Configuring User Archives
- General Runtime Functions
- Archive-specific Runtime Functions

You will find the functions in an alphabetical order in the groups.

When a data record is created, no check is made for completeness or correctness. In particular, a field must not be left blank.

See also

Standard Functions of the WinCC Script Language (Page 365)

2.3.7.2 Standard functions for user archives

uaGetLastError

Description:

The functions of the WinCC script language returns a BOOL value, whereby TRUE corresponds to error-free processing. If FALSE is returned, the error of the last function can be read with "uaGetLastError()" and "uaGetLastHResult()".

If "uaGetLastError()" is called after running several functions, "uaGetLastError()" will return the error that occurred last. To be certain of the function in which the error occurred, "uaGetLastError()" and "uaGetLastHResult()" should be called every time a function returns FALSE.

Example:

```
if ( uaArchiveGetFieldValueLong ( hArchive, Index, &IntValue ) ==  
TRUE )
```

```
printf( "Field Value = %u\n", IntValue );  
else  
printf("Error calling uaArchiveGetFieldValueLong: %d / %08lx\n",  
uaGetLastError(), uaGetLastHResult());
```

With functions, that do not return a value (VOID), there should be a request via `uaGetLastError()` in any case.

Example:

```
uaArchiveGetFilter(hArchive, pszFilter, cMaxLen);  
INT nUAError = uaGetLastError ( );  
if ( UA_ERROR_SUCCESS != nUAError)  
{  
    printf( "Filter = [%s]\n", pszFilter );  
}  
else  
{  
    printf("Error calling uaArchiveGetFilter: %d, hr=0x%08lX\n",  
nUAError, uaGetLastHResult());  
}  
  
INT uaGetLastError()
```

Return Value:

Error status of the last function executed. The following errors can be returned by "uaGetLastError()":

- UA_ERROR_SUCCESS
- UA_ERROR_GENERIC
- UA_ERROR_CONNECT_FAILED
- UA_ERROR_OPEN_FAILED
- UA_ERROR_CLOSE_FAILED
- UA_ERROR_REQUERY_FAILED
- UA_ERROR_MOVE_FAILED
- UA_ERROR_INSERT_FAILED
- UA_ERROR_UPDATE_FAILED
- UA_ERROR_DELETE_FAILED
- UA_ERROR_IMPORT_FAILED
- UA_ERROR_EXPORT_FAILED
- UA_ERROR_READ_FAILED

UA_ERROR_WRITE_FAILED
UA_ERROR_GET_FAILED
UA_ERROR_SET_FAILED
UA_ERROR_INVALID_NAME
UA_ERROR_INVALID_TYPE
UA_ERROR_INVALID_NUMRECS
UA_ERROR_INVALID_COMMTYPE
UA_ERROR_INVALID_LENGTH
UA_ERROR_INVALID_PRECISION
UA_ERROR_NULL_POINTER
UA_ERROR_INVALID_POINTER
UA_ERROR_INVALID_HANDLE
UA_ERROR_INVALID_INDEX
UA_ERROR_SERVER_UNKNOWN

These error constants, as well as the predefines of the user archives routines, are located in CCUACAPI.H.

uaGetLastHResult

Description:

Reads the last occurred COM error. This function serves mainly the diagnosis of incompatibilities during COM implementation or rather to detect registration and communication errors.

This function is basically to be used in addition to UAGetLastError, if a user archive function (e.g. uaConnect) signals an error with "FALSE".

```
LONG uaGetLastHResult()
```

Return Value:

Last occurred COM error

2.3.7.3 Functions for Configuring User Archives

Functions for Configuring User Archives

Functions for Configuration

These functions serve for the configuration of user archives.

Function	Description
uaAddArchive	Adds a new user archive.
uaAddField	Adds a new field
uaGetArchive	Reads the archive configuration
uaGetField	Reads the field configuration
uaGetNumArchives	Finds the number of configured archives
uaGetNumFields	Finds the number of fields
UaQueryConfiguration	Establishes a connection to the User Archives configuration
uaReleaseConfiguration	Terminates the connection to the configuration
uaRemoveAllArchives	Deletes all archives
uaRemoveAllFields	Deletes all fields
uaRemoveArchive	Deletes an archive
uaRemoveField	Deleting a Field
uaSetArchive	Writes the archive configuration
uaSetField	Sets the field configuration

uaAddArchive

Description:

Adds a new user archive. This corresponds to the configuration of a new user archive using the User Archives editor.

```
LONG uaAddArchive (  
    UAHCONFIG hConfig,  
    UACONFIGARCHIVE* pArchive )
```

Parameters:

```
UAHCONFIG hConfig,
```

Configuration handle for the user archive. This handle is generated with "uaQueryConfiguration".

```
UACONFIGARCHIVE* pArchive
```

Pointer on the buffer for the storage of the user archive configuration.

Return Value:

Index of the new user archive. An error corresponds to "-1".

uaAddField**Description:**

Adds a new data field.

```
LONG uaAddField (  
    UAHCONFIG hConfig,  
    long lArchive,  
    UACONFIGFIELD* pField )
```

Parameters:

UAHCONFIG hConfig,

Configuration handle of the user archive. This handle is generated with "uaQueryConfiguration".

long lArchive,

Archive Index (0 to (uaGetNumArchives()-1))

UACONFIGFIELD* pArchive

Pointer on the buffer of the field configuration.

Return Value:

Index of the new field. A value of "-1" indicates an error.

See also

Structure for Programming the "uaCONFIGFIELD" Handle (Page 390)

Structure for Programming the "uaCONFIGFIELDW" Handle (Page 391)

uaGetArchive**Description:**

Reads the user archive configuration.

```
BOOL uaGetArchive (  
    UAHCONFIG hConfig,  
    long lArchive,
```

```
UACONFIGARCHIVE* pArchive )
```

Parameters:

```
UAHCONFIG hConfig,
```

Configuration handle of the user archive. This handle is generated with "uaQueryConfiguration".

```
long lArchive,
```

Archive Index (0 to (uaGetNumArchives()-1))

```
UACONFIGARCHIVE* pArchive
```

Pointer on the buffer for receiving the user archive configuration.

Return Value:

TRUE : Access to the user archive was successful

FALSE : Error

uaGetField

Description:

Reads the field configuration.

```
BOOL uaGetField (
```

```
UAHCONFIG hConfig,
```

```
long lArchive,
```

```
long lField,
```

```
UACONFIGFIELD* pField )
```

Parameters:

```
UAHCONFIG hConfig,
```

Configuration handle of the user archive. This handle is generated with "uaQueryConfiguration".

```
long lArchive,
```

Archive Index (0 to (uaGetNumArchives()-1))

```
long lField,
```

The field number, where lField = 0 is addressing the first field.

```
UACONFIGFIELD* pArchive
```

Pointer on the buffer for receiving the field configuration.

Return Value:

TRUE : Access to the user archive was successful

FALSE : Error

uaGetNumArchives**Description:**

Reads the number of user archives currently configured.

```
LONG uaGetNumArchives (  
    UAHCONFIG hConfig )
```

Parameters:

UAHCONFIG hConfig

Configuration handle of the user archive. This handle is generated with "uaQueryConfiguration".

Return Value:

Number of user archives currently configured. In case of an error, -1 will be returned.

uaGetNumFields**Description:**

Supplies the number of the configured fields. The "ID", "Last User" and "Last Access" fields are not included. In the configuration calls, the indexes are indicated from 0 to "uaGetNumFields() -1".

```
LONG uaGetNumFields (  
    UAHCONFIG hConfig,  
    long lArchive )
```

Parameters:

UAHCONFIG hConfig,

Configuration handle of the user archive. This handle is generated with "uaQueryConfiguration".

long lArchive,

Archive Index (0 to (uaGetNumArchives()-1))

Return Value:

Number of the configured fields. In case of an error, -1 will be returned.

UaQueryConfiguration

Description:

Establishes a connection to the user archive for the configuration.

```
BOOL uaQueryConfiguration (  
    UAHCONFIG* phConfig )
```

Parameters:

UAHCONFIG* phConfig,
Pointer to the handle of the archive.

Return Value:

TRUE : Access to the user archive was successful
FALSE : Error

uaReleaseConfiguration

Description:

Terminates connection to user archives (configuration).

```
BOOL uaReleaseConfiguration (  
    UAHCONFIG hConfig,  
    BOOL bSave )
```

Parameters:

UAHCONFIG hConfig

Configuration handle of the user archive. This handle is generated with "uaQueryConfiguration".

BOOL bSave

Saves changes made to the configuration before terminating the connection to user archive.

TRUE = Save Changes, FALSE = Discard Changes

Warning: Save changes (bSave = TRUE) may only be used when Runtime is not active! You can check whether Runtime is active by requesting ualsActive().

Return Value:

TRUE : Successful termination of the connection

FALSE : Error

uaRemoveAllArchives**Description:**

Deletes all user archives which are not currently used in views.

```
BOOL uaRemoveAllArchives  
( UAHCONFIG hConfig )
```

Parameters:

UAHCONFIG hConfig

Configuration handle of the user archive. This handle is generated with "uaQueryConfiguration".

Return Value:

TRUE : Successful deletion

FALSE : Error

Comment

After that, you can query by means of "uaGetNumArchives()" whether all archives were deleted.

uaRemoveAllFields**Description:**

Removes all fields.

```
BOOL uaRemoveAllFields (  
    UAHCONFIG hConfig,  
    long lArchive )
```

Parameters:

UAHCONFIG hConfig,

Configuration handle of the user archive. This handle is generated with "uaQueryConfiguration".

long lArchive,

Archiv Index (0 to (uaGetNumArchives()-1))

Return Value:

TRUE : Successful deletion of the fields

FALSE : Error

uaRemoveArchive

Description:

uaRemoveArchive deletes the entire user archive configured.

```
BOOL uaRemoveArchive (  
    UAHCONFIG hConfig,  
    long lArchive )
```

Parameters:

UAHCONFIG hConfig,

Configuration handle of the user archive. This handle is generated with "uaQueryConfiguration".

long lArchive,

Archive Index (0 to (uaGetNumArchives()-1))

Return Value:

TRUE : User archive has been deleted successfully

FALSE : Error

uaRemoveField

Description:

Removes a field.

```
BOOL uaRemoveField (  
    UAHCONFIG hConfig,  
    long lArchive,  
    long lField )
```

Parameters:

UAHCONFIG hConfig,
Configuration handle for the user archive. This handle is generated with "uaQueryConfiguration".
long lArchive,
Archive Index (0 to (uaGetNumArchives()-1))
long lField,
The field number, where lField = 0 is addressing the first field.

Return Value:

TRUE : Field has been deleted successfully
FALSE : Error

uaSetArchive**Description:**

Defines the configuration of a user archive.

```
BOOL uaSetArchive (  
    UAHCONFIG hConfig,  
    long lArchive,  
    UACONFIGARCHIVE* pArchive  
)
```

Parameters:

UAHCONFIG hConfig,
Configuration handle for the user archive. This handle is generated with "uaQueryConfiguration".
long lArchive,
Archive Index (0 to (uaGetNumArchives()-1))
UACONFIGARCHIVE* pArchive
Pointer to the pArchive buffer with the user archive configuration.

Return Value:

TRUE : Access to the user archive was successful
FALSE : Error

uaSetField

Description:

Sets the field configuration.

```
BOOL uaSetField (
    UAHCONFIG hConfig,
    long lArchive,
    long lField,
    UACONFIGFIELD* pField )
```

Parameters:

UAHCONFIG hConfig,

Configuration handle for the user archive. This handle is generated with "uaQueryConfiguration".

long lArchive,

Archive Index (0 to (uaGetNumArchives()-1))

long lField,

The field number, where lField = 0 is addressing the first field.

UACONFIGFIELD* pField

Pointer on the buffer of the field configuration.

Return Value:

TRUE : Access to the user archive was successful

FALSE : Error

Structure for Programming the "uaCONFIGFIELD" Handle

Structure "uaCONFIGFIELD"

```
typedef struct tagUACONFIGFIELD
{
    LONG lArchiveId; // Unique ID of Archive
    LONG lArchiveId; // Unique ID of Archive
    LONG lPosition; // Position (Sequence) of the Archives
    CHAR szName[UA_MAXLEN_NAME+1]; // Archive name max. 20 characters
    CHAR szAlias[UA_MAXLEN_ALIAS+1]; // Alias name max 50 characters
```

```

LONG lType; //Archive type

LONG lLength; /* Maximum number of characters if the data field is of STRING type; else
not used */

LONG lPrecision; // Is used internally; need not be populated

CHAR szMinValue[UA_MAXLEN_VALUE+1]; /* Minimum for data fields that are not
STRING or DATE type; else not used */

CHAR szMaxValue[UA_MAXLEN_VALUE+1]; /* Maximum for data fields that are not
STRING or DATE type; else not used */

CHAR szStartValue[UA_MAXLEN_VALUE+1]; // Start value

CHAR szDMVarName[UA_MAXLEN_DMVARNAME+1]; /* Tag from the Data Manager (is used
for archives with communication via WinCC tags) */

DWORD dwReadRight; // Read access rights

DWORD dwWriteRight; // Write access rights

DWORD dwFlags; // Last access

} UA_CONFIGFIELD;

```

See also

uaAddField (Page 383)

Structure for Programming the "uaCONFIGFIELDW" Handle**Structure "uaCONFIGFIELDW"**

```

typedef struct tagUA_CONFIGFIELDW
{
    LONG lArchiveId; // Unique ID of Archive
    LONG lArchiveId; // Unique ID of Archive
    LONG lPosition; // Position (Sequence) of the Archives
    WCHAR wszName[UA_MAXLEN_NAME+1]; // Archive name max. 20 characters
    WCHAR wszAlias[UA_MAXLEN_ALIAS+1]; // Alias name max. 50 characters
    LONG lType; //Archive type
    LONG lLength; /* Maximum number of characters if the data field is of STRING type; else
not used */
    LONG lPrecision; // Is used internally; need not be populated
    WCHAR wszMinValue[UA_MAXLEN_VALUE+1]; /* Minimum for data fields of type that are
not STRING or DATE, else not used */

```

```

WCHAR wszMaxValue[UA_MAXLEN_VALUE+1]; /* Maximum for data fields that are not of
STRING or DATE type; else not used */

WCHAR wszStartValue[UA_MAXLEN_VALUE+1]; // Start value

WCHAR wszDMVarName[UA_MAXLEN_DMVARNAME+1]; /* Tag from the Data manager (is
used in archives for communication via WinCC tags) */

DWORD dwReadRight; // Read access rights

DWORD dwWriteRight; // Write access rights

DWORD dwFlags; // Last access

} UA_CONFIGFIELD

```

See also

[uaAddField](#) (Page 383)

Structure for Programming the "uaAddArchive" Handle**Structure "uaAddArchive"**

```

typedef struct tagUAADDARCHIVE
{
    LONG lArchiveId; // Unique ID of Archive
    LONG lPosition; // Position (Sequence) of the Archives
    CHAR szName[UA_MAXLEN_NAME+1]; // Archive name max. 20 characters
    CHAR szAlias[UA_MAXLEN_ALIAS+1]; // Alias name max 50 characters
    LONG lType; UA_ARCHIVETYPE_UNLIMITED // Archive type unlimited
    UA_ARCHIVETYPE_LIMITED // Archive type limited
    LONG lNumRecs; // max. number of data sets
    LONG lCommType;
    UA_COMMTYPE_NONE // no communication
    UA_COMMTYPE_RAW // Communication via raw data
    UA_COMMTYPE_DIRECT // Communication via DM tags
    CHAR szPLCID[UA_MAXLEN_PLCID+1]; // PLCID of raw data tag
    CHAR szDMVarName[UA_MAXLEN_DMVARNAME+1]; // Name of the raw data tag
    CHAR szIDVar[UA_MAXLEN_DMVARNAME+1]; // Control tag ID
    CHAR szJobVar[UA_MAXLEN_DMVARNAME+1]; // Control tag job
    CHAR szFieldVar[UA_MAXLEN_DMVARNAME+1]; // Control tag field
}

```

```

CHAR szValueVar[UA_MAXLEN_DMVARNAME+1]; // Control tag value
DWORD dwReadRight; // Read access rights
DWORD dwWriteRight; // Write access rights
DWORD dwFlags; UA_ARCHIVEFLAG_ACCESS // Last access
UA_ARCHIVEFLAG_USER // Last user
} UAADDARCHIVE;

```

Note**LONG lArchiveId; //Unique ID of archive**

When indicating the value: 0, a unique ID is automatically used and returned by "AddArchive". This returned ID is then specified for "AddField" (When -1 is returned as an ID, the archive could not be created).

Structure for Programming the "uaCONFIGARCHIVEA" Handle**Structure "uaCONFIGARCHIVE"**

```

typedef struct tagUACONFIGARCHIVE
{
    LONG lArchiveId; // Unique ID of Archive
    LONG lPosition; // Position (Sequence) of the Archives
    CHAR szName[UA_MAXLEN_NAME+1]; // Archive name max. 20 characters
    CHAR szAlias[UA_MAXLEN_ALIAS+1]; // Alias name max 50 characters
    LONG lType; UA_ARCHIVETYPE_UNLIMITED // Archive type unlimited
    UA_ARCHIVETYPE_LIMITED // Archive type limited
    LONG lNumRecs; // max. number of data sets
    LONG lCommType;
    UA_COMMTYPE_NONE // no communication
    UA_COMMTYPE_RAW // Communication via raw data
    UA_COMMTYPE_DIRECT // Communication via DM tags
    CHAR szPLCID[UA_MAXLEN_PLCID+1]; // PLCID of raw data tag
    CHAR szDMVarName[UA_MAXLEN_DMVARNAME+1]; // Name of the raw data tag
    CHAR szIDVar[UA_MAXLEN_DMVARNAME+1]; // Control tag ID
    CHAR szJobVar[UA_MAXLEN_DMVARNAME+1]; // Control tag job
    CHAR szFieldVar[UA_MAXLEN_DMVARNAME+1]; // Control tag field

```

2.3 User Archive Scripts

```

CHAR szValueVar[UA_MAXLEN_DMVARNAME+1]; // Control tag value

DWORD dwReadRight; // Read access rights

DWORD dwWriteRight; // Write access rights

DWORD dwFlags; UA_ARCHIVEFLAG_ACCESS // Last access

UA_ARCHIVEFLAG_USER // Last user

} UACONFIGARCHIVE;

```

Note

LONG IArchiveId; //Unique ID of archive

When indicating the value: 0, a unique ID is automatically used and returned by "ConfigArchive". This returned ID is then specified for "ConfigField" (When -1 is returned as an ID, the archive could not be created).

2.3.7.4 General Runtime Functions

General Runtime Functions

Functions for User Archives

These functions open and close user archives and views for the runtime operation.

Function	Description
uaConnect	Establishing a Connection to the User Archives. This connection is valid for all user archives in runtime
uaDisconnect	If a connection to User Archives (runtime) exists, it will be terminated.
uaGetLocalEvents	Reads local events
uaIsActive	Determines, if runtime is active
uaOpenArchives	Determines the number of user archives open
uaOpenViews	Determines the number of open views
uaQueryArchive	Establishes a connection to a user archive
uaQueryArchiveByName	Establishes a connection to a user archive using the archive name
uaReleaseArchive	Terminates the connection to the user archive
uaSetLocalEvents	Sets local events
uaUsers	Finds the number of active connections or active users.

uaConnect

Description:

Establishes connection to User Archives (Runtime).

```
BOOL uaConnect (
    UAHCONNECT* phConnect )
```

Parameters:

UAHCONNECT* phConnect

Pointer on the handle of the newly connected user archive.

Return Value:

TRUE : A user archive has been successfully connected

FALSE : Error

uaDisconnect

Description:

If a connection to User Archives (Runtime) exists, it will be terminated.

```
BOOL uaDisconnect (
    UAHCONNECT hConnect )
```

Parameters:

UAHCONNECT hConnect

Handle of the connected User Archive (Runtime). This handle is generated with "uaConnect".

Return Value:

TRUE : Successful disconnection of a user archive

FALSE : Error

uaGetLocalEvents

Description:

Reads local events

```
BOOL uaGetLocalEvents
```

```
( UAHCONNECT hConnect )
```

Parameters:

UAHCONNECT hConnect

Handle of the User Archive. This handle is generated with "uaConnect".

Return Value:

Local event (bLocalEvents)

uaIsActive

Description:

Examines if Runtime is active.

```
BOOL uaIsActive (  
    UAHCONNECT hConnect )
```

Parameters:

UAHCONNECT hConnect

Handle of the User Archive. This handle is generated with "uaConnect".

Return Value:

TRUE : One or more user archives are actively open in Runtime

FALSE : No user archive in Runtime

uaOpenArchives

Description:

Queries how many user archives are open in Runtime.

```
LONG uaOpenArchives (  
    UAHCONNECT hConnect )
```

Parameters:

UAHCONNECT hConnect

Handle of the User Archive. This handle is generated with "uaConnect".

Return Value:

Number of currently open user archives.

uaOpenViews**Description:**

Queries how many user archive views are open in Runtime.

```
LONG uaOpenViews (  
    UAHCONNECT hConnect )
```

Parameters:

UAHCONNECT hConnect

Handle of the view. This handle is generated with "uaConnect".

Return Value:

Number of views currently open.

uaQueryArchive**Description:**

Establishes a connection to the user archive for Runtime operation. UaQueryArchive creates the handle UAHARCHIVE.

```
BOOL uaQueryArchive (  
    UAHCONNECT hConnect,  
    LONG lArchive,  
    UAHARCHIVE* phArchive )
```

Parameters:

UAHCONNECT hConnect

Handle of the connected user archive (Runtime). This handle is generated with "uaConnect".

LONG lArchive

Archiv Index (0... uaGetNumArchives() -1)

UAHARCHIVE* phArchive

Pointer to the handle of the archive.

Return Value:

TRUE : Successful generation of the handle to the user Archive

FALSE : Error

Comment:

If you use User Archives functions in a client project, which views redundant server pairs, the user archives connection cannot automatically be switched to the new master when changing masters. In this case, all user archives calls deliver the LastError

UA_ERROR_SERVER_UNKNOWN = 1004. As a result, user programs must run a new uaQueryArchive() or uaQueryArchiveByName() and uaArchiveOpen().

uaQueryArchiveByName

Description:

Establish a connection to a user archive for Runtime operation using the archive name.

UaQueryArchiveByName creates the handle UAHARCHIVE to the user archive.

```
BOOL uaQueryArchiveByName (
    UAHCONNECT hConnect,
    LPCSTR pszName,
    UAHARCHIVE* phArchive )
```

Parameters:

UAHCONNECT hConnect

Handle of the connected user archive (Runtime). This handle is generated with "uaConnect".

LPCSTR pszName

Name of the user archive. With a client project, you can add a server prefix with '::' as a separator to the archive name, if one other than the default server is used.

UAHARCHIVE* phArchive

Pointer to the handle of the archive.

Return Value:

TRUE : Handle to the user archive has been generated successfully

FALSE : Error

Comment:

If you use User Archives functions in a client project, which views redundant server pairs, the user archives connection cannot automatically be switched to the new master when changing masters. In this case, all user archives calls deliver the LastError

UA_ERROR_SERVER_UNKNOWN = 1004. As a result, user programs must run a new uaQueryArchive() or uaQueryArchiveByName() and uaArchiveOpen().

uaReleaseArchive

Description:

Terminates the connection to the current user archive.

```
BOOL uaReleaseArchive (
    UAHARCHIVE hArchive )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

Return Value:

TRUE : Disconnection from the user archive completed successfully.

FALSE : Error

Comment

The handle "hArchive" must be set to "NULL" with successful disengagement, so that the error "UA_ERROR_INVALID_HANDLE" will certainly appear if the no longer valid handle is continuously used, without the respective function starving in the COM interfaces for a long time.

uaSetLocalEvents

Description:

Sets local events.

```
void uaSetLocalEvents (
    UAHCONNECT hConnect
    BOOL bLocalEvents )
```

Parameters:

UAHCONNECT hConnect

Handle of the User Archive. This handle is generated with "uaConnect".

BOOL bLocalEvents

Local Event

uaUsers

Description:

The Runtime function "uaUsers" returns the number of all users connected to the user archive using "uaConnect". Please note that this will also take into consideration WinCC internal calls to the user archive, aside from the calls initiated by the user (say, from scripts, etc.).

```
LONG uaUsers (  
    UAHCONNECT hConnect )
```

Parameters:

UAHCONNECT hConnect

Handle of the User Archive. This handle is generated with "uaConnect".

Return Value:

Number of active connections or users.

2.3.7.5 Archive-specific Runtime Functions

Archive-specific Runtime Functions

User Archive Functions

These functions are used to operate the user archives and views in runtime.

Function	Description
uaArchiveClose	Terminates the connection to the current user archive.
uaArchiveDelete	Deletes a data record from the current user archive
uaArchiveExport	Exports the current user archive
uaArchiveGetCount	Reads the number of data records.
uaArchiveGetFieldLength	Reads the length of the current field
uaArchiveGetFieldName	Reads the name of the current field
uaArchiveGetFields	Reads the number of fields
uaArchiveGetFieldType	Reads the type of the current field
uaArchiveGetFieldValueDate	Reads date and time and places it in the current data field
uaArchiveGetFieldValueDouble	Reads the "Double" value of the current data field
uaArchiveGetFieldValueFloat	Reads the "Float" value of the current data field
uaArchiveGetFieldValueLong	Reads the "Long Int" of the current data field

uaArchiveGetFieldValueString	Reads the "String" of the current data field
uaArchiveGetFilter	Reads the filter of the current data field
uaArchiveGetID	Reads the ID of the current data field
uaArchiveGetName	Reads the name of the current data field
uaArchiveGetSort	Reads the sorting of the current data field
uaArchiveImport	Import User Archive
uaArchiveInsert	Inserts a new data record into the user archive
uaArchiveMoveFirst	Goes to the first data record
uaArchiveMoveLast	Goes to the last data record
uaArchiveMoveNext	Goes to the next data record
uaArchiveMovePrevious	Goes to the previous data record
uaArchiveOpen	Establishes a connection to the current user archive
uaArchiveReadTagValues	Reads tag values
uaArchiveReadTagValuesByName	Reads tag values based on name
uaArchiveRequery	New Query
uaArchiveSetFieldValueDate	Writes the current data field
uaArchiveSetFieldValueDouble	Writes the "Double" value of the current data field
uaArchiveSetFieldValueFloat	Writes the "Float" value of the current data field
uaArchiveSetFieldValueLong	Writes the "Long Int" of the current data field
uaArchiveSetFieldValueString	Writes the "String" of the current data field
uaArchiveSetFilter	Sets the filter
uaArchiveSetSort	Sets the sort criteria
uaArchiveUpdate	Updates the user archive which is open.
uaArchiveWriteTagValues	Writes the values of the current data record into a tag
uaArchiveWriteTagValuesByName	Writes the values of the current data record into a tag based on name

uaArchiveClose

Description:

Terminates the connection to the current user archive.

```
BOOL uaArchiveClose (
    UAHARCHIVE hArchive )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

Return Value:

TRUE : User archive has been closed successfully

FALSE : Error

uaArchiveDelete

Deleting archives

Standard functions of the WinCC Script language!AL(UAEditorScriptAllgemein,3,"","")

References of the standard functions for User Archivs!AL(UAEditorReferenzenScript,3,"","")

Description:

Deletes the data from a user archive. The configured user archive, however, is retained.

```
BOOL uaArchiveDelete (  
    UAHARCHIVE hArchive,  
    LPCSTR pszWhere )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LPCSTR pszWhere

This string contains the SQL selection expression. It defines, which data records are to be deleted. The expression is identical to the one in the "DELETE FROM <archive> WHERE pszWhere" SQL statement.

Warning! If this string is empty, the entire user archive will be deleted.

Return Value:

TRUE : User archive has been deleted successfully

FALSE : Error

uaArchiveExport

Description:

Exports the current user archive to another archive in CSV format.

```
BOOL uaArchiveExport (  
    UAHARCHIVE hArchive,
```

```
LPCSTR pszDestination,
LONG lType,
LONG lOptions )
```

Parameter:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LPCSTR pszDestination

File name of the target archive. When calling the function on clients, the path specification refers to the server computer.

LONG lType

Data format of the target archive. Two formats are available:

UA_FILETYPE_DEFAULT = 0: Default file format = CSV

UA_FILETYPE_CSV = 1: CSV file format

LONG lOptions

Options

Reserved for later expansions. Must be 0.

Return Value:

TRUE : User archive has been exported successfully

FALSE : Error

uaArchiveGetCount**Description:**

Reads the number of data records.

```
LONG uaArchiveGetCount(
    UAHARCHIVE hArchive,
    LONG * plCount )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG plCount

Pointer to a tag in which the number of data records should be stored.

Return Value:

Number of data records.

0 = Archive is empty or fault has occurred. The query must be made using `uaGetLastError()`.

uaArchiveGetFieldLength

Description:

Reads the length of a field in the current data record.

```
LONG uaArchiveGetFieldLength(  
    UAHARCHIVE hArchive,  
    LONG lField )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG lField

The field number, where lField = 1 is addressing the first field.

Return Value:

Length of the current field.

uaArchiveGetFieldName

Description:

Reads the name of a field in the current data record.

```
VOID uaArchiveGetFieldName (  
    UAHARCHIVE hArchive,  
    LONG lField,  
    LPCSTR pszName,  
    LONG cMaxLen )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG lField

The field number, where lField = 1 is addressing the first field.

LPCSTR pszName

Field Name

LONG cMaxLen

Maximum Length

uaArchiveGetFields**Description:**

Reads the number of configured data fields, which also includes the "ID", "Last User" and "Last Access" fields. In the Runtime calls, the indexes of the configured fields are indicated 1 to N. The field ID has the index 0. The fields "Last User" and "Last Access" are appended to the end of the configured fields.

```
LONG uaArchiveGetFields (
    UAHARCHIVE hArchive )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

Return Value:

Number of the configured fields.

uaArchiveGetFieldType**Description:**

Reads the type of a field in the current data record.

```
LONG uaArchiveGetFieldType (
    UAHARCHIVE hArchive,
    LONG lField )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG lField

The field number, where lField = 1 is addressing the first field.

Return Value:

Type of the current field.

The symbolic definitions of the field types are:

UA_FIELDTYPE_INTEGER

UA_FIELDTYPE_DOUBLE

UA_FIELDTYPE_STRING

UA_FIELDTYPE_DATETIME

uaArchiveGetFieldValueDate

Description:

Reads the date and time of a field in the current data record.

```
BOOL uaArchiveGetFieldValueDate (
    UAHARCHIVE hArchive,
    LONG lField,
    LPSYSTEMTIME pstDateTime )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG lField

The field number, where lField = 1 is addressing the first field.

LPSYSTEMTIME pstDateTime

Pointer on the tag of the SYSTEMTIME type.

Return Value:

TRUE : Successful reading of date and time

FALSE : Error

uaArchiveGetFieldValueDouble

Description:

Reads the Double value of a field in the current data record.

```
BOOL uaArchiveGetFieldValueDouble (
    UAHARCHIVE hArchive,
    LONG lField,
    double* pdValue )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG lField

The field number, where lField = 1 is addressing the first field.

double* pdValue

Pointer on the tag of the current field content.

Return Value:

TRUE : Successful reading of the field value

FALSE : Error

uaArchiveGetFieldValueFloat

Description:

Reads the floating value of a field in the current data record.

```
BOOL uaArchiveGetFieldValueFloat (
    UAHARCHIVE hArchive,
    LONG lField,
    FLOAT* pfValue )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG lField

The field number, where lField = 1 is addressing the first field.

Float* pfValue

Pointer on the Float tag of the current field content.

Return Value:

TRUE : Successful reading of the field value

FALSE : Error

uaArchiveGetFieldValueLong

Description:

Reads the Long Integer value of a field in the current data record.

```
BOOL uaArchiveGetFieldValueLong (
    UAHARCHIVE hArchive,
    LONG lField,
    LONG* pdValue )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG lField

The field number, where lField = 1 is addressing the first field.

LONG* pdValue

Pointer on the Long tag of the current field content.

Return Value:

TRUE : Successful reading of the field value

FALSE : Error

uaArchiveGetFieldValueString

Description:

Reads the string of a field in the current data record.

```
BOOL uaArchiveGetFieldValueString (
```



```
UAHARCHIVE hArchive,  
LONG lField,  
LPSTR pszString,  
LONG cMaxLen )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG lField

The field number, where lField = 1 is addressing the first field.

LPCSTR pszString

Field value as string.

LONG cMaxLen

Maximum length of the string.

Return Value:

TRUE : Successful reading of the field value

FALSE : Error

uaArchiveGetFilter**Description:**

Reads the filter of the current data record. Additional information can be found in the appendix under "SQL Statements".

```
VOID uaArchiveGetFilter (  
    UAHARCHIVE hArchive,  
    LPSTR pszFilter,  
    LONG cMaxLen )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LPSTR pszFilter

Read filter.

LONG cMaxLen

Maximum Length

uaArchiveGetID

Description:

uaArchiveGetID reads the ID of the user archive.

```
LONG uaArchiveGetID (  
    UAHARCHIVE hArchive )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

Return Value:

ID of the User Archive

uaArchiveGetName

Description:

Reads the name of the user archive.

```
VOID uaArchiveGetName (  
    UAHARCHIVE hArchive,  
    LPSTR pszName,  
    LONG cMaxLen )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LPSTR pszName

Pointer on the buffer for the user archive name.

LONG cMaxLen

Maximum Length

Example:

```
char Filling [40];
uaArchiveGetName( hArchive, Filling, 39 );
```

uaArchiveGetSort**Description:**

uaArchiveGetSort reads the sorting of the user archive. Additional information can be found in the appendix under "SQL Statements".

```
VOID uaArchiveGetSort (
    UAHARCHIVE hArchive,
    LPSTR pszSort,
    LONG cMaxLen )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LPCSTR pszSort

Sort ...

LONG cMaxLen

Maximum Length

uaArchiveImport**Description:**

uaArchiveImport imports a user archive using the CSV data format. The structure of the target archive must be identical to the imported CSV archive.

```
BOOL uaArchiveImport (
    UAHARCHIVE hArchive,
    LPCSTR pszSource,
    LONG lType,
    LONG lOptions )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LPCSTR pszSource

File name of the source archive.

LONG lType

Data format of the source archive. Two formats are available:

UA_FILETYPE_DEFAULT = 0: Default file format = CSV

UA_FILETYPE_CSV = 1: CSV file format

LONG lOptions

Reserved for later expansions. Must be 0.

Return Value:

TRUE : User archive has been imported successfully

FALSE : Error

uaArchiveInsert

Description:

Inserts the local data record buffer into the current database. To have useful data in the new data record, the fields of the local data record buffer must be written with the "uaArchiveSetFieldValue..." functions before calling "uaArchiveInsert".

The internal column "ID" must be written into the current data set with the function "uaArchiveSetFieldValueLong".

```
BOOL uaArchiveInsert (
    UAHARCHIVE hArchive )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

Return Value:

TRUE : Successful insertion of the data record

uaArchiveMoveFirst

Description:

Goes to the first data record.

```
BOOL uaArchiveMoveFirst (  
    UAHARCHIVE hArchive )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

Return Value:

TRUE : Successful movement in the user archive

FALSE : Error

uaArchiveMoveLast

Description:

Goes to the last data record.

```
BOOL uaArchiveMoveLast (  
    UAHARCHIVE hArchive )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

Return Value:

TRUE : Successful movement in the user archive

FALSE : Error

uaArchiveMoveNext

Description:

Goes to the next data record.

```
BOOL uaArchiveMoveNext (
    UAHARCHIVE hArchive )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

Return Value:

TRUE : Successful movement in the user archive
FALSE : Error

uaArchiveMovePrevious

Description:

Goes to the previous data record.

```
BOOL uaArchiveMovePrevious (
    UAHARCHIVE hArchive )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

Return Value:

TRUE : Successful movement in the user archive
FALSE : Error

uaArchiveOpen

Description:

uaArchiveOpen opens an existing user archive. uaArchiveOpen must always be called when the read or write operation must be executed in connection with this user archive. uaArchiveOpen must, for example, be called before calling the uaArchiveMoveFirst, uaArchiveMoveLast, uaArchiveMoveNext, uaArchiveMovePrevious, uaArchiveDelete, uaArchiveUpdate, uaArchiveInsert functions.

Note**Sort and filter user archives**

You can use the "uaArchiveSetSort" and "uaArchiveSetFilter" functions on a user archive without opening the user archive using "uaArchiveOpen".

```
BOOL uaArchiveOpen (
    UAHARCHIVE hArchive )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

Return Value:

TRUE : User archive has been opened successfully

FALSE : Error

uaArchiveReadTagValues**Description:**

Reads the current value from the field tag.

```
BOOL uaArchiveReadTagValues (
    UAHARCHIVE hArchive,
    LONG* pnFields,
    LONG cFields,
    LONG lOptions )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG* pnFields

Reserved for future applications (NULL).

LONG cFields

Number of field indices transferred (size of array pnFields).

Reserved for future applications (0).

LONG lOptions

Reserved for future applications (0).

In the case of all other values of lOptions, the data is inserted at the position of the pointer.

Return Value:

TRUE : Successful reading in the user archive

FALSE : Error

uaArchiveReadTagValuesByName

Description:

Reads the tag values in the current data.

```
BOOL uaArchiveReadTagValuesByName (
    UAHARCHIVE hArchive,
    LPCSTR pszFields,
    LONG lOptions )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LPCSTR pszFields

Reserved for future applications (NULL).

LONG lOptions

Reserved for future applications (0).

Return Value:

TRUE : Successful reading in the user archive

FALSE : Error

uaArchiveRequery

Description:

After calling uaArchiveSetFilter and uaArchiveSetSort, the archive must be reloaded with uaArchiveRequery.

Note**Sort and filter user archives**

You can use the "uaArchiveSetSort" and "uaArchiveSetFilter" functions on a user archive without opening the user archive using "uaArchiveOpen". In case, do not call the "uaArchiveRequery" function.

Call up the uaArchiveRequery in the following cases:

- If you have made entries via the User Archive Table Control.
- If you have made entries in the Editor User Archives that are to be transferred to the table field.

```
BOOL uaArchiveRequery(  
    UAHARCHIVE hArchive )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

Return Value:

TRUE : Successful Requery

FALSE : Error

uaArchiveSetFieldValueDate**Description:**

Writes the date and time into a field of the current data record.

```
BOOL uaArchiveSetFieldValueDate (  
    UAHARCHIVE hArchive,  
    LONG lField,  
    LPSYSTEMTIME pstDateTime )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG lField

The field number, where IField = 1 is addressing the first configured field. IField = 0 addresses the ID field.

LPSYSTEMTIME pstDateTime

Date and Time

Return Value:

TRUE : Successful writing of date and time

FALSE : Error

uaArchiveSetFieldValueDouble

Description:

Writes a Double value into a field of the current data record.

```
BOOL uaArchiveSetFieldValueDouble (
    UAHARCHIVE hArchive,
    LONG lField,
    double dValue )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG lField

The field number, where IField = 1 is addressing the first configured field. IField = 0 addresses the ID field.

double dValue

Field Value

Return Value:

TRUE : Successful writing of the field value

FALSE : Error

uaArchiveSetFieldValueFloat

Description:

Writes a Float value into a field of the current data record.

```
BOOL uaArchiveSetFieldValueFloat (
    UAHARCHIVE hArchive,
    LONG lField,
    float fValue )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG lField

The field number, where lField = 1 is addressing the first configured field. lField = 0 addresses the ID field.

float fValue

Field Value

Return Value:

TRUE : Successful writing of the field value

FALSE : Error

uaArchiveSetFieldValueLong**Description:**

Writes a Long Integer value into a field of the current data record.

```
BOOL uaArchiveSetFieldValueLong (
    UAHARCHIVE hArchive,
    LONG lField,
    LONG dValue )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG lField

The field number, where lField = 1 is addressing the first configured field. lField = 0 addresses the ID field.

LONG dValue

Field Value

Return Value:

TRUE : Successful writing of the field value

FALSE : Error

uaArchiveSetFieldValueString

Description:

Writes a String into a field of the current data record.

```
BOOL uaArchiveSetFieldValueString (
    UAHARCHIVE hArchive,
    LONG lField,
    LPCSTR pszString )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG lField

The field number, where lField = 1 is addressing the first configured field. lField = 0 addresses the ID field.

LPCSTR pszString

Field Value

Return Value:

TRUE : Successful writing of the field value

FALSE : Error

uaArchiveSetFilter

Description:

Sets the filter. You can call up the function without opening the archive using "uaArchiveOpen".

Note

If you have opened the user archive using "uaArchiveOpen", then reload the user archive after filtering using "uaArchiveRequery". Additional information can be found in the appendix under "SQL Statements".

```
VOID uaArchiveSetFilter (
    UAHARCHIVE hArchive,
    LPSTR pszFilter )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LPSTR pszFilter

Filter to be set.

uaArchiveSetSort**Description:**

Sets the user archive's sorting. You can call up the function without opening the archive using "uaArchiveOpen".

Note

If you have opened the user archive using "uaArchiveOpen", then reload the user archive after sorting using "uaArchiveRequery". Additional information can be found in the appendix under "SQL Statements".

```
BOOL uaArchiveSetSort (
    UAHARCHIVE hArchive,
    LPSTR pszSort )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LPCSTR pszSort

Sort ...

Return Value:

TRUE : Successful setting of the sorting

FALSE : Error

uaArchiveUpdate

Description:

Updates the user archive which is open. All data changes of a user archive are incorporated into the database. The configuration of the user archive remains unchanged.

```
BOOL uaArchiveUpdate (  
    UAHARCHIVE hArchive )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

Return Value:

TRUE : User archive has been updated successfully

FALSE : Error "Update_failed" = 106

This error occurs at a consistency violation, for example:

The "Field must contain a Value" flag is set in a field,

but no value has been placed in this field.

uaArchiveWriteTagValues

Description:

Writes the values of the current data record into the tags.

```
BOOL uaArchiveWriteTagValues (  
    UAHARCHIVE hArchive,  
    LONG* pnFields,  
    LONG cFields,  
    LONG lOptions )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LONG* pnFields

Reserved for future applications (NULL).

LONG cFields

Reserved for future applications (0).

LONG lOptions

Reserved for future applications (0).

Return Value:

TRUE : Successful reading in the user archive

FALSE : Error

uaArchiveWriteTagValuesByName**Description:**

Writes the values of the current data record into the tags. The access is based on the names of the user archive and field.

```
BOOL uaArchiveWriteTagValuesByName (
    UAHARCHIVE hArchive,
    LPCSTR pszFields,
    LONG lOptions )
```

Parameters:

UAHARCHIVE hArchive

Handle of the User Archive. This handle is generated with "uaQueryArchive" or "uaQueryArchiveByName".

LPCSTR pszFields

Reserved for future applications (NULL).

LONG lOptions

Reserved for future applications (0).

Return Value:

TRUE : Successful reading in the user archive

FALSE : Error

2.4 WinCC UserArchiveControl

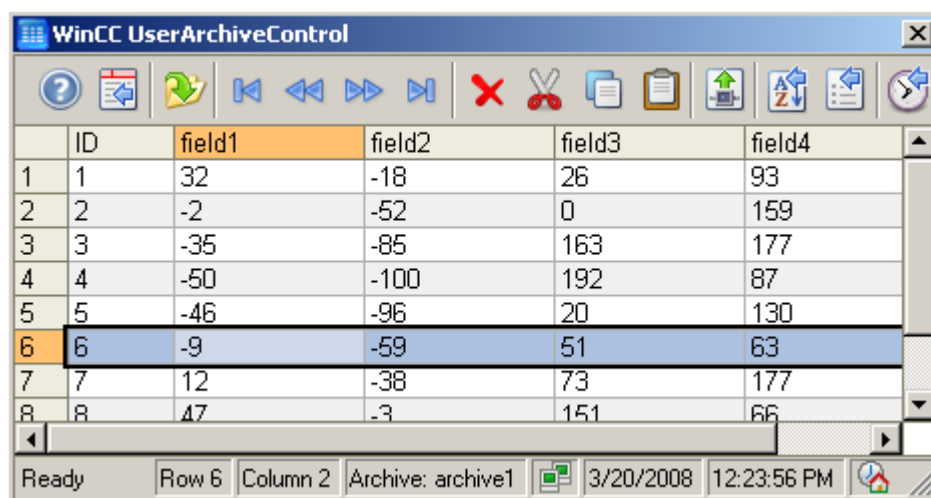
2.4 Resources

2.4.1 WinCC UserArchiveControl

Functional scope

The WinCC UserArchiveControl provides access capability to archives and views of user archives. In runtime, you can:

- Create, delete or modify new data records
- Browsing in user archives
- Read and write tags for direct tag link
- Import and export user archive
- Define selection criteria to display a certain portion of the user archive only
- Define sorting conditions for the displayed columns



	ID	field1	field2	field3	field4
1	1	32	-18	26	93
2	2	-2	-52	0	159
3	3	-35	-85	163	177
4	4	-50	-100	192	87
5	5	-46	-96	20	130
6	6	-9	-59	51	63
7	7	12	-38	73	177
8	8	47	-3	151	66

Ready | Row 6 | Column 2 | Archive: archive1 | 3/20/2008 12:23:56 PM

Properties

You connect a UserArchiveControl with a selected user archive or a view in the configuration. For access, the user archive or the view must be enabled. If you remove the access protection, you must connect the WinCC UserArchiveControl with the user archive again in the configuration dialog.

Access protection for a user archive or field is queried on opening a screen of the UserArchiveControl:

- If the user has no authorization for reading the user archive, no data is shown but the column headings in the table are shown.
- If the user has no authorization for reading a field, the respective column is not shown in the table.
- If the user has no authorization for writing in the user archive, the user cannot edit the data in the table.
- If the user has no authorization for writing in a field, the user cannot edit the respective column in the table.

Access protection for the control tag of a protective archive must be configured separately with the object properties e.g. a picture, an IO field or a button.

2.4.2 Configuring the UserArchiveControl

2.4.2.1 How to configure the UserArchiveControl

Introduction

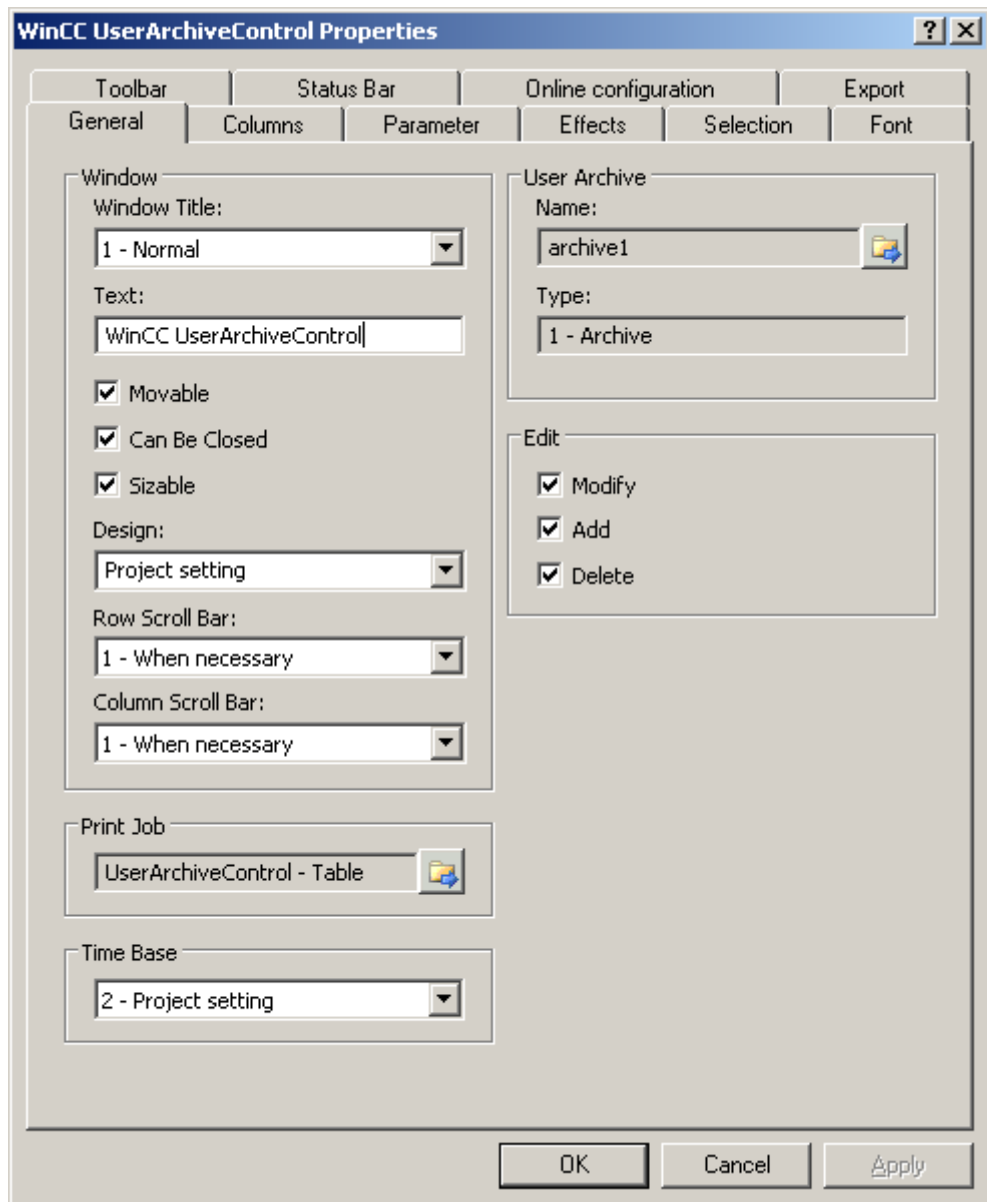
The values of the user archive are shown in runtime in an ActiveX control. You can configure a WinCC UserArchiveControl for this in the Graphics Designer.

Requirements

- You have configured an archive or a view in the user archive.

Configuration steps

1. Link the WinCC UserArchiveControl to a Graphics Designer picture.
2. Configure the basic properties for the UserArchiveControl on the "General" tab.
 - The table window properties
 - The general properties of the control
 - The time base of the control
 - The editing capabilities of the contents in control



3. Connect the UserArchiveControl with an archive or a view of the user archive.
4. Define the content of the table of the UserArchiveControl, in which you configure the selected columns from the user archive.
5. Configure the display and properties of the table on the "Parameter", "Display" and "Marker" tabs.
6. Configure the toolbar and the status bar of the table window on the respective tabs
7. Save the configuration.

2.4.2.2 How to define the contents of the UserArchiveControl

Introduction

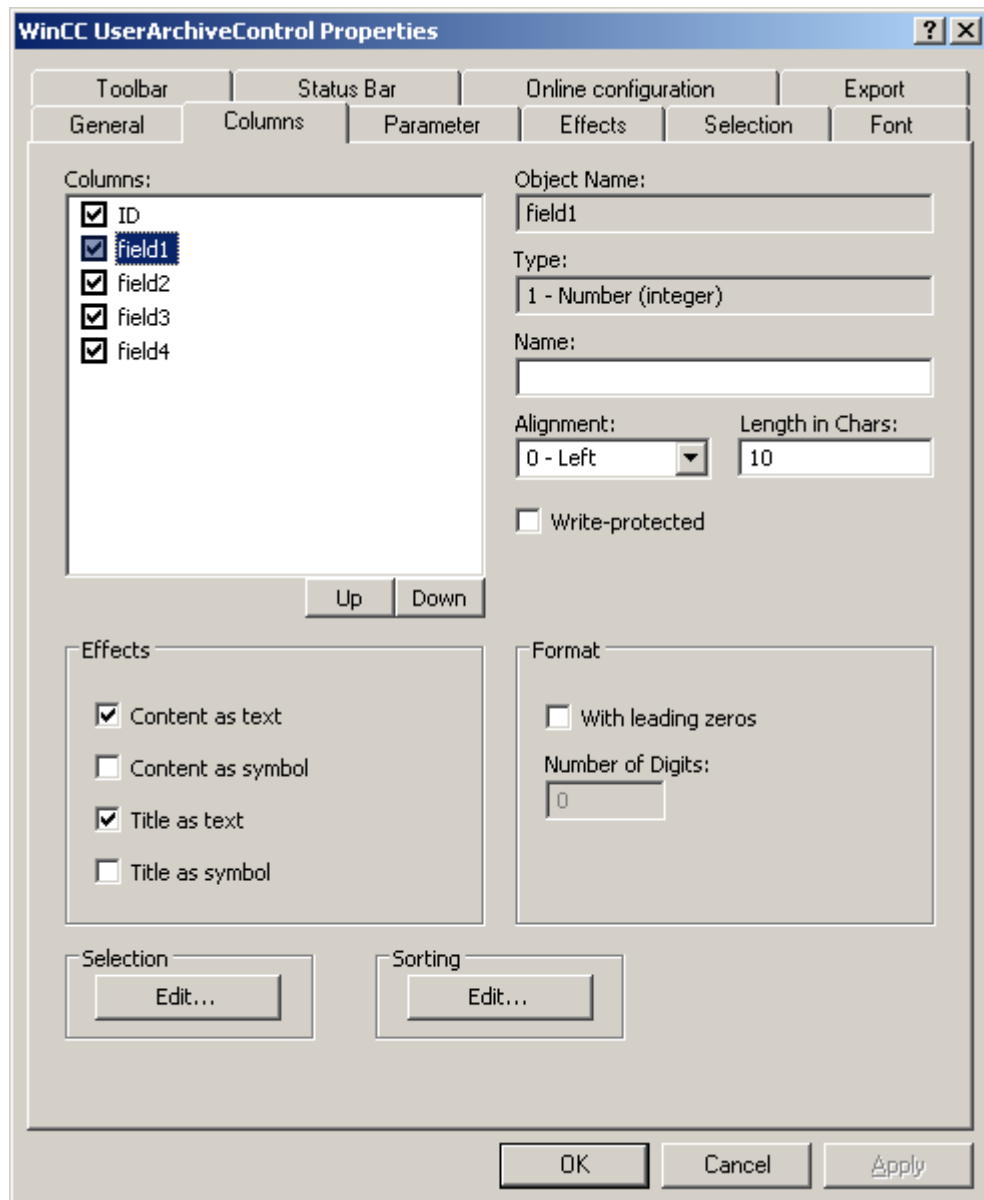
The WinCC UserArchiveControl shows the data of the connected user archive in a table. The displayed content of the table is determined by the selected columns of the user archive and the selection of column content.

Requirements

- You have created one or more user archives or views.
- You have connected the UserArchiveControl with a user archive or a view.

Configures the columns of the user archive.

1. Go to the "Columns" tab.



2. In the "Columns" list, you can see the fields of the connected archive or view. If the box in front of the name of the column is checked, the column is displayed in the table. Deactivate the check box if you do not want it to appear.
3. Determine the order of the columns in the table using the "Up" and "Down" buttons.
4. Select a column to configure the properties and the format.
5. If necessary, change the width of the column in the table. Enter a value in the "Length in chars" field.

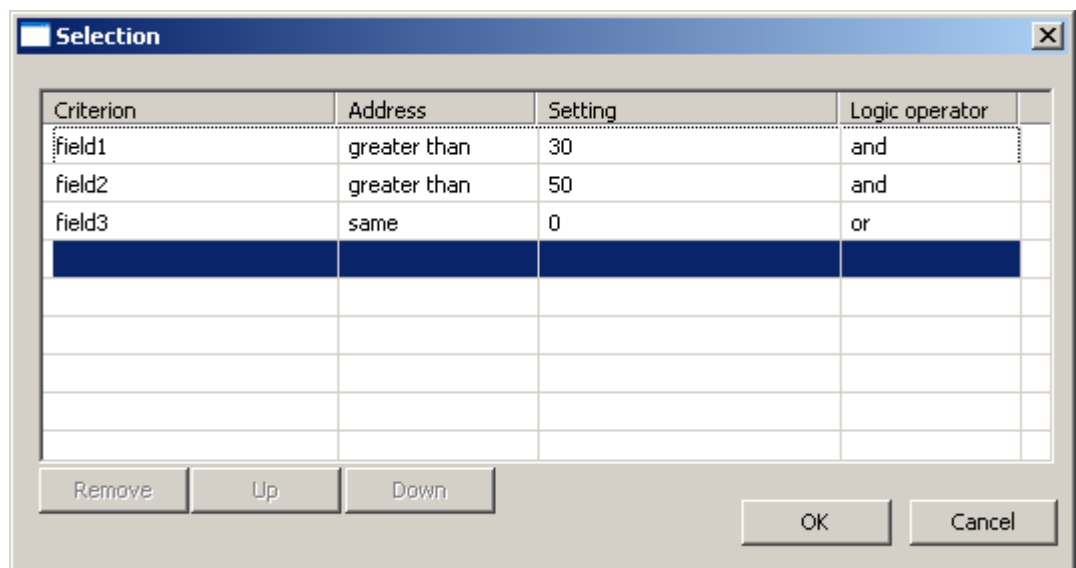
6. Some columns can also show the content and the header as a symbol. Determine how these columns are displayed in the "Display" field. Text and symbols can be displayed at the same time.
7. Save the configuration.

Selection of column content that will be displayed in the table

Configure criteria used to display the content in the columns in the "Selection" area.

Procedure

1. Click "Edit...". The selection dialog is opened.

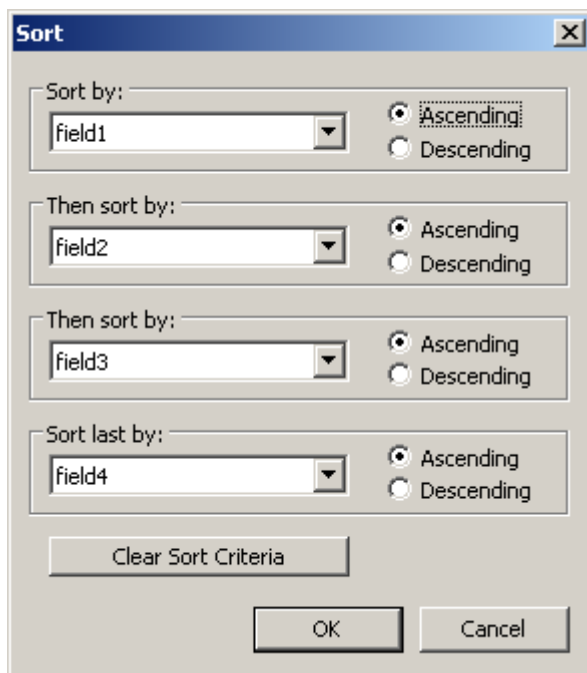


2. Specify the criteria for the display. More information on the selection of columns can be found under .
3. Click "OK" to close the selection dialog. The selection is applied at Runtime start in the table of the UserArchiveControl.

Configuring the Sorting of Columns

You configure the sorting of the columns in the table of the UserArchiveControl in the "Sorting" area. You can also specify the sorting criteria in Runtime using the key functions.

1. Click "Edit...". The sorting dialog opens.



2. Set a sorting sequence. More information on the sorting of columns can be found under .
3. Click "OK" to close the sorting dialog.
4. Save the configuration of the content of the UserArchiveControl.

2.4.2.3 How to configure the display for the table

How to configure the properties of the table elements

Introduction

You can adjust the properties of the table elements in the WinCC controls to suit requirements.

Prerequisite

- You opened Graphics Designer and configured a picture with a WinCC control.
- The configuration dialog of the WinCC control is open.

Procedure

1. Go to the "Parameters" tab.

The screenshot shows the 'Parameters' tab of the WinCC UserArchiveControl configuration dialog. It is divided into four sections: 'Column Header', 'Sort', 'Row Label', and 'Table Content'. In the 'Column Header' section, 'Show' is checked, 'Shorten content' is unchecked, 'Width can be changed' is checked, and 'Alignment' is set to '0 - Left'. In the 'Sort' section, 'Sort by column header' is set to '2 - With double-click', 'Sorting order on mouse click' is set to '0 - Up/down/none', 'Show sorting symbol' is checked, 'Show sorting index' is checked, and 'Use sorting key' is unchecked. In the 'Row Label' section, 'Show' is checked and 'Alignment' is set to '0 - Left'. In the 'Table Content' section, 'Show empty columns' is checked, 'Show empty rows' is checked, and 'Shorten content' is unchecked.

2. Specify the properties for of
 - Column header
 - Line labeling
 - Sorting
 - Table content
3. Save the configuration.

How to configure the colors of the table elements

Introduction

You can adjust the colors of the table elements in the WinCC controls to suit requirements.

Prerequisite

- You opened Graphics Designer and configured a picture with a WinCC control.
- The configuration dialog of the WinCC control is open.

Procedure

1. Go to the "Display" tab.

The screenshot shows the 'Display' configuration tab for the WinCC UserArchiveControl. It is organized into four panels:

- Table Content:**
 - Row Color 1:** Background (white) and Font (black).
 - Row color 2:** Checked checkbox. Background (light gray) and Font (black).
 - Background:** Color (white).
- Table Title:**
 - Font color of the title:** Background (light beige) and Font (black).
 - Shading color:** Unchecked checkbox. Light (white) and Dark (dark gray).
 - Selection color:** Checked checkbox. Background (orange) and Font (black).
- Dividing Lines:**
 - Color of dividing line:** Content (gray) and Title (dark gray).
 - Weight:** Set to 1.
 - Horizontal:** Checked checkbox.
 - Vertical:** Checked checkbox.
- General:**
 - Border:** Color (black) and Weight (0).
 - Window Dividing Lines:** Color (dark gray) and Weight (1).

2. Define the colors for the background or text here for:
 - Table content. You can define different colors for even and odd line numbers to improve differentiation between both.
 - Contents of the table header
 - Separating lines in the table and for table headers
3. Define the color and the line weight in the "General" area in terms of:
 - Control borders
 - Window dividers for control elements
4. Save the configuration.

How to configure the marking of the selected cells and lines

Introduction

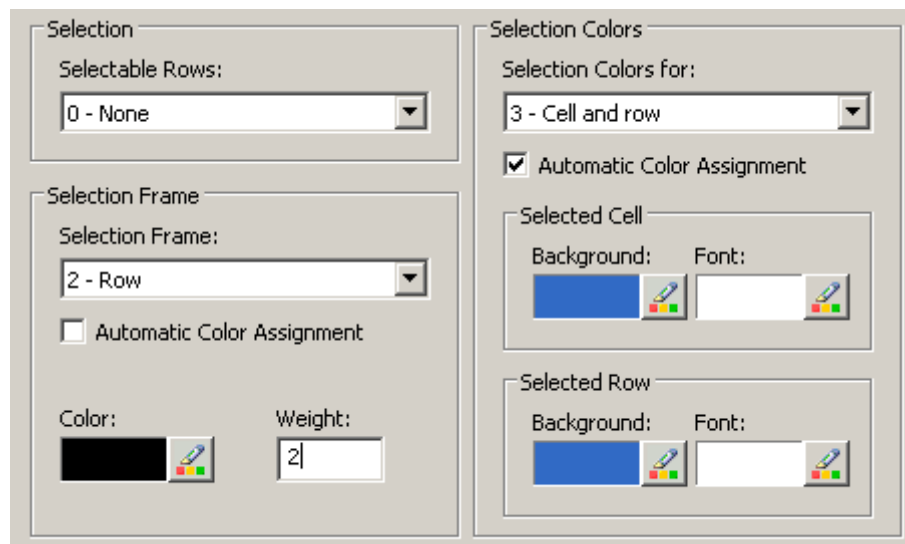
You can customize the marking of the selected cells and rows in the WinCC control to suit requirements.

Prerequisite

- You opened Graphics Designer and configured a picture with a WinCC control.
- The configuration dialog of the WinCC control is open.

Procedure

1. Go to the "Marker" tab.



2. Define whether to select rows or only cells using the mouse.
3. Configure the properties of the selection rectangle that can be displayed around selected table cells or rows.
4. Configure the marking color for selectable cells and/or rows. The system colors are used for marking with "Automatic coloring" property.
5. Save the configuration.

How to configure sorting via the column heading

Introduction

You can adjust the sorting order by means of table column header in the WinCC controls to suit requirements.

Prerequisite

- You opened Graphics Designer and configured a picture with a WinCC control.
- The configuration dialog of the WinCC control is open.

Procedure

1. Go to the "Parameters" tab.

The screenshot shows the 'Parameters' tab of the WinCC UserArchiveControl configuration dialog. It is divided into four sections: 'Column Header', 'Sort', 'Row Label', and 'Table Content'.
- **Column Header:** Contains checkboxes for 'Show' (checked), 'Shorten content' (unchecked), and 'Width can be changed' (checked). Below these is an 'Alignment:' dropdown menu set to '0 - Left'.
- **Sort:** Contains a 'Sort by column header:' dropdown set to '2 - With double-click'. Below it is a 'Sorting order on mouse click:' dropdown set to '0 - Up/down/none'. At the bottom are checkboxes for 'Show sorting symbol' (checked), 'Show sorting index' (checked), and 'Use sorting key' (unchecked).
- **Row Label:** Contains a 'Show' checkbox (checked) and an 'Alignment:' dropdown set to '0 - Left'.
- **Table Content:** Contains checkboxes for 'Show empty columns' (checked), 'Show empty rows' (checked), and 'Shorten content' (unchecked).

2. Define whether to enable sorting and the sorting method by column header. In WinCC AlarmControl, you can only sort by column header if the "Auto-Scrolling" is disabled. You can deactivate "Auto-Scrolling" either in the "General" tab, or using the "Autoscroll" toolbar icon of the WinCC AlarmControl.
3. Determine the sorting order by mouse click on the column header. Select ascending, descending or no sorting order.
4. Configure the sorting icon and index to be displayed in the column header with right justification. These show the sorting order and sequence of the columns.
5. Activate the "Use sorting button" to display the sorting icon as sorting button above the vertical scroll bar. Click this sorting button to activate a configured sorting order for the column selected. The sort button is not displayed if a vertical scroll bar is missing.
6. Save the configuration.

2.4.2.4 How to configure the toolbar and the status bar

Introduction

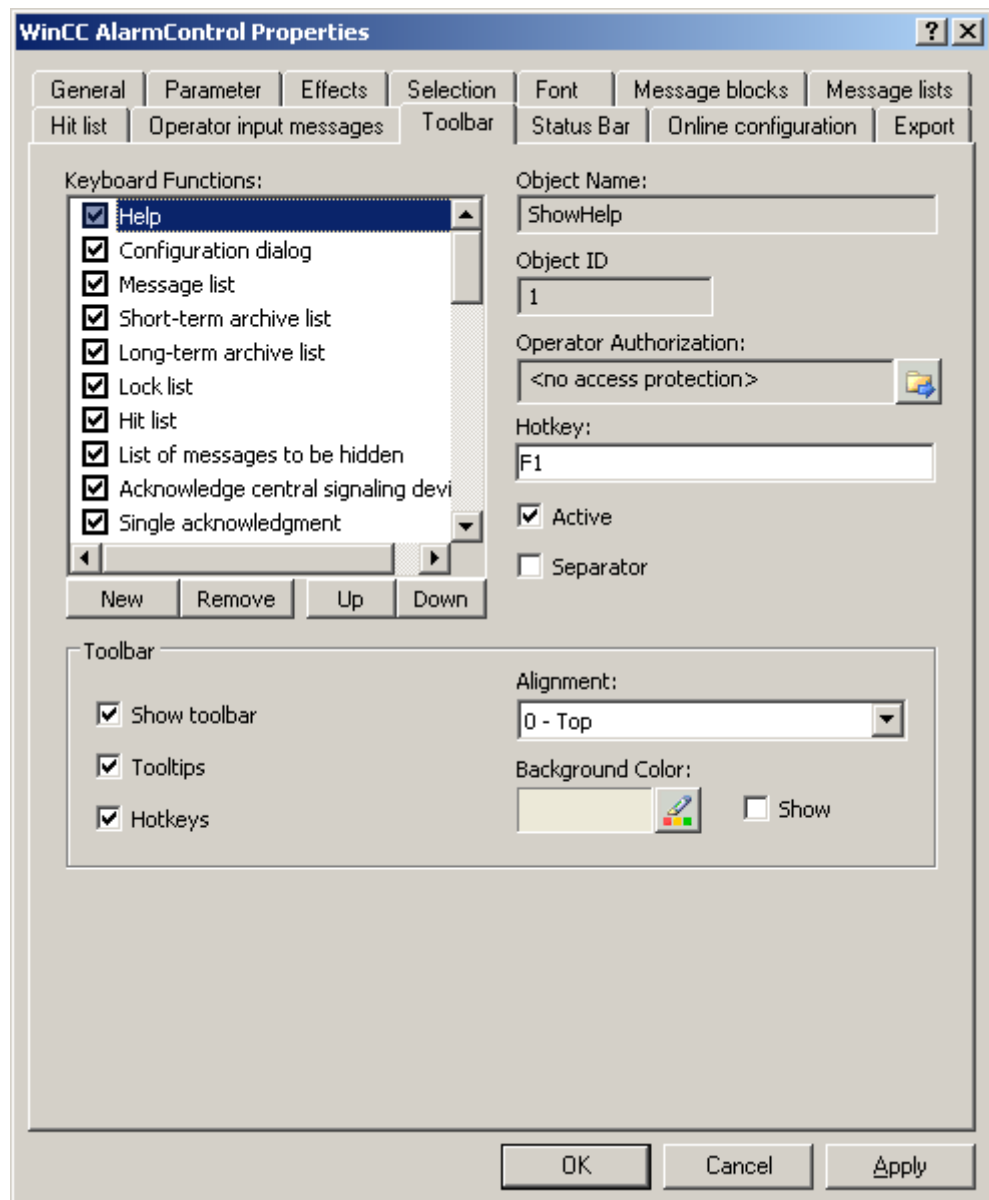
The WinCC controls are operated at runtime using the functions of the toolbar buttons. The status bar contains information pertaining to the current status of the WinCC control. You can adapt the toolbar and the status bar for all WinCC controls when configuring, or at runtime.

Prerequisite

- A picture showing a WinCC control is opened in Graphics Designer when configuring.
- The WinCC control is assigned the "Configuration dialog" button function for opening the configuration dialog in Runtime.
- The configuration dialog of the WinCC control is open.

How to configure the toolbar

1. Go to the "Toolbar" tab. In the WinCC AlarmControl, for example:

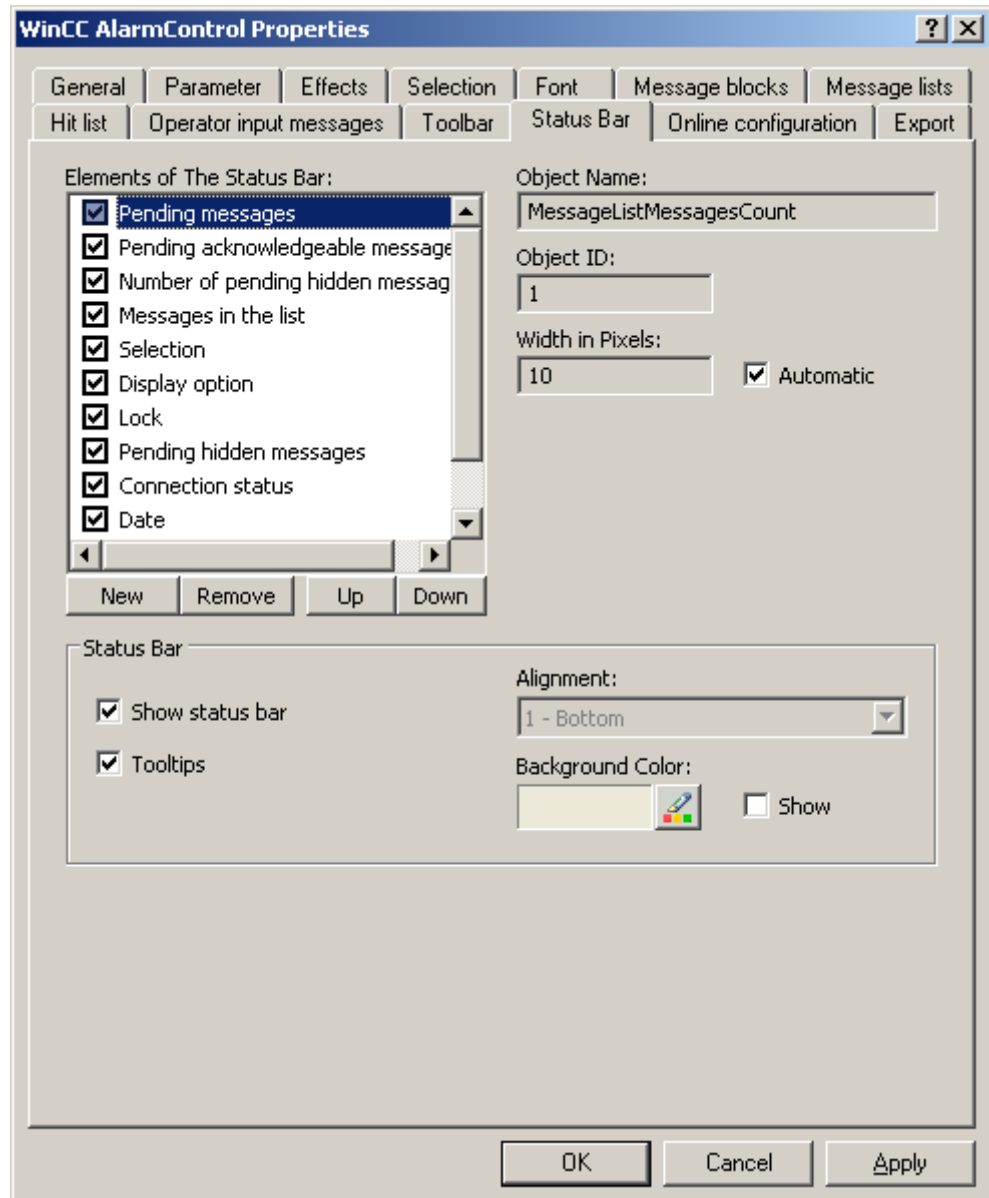


2. In the list, activate the button functions you require for operating the WinCC control in Runtime. For information on the button functions, refer to the description of the corresponding WinCC control at "Operation in runtime".

3. Determine the sorting order for displaying the button functions in the toolbar. Select the button functions from the list and move the functions using the "Up" and "Down" buttons.
4. Define a hotkey for the functions of the toolbar buttons.
5. Any button functions assigned operator authorizations are only available in Runtime to authorized users.
6. An activated button function is displayed in runtime if you deactivate its "Active" option, however, it cannot be operated.
7. You can set separators between the button functions. Activate the "Separator" option for the button function to be restricted by separator.
8. Configure the general properties of the toolbar, e.g. alignment or background color.

How to configure the status bar

1. Go to the "Status bar" tab. In the WinCC AlarmControl, for example:



2. Activate the elements required in runtime in the list of status bar elements. For further information on status bar elements, refer to the descriptions of the corresponding WinCC control at "Operation in runtime".
3. Determine the sorting order for displaying of the status bar elements. Select the elements from the list and move these using the "Up" and "Down" buttons.
4. To resize the width of a status bar element, deactivate the "Automatic" option and enter a pixel value for the width.
5. Configure the general properties of the status bar, e.g. alignment or background color.

2.4.2.5 How to export runtime data

Introduction

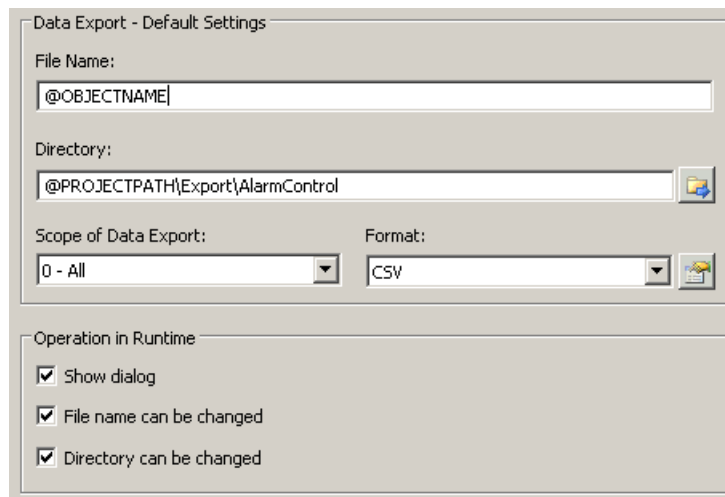
The runtime data shown in the WinCC controls can be exported using a button function. Set up operation of the data export in runtime in the configuration dialog.

Requirements

- A picture showing a WinCC control is opened in Graphics Designer when configuring.
- The configuration dialog of the WinCC control is open.

How to configure the operation of the data export

1. Go to the "Export" tab.



2. A standard file name and a standard directory are already entered in the "Data export default settings." In this case for AlarmControl. If necessary, define a file name and a directory for the export file.
3. CSV is currently available as data format. Click on  to define the separator in the CSV file.
4. Define the scope of the data export:
 - All runtime data is exported
 - Selected runtime data is exported. This data export is only possible in WinCC controls with tabular display.
5. Configure the operation of the data export in runtime. Define:
 - whether users are allowed to rename the file, or change the directory.
 - whether to display the "Data export default settings" dialog in Runtime.
6. If "Display dialog" is deactivated, the data for operation of the "Export data" button function is immediately exported to the defined export file.

7. Save the configuration.
8. Go to the "Toolbar" tab to activate the "Export data" button function for runtime.

Results

You can export all or selected data to a defined file at runtime using button function .

2.4.2.6 How to define the effect of the online configuration

Introduction

Users can parameterize the WinCC controls in Runtime. You must define the runtime effects of the online configuration.

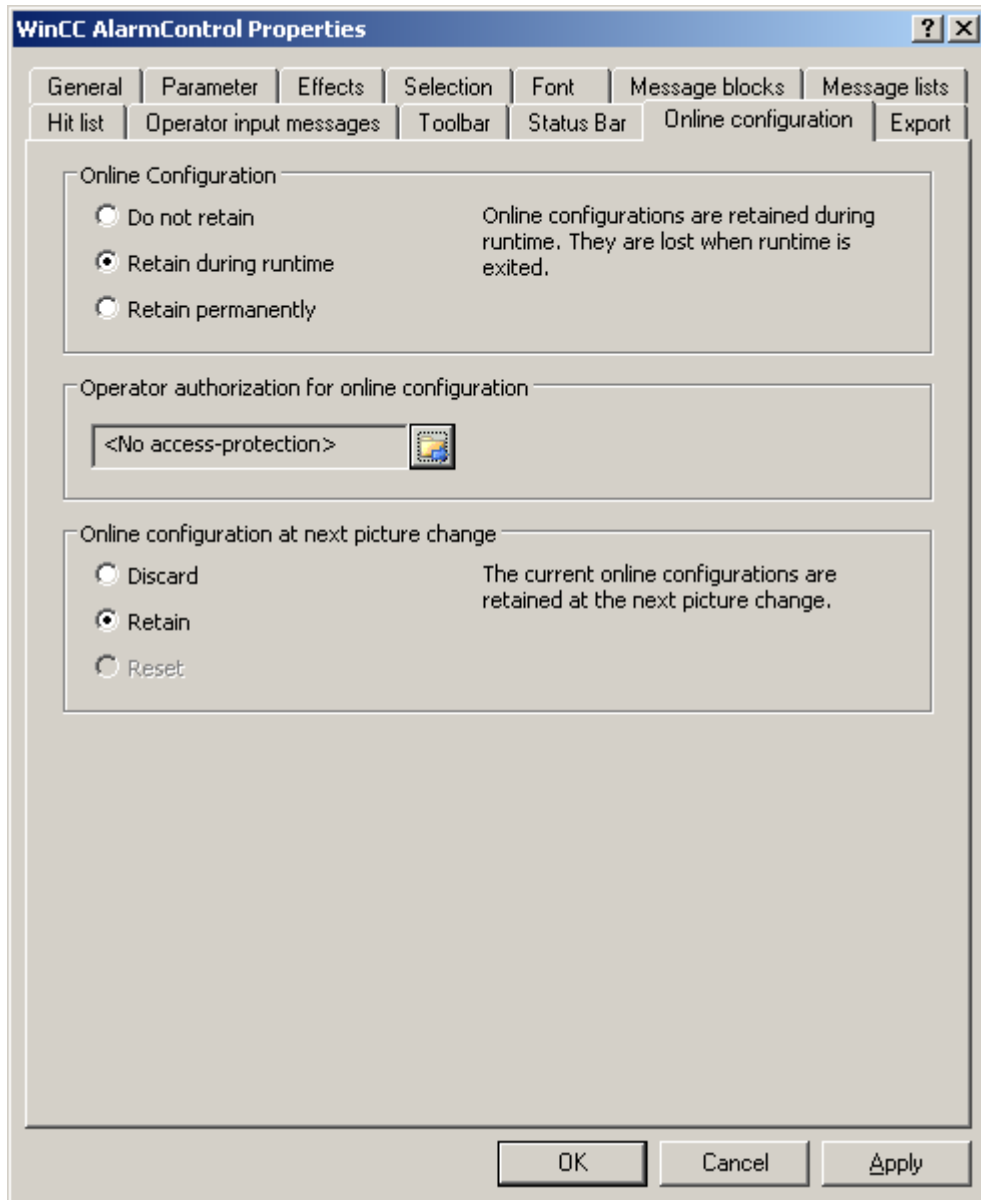
Changes configured at runtime are saved to a separate picture in the configuration system. The original picture configuration is retained in the configuration system.

Prerequisite

- A picture showing a WinCC control is opened in Graphics Designer when configuring.
- The configuration dialog of the WinCC control is open.

Procedure

1. Go to the "Online configuration" tab. For example, in OnlineTrendControl:



2. The option buttons of the "Online configuration" field for setting online configuration defaults are only available in the configuration system. The option buttons are not available in runtime.

Select one of the three effects of the online configuration:

- "Do not retain". The online configurations are not retained in Runtime. This default setting disables all options for runtime users. Online configurations are lost at the next picture change and on activation/deactivation of the project.
- "Retain during runtime". This default setting enables the "discard", "retain" or "reset" options for runtime users. Online configurations are retained at the next picture change

if the "retain" option is enabled, however, these are lost on activation/deactivation of the project.

- "Retain permanently". This default setting enables the "discard", "retain" or "reset" options for runtime users. Online configurations are retained at the next picture change and on activation/deactivation of the project if the "retain" option is enabled.

3. Define corresponding user authorizations for online configuration.

4. The option buttons of the "Online configuration on next picture change" can be enabled for operation in the configuration system and at runtime by setting the "retain at runtime" and "retain permanently" defaults. The "reset" operation is only available in runtime, because the configuration system contains the original configuration.

Select one of three effects of the online configuration at the next picture change:

- Select "discard" if to discard the online configuration at the next picture change.
- Activate "retain" to activate the online configuration based on default settings at the next picture change or on activation/deactivation of the project.
- Activate "reset" to activate a picture which is older than the version stored in the configuration system for runtime. All online changes are lost.

Note

The picture is also replaced at runtime if you save it in Graphics Designer, or when loading deltas in online mode. All online changes are lost.

5. Save the configuration.

Note

The different configurations are only activated for new users after you performed a picture change.

2.4.2.7 How to make the toolbar for the UserArchiveControl dynamic

Introduction

The default functions for operating the WinCC UserArchiveControl are no longer supported for the new WinCC UserArchiveControl as of WinCC V7.0. You can use the WinCC types of dynamics to e.g. operate a key function of the toolbar with a script.

Overview

With WinCC controls as of V7.0, special functions are no longer required for operating the control with toolbar dynamics. The previously used standard functions "Tlg..." are no longer supported.

If you do not want to operate the control via the toolbar, you can write the "ID" for the desired button in the "ToolbarButtonClick" object property with an optional type of dynamics.

The "ID" of a button of the toolbar can be determined:

- with the table on page "Operation of the UserArchiveControl in runtime".
- in the configuration dialog of the UserArchiveControl on the "Toolbar" tab via field "Object ID".

Example: Open the configuration dialog for the control

In order to open the configuration dialog of the control, dynamics are possible as follows:

- VBScript:
 - ScreenItems("Control1").ToolBarButtonClick = 2
 - As an alternative to the property "ToolBarButtonClick", there are also methods in VBS for operating the toolbar: ScreenItems("Control1").ShowPropertyDialog
 - Or, with the following notation with the support of "Intellisense":
Dim obj
Set obj = ScreenItems("Control1")
obj.ShowPropertyDialog
- C script:
 - SetPropWord(lpszPictureName, "Control1", "ToolBarButtonClick", 2);
- Direct connection
 - In the dialog for the direct connection for the source, enter "2" as a constant
 - Select the property "ToolBarButtonClick" for the object "Control1" for the target "Object in picture"

See also

Operating the UserArchiveControl in runtime (Page 442)

2.4.3 Operation in runtime

2.4.3.1 Operating the UserArchiveControl in runtime

Introduction

The buttons on the toolbar are used to operate the WinCC UserArchiveControl during runtime. If you do not want to operate the table window via the toolbar, you can write the "ID" for the desired button in the "ToolBarButtonClick" object property with an optional type of dynamics.

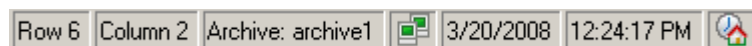
Overview

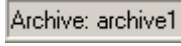
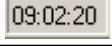
Symbol	Description	ID
	"Help" Calls up the help on WinCC UserArchiveControl.	1
	"Configuration dialog" Opens the configuration dialog, in which you can change the properties of the UserArchiveControl.	2
	"Select data connection" Opens a dialog box in which you can select a user archive. The content of the selected user archive is displayed in the table of the UserArchiveControl.	3
	"First line" The first value of the user archive is displayed in the table via the button.	4
	"Previous line" The previous value of the user archive is displayed in the table via the button.	5
	"Next line" The next value of the user archive is displayed in the table via the button.	6
	"Last line" The last value of the user archive is displayed in the table via the button.	7
	"Delete lines" The content of the marked lines are deleted.	8
	"Cut lines" The content of the marked lines are cut out.	9
	"Copy lines" The content of the marked lines are copied.	10
	"Insert lines" The content of the copied or cut-out lines is inserted starting from the marked line.	11
	"Read tags" This button is used for reading the content of the connected WinCC tags and writing to the columns. In order to use the button, the communication type "Communication via WinCC tags" must be activated in the user archive. The columns must be connected with tags.	12
	"Write tags" This button is used for writing the content of the columns into the connected WinCC tags. In order to use the button, the communication type "Communication via WinCC tags" must be activated in the user archive. The columns must be connected with tags.	13
	"Import Archive" The content of a user archive is imported into the table of the UserArchiveControl with the button. The user archive must exist as a CSV file in the "ua" directory of the project folder.	14
	"Export archive" This button exports the content of the table of the UserArchiveControl. The user archive exists as a CSV file in the "ua" directory of the project folder.	15

	"Sort dialog" Opens a dialog for setting user-defined sort criteria for the displayed user archive columns.	16
	"Selection dialog" Defining selection criteria for the columns of the user archive to be displayed in the table.	17
	"Print" Starts the printout of the displayed values. The print job used for printing is defined in the configuration dialog on the "General" tab.	18
	"Export data" This button is used for exporting all or the selected runtime data into a "CSV" file. If the option "Display dialog" is active, a dialog opens in which you can view the settings for exporting and can start the export. With the respective authorization, you are also allowed to select the file and the directory for the export. If a dialog is displayed, the export of the data to the predefined file starts immediately.	20
	"Time base dialog" Opens a dialog for setting the time base for the times used in the user archive.	19
	"User-defined 1" Shows the first key function created by the user. The function of the button is user-defined.	1001

Possible elements of the status bar

The following elements can appear in the status bar of the UserArchiveControl:



Symbol	Name	Description
	Archive name	Displays the name of the selected user archive.
	Row	Shows the number of the marked line.
	column	Shows the number of the marked column.
	Date	Shows the system date.
	Time	Shows the system time.
	Time Base	Shows the time base used in the display of times.

How to navigate in the table of the WinCC UserArchiveControl

You can navigate in the table as follows:

- You enter the next cell with the "ENTER" key or with the "Right" cursor key.
- You enter the previous cell with "SHIFT+ENTER" key or with the "Left" cursor key.

- You enter the next line by clicking with the mouse in the line or with the "Down" cursor key.
- You enter the previous line by clicking with the mouse in the line or with the "Up" cursor key.

2.4.3.2 To process the data in the UserArchiveControl:

Introduction

You can edit data in the WinCC UserArchiveControl. The following options are available:

- Enter new data
- Change existing data
- Delete lines
- Cut-out, copy and insert lines

Requirements

- You have permitted editing in the configuration dialog on the "General" tab.
- You have deactivated the "Write-protected" property for the column to be edited in the configuration dialog on the "Columns" tab.
- The "ID" column cannot be edited.
- If the UserArchiveControl is connected with a view, you cannot delete or cut out a line.

Entering new data in the table.

1. Click on  to move to the last line. The line is marked.
2. Double-click on the first cell of the marked line. You can also press on "F2", "Alt+Enter" or "Ctrl+Enter" in the cell.
3. You enter the values in the cells one after the other and confirm each time by pressing Enter. After you have entered all values in the line and have marked another line, the new data record is written to the user archive. You move to another line by clicking with the mouse, with the "ENTER" key or with the "Up" and "Down" cursor keys.
4. You can copy the data of a marked line with "CTRL+C" or "CTRL+X" into the clipboard. The copied data is inserted into a marked line with "CTRL+V".

Changing existing data in the table

1. Click on  or  to move to the desired line. You can also use the scrollbars to move to the desired line.
2. Double-click on the desired cell of the marked line. You can also press on "F2", "Alt+Enter" or Ctrl+Enter" in the cell.
3. You enter the values in the cells one after the other and confirm each time by pressing Enter. After you have entered all values in the line and have marked another line, the changed data record is written to the user archive.

Deleting a line in the table

1. Click on  or  to move to the desired line. You can also use the scrollbars to move to the desired line.
2. Click on  to delete the marked line.

Cutting, copying and inserting lines

1. Click on  or  to move to the desired line. You can also use the scrollbars to move to the desired line.
2. Click on  or  to cut or copy the data of the line. As an alternative, you can also use key combination "CTRL+ALT+X" or "CTRL+ALT+C".
3. Go to the desired line into which you want to copy the data. Click on  to insert the cut-out or copied data. If you do not want to overwrite the data of the marked line, move into the last line to insert the data.

2.4.3.3 How to select the data of the user archive

Introduction

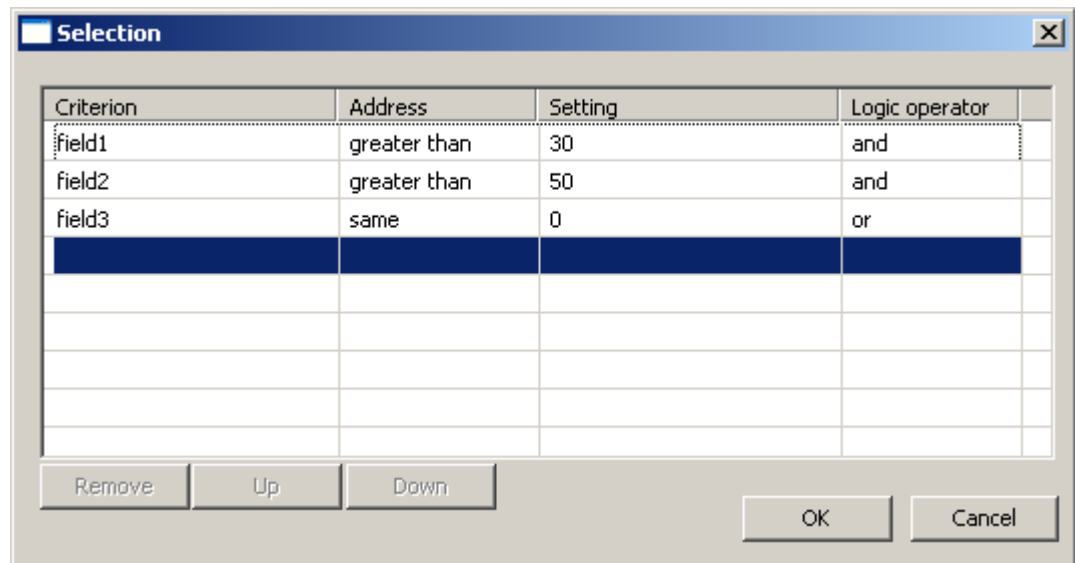
The content of the user archive that you want to display or export into the table of the WinCC UserArchiveControl can be defined in runtime via the selection dialog. You define the selection criteria concerning the displayed user archive columns in the selection dialog.

Prerequisite

- You have configured the button function "Selection dialog" on the "Toolbar" tab of the UserArchiveControl.

Procedure

- Click on  Runtime. The "Selection" dialog opens.



- Double-click in the first empty line in the "Criteria" column. The list with the columns of the user archive is displayed. Select the desired columns, e.g. "field1".
- Double-click in the "Operand" column to select an operand.
- Double-click in the "Setting" column to enter a comparison value.
- Double click in the "Logic operator" column to select an "AND" or "OR" function.
- Repeat the procedure if you want to define further criteria.
- Click "OK" to close the selection dialog. The selection is displayed in the table of the UserArchiveControl.

NOTICE

Ensuring the display of column content

Make sure of proper use of the settings and connections of criteria.

Incorrect links can lead to data of the connected user archive not being displayed in the UserArchiveControl.

Each criteria must be tested separately and then each of the linked criteria needs to be tested before linking criteria. Check that all expected contents are also displayed in combination.

This ensures that the selection is completely displayed in the UserArchiveControl.

2.4.3.4 How to sort the display of user archive data

Introduction

During runtime, you can sort the data in the UserArchiveControl by column. You sort the columns either via the "Sort dialog" button function or directly via the column headers.

Note

You can also specify the sort criteria during configuration in the UserArchiveControl by clicking the "Edit..." button under "Sorting" on the "Columns" tab.

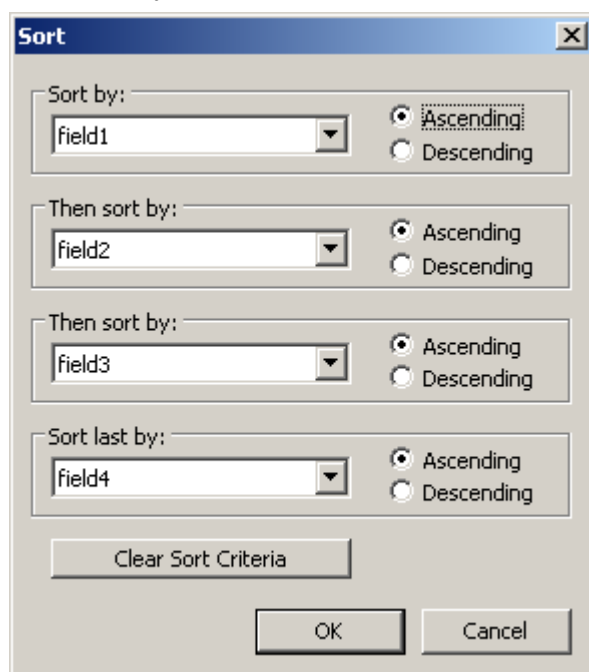
How to sort with the Sort dialog

Requirements

- You have configured the button function "Sort dialog" on the "Toolbar" tab of the UserArchiveControl.

Procedure

1. Click the "Sort Dialog" button .
2. In the "Sort By" field, select the column of the connected user archive, by which to sort first. Select the relevant check box to specify sorting in ascending or descending order. If you want to sort by more than one column, select the other columns in the desired order in the "Then sort by" lists.



How to sort the column contents with the column heading

When sorting using the column header, you are able to specify the sort order for more than four columns. A sorting icon and sorting index, displayed with right-justification in the column heading, show sorting order and sorting sequence of the column contents.

Requirements

- You have permitted the sorting in the list field "Sorting with column heading" by clicking or double clicking on the WinCC UserArchiveControl on the "Parameter" tab.
- You have activated the "Show sorting icon" and "Show sorting index" checkboxes.

Procedure

1. Click the column header of the column you want to sort as first column. The sorting index "1" is displayed, and the sorting icon points upwards for ascending sort order.
2. If you want to sort in descending order, click the column header again.
3. If the sorting order has been defined with "up/down/none", you can undo the sorting of the column with a third click.
4. If you want to sort several message blocks, click the respective header columns in the desired sequence.

2.5 Before WinCC V7: WinCC User Archives Table Element

2.5 Resources

2.5.1 User Archives Table Element

2.5.2 Functionality

Functional scope

The User Archives Table element provides options to access user archives and views of the user archives. Using the User Archives Table Element, during runtime you can:

- Create, delete or modify new data records
- Browse in user archive
- Read and write tags for direct tag link

- Import and export user archive and
- Define filter and sorting conditions

The screenshot displays the WinCC User Archives Table Element interface. At the top, there is a toolbar with various icons for table manipulation. Below the toolbar is a table with the following data:

	ID	Recipes	Water	sugar	Caffeine	Color_7	Phosphoric
1	1	Calif Coke	90	10	10	10	15
2	2	Coke	80	30	15	20	20

Below the table, there is an "Input mask Coke" section. It contains several input fields for editing the data of the selected record (ID 1):

ID	1	Water	90
Recipes	Calif Coke	sugar	10
LastAccess	18.05.99 16:49:50	Caffeine	10
		Coloring7	10
		Phosphoric_acid	15

At the bottom of the interface, there is a status bar with the text "Finished" and a navigation area showing "Rec 1/5", "Row 1", and "Col 1".

Views

The User Archives Table Element offers two views: the table view and the form view.

- The Table View
The Table View is used for a tabular display of the User Archive. Each record occupies one row, the data fields of a record are displayed as columns.
- The Form View
The Form View offers a user interface that can be designed by the user. The Form View of User Archives offers three field types: static texts, input fields and buttons.

Note

During configuration, a User Archives Table Element is connected to a selected user archive or form and can then only access that user archive or form. For access, the user archive / form must be enabled (access protection). Specific authorizations can be assigned to the control in the User Administrator.

If this access protection is canceled, the control must be reconnected in the Graphics Designer to the user archive so that control detects the canceled access protection.

Access protection for an archive or field is queried on opening a screen of a User Archive Table Elements. Access protection for the control tags of a protected archive must be implemented separately via the object properties, e.g. of the picture, I/O field or button.

See also

Configuring a User Archives Table Element (Page 451)

2.5.3 Configuration of User Archives Table Element

2.5.3.1 Configuring a User Archives Table Element

Procedure

To configure a WinCC User Archives Table Element, proceed as follows:

1. Configure a user archive using the Editor User Archives or by using the functions of the WinCC script language. In the description of the Editors User Archives, you can know how the user archive "Cola" has been configured.
2. Place a new User Archives Table Element in an image of the Graphic Designer.
3. Configure the properties of the User Archives Table Element.
4. Configure a User Archives Form View.

See also

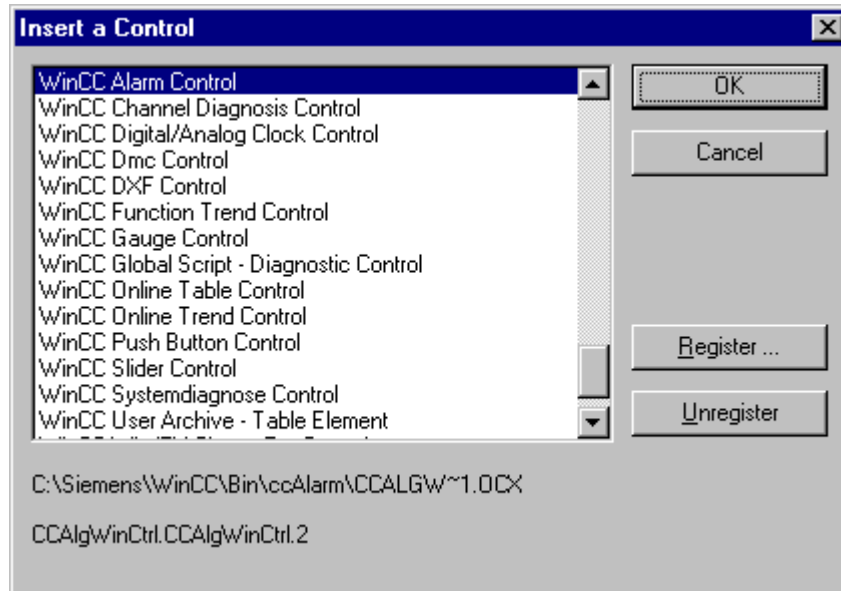
Create new user archive (Page 317)

2.5.3.2 Place the User Archive Table Element in a process screen

Procedure

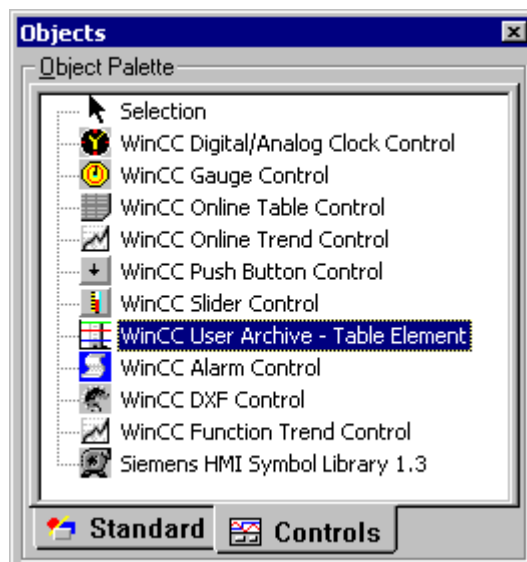
To set up a User Archives Table Element in a process screen, you need to configure it in the Graphics Designer. This is done through the following steps:

1. Select the "Smart Objects" object group from the object palette.
2. Click the "Control" object and drag a window of an adequate size into the image area.
3. In the "Add Control" selection dialog that is now displayed, select the "WinCC User Archive Table Element" option and confirm the selection with OK.



Alternative Procedure

- In the "Controls" tab of the object palette, some standard controls will be displayed to you for selection in the Object Palette window.
- Select the WinCC User Archive Table Element.



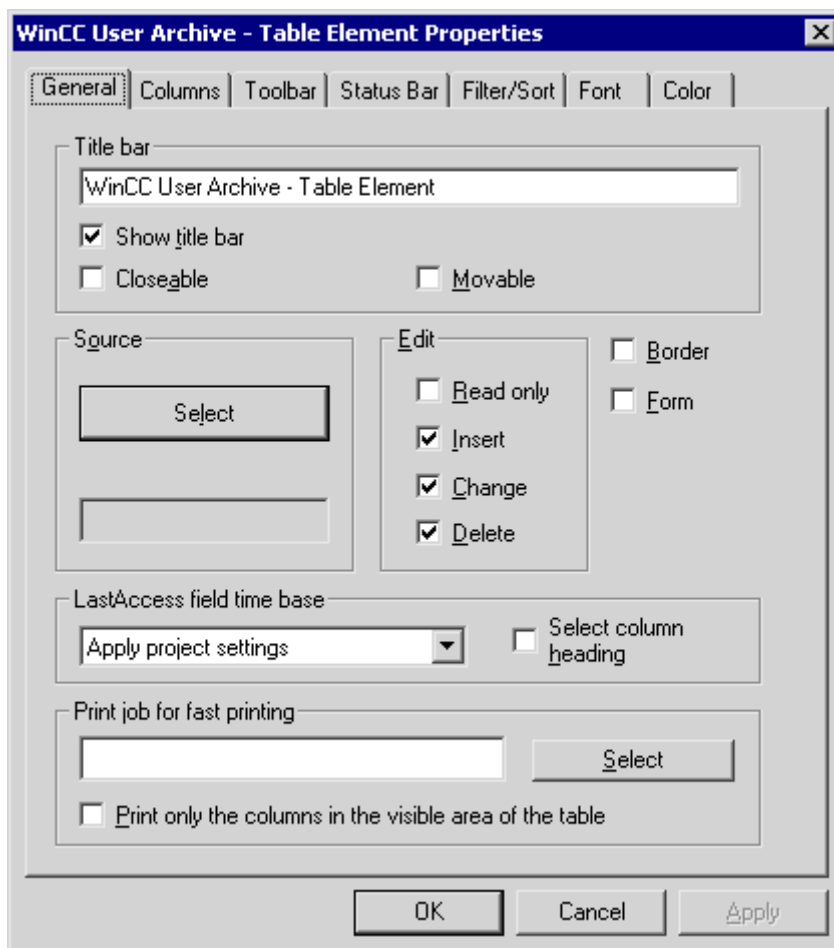
See also

Define properties of the User Archives Table Element (Page 453)

2.5.3.3 Define properties of the User Archives Table Element**Procedure**

The following guideline gives you details of how you can configure a User Archives Table Element for the user archive "Cola" in the Graphics Designer using the "Properties of WinCC User Archive Table Element" dialog box.

1. Double-click in the area of the "WinCC User Archive Table Element". You will see the "WinCC User Archive Table Element Properties" dialogbox with the "General" tab.



2. In the Source input field, define the archive or the view that is to be displayed in the Control. Click "Select" and select the user archive "Cola" in the Package Browser dialog.
3. You can define the access type during runtime in the Edit field. The "Add", "Modify" and "Delete" access types are enabled by default. Instead, you can also activate "Ready-Only".

4. Using the "Border" checkbox you can define whether the Control dialog is to be displayed with or without frame. Activate these options.
5. You can accept all the pre-settings in the other tabs without making any changes.

See also

"General" tab (Page 456)

Delete the User Archives Table Element (Page 454)

2.5.3.4 Delete the User Archives Table Element

Procedure

The User Archives Table Element is deleted in two steps in the Graphic Designer:

1. Click to select the User Archives Table Element to be deleted
2. Press Delete key or select "Edit - Delete" menu.

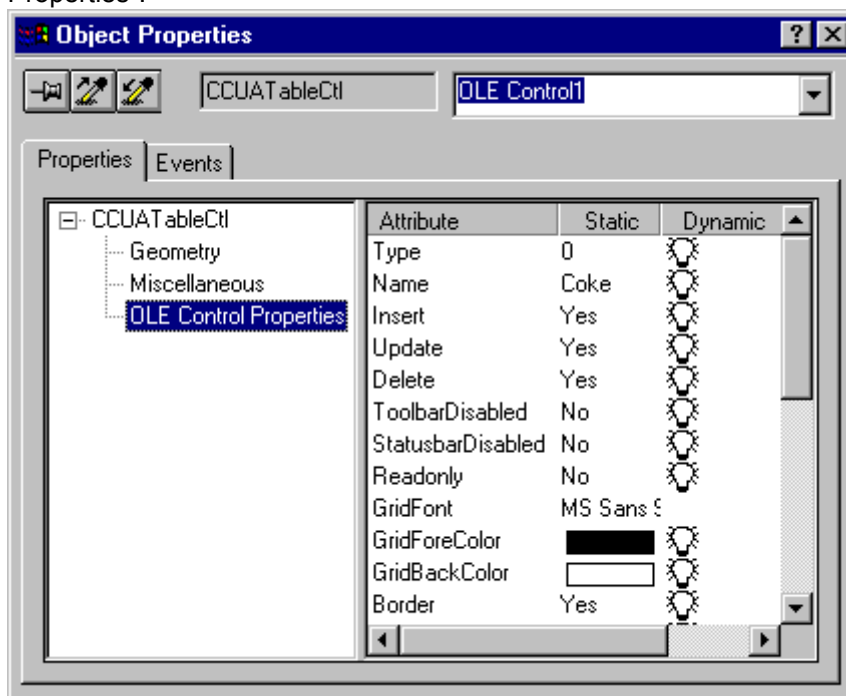
NOTICE
The delete action is executed immediately without any warning! You can undo the delete action only using "Edit - Undo" menu or "Ctrl+Z".

2.5.4 Properties of WinCC User Archives Table Element

2.5.4.1 Properties of WinCC User Archives Table Element

Procedure

1. You can modify the attributes of a User Archives Table Element by right-clicking the object and selecting the "Properties" menu item from the pop-up menu which opens. You can edit the statics of the Filter, Form, PressTBBButton and Sort attributes. To avoid inconsistencies in the database, make changes to the other object properties via the "Properties of WinCC User Archive - Table Element" dialog box (double-click Control).
2. In the "Properties" tab of the opened "Object Properties" box, select the group "Control Properties".



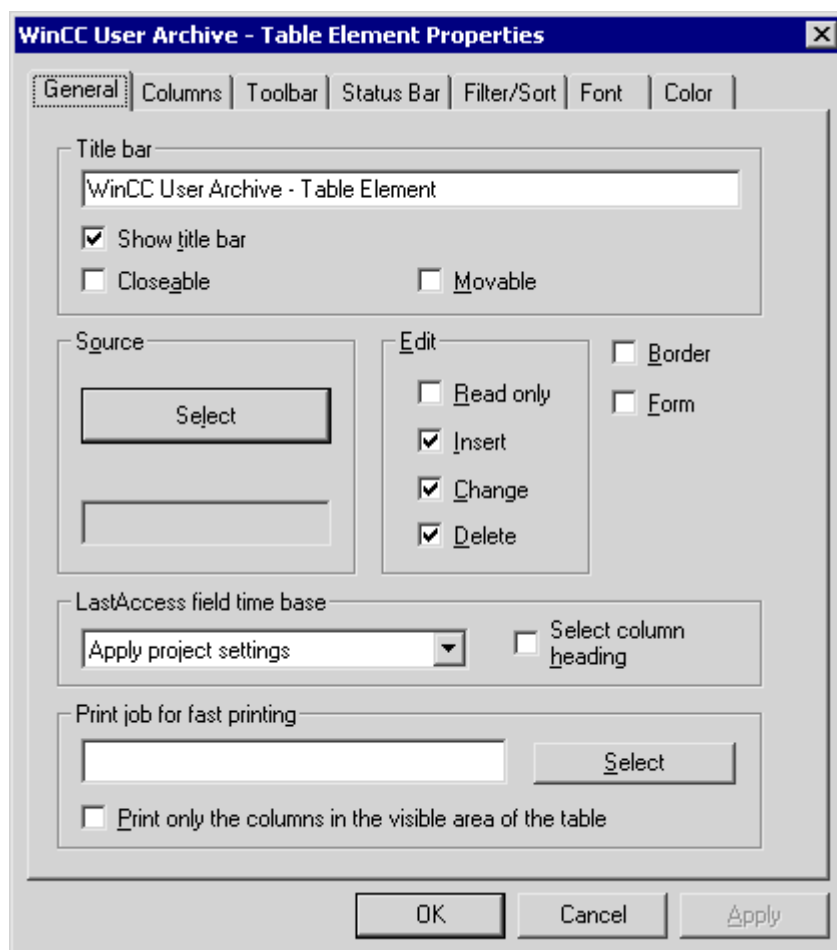
The User Archives Table Element is normally configured in the Graphics Designer by double-clicking one of the controls. You can make the desired changes in the dialog box that opens. As the existing user archives, views, tags, etc. in the dialog boxes of the different tabs are offered for selection, you can easily and safely make all the changes.

See also

Configuring a User Archives Table Element (Page 451)

2.5.4.2 "General" tab

Configuration



Icon	Description
Title Bar	Define the window title in the "Title bar" field. You define here whether the title bar can be displayed, the window can be closed or moved.
Source	Click the "Select" button to go to the Package Browser where you can select a user archive or view that has been configured earlier.
Edit	You can define the access type during runtime in the Edit field. The access types "Add", "Modify" and "Delete" are released for the user archives when you uncheck the "Read-Only" checkbox. For views, only the "Modify" checkbox is released.
Border	Using the "Border" checkbox you can define whether the Control dialog is to be displayed with or without frame.
Form	Use this checkbox to define whether the form view should be the start view in the control window.

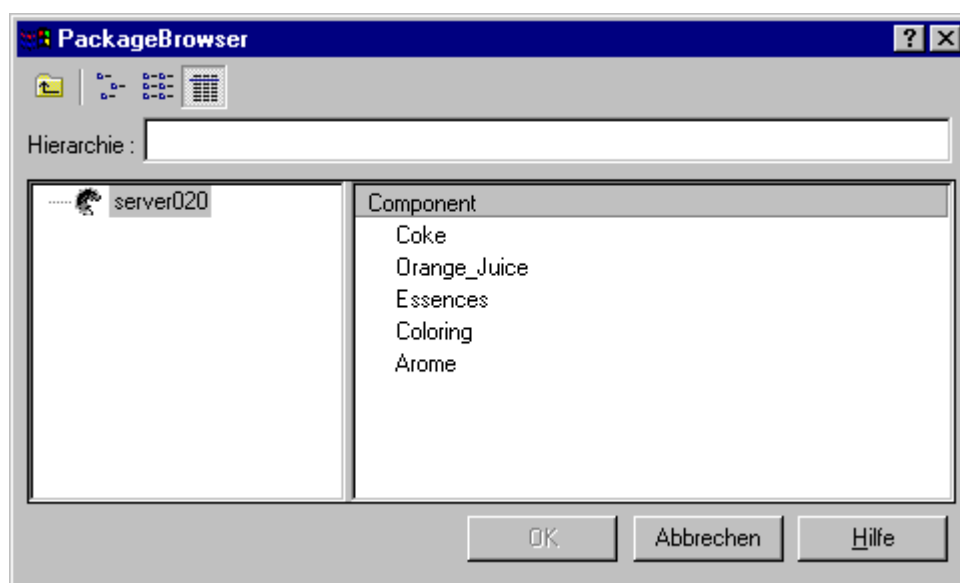
Icon	Description
Time base for the Last Access field	In this selection field you define the time basis for the time display in the Last Access field.
Print job for quick printing	In this field you define the print job that is to be used for printing the displayed data.

Note

If the configuration of the user archive is changed in the Editor User Archives, for e.g. the access protection is removed, then the Control in the Graphics Designer must be linked again to this user archive. The Control can then detect the modified archive configuration.

The Package Browser

The Package Browser is activated by clicking the Select button in the Properties dialog of the User Archives Table Element. You can select from the user archives and views that have already been configured.



In a WinCC client, you can select in the Navigation field of the Package Browser those servers whose packages have been loaded and where a user archive has been configured using tags. In a project of the WinCC client, you can access the user archives of all the servers linked in the project. Some user archives are not meant for WinCC Client. The path to the selected server is displayed in the Hierarchy area. It can be edited so that you can manually enter the path to the desired server.

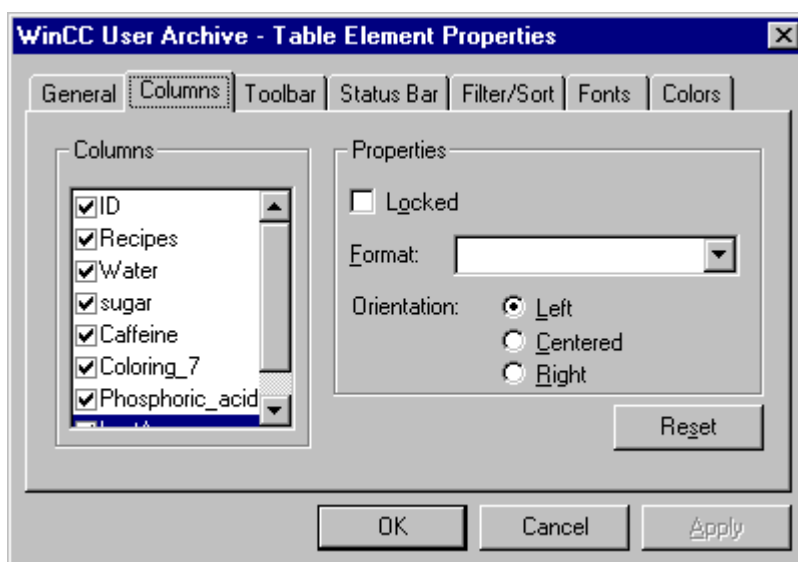
If the required server is not in the default list, a package of this server must be loaded by using the server data function. You can find additional information about WinCC Client functionalities in the WinCC Explorer Help.

Note

If the Control is not linked to an existing user archive or view, then the error message "Error while connecting the data!" is displayed when you change to runtime.

See also

List of properties for the User Archives Table Element (Page 480)

2.5.4.3 "Columns" tab**Configuration**

Icon	Description
Columns	In the Columns input field you can define which fields inserted in the Editor User Archives are to be displayed in the process screen.
Features	In the Properties input field, you can define the properties of the field currently selected in the Columns field.
Blocked	You can use the Blocked checkbox to protect the select field from being overwritten.
Format	Use the Format field to define the way the values are displayed: • Fixed (Fixed point number "%.2f") • Scientific (Exponential display "%e") • Date (only date output "%x") • Time (only time output "%X") • TimeStamp (Output date and time "%c") A date field is displayed in the date format that is set in the operating system.

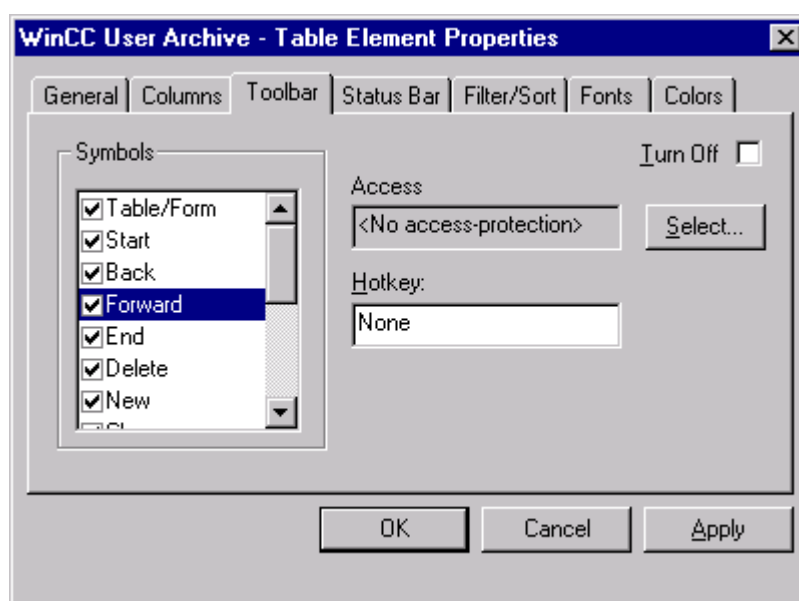
Icon	Description
Alignment	In the "Alignment" field you can select between Left, Centered and Right.
Reset	Use Reset button to reestablish the previous setting.

Note

In the Format field you can also format the decimal places (for e.g. "%3f" for three decimal places) or the hexa-decimal format "%x" for integer values.

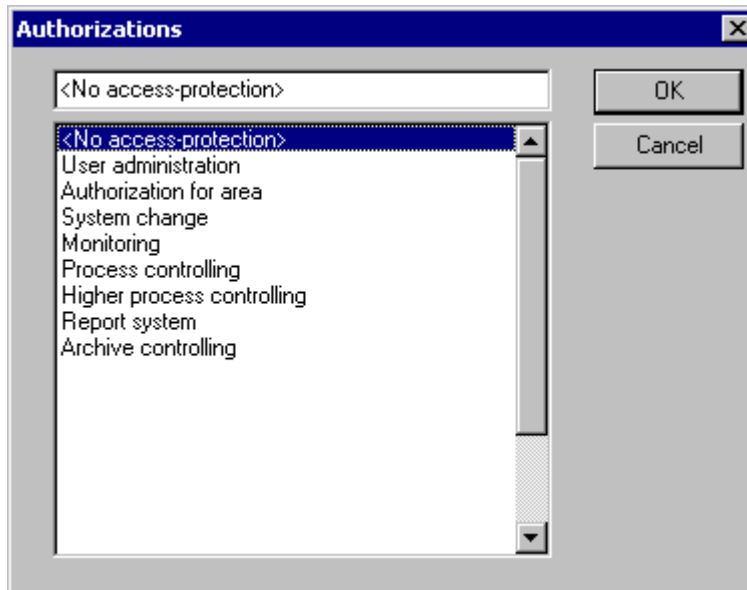
See also

List of properties for the User Archives Table Element (Page 480)

2.5.4.4 "Toolbar" Tab**Configuration**

Icon	Description
Icons	Under "Icons" you define the icons that are to be included in the toolbar.
Access rights	The access rights for the selected symbol are displayed in the "Access Rights" field.
Selection	Click the "Selection" button to display the "Authorizations" dialog box where you can define the desired access.
Switch off	The "Turn Off" field is used to turn or turn off the toolbar display.
Hotkey	You can assign hotkeys to individual functions in the Hotkey field.

Permissions



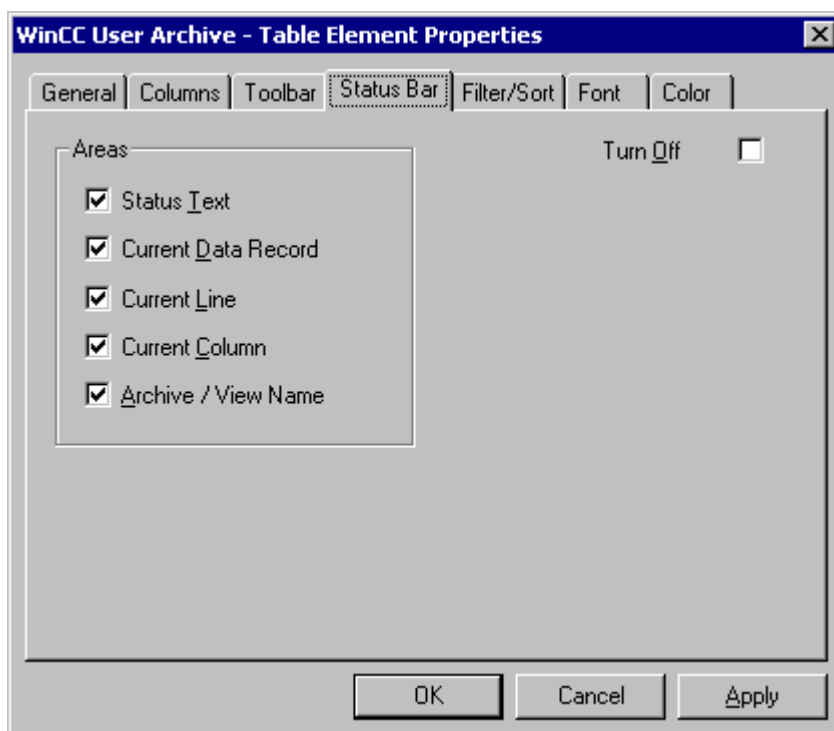
You can define the desired authorizations in the Authorizations dialog box. The authorizations displayed in the dialog have earlier been configured in the User Administrator.

See also

List of properties for the User Archives Table Element (Page 480)

2.5.4.5 "Status Bar" Tab

Configuration



Icon	Description
Areas	In the "Ranges" checkbox you can define which elements are to be included in the status bar of the control.
Switch off	The "Turn Off" field is used to turn or turn off the status bar display.

The status bar is displayed as follows when all the areas of the status bar are activated:

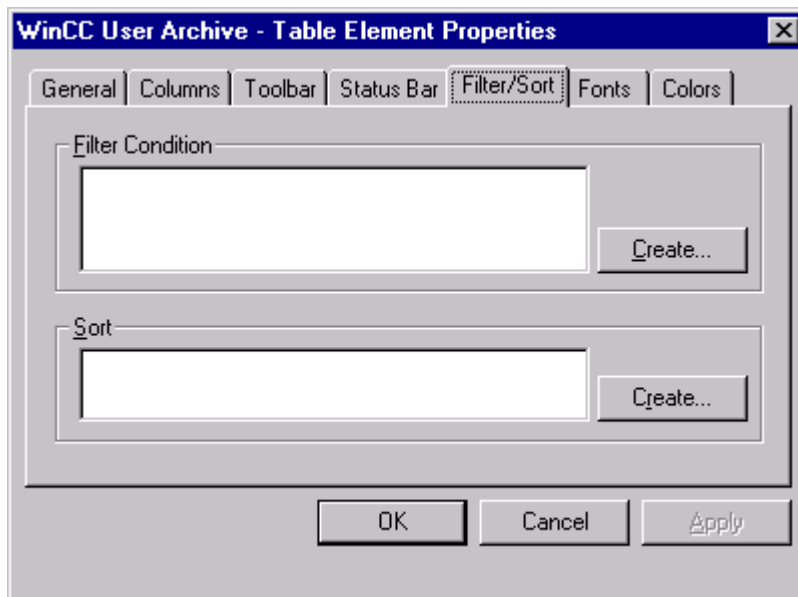
Finished	Rec 1/1	Row 1	Col 1

See also

List of properties for the User Archives Table Element (Page 480)

2.5.4.6 "Filter/ Sorting" tab

Configuration



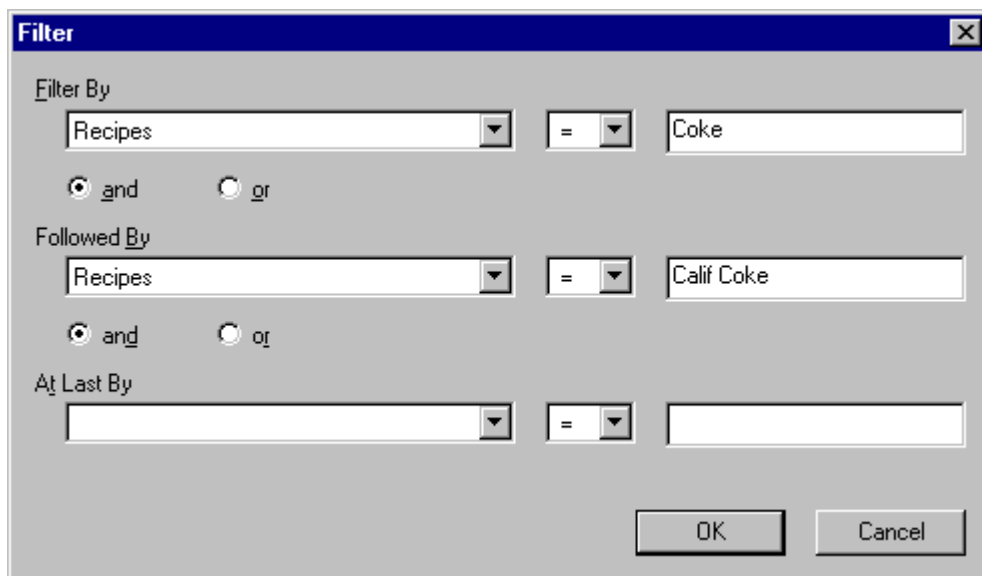
Filter criterion

You define the filter criteria in the "Filter Criterion" dialog. Enter directly the rules for the filter criteria. The conditions are formulated in the database programming language SQL (Structured Query Language). You will find a description of the SQL with many practical examples in the Appendix.

Example: FieldC > 100

All data sets, which have a value greater than 100 in the "FieldC" column, are selected.

When you click the "Create..." button you will get an automated filter screen where you can define the filter criteria.

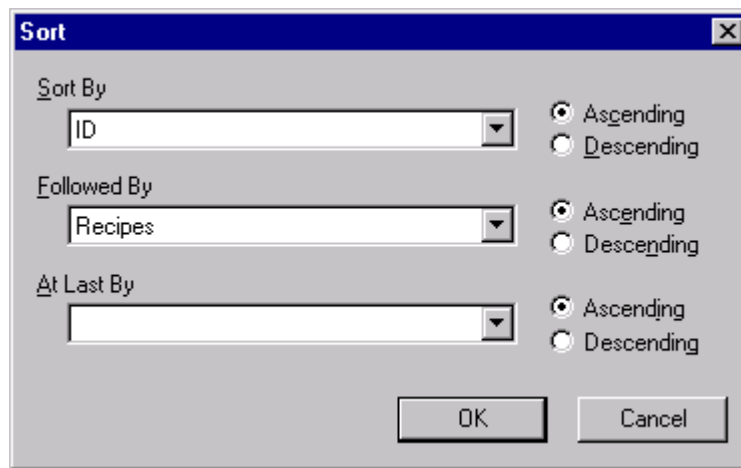


In the "Filter by" row you can define the filter criteria; in the left selection field you will see a display of all data fields of the user archive. You can use the "Followed by" and "Lastly after" rows to define the following filter criteria. The filters are processed in this sequence.

Sort order

You define the sort criteria in the "Sort..." dialog. Enter the sorting rules directly in the database programming language SQL.

When you click the "Create..." button you will get an automated filter screen where you can define the sort criteria.



In the "Sort By" selection field you can define the sort criteria; all data fields of the user archive are offered for selection. You can use the "Followed by" and "Lastly after" selection fields to define the following sort criteria. The filters are processed in this sequence. Sorting is done in an ascending order when you click "Ascending"; in a descending order when you click "Descending".

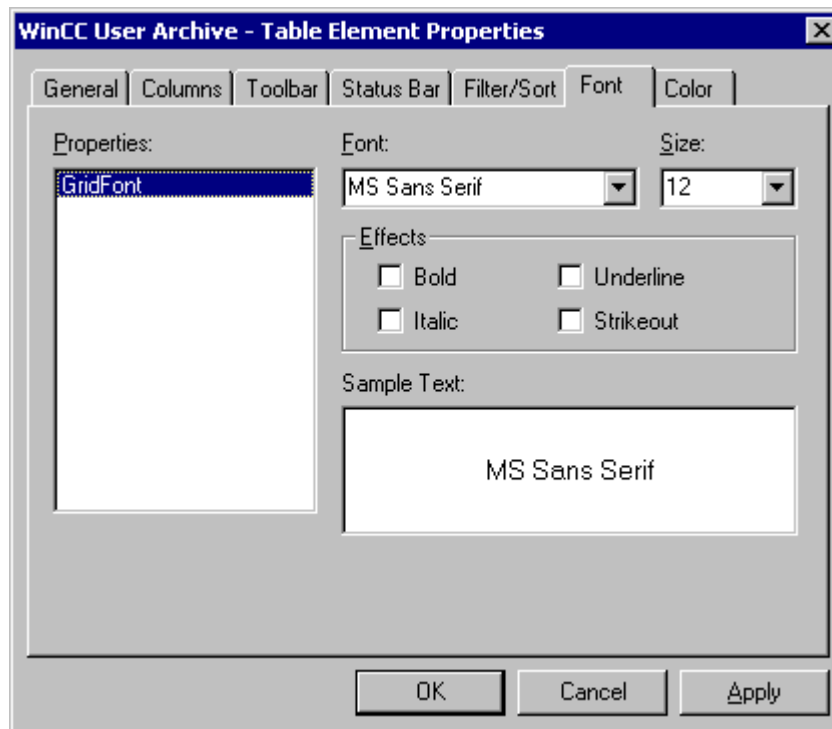
See also

List of properties for the User Archives Table Element (Page 480)

The SQL language (Page 361)

2.5.4.7 "Fonts" Tab

Configuration



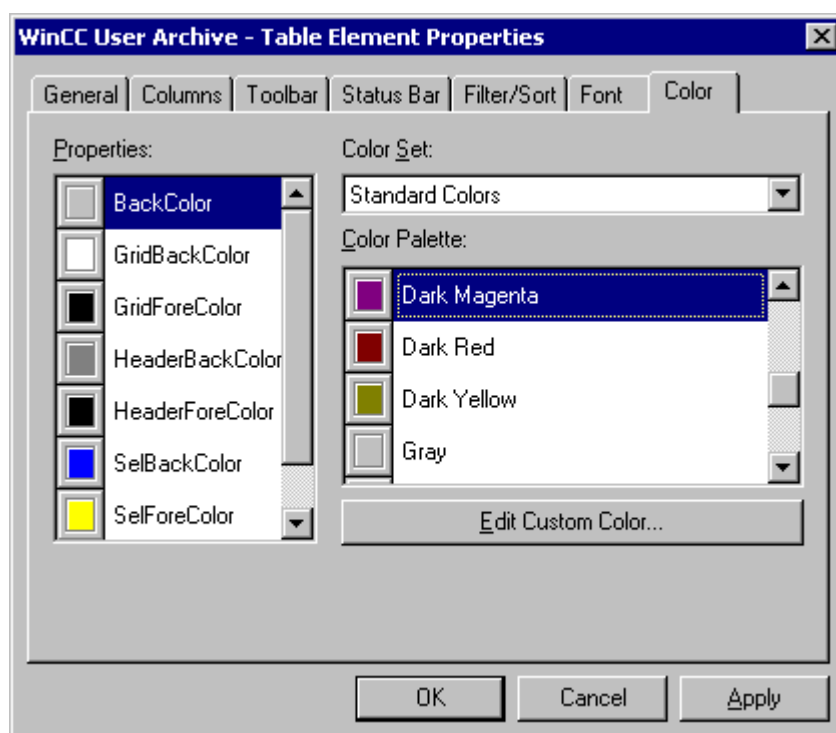
In the "Fonts" tab you define the font to be used in the Control.

See also

List of properties for the User Archives Table Element (Page 480)

2.5.4.8 "Colors" tab

Configuration



In the "Colors" tab you define the colors to be used in the Control.

See also

List of properties for the User Archives Table Element (Page 480)

2.5.5 Configuring a Form View

2.5.5.1 Configuring a Form View

Procedure

The form of the User Archive Table Element may be configured by the user himself in Graphics Designer and is used for editing and displaying user archive data in Runtime.

A configured User Archives Table Element is a prerequisite for creating a form view.

The following guideline will show you the configuration of a new form view in the Graphics Designer.

2.5 Before WinCC V7: WinCC User Archives Table Element

1. Press the Control key and simultaneously double-click the User Archives Table Element. You will get a table view of the Controls. You can now define the width of the individual columns for Runtime.

ID	Recipes	Water	sugar	Caffeine	Coloring7	Phosphoric_acid
...						

2. Use the  icon to switch between form and table view. Click this icon to go to the form view.

You can now start configuring a form.

We will now create a form:

Input mask Coke

ID	1	Water	90
Recipes	Calif Coke	sugar	10
LastAccess	18.05.99 16:49:50	Caffeine	10
		Coloring7	10
		Phosphoric_acid	15

Note

With a right-click in the empty form you can use the function "Create, all" from the pop-up menu to automatically generate the form fields for all the data fields existing in the user archive. For each data field, a text field with the corresponding alias name is also inserted for each data field. The "Create, selected" option is used to generate form fields only for the columns that are selected in the "Columns" tab.

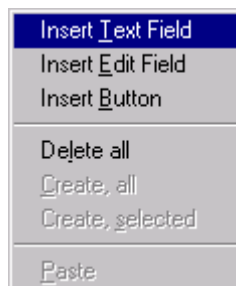
Note

The User Archive Table Element does not support zooming functionality. The configuration of the zoom functionality can lead to display problems in runtime.

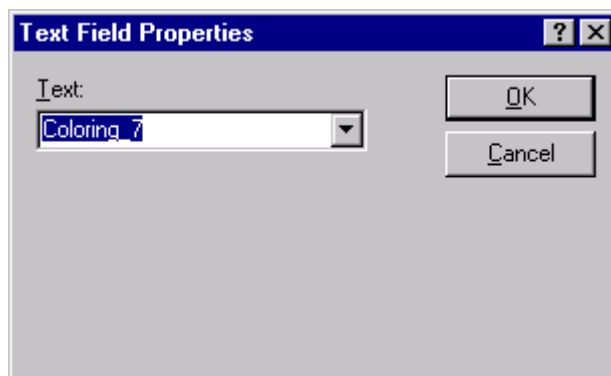
2.5.5.2 Insert "Text" form field

Procedure

1. Open the Form view if you haven't already done so.
2. To insert a new "Text" form field, click with the right mouse key in the working area of the User Archives Table Element in the Graphics Designer at the place where you want to position the text. You will see the following list box:



3. When you select "Add Text Field" you will go to the "Text Field Properties" dialog box.



4. You can enter the desired text in the Text field. Enter here text "Input form Cola" as title for the form.

Note

If you expand the selection field via "Text" you will get a display of all field names of an archive as static text. If text references for language switching have already been entered as Text References in the Text library, the same are offered for selection.

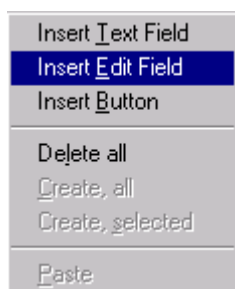
See also

Insert "Edit" form field (Page 468)

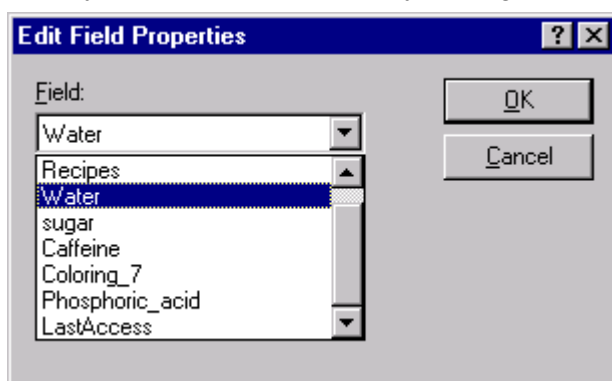
2.5.5.3 Insert "Edit" form field

Procedure

1. Open the Form view if you haven't already done so.
2. To insert a new "Edit" form field, click with the right mouse key in the working area of the User Archives Table Element in the Graphics Designer at the place where you want to position the Edit field. You will see the following list box:



3. When you select "Add Edit Field" you will go to the "Edit Field Properties" dialog box:



In the dialog of the selection field, you can select from all the configured fields of the user archive.

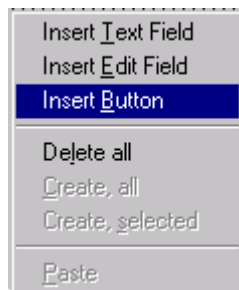
4. Select "Water". You can now insert other editing fields such as Sugar, Dyestuff 7, Caffeine and Phosphoric Acid.

See also

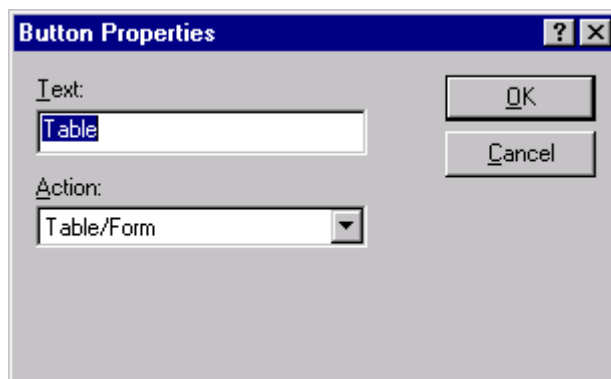
Insert "Button" form field (Page 469)

2.5.5.4 Insert "Button" form field**Procedure**

1. Open the Form view if you haven't already done so.
2. To insert a new "Button", click with the right mouse key in the working area of the User Archives Table Element in the Graphics Designer at the place where you want to position the button. You will see the following list box:



3. When you select "Add Button" you will go to the "Button Properties" dialog box:



4. In the Text field you can define the text that is to be displayed as label on the new button. Enter the text "Table View".
5. In the Action field you can select an icon of the Form view. Your newly configured button will execute the same action as the corresponding icon in the toolbar. Select "Form" to switch to table view.

Note

You can link all functions of the toolbar from the form view to a button. You can also design the size and layout of the buttons to operate some functions of the toolbar via a touch screen.

See also

Edit form fields subsequently (Page 470)

2.5.5.5 Edit form fields subsequently

Procedure

1. To subsequently modify a form field, click the pre-configured form field with the right mouse key and then click the "Properties" button.
or
Double-click the pre-configured form field.

You will see the corresponding dialog for modifying the form field as described in the chapters on Text, Edit and Button form fields.

See also

Delete form fields (Page 470)

2.5.5.6 Delete form fields

Procedure

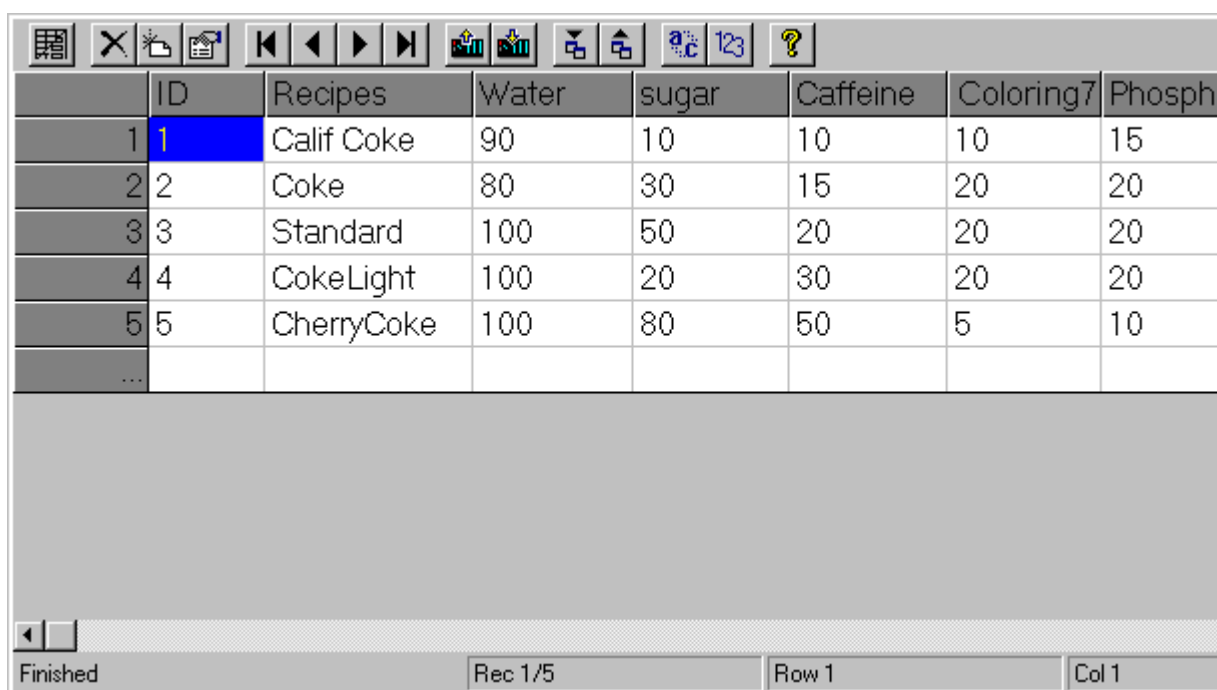
1. To delete a form field, click the pre-configured form field with the right mouse key.
 2. Click on the menu item "Delete".
- The form field is deleted. Do not use the Delete key as this will delete the entire Control.

2.5.6 User Archives Table Elements in Runtime

2.5.6.1 Table of User Archives Table Element

Application

The table of the User Archive Table Element is used for displaying and entering user archive data during Runtime in table format. Several lines of text within one cell may be accessed by hitting the key combination <CTRL+ ENTER>. Several lines of text within one cell will be displayed as one line in the table view, all lines are summarized in one single line.



	ID	Recipes	Water	sugar	Caffeine	Coloring7	Phosph
1	1	Calif Coke	90	10	10	10	15
2	2	Coke	80	30	15	20	20
3	3	Standard	100	50	20	20	20
4	4	CokeLight	100	20	30	20	20
5	5	CherryCoke	100	80	50	5	10
...							

Finished Rec 1/5 Row 1 Col 1

The table and form window of the control is operated using the toolbar:



Processing inside a table takes place in the same way as the processing of the table window in the Editor User Archive.

Note

If one or more values are changed in the control table, you must exit the data record, i.e. change to another table cell or row so the value is accepted into the database and be updated in other displays.

Actions in the WinCC script language must take care of the selection of records. You cannot select records using the control.

See also

Configuring a User Archives Table Element (Page 451)

2.5.6.2 The User Archives Table Element Form

Application

The form of the User Archive Table Element may be configured by the user himself in Graphics Designer and is used for displaying and entering user archive data in Runtime. Text may be accessed and displayed in several lines.

Input mask Coke

ID	1	Water	90
Recipes	Calif Coke	sugar	10
LastAccess	18.05.99 16:49:50	Caffeine	10
		Coloring7	10
		Phosphoric_acid	15

Finished Rec 1/5 Row 1 Col 1

Note

If you change one or more values in a user archive table element form, then you must scroll to another record after completing the data entry so that the value is accepted in the database and updated in other displays.

2.5.6.3 Toolbar of WinCC User Archives Table Element

Functions

The toolbar offers the following options:

The screenshot shows the WinCC User Archives Table Element interface. At the top is a toolbar with icons for switching, deleting, inserting, modifying, navigating, reading/writing tags, importing/exporting, defining filter and sort criteria, and setting the time base. Below the toolbar, the title "Input mask Coke" is displayed. The main area contains input fields for various attributes: ID (1), Recipes (Calif Coke), LastAccess (18.05.99 16:49:50), Water (90), sugar (10), Caffeine (10), Coloring7 (10), and Phosphoric_acid (15). At the bottom, a status bar shows "Finished", "Rec 1/5", "Row 1", and "Col 1".

Icon	Description
	Switching
	Delete record
	Insert a new record
	Modify existing field
	Browse in Table window
	Read and write tags
	Import and export user archive
	Define filter criterion
	Define sort criterion
	Time base for the "LastAccess" field

Icon	Description
	Printing
	Request Help

Switching

Use the icon to switch between form and table view.

Delete record

The selected record is deleted.

Insert a new record

You enter the values of the data fields one after the other and confirm each time by pressing Enter. After entering all data fields, the new record is created with the inserted values.

Modify existing field

After clicking this icon, click the field you want to modify. This will display the text marker - you can now see that the field can be edited. As soon as the "Modify existing field" is active, you can modify the Used Archives Table Element in the Modify mode. You can then move the cursor in the table and make the changes immediately. If the "Modify" mode is switched off, you can make changes only by pressing the F2 hotkey or double-clicking the field to be modified.

Browse in Table window

You can use these buttons to scroll or browse backward and forward in the table window and jump to the start or end of the user archive.

Read and write tags

These buttons are used to read and write WinCC tags.

While setting up the user archive in the "Communication" tab of the "Archive Properties" dialog box, you can select the communication type "Communication via WinCC tags".

Import and export archives

After clicking one of these buttons, the user archives are imported or exported in CSV format (Coma Seperated Value).

NOTICE

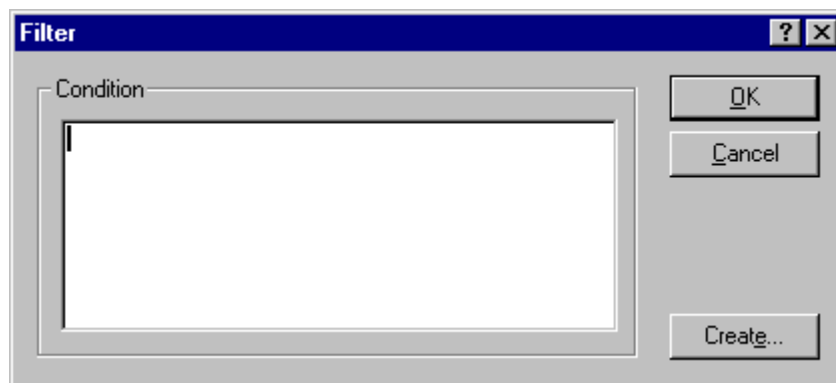
Before reading them in Excel, you need to specify data type as CSV because otherwise Excel will not read the CSV file exported from WinCC correctly.

Note

With a multi-user project the following has to be considered: If there is a user archive on the server, e.g. at "c:\Projects\Test\UA", it is enabled with this specified path. The client maps the enablement via a network drive e.g. "I:\Test\UA". Thereafter, the standard path of the User Archive is on the client "I:\Test\UA". However, this directory does not exist on the server with this description. If you want to import / export user archive data, you have to change the standard path on the client, in our example to "C:\Projects\Test\UA".

Define filter criterion

Use this option to enter filter criteria. All displayed data is exported. To export a subset, you need to formulate the filter criteria in such a way that only the desired data is displayed. You can then export this filtered data.

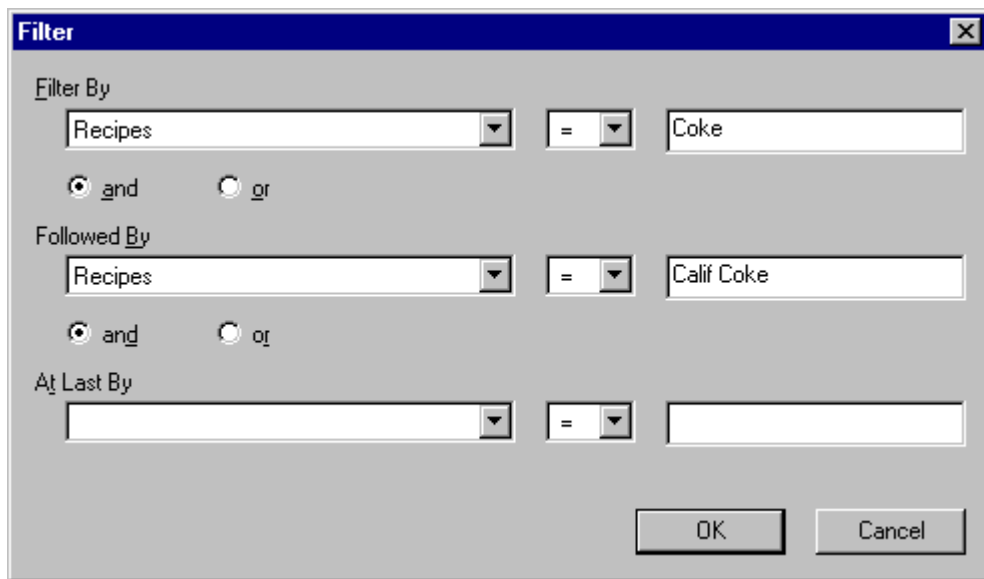


The conditions are formulated in the database programming language SQL (Structured Query Language). You will find a description of the SQL with many practical examples in the Appendix. For more details refer concerned technical literature.

Example: ID < 100

Only data fields with IDs 1 to 99 are selected; all other data fields are not displayed.

When you click the "Create..." button you will get an automated filter screen where you can define the filter criteria.

The 'Filter' dialog box has a title bar with a close button. It contains three filter criteria sections. The first section, 'Filter By', has a dropdown menu showing 'Recipes', an equals sign in a dropdown, and a text box containing 'Coke'. Below this are radio buttons for 'and' (selected) and 'or'. The second section, 'Followed By', has a dropdown menu showing 'Recipes', an equals sign in a dropdown, and a text box containing 'Calif Coke'. Below this are radio buttons for 'and' (selected) and 'or'. The third section, 'At Last By', has an empty dropdown menu, an equals sign in a dropdown, and an empty text box. At the bottom right are 'OK' and 'Cancel' buttons.

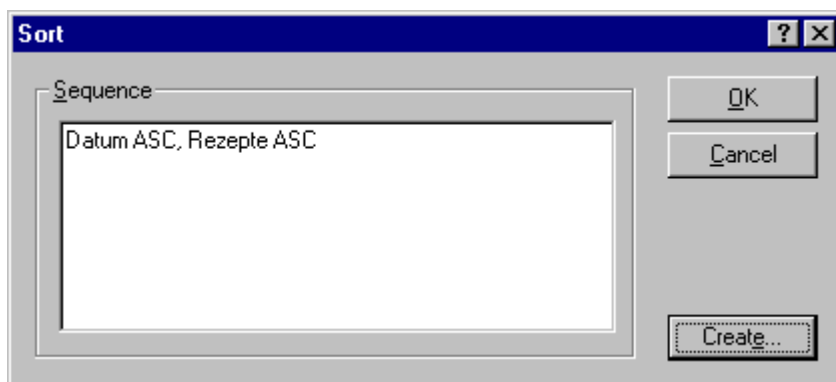
In the "Filter by" row you can define the filter criteria; in the left selection field you will see a display of all data fields of the user archive. You can use the "Followed by" and "Lastly after" rows to define the following filter criteria. The filters are processed in this sequence.

Note

The filter conditions defined here are temporary, i.e. after building up a fresh screen, the filter criteria defined in the Properties dialog are again valid.

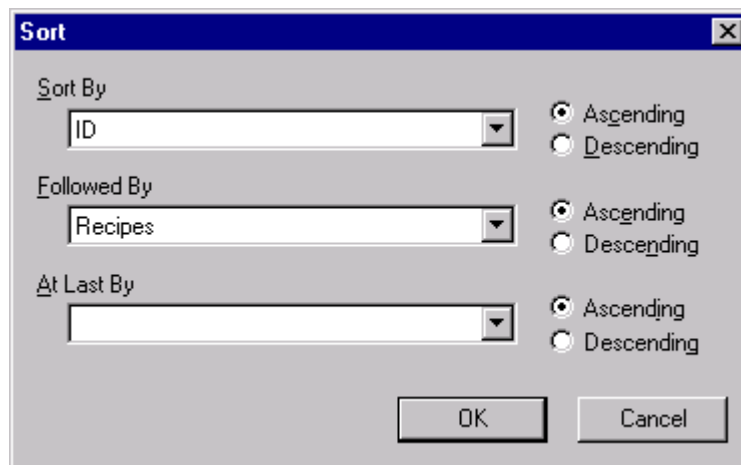
Define sort criterion

Use this option to enter sort criteria.

The 'Sort' dialog box has a title bar with a question mark and a close button. It contains a 'Sequence' label above a text box that contains 'Datum ASC, Rezept ASC'. To the right of the text box are 'OK' and 'Cancel' buttons. At the bottom right is a 'Create...' button.

The rules for sorting are directly specified in the database programming language SQL. Also refer the description of SQL in the Appendix. For more details refer concerned technical literature.

When you click the "Create..." button you will get an automated filter screen where you can define the sort criteria.



In the "Sort By" selection field you can define the sort criteria; all data fields of the user archive are offered for selection. You can use the "Followed by" and "Lastly after" selection fields to define the following sort criteria. The filters are processed in this sequence. Sorting is done in an ascending order when you click "Ascending"; in a descending order when you click "Descending".

Note

The sort conditions defined here are temporary, i.e. after building up a fresh screen, the filter criteria defined in the Properties dialog are again valid.

Time base for the Last Access field

You can use this option to change the time base for the "Last Access" field.

Printing

This option starts printing the displayed values.

Request Help

Click the Help button to request Help for the User Archives Table Element.

2.5.6.4 Operating the Control using Dynamized Objects

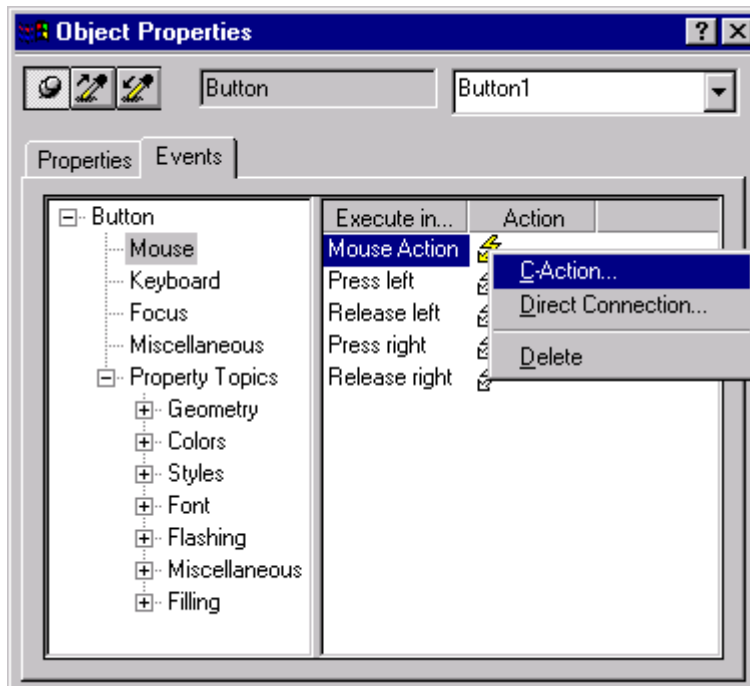
Operating options

The User Archives Table Element offers you the option of shifting the functions of all the toolbar buttons to self-defined buttons or I/O fields. You can define the size and appearance of each of the buttons so that you can operate the Table Element say by using a touch screen.

Example for using the "Press TB Button" attribute

You need to run the following steps to connect the User Archives Table Element with a button:

1. Create a button in the Graphics Designer and call it by right-clicking the object properties.
2. In the "Event" tab, select the Mouse option. Click in the right window to select "Mouse click". After right-clicking the arrow in the "Action" column, you see a dialog box; select the "Direct Connection" option.



3. In the "Source" area, select "Constants" and enter a constant here, for e.g. "Form" (see further below for an overview of the constants available for the User Archives Table Element).
4. In the "Target" area, click the "Object in Image" option and select the table element to be linked in the Object Selection field. In the Properties box, select "PressTBButton" and confirm the dialog by clicking OK.
5. Save the picture in the Graphics Designer and go to Runtime. When you now activate the configured button "Form", the control display changes from the table view to the form view and vice-versa.

Constants for direct link to the User Archives Table Element

For the above direct links to the User Archives Table Element you have one constant for each button of the control. You can assign the individual buttons according to the following table.

Constant	Corresponding button
Form	
Delete	

Constant	Corresponding button
New	
Edit	
First	
Previous	
Next	
Load	
WriteVar	
ReadVar	
Import	
Export	
Filter	
Sort	
Timezone	
Print	
Help	

Note

While operating the table window using the keyboard, the cell cursor for the selected record cell is no longer visible when you press the keys "Tab" and "Position 1". To bring back the display to the last edited record, insert a button according to the steps mentioned above and select the "VTB_Focus" constant. When you press this button, the cell cursor jumps back to its last position.

See also

List of properties for the User Archives Table Element (Page 480)

2.5.6.5 List of properties for the User Archives Table Element

Overview

You can set the following properties for the user archives table control:

Attribute	Description	can be made dynamic
BackColor	Defines the background color of the table window in the user archives table element. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. In the tab "Colors" you can edit these settings.	no
Border	Determines whether the form view of the User Archives Table Elements in Runtime is displayed with or without a border. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. In the tab "General" you can edit these settings.	no
Buttons	Defines that the pointers generated by the software of the buttons activated in the toolbar are output. In order to avoid inconsistencies in the database, the static of this attribute must not be edited.	no
Caption	Defines the labeling of the title bar in the user archives table element.	no
Closable	Determines whether the user archives table element can be closed via the "X" in the title bar.	no
Delete	Determines whether deletion processes are allowed in the user archives table element in Runtime. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. There is a checkbox for this in the area "Edit" on the "General" tab.	no
Filter	Defines filter conditions for the database. The conditions are formulated in the database programming language SQL (Structured Query Language). Example: FieldC>100 All data sets, which have a value greater than 100 in the "FieldC" column, are selected. You can also enter these filter conditions in the tab "Filter/Sort".	yes, with the name Filter
Form	Defines the view of the user archives table elements when starting in Runtime: Status "Yes": Outputs the form view Status "No": outputs the table view. You can also change these settings in the "General" tab	yes, with the name Form
GridBackColor	Defines the background color of the data set in the user archives table element. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. In the tab "Colors" you can edit these settings.	no
GridFont	Defines the font in the user archives table element. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. In the tab "Fonts" you can edit these settings.	no

2.5 Before WinCC V7: WinCC User Archives Table Element

Attribute	Description	can be made dynamic
GridForeColor	Defines the font color of the data set in the user archives table element. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. In the tab "Colors" you can edit these settings.	no
HeaderBackColor	Defines the background color of the header and the column with the consecutive line number in the user archives table element. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. In the tab "Colors" you can edit these settings.	no
HeaderForeColor	Defines the font color of the header and the column with the consecutive line number in the user archives table element. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. In the tab "Colors" you can edit these settings.	no
Insert	Defines whether entries can be made in the user archives table element. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. There is a checkbox for this in the area "Edit" on the "General" tab.	no
LocaleSpecificSettings	Defines the language-specific response of the texts and fonts, which you configure in the properties dialog. Value = "Yes": You can assign separate texts and fonts for each Runtime language. To do so, select a language in the "View/Language" menu of the Graphics Designer and choose the desired font in the controls. Value = "No": You cannot define language-specific texts and fonts. The configuration of the control always applies to all available Runtime languages.	yes, with the name LocaleSpecificSettings
Movable	Defines whether the user archives table element can be moved.	no
Name	Defines which user archive or which view is displayed. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. In the area "Source" of the tab "General" you will see a selection of all configured user archives and views.	no
PressTBButton	Connects all buttons of the toolbar of the table elements with self-defined buttons or I/O fields.	yes, with the name Press TB Button
PrintJob	Specifies which layout should be used for the print output.	no
PrintVisColsOnly	Defines whether only the currently visible columns should be printed in draft print mode.	no
Read only	Defines whether the user archives table element can be edited or only read in Runtime. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. There is a checkbox for this in the area on the "General" tab.	no
SelBackColor	Defines the background color of the selected data set in the user archives table element. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. In the tab "Colors" you can edit these settings.	no

2.5 Before WinCC V7: WinCC User Archives Table Element

Attribute	Description	can be made dynamic
SelfForeColor	Defines the font color of the selected data set in the user archives table element. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. In the tab "Colors" you can edit these settings.	no
SelectedID	Displays the ID of the data set selected in the control window. SelectedID = "0": if no valid data set has been selected, e.g. during the connection error SelectedID = "-1", if the edit line is selected.	no
Sort	Defines the sorting conditions for the database. The conditions are formulated in the database programming language SQL (Structured Query Language). You can also enter your filter conditions in the "Filter/Sort" Tab.	yes, with the name Sort
StatusbarDisabled	Defines whether the status bar in the user archives table element is activated in Runtime. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. There is a checkbox "Turn Off" for this in the area on the "Status Bar" tab.	no
StatusbarShowArc	Defines whether the archive name is shown in the status bar of the user archives table element.	yes, with the name StatusbarShowArc
StatusbarShowCol	Defines whether the consecutive number of the currently selected data set column is shown in the user archives table element status bar. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. You can change this setting in "Current column" in the "Status Bar" tab.	no
StatusbarShowRecord	Defines whether the field coordinates of the currently selected data set column is shown in the user archives table element status bar. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. You can change this setting in "Current Data Record" in the "Status Bar" tab.	no
StatusbarShowRow	Defines whether the consecutive number of the currently selected data record line is shown in the user archives table element status bar. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. You can change this setting in "Current Line" in the "Status Bar" tab.	no
StatusbarShowText	Defines whether the current status of the database is shown in the user archives table element status bar. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. You can change this setting in "Status text" in the "Status Bar" tab.	no
Titleline	Defines whether the title bar is shown in the user archives table element.	no
TimeZone	Determines the time base used for the display of times in Runtime. The time base is set via the following numeric values: Value = 0: Apply project settings Value = 1: Server's time zone Value = 2: Local time zone Value = 3: Coordinated Universal Time (UTC) We recommend applying the default configuration "Apply Project Settings". This means that the display is operated at same time zone as the rest of the project.	no

Attribute	Description	can be made dynamic
TimeZoneMark	Determines whether the column heading of the LastAccess field should include the set time zone. The following acronyms are used for the time zone: LOC: Local time zone UTC: Coordinated universal time SVR: Server's time zone	no
ToolbarDisabled	Defines whether the toolbar in the user archives table element is activated in Runtime. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. There is a checkbox "Turn Off" for this in the area on the "Toolbar" tab.	no
Type	Defines whether a user archive or a view is displayed in the user archives table element Value Type= 0: Stands for a user archive Value Type= 1: Stands for a view. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. In the area "Source" of the tab "General" you will see a selection of all configured user archives and views.	no
Update	Defines whether changes can be made in the user archives table element. In order to avoid inconsistencies in the database, the static of this attribute should only be edited in the dialog box of the property dialog. There is a checkbox for this in the area "Edit" on the "General" tab.	no

2.5.6.6 Overview of the dynamizable properties in the layout

Filter

You can use the "Filter" property to define filter conditions for the database. The conditions must be formulated in the database programming language SQL.

Format: SQL text

Tag type: Text Tag

Sort

You can use the "Sort" property to define sorting criteria for the database. The criteria must be formulated in the database programming language SQL.

Format: SQL text

Tag type: Text Tag

TimeZone

Determines the time base used for the display of times in runtime.

Format: Number

Value	Description
0	Local time zone
1	Server Time Zone

2.5 Before WinCC V7: WinCC User Archives Table Element

Value	Description
2	Coordinated Universal Time (UTC)
3	Apply project Settings

Variable type: All tag types, except binary, text and raw data tags

COM Provider in the Layout Editor

3 Resources

3.1 COM Provider in the Layout Editor

Contents

In the report system of WinCC, an user-specific report object can be integrated with the aid of a COM interface. Therefore data which does not originate in WinCC can be output in a WinCC report. The documentation provides information about:

- Application options of the COM Provider
- Integration of COM Provider into WinCC
- Outputting data with the aid of a COM object
- Details of the COM interface for reporting

3.2 Working with COM Server Objects

Introduction

To integrate user-specific data in a WinCC log, you can integrate a COM server in the report system. This COM server provides a COM object in the object selection that is selected in the page layout editor and added to a page layout. The COM object then provides the user-specific data for output in the log. The COM object can be of the type text, table, or picture.

At this point, no comment can be made about the COM object itself. Information about it is provided by the writer of the COM server.

Integration of COM Server Objects

The following steps are necessary to use a COM object in the report system:

1. Register COM object
2. If necessary, depending on the COM object, run the Registry file of the COM object.
3. Insert COM object in the object palette of the page layout editor by entering it in the registry
4. Select COM object in the object selection
5. Position COM object in the layout and parameterize it

3.3 How to Output Data from a COM Server in a Report

Steps 1 to 3 must be carried out on every server and WinCC client on which you want to use such a COM object. These steps must also be carried out if the COM object should only be used in Runtime.

To create a COM server for the report system of WinCC, the Type Library "WinCCProtProvider.tlb" must be registered on the development computer. The Type Library is automatically registered on a computer with WinCC installed. To register on a computer without WinCC installed there is the following option:

Copy the "WinCCProtProvider.tlb" file from a computer with WinCC installed. The file is located in directory "..\Siemens\WinCC\Interfaces. Add the file to the target computer and register it.

A correctly integrated COM object is displayed in the object selection of the page layout editor on the "COM Server" tab. From there it can be selected and added to the layout.

For more information, refer to chapter "Details of the COM Interface for Reporting".

See also

How to Output Data from a COM Server in a Report (Page 486)

Example of an integration of a COM server (Page 487)

Details of the COM Interface for Reporting (Page 488)

3.3 How to Output Data from a COM Server in a Report

Introduction

To integrate user-specific data in a WinCC log, you can integrate a COM server in the report system. This COM server makes a log object available in the object palette that can be selected in the page layout editor and inserted in a page layout. The COM object then provides the user-specific data for output in the log.

Available Log Objects

COM object defined by the user	Serves to output data from data sources of the user in a WinCC log.

Requirements

- Knowledge of how to create layouts and insert log objects

Procedure

1. Create a new page layout, and open it in the page layout editor.
2. In the object palette on the COM Server tab, select a COM object integrated by the user, and drag it to the desired size in the working area.

3. The creator of the COM object makes specifications available on the connection and selection of the data.
4. Configure the COM object in accordance with the specifications made there.
5. Save the layout.
6. Create a print job, and select the configured page layout there.
7. Start output by means of the print job in WinCC Explorer or by means of a configured call in a WinCC picture, for example.

Output options

You receive information on possible output options from whoever wrote the COM object.

See also

Working with COM Server Objects (Page 485)

Details of the COM Interface for Reporting (Page 488)

Example of an integration of a COM server (Page 487)

3.4 Example of an integration of a COM server

Introduction

There are two examples provided on the WinCC CD-ROM, each containing a COM server. One example is written in Visual Basic, the second one is written in Visual C. The examples are in zip format under "Options\ODK\Samples" on the WinCC CD-ROM. The "CCProtTableServerExampleVB.zip" file contains the example in Visual Basic. "CCProtPicturerExampleCPP.zip" file contains the example in Visual C++.

Procedure

Unpack the compressed file into a temporary directory. Integrate the files into the system.

1. Provide the Type Library
2. Compile example
3. Register COM server
4. Integrate COM server in the report system

To integrate the COM Server in the system, you require extensive programming skills that cannot be provided in this documentation. Chapter "COM Interface Requirements for Reporting" contains instructions regarding the formal COM server requirements.

The "*.REG" files supplied with the examples are not part of the sample projects in Visual Basic or Visual C. Instead they are used to insert the COM Servers into the object palette of the page layout editor.

See also

Details of the COM Interface for Reporting (Page 488)

How to Output Data from a COM Server in a Report (Page 486)

3.5 Details of the COM Interface for Reporting

Introduction

This chapter provides information and the requirements of a COM Interface for reporting. Further instructions can be found in the chapters "COM Server Data Output" and "Example of the Integration of a COM Server".

Call Interface

An object can/must provide the following COM interfaces so that the WinCC report system can use this object:

```
interface IWinCCProtProvider : IDispatch
{
    HRESULT Register([in]IDispatch* pIDispWinCCProtReportParams);
    HRESULT Unregister();
    HRESULT GetName([out, retval]BSTR* pName);
    HRESULT ShowPrivateDialog([in]long hwndParent, [out, retval]BOOL* pfOK);
    HRESULT SetPrivateData([in]VARIANT PrivateInfo);
    HRESULT GetNameOfPrivateData([out, retval]BSTR* pPrivateInfoName);
    HRESULT GetPrivateData([out, retval]VARIANT* pPrivateData);
};

interface IWinCCProtProviderText : IDispatch
{
    HRESULT GetText([out, retval]BSTR* pName);
};

interface IWinCCProtProviderTable : IDispatch
{
    HRESULT GetNumCols([out, retval]int* pnNumCols);
    HRESULT GetNumLines([out, retval]int* pnNumLines);
    HRESULT GetText([in]int nLine, [in]int nCol, [out, retval]BSTR* pName);
    HRESULT HasHeader([out, retval]BOOL* pfHasHeader);
    HRESULT GetHeader([in]int nCol, [out, retval]BSTR* pName);
};
```



```
};
Interface IWinCCProtProviderPicture : IDispatch
{
    HRESULT Draw( [in]long hdc, [in]int left, [in]int top,
[in]int right, [in]int bottom);
};
```

The object must support the interface IWinCCProtProvider and one of the interfaces IWinCCProtProviderText, IWinCCProtProviderTable and IWinCCProtProviderPicture.

CR+ is used as line feed characters at the IWinCCProtProviderText interface.

LF (CR = "Carrage return" and LF = "line feed").

Interface IWinCCProtProvider

Register	Is called after starting the COM Server to pass a pointer to IWinCCProtReportParams to the server.
Unregister	Is called to instrct the COM Server to rerelease the pointer to the interface IWinCCProtReportParams.
GetName	Returns the name of the COM Server to display it on the configuration interface.
ShowPrivateDialog	Open the COM Server selection dialog box.
SetPrivateData	Passes on the SelCrit data saved in the layout to the COM Server.
GetPrivateData	Reads the SelCrit data from the COM Server to save it in the layout.
GetNameOfPrivateData	Returns the name of the selection criterion to display it on the configuration interface.

Interface IWinCCProtProviderTable

GetNumCols	Returns the number of columns to be printed in the report.
GetNumLines	Returns the number of rows which should be printed in the report.
HasHeader	Returns the information as to whether a table header should be printed in the report.
GetHeader	Returns the table header text to be printed in the report.
GetText	Returns the text to be printed in the report.

Special aspects of the data:

The control characters for colors, alignment, etc. always precede the output text and can be combined with each other (e.g. "<U>output text"). They are not case-sensitive.

<END>	Concludes the interpretation of control sequences. The rest of the text is accepted as specified.
<COLOR=#rrggbb>	Font color in hexadecimal notation (default = as set for the table)
<BGOLOR=#rrggbb>	Background color in hexadecimal notation (default = as set for the table)
	Bold
<U>	Underlined
<I>	Italic
<STRIKE>	Strike-through
<ALIGN=left>	Left aligned
<ALIGN=center>	Centered
<ALIGN=right>	Right aligned

Interface IWinCCProtProviderText

GetText Returns the text to be printed in the report

Interface IWinCCProtProviderPicture

Draw Hands over a handle to the device connect to the COM Server and coordinated in which you can draw.

Draw the output field here in the specied device context. It is an enhanced metafile. Drawing is done in MM_HIMETRIC mode.

Interface to Read Parameters from the Report Context

The report system provides an interface to read parameters from the report context (settings from the report system and the job properties).

Interface IWinCCProtReportParams : IDispatch

```
{
HRESULT GetParameter ( [in]BSTR PropertyName, [out]VARIANT* Value );
};
```

Currently the following properties can be read:

TimeFrom	Parameter from the print job
TimeTo	Parameter from the print job
PrivateSelCrit	Is used e.g. when the "Print" button in WinCC Control is pressed. Using this PrivateSelCrit, the current selections of the WinCC Control are sent to the COM server.
ProjectName	WinCC project name

LCID_APP	Currently set language in the application which the COM server calls (PrintIt/ProtCS). The WinCC? system language might be different because the Runtime language is not actively supported by WinCC.
LCID_RT	Current Runtime language of WinCC. This setting is only visible in Runtime.

Registry Entries

The entries in the Registry are made automatically depending on the COM object or must be made in the Registry by calling a Registry file. The Registry file must be supplied by the COM object. Without these Registry entries, a COM object is not available to the report system, even if the COM object is registered. The COM objects are registered by the user.

Example of Registry entries of COM server:

```
HKEY_LOCAL_MACHINE\SOFTWARE\SIEMENS\WinCC\Report Designer
\ReportClientDLLs\{4BF175C2-8BFF-11D0-840D-0080AD1374C8} (GUI-ID of the COM
object as a unique key)
"DllClientGUID"="{4BF175C2-8BFF-11D0-840D-0080AD1374C8}" (GUID of the COM object)
"DllFileName"="CCPComProvider.dll"
"NeedsRuntime"="NO"
„RunsOnServer„=„YES„
„RunsOnServer„=„YES„
„RunsOnMultiClient„=„YES„
"UseReportDesignerObjTab"="COM-Server"
```

Behavior of the Application Objects

The calls in the COM Server are timed.

Debugging support:

So that debugging COM servers is not unnecessarily made more difficult by the timeout behavior, the timeout period can be set. If the Registry Key is not defined, the default value (10000 milliseconds) is used.

```
HKEY_CURRENT_USER\Software\SIEMENS\WINCC\ReportSystem\TimeOuts\
```

InvokeTimeOut (vom Typ DWORD) --> TimeOut period in milliseconds

If timeout period 0xffffffff (-1) is entered as value, the report system waits infinitely for the function.

See also

How to Output Data from a COM Server in a Report (Page 486)

Example of an integration of a COM server (Page 487)

OPC - Open Connectivity

4 Resources

4.1 OPC - Open Connectivity

Contents

The OPC standard software interface allows devices and applications from various manufacturers to be combined with one another in a uniform manner.

WinCC can be used as an OPC server or an OPC client. The "OPC" channel represents the OPC client application of WinCC.

This chapter describes

- the OPC servers of WinCC
- how to use OPC in WinCC
- how to set up various OPC DA links
- how to configure the access to the WinCC message system
- how the WinCC message system appears on the OPC A&E
- how to set up access to the WinCC archive system

4.2 Functionality of OPC

Introduction

OPC (Open Connectivity) is a uniform and manufacturer-independent software interface. The OPC interface is based on Microsoft Windows COM (Component Object Model) and DCOM (Distributed Component Object Model) technology. OPC XML, on the other hand, is based on the Internet standards XML, SOAP, and HTTP.

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 that 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.

Previously, applications which accessed process data were tied to the access protocols of the communication network. The OPC standard software interface allows devices and applications from various manufacturers to be combined with one another in a uniform manner.

The OPC client is an application that accesses process data, messages and archives of an OPC server. Access is through the OPC software interface.

An OPC server is a program that provides the applications from various manufacturers with a standard software interface. The OPC server is the intermediate layer 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. Currently, Windows 2000, Windows XP, Windows 2003 Server and Windows VISTA have these software interfaces.

XML

Communication via DCOM is restricted to local networks. Data exchange via XML operates with SOAP (Simple Object Access Protocol). SOAP is a platform-independent XML-based protocol. SOAP can be used to enable applications to communicate with each other via the Internet or in heterogenic computer networks by means of HTTP (HyperText Transfer Protocol).

4.3 OPC Specifications

Introduction

The OPC standard software interface was defined by the OPC Foundation. The OPC Foundation is an alliance of leading companies in the field of industrial automation. The OPC servers of WinCC support the following specifications.

- OPC Data Access 1.0, 2.05a and 3.0
- OPC XML Data Access 1.01
- OPC Historical Data Access 1.20
- OPC Alarm & Events 1.10

OPC Data Access (OPC DA)

OPC Data Access (OPC DA) is the specification for the management of process data. The WinCC OPC DA server conforms to OPC DA specifications 1.0, 2.05a and 3.0.

OPC eXtensible Markup Language DA (OPC XML DA)

The OPC XML standard supports communication over the Internet by means of platform-independent protocol. Clients are no longer restricted to a Windows environment (DCOM). Other operating systems such as LINUX can monitor and exchange OPC data on the Internet using the HTTP protocol and SOAP interface.

Data access via OPC XML has a functional scope similar to OPC Data Access. Change-related feedback messages concerning data modification, as with the DCOM interfaces, are not designed for OPC XML due to their loose Internet connection.

OPC Historical Data Access (OPC HDA)

OPC Historical Data Access (OPC HDA) is the specification for the management of archive data. The specification is an extension of the OPC Data Access specification. The WinCC OPC HDA server of WinCC V 6.2 or higher conforms to OPC HDA specification 1.20.

OPC Alarms & Events (OPC A&E)

OPC Alarm & Events is a supplemental specification for the transmission of process alarms and events. The WinCC OPC A&E server of WinCC V 7.0 or higher conforms to OPC A&E specification 1.10.

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

4.4 Compatibility

Introduction

The OPC Foundation provides its members with a software tool for an automated compliance test of OPC server products.

The results submitted are published on the OPC Foundation Web site. The results can be called up from there using the search term "OPC Self-Certified Products".

This tool checks the conformity of WinCC OPC servers with the OPC specifications of the OPC Foundation.

In addition, a regular test is also carried out to determine whether the WinCC OPC server and WinCC OPC DA client are compatible with other OPC products.

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

4.5 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 can be selected for installation:

- WinCC OPC DA server
- WinCC OPC XML DA Server
- WinCC OPC HDA server
- WinCC OPC A&E Server

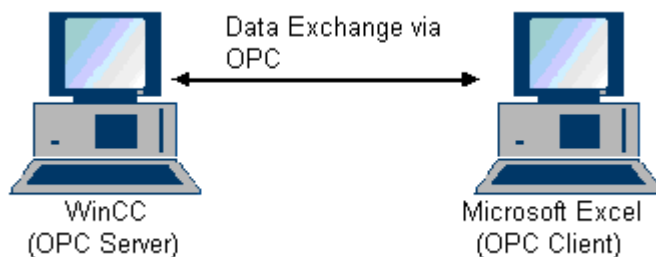
Licensing:

For utilization of the WinCC OPC HDA, the WinCC-OPC-XML-DA server and the WinCC OPC A&E server, a license must be purchased. The "Connectivity Pack" license must be installed on the WinCC server which is utilized as the WinCC OPC HDA server, WinCC OPC XML DA server or WinCC OPC A&E server. For more information, see chapter "Licensing".

Applications

WinCC as OPC DA Server

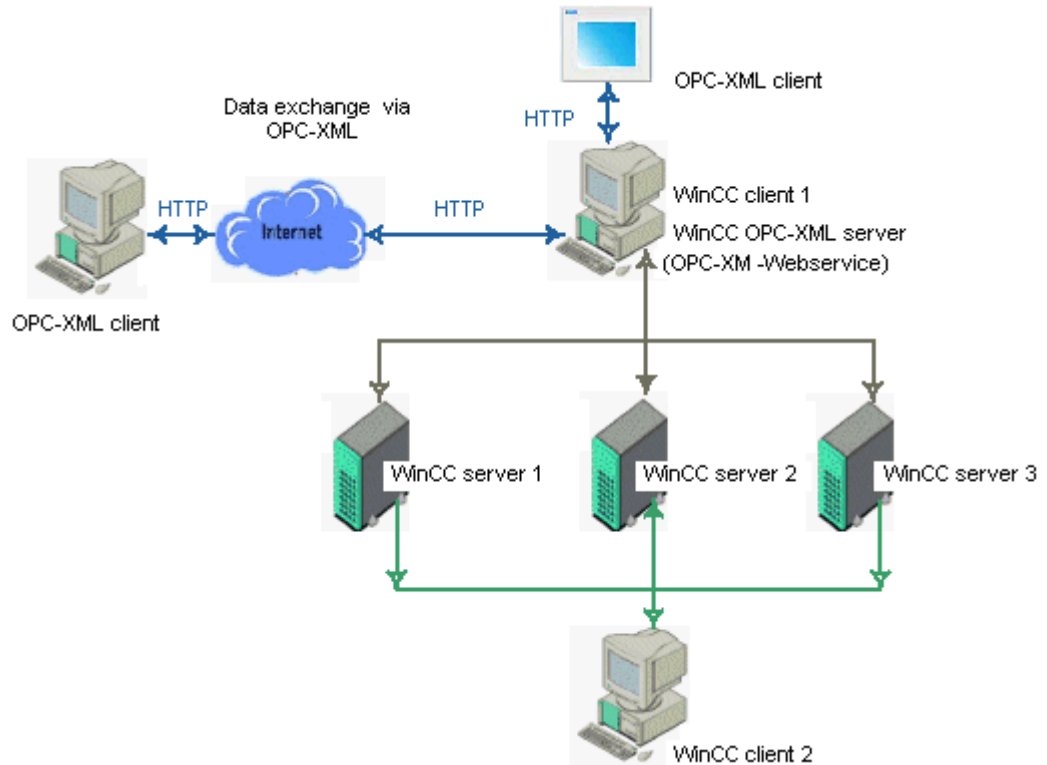
The WinCC OPC DA server makes the data from the WinCC project available to other applications. These applications may be run on the same computer or on computers in a network environment. In this way WinCC tags can for example be exported to Microsoft Excel.



WinCC as OPC-XML-DA-Server

Within a distributed system, WinCC clients have views of several WinCC servers. The WinCC OPC XML DA server provides the OPC XML client with the OPC process data as a web page. The web page can be accessed via the Internet using HTTP. The OPC XML client is no longer

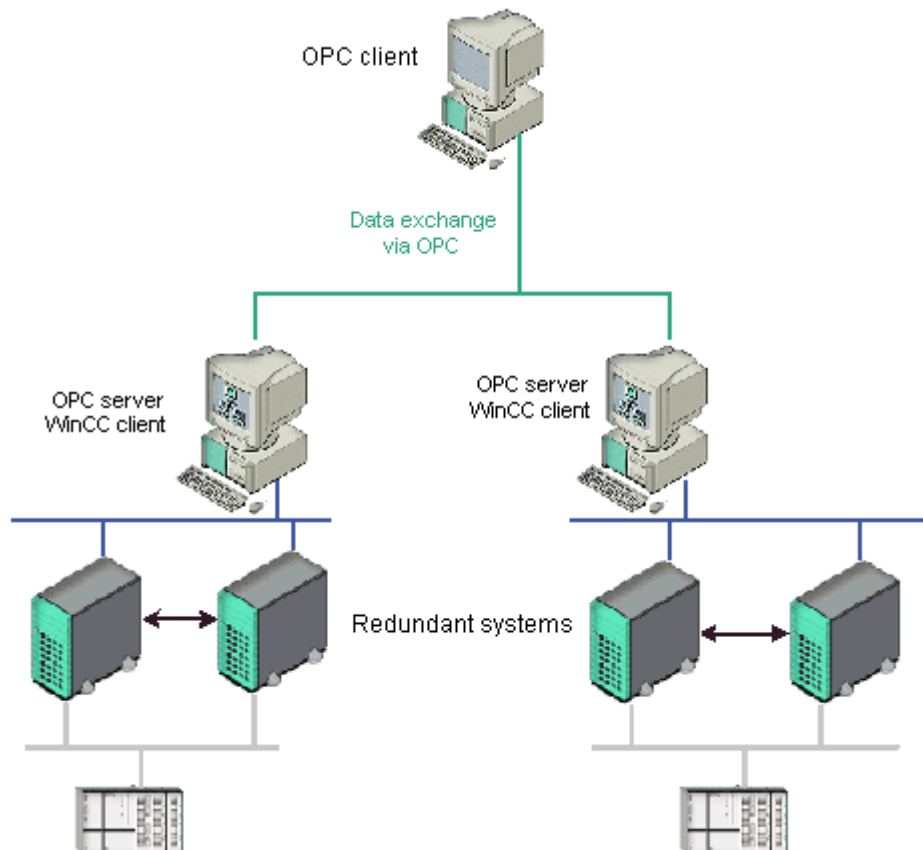
limited to the local network. In this way, OPC XML clients can access WinCC Runtime data via any platform as well as through an intranet or the Internet.



WinCC OPC-Servers in a Redundant System

In a redundant system, the WinCC servers monitor one another in Runtime for early recognition of a server failure. The WinCC OPC-Servers make WinCC Runtime data available to the OPC client, using the OPC software interface.

Any software program based on the corresponding OPC specification may be used as an OPC client. It is thus possible to use the OPC client for the central monitoring of various redundant systems. By creating proprietary OPC clients, most user-specific requirements can be met.



4.6 WinCC OPC XML DA Server

4.6.1 Functionality of WinCC OPC XML DA server

Introduction

The OPC XML DA server from WinCC is realized as a web service of the Microsoft Internet Information Server (IIS).

The WinCC OPC XML DA server provides the OPC XML client with the OPC process data as a web page. The web page can be accessed via the Internet using HTTP. The address of the WinCC OPC XML DA Server is: `<http://<xxx>/WinCC-OPC-XML/DAWebservice.asmx>`

The WinCC OPC XML DA server is not visible in WinCC. When an OPC XML client requests data, the Web service is automatically started by the Web server.

To establish successful OPC communication, the following must be observed:

- The WinCC project of the WinCC OPC XML DA server must be activated.
- The computer of the WinCC OPC XML DA server must be capable of being accessed via HTTP.

Licensing

In order to operate the WinCC OPC XML DA server, the following licenses must be installed on each WinCC computer implemented as an OPC XML server:

- WinCC Basic System
- WinCC Option Connectivity Pack

Special features of tags of "String" type

If you use tags of "string" type that logically stand for floating point values, there can be problems when OPC clients want to write and read.

Description

An OPC client writes to a string tag and does not enter the new value in the form of a string but as float, double or decimal.

Problem

The comma that indicated the decimal value (in German) can be lost. This will result in a wrong value.

This also affects the read access to string tags if the read value is requested in the float, double or decimal format.

Remedy

Only use the respective floating point tags for the floating point values. Access string tags only in string format.

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

4.6.2 Installation

4.6.2.1 Installation

Introduction

OPC XML enables access to the process tags via the Internet. In order to operate OPC XML, a number of additional software components must be installed.

Note

It is essential to observe the sequence of installation steps as described here. Otherwise, there might be problems during installation.

Installing WinCC-OPC-XML-Server under Windows 2000 and Windows XP

1. Installation of the Internet Information Server (IIS)
2. Installation of the ASP.NET under Windows Server 2003
3. Installation Microsoft .NET-Framework V2.0
4. Installation of the WinCC OPC XML Server Using WinCC Setup

4.6.2.2 Installing the Internet Information Service (IIS)

Introduction

The Internet Information Service (IIS) must be installed prior to the installation of WinCC OPC XML Server in Windows 2000 Professional, Windows 2000 Server, Windows 2003 Server or Windows XP.

Note

To install the Internet Information Service in Windows, you must have the necessary write access rights to the registry database. To do this, you must have administrator rights.

Procedure

1. Open the "Add or Remove Programs" dialog and click "Add or Remove Windows Components". The following dialog is opened.
2. Select the check box preceding the "Internet Information Service (IIS)" item in the selection box. On Windows 2003 servers, IIS is accessible at "Windows Components" > "Application Server" > "Internet Information Service". You must also install the ASP.NET under Windows Server 2003



3. Click "Next". The required data is transferred and the necessary configuration is completed by Windows.
4. Close the Assistant by clicking "Finish".

4.6.2.3 Installing Microsoft .NET Framework

Microsoft .NET Framework is the software required for the use of .NET applications. It is also required to operate the OPC XML web service.

Approximately 80 MB of memory space is required on the hard disk for the installation.

Microsoft .NET Framework 2.0 is available from the following sources:

- Download from <http://msdn.microsoft.com/downloads>.
- Microsoft Windows 2003 Server CD

Install the software as instructed by Microsoft.

4.6.2.4 Installing WinCC OPC XML DA Server

Requirement

- Internet Information Server (IIS)
- Installation of the ASP.NET under Windows Server 2003
- Microsoft .NET Framework 2.0

Installation

The WinCC OPC XML DA server can be selected during the installation of WinCC. For more information, refer to the WinCC Information System in chapter "Installation Notes" > "Installing WinCC".

The following settings must be made during the installation:

- Create virtual directory "WinCC-OPC-XML".
- Define the access rights for the directory.

See also

How to Test the Installation (Page 572)

Defining the Security Settings with IIS (Page 570)

4.7 WinCC OPC DA server

4.7.1 Functionality of the WinCC OPC DA Server

Introduction

The WinCC OPC DA Server supports OPC Data Access specifications 1.0, 2.05a and 3.0. This has been confirmed by the compliance test. The WinCC OPC DA Server of WinCC V6.0 SP2 or higher supports the OPC Data Access 3.0 specification.

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 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.

Installation

The WinCC OPC DA server can be selected during the installation of WinCC. After installation, the WinCC OPC DA server is immediately usable without any further configuration.

The WinCC OPC DA server can be implemented on a WinCC server or a WinCC client.

Configuration tip

In your WinCC project you can summarize for structuring purposes tags in tag groups. The tags should not have the same name as the group.

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

Compatibility (Page 495)

Querying the OPC DA Server Name (Page 504)

Using Multiple OPC DA Servers (Page 503)

Example of WinCC to WinCC Connection (Page 506)

Example of WinCC - SIMATIC NET FMS OPC Server Connection (Page 510)

Example of a WinCC - SIMATIC NET S7 OPC Server Connection (Page 512)

Example of the WinCC - Microsoft Excel Connection (Page 518)

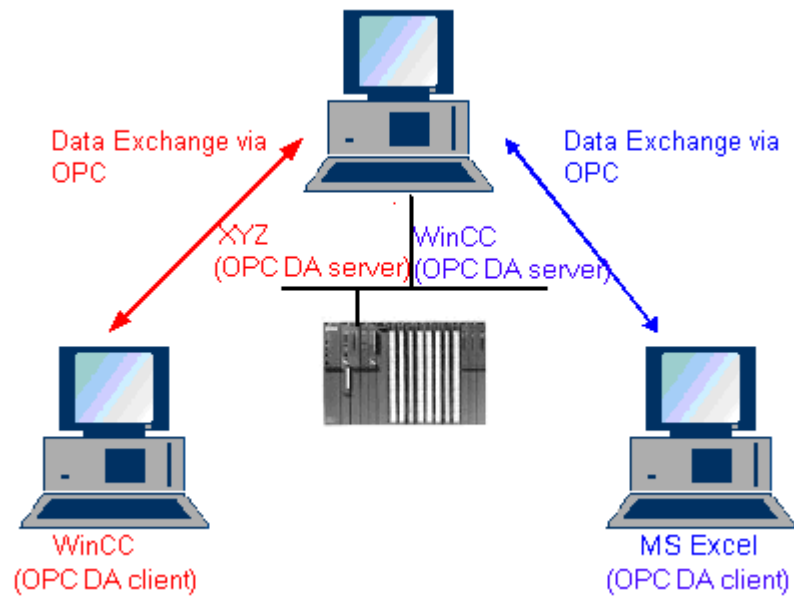
4.7.2 Using Multiple OPC DA Servers

Introduction

More than one OPC DA server may be installed on a computer, and any number may work in parallel.

In this way, the OPC DA server of WinCC and the OPC DA server of another (third-party) provider may be operated independently of one another on the same computer.

The WinCC OPC DA client can access the process data of the automation device via the OPC server of the third-party provider. The OPC DA client of Microsoft Excel can use the WinCC OPC DA server to access the WinCC data.



There are a number of OPC DA servers available from various manufacturers. Each of these OPC DA servers has a unique name (ProgID) for identification. OPC DA clients must use this name to address the OPC server.

The OPC Item Manager can be used to query the name of the OPC DA server. The OPC DA server of WinCC V 7 is named: "OPCServer.WinCC".

See also

Querying the OPC DA Server Name (Page 504)

4.7.3 Querying the OPC DA Server Name

Introduction

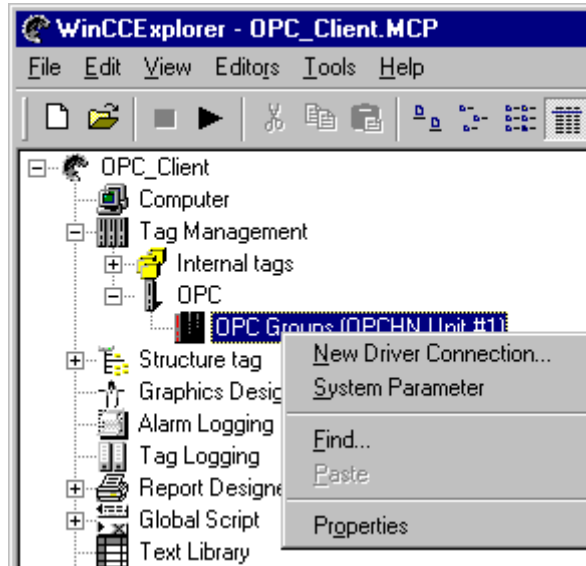
Multiple OPC DA servers can be installed on a single computer. The OPC Item Manager displays the names of the OPC DA servers available to the workstation in a selection window. These OPC DA servers can be run on the same computer or on computers in the network environment.

Requirement

Add the "OPC" channel to the WinCC project of the WinCC OPC DA client.

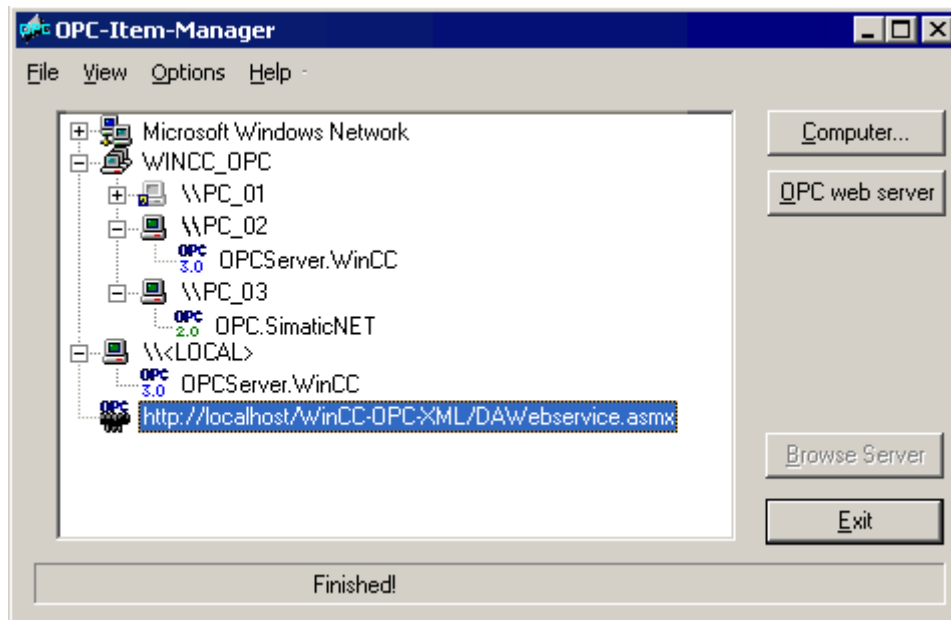
Procedure

1. In the shortcut menu of the channel unit "OPC Groups(OPCHN Unit#1)" on the WinCC OPC DA client, select "System Parameters".



The OPC Item Manager is opened.

2. In the navigation window of the OPC Item Manager, select the name of the computer you wish to access.
3. The OPC Item Manager displays the names of the OPC DA servers that available to your computer in a selection window.



4.7.4 Examples of OPC DA Connections

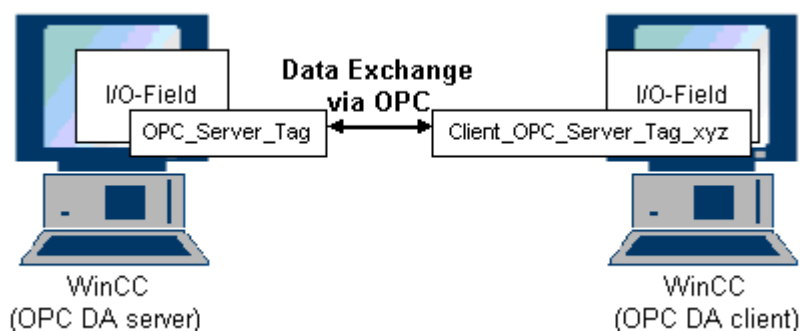
4.7.4.1 WinCC - WinCC Connection

Example of WinCC to WinCC Connection

Introduction

When establishing a WinCC - WinCC connection, data are exchanged between the WinCC OPC DA server and client by means of the "OPC_Server_Tag" WinCC tag. The "Client_OPC_Server_Tag_xyz" WinCC tag on the client reads the "OPC_Server_Tag" WinCC tag on the server. If the value of the "OPC_Server_Tag" tag on the WinCC OPC server changes, the value of the "Client_OPC_Server_Tag_xyz" WinCC tag on the WinCC OPC DA client also changes. Changes on the client are also reflected on the server.

Tag values are displayed in I/O fields on both computers.



Requirements

- Two computers with WinCC projects.
- Both computers must be accessible via their IP addresses.

Configuration Procedure

The following configurations are required to establish a WinCC - WinCC connection:

1. Configuring a WinCC Project on a WinCC OPC DA Server
2. Configuring a WinCC Project on a WinCC OPC DA Client

See also

How to Configure a WinCC Project on a WinCC OPC DA Server (Page 507)

Configuring the WinCC Project on the WinCC OPC DA Client (Page 507)

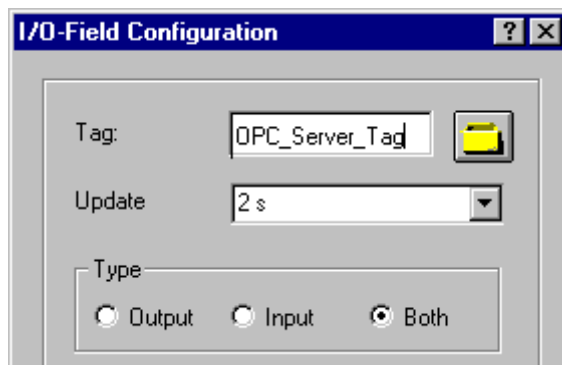
How to Configure a WinCC Project on a WinCC OPC DA Server

Introduction

In this section, a WinCC tag is created in the WinCC project of the WinCC OPC DA server and displayed in an I/O field.

Procedure

1. Select "New Tag" from the shortcut menu of the "Internal Tags" icon on the WinCC OPC DA server. Create a new tag called "OPC_Server_Tag" of the "signed 16-bit value" type.
2. Launch the Graphics Designer and open a new picture.
3. Add an I/O field to the picture. Select the "I/O field" object from the object list under "Smart Objects". The "I/O Field Configuration" dialog is opened.



4. Enter the name "OPC_Server_Tag" in the "Tag" field.
5. Set the update to "2s" and the field type to "I/O field".
6. Click "OK" to close the dialog and save the picture.
7. Enable the WinCC project by clicking the "Activate" button in the Graphics Designer.

See also

Configuring the WinCC Project on the WinCC OPC DA Client (Page 507)

Configuring the WinCC Project on the WinCC OPC DA Client

Introduction

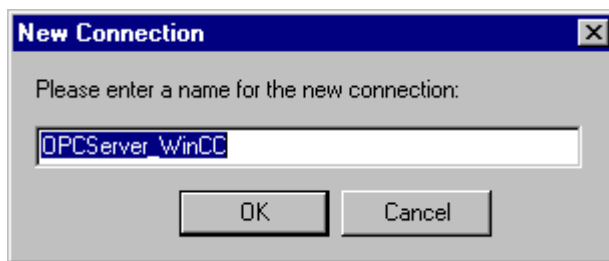
In this section, a WinCC tag is created on the WinCC OPC DA client, in order to read a WinCC tag on the WinCC OPC DA server. The tag value is displayed in an I/O field.

Requirements

- Add the "OPC" channel to the WinCC project of the WinCC OPC DA client.
- Configure an internal tag named "OPC_Server_Tag" of the data type "signed 16-bit value" in the WinCC project of the WinCC OPC DA server.
- Enable the WinCC project of the WinCC OPC DA server.

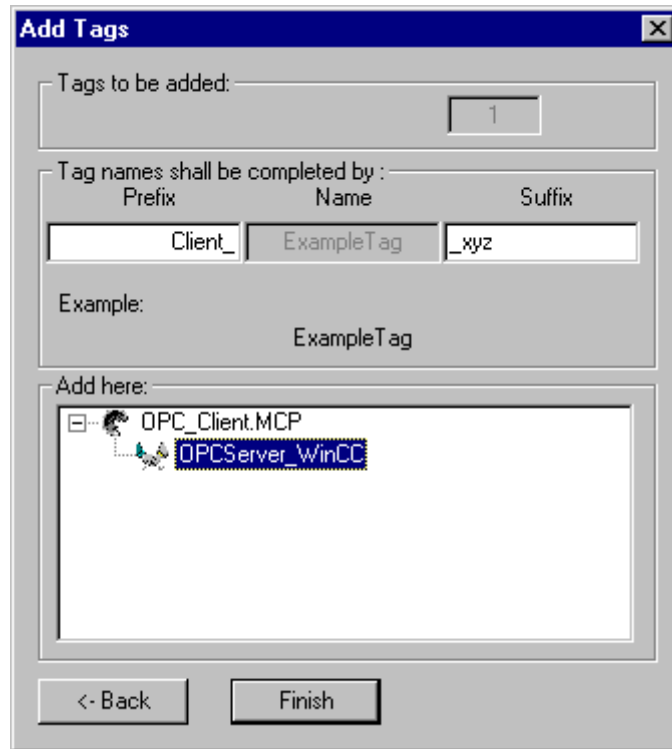
Procedure

1. In the shortcut menu of the channel unit "OPC Groups(OPCHN Unit#1)" on the WinCC OPC DA client, select "System Parameters". The OPC Item Manager is opened.
2. Choose the name of the computer to be used as the OPC DA server from the selection dialog. Select "OPCServer.WinCC" from the list. Click the "Browse Server" button. The "Filter Criteria" dialog is opened.
3. Click the "Next->" button in the "Filter Criteria" dialog. Select the "OPC_Server_Tag" tag in the "OPCServer.WinCC ..." dialog. Click the "Add Items" button.
4. If a connection to the OPC DA server already exists, continue with step 5.
If no connection has been configured, a corresponding message is displayed.
Click "Yes". The "New Connection" dialog is displayed.



Enter "OPCServer_WinCC" as the name of the connection. Click "OK".

5. The "Add Tags" dialog is displayed.
Enter "Client_" in the prefix field and "_xyz" in the suffix field. Select connection "OPCServer_WinCC". Click "Finish".



6. Click the "<- Back" button in the "OPCServer.WinCC ..." dialog. In the "OPC Item Manager", click "Exit" to close the OPC Item Manager.
7. Launch the Graphics Designer and open a new picture. Add an I/O field to the picture. Select the "I/O field" object from the object list under "Smart Objects". The "I/O Field Configuration" dialog is opened.
8. Enter the name "Client_OPC_Server_Tag_xyz" in the "Tag" field. Set the update to "2 s". Set the field type to "I/O field". Close the dialog and save the picture. Enable the WinCC project by clicking the "Activate" button in the Graphics Designer.
9. The value of the configured tags is displayed in the I/O field on both the WinCC OPC DA server and the client. Enter a new value in the I/O field on the WinCC OPC DA server. The new value is displayed in the I/O field on the WinCC OPC DA client.

See also

How to Configure a WinCC Project on a WinCC OPC DA Server (Page 507)

4.7.4.2 WinCC - SIMATIC NET FMS OPC Server Connection

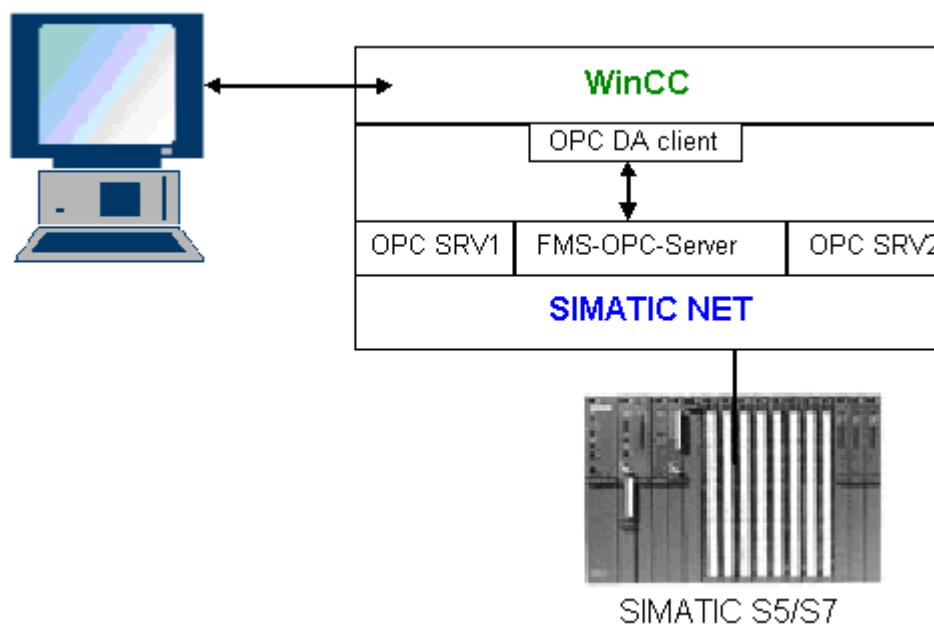
Example of WinCC - SIMATIC NET FMS OPC Server Connection

Introduction

During the installation of SIMATIC NET, you can select the OPC server to be installed. In the following example, a connection between WinCC and SIMATIC NET FMS OPC server is configured. Data from the automation device is made available to WinCC through the SIMATIC NET FMS OPC server.

In this example, WinCC is used as the WinCC OPC DA client. The OPC Item Manager displays the indexes of the object list configured for the automation device.

The current value of the tag is displayed in an I/O field. As soon as the value of the tags on the SIMATIC NET FMS OPC server changes, the new value is reflected on the process picture on the WinCC OPC DA client. Conversely, a value entered in the I/O field is sent to the automation device.



Requirements

- A computer with WinCC, SIMATIC NET software.
- A configured SIMATIC NET FMS OPC server. For additional information regarding the setup of SIMATIC NET S7 OPC servers refer to the SIMATIC NET documentation.

Configuration steps

The following configuration is required in the WinCC project of the WinCC OPC DA client:

1. Configuring a WinCC - SIMATIC NET FMS OPC server connection

Communication Manual

The communication manual contains additional information and extensive examples for the channel configuration. This manual is available for download on the Internet:

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

Search by order number:

- A5E00391327

How to Configure the WinCC - SIMATIC NET FMS OPC Server Connection

Introduction

In this section, a WinCC tag that accesses an FMS index is configured in the WinCC project of the WinCC OPC DA client. The tag value is displayed in an I/O field.

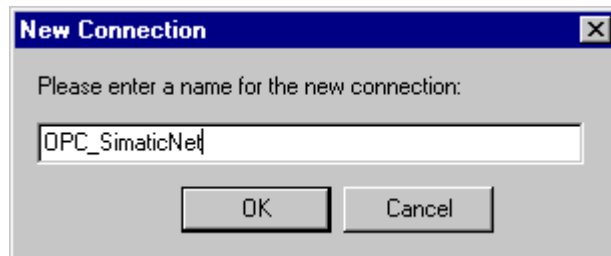
Requirement

- Add the "OPC" channel to the WinCC project of the WinCC OPC DA client.

Procedure

1. In the shortcut menu of the channel unit "OPC Groups(OPCHN Unit#1)" on the WinCC OPC DA client, select "System Parameters". The OPC Item Manager is opened.
2. Choose the name of the computer to be used as the OPC DA server from the selection dialog. Select "OPC.SIMATICNet" from the list.
Click the "Browse Server" button. The "Filter Criteria" dialog is opened.
3. Click the "Next->" button in the "Filter Criteria" dialog. The "OPC.SIMATICNet.." dialog is opened. All FMS indexes configured are displayed in a selection list. Select an index. Click the "Add Items" button.

4. If a connection to the SIMATIC NET FMS OPC server already exists, continue with step 5.
If no connection has been configured, a corresponding message is displayed.
Click "Yes". The "New Connection" dialog is displayed.



Enter "OPC_SlimaticNET" as the name of the connection. Click "OK".

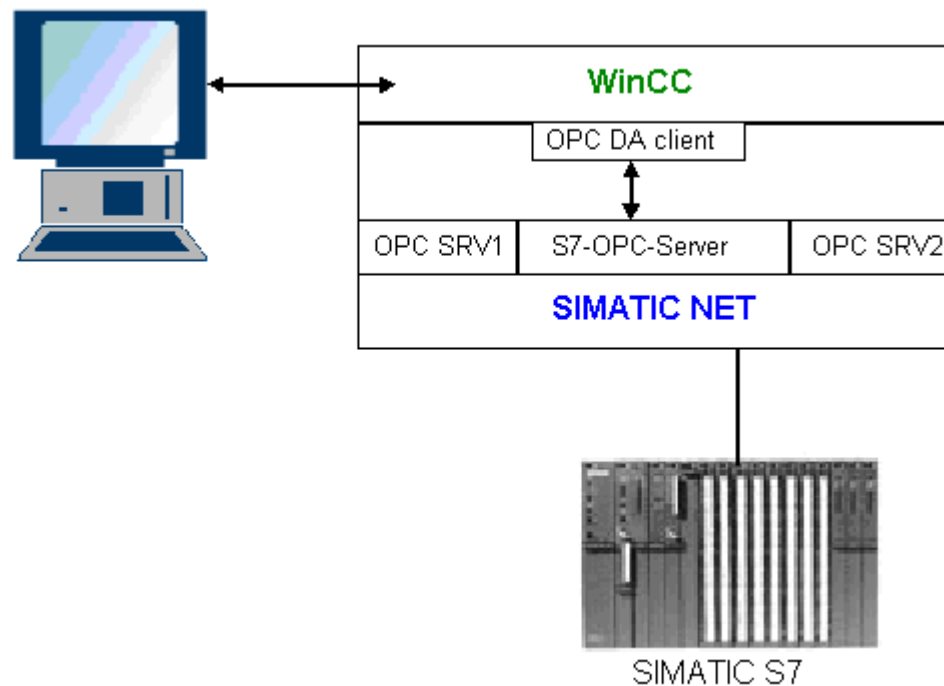
5. The "Add Tags" dialog is opened.
Enter "Client_" in the prefix field and "_xyz" in the suffix field. Select the connection "OPC_SimaticNET". Click "Finish".
6. Click the "<- Back" button in the "OPC.SIMATICNet .." dialog. In the "OPC Item Manager", click "Exit" to close the OPC Item Manager.
7. Launch the Graphics Designer and open a new picture. Add an I/O field to the picture. Select the "I/O field" object from the object list under "Smart Objects". The "I/O Field Configuration" dialog is opened.
8. Enter the name of the tags in the "Tag" field. Set the update to "2s". Set the field type to "I/O field".
9. Click "OK" to close the dialog and save the picture. Enable the WinCC project by clicking the "Activate" button in the Graphics Designer.
10. The current value of the FMS index is shown in the I/O field. The value is updated every two seconds. Enter a value in the I/O field. The changed value is passed to the automation device.

4.7.4.3 WinCC - SIMATIC NET S7-OPC Server Connection

Example of a WinCC - SIMATIC NET S7 OPC Server Connection

During the installation of SIMATIC NET, you can select the OPC server to be installed. In the following example, a WinCC - SIMATIC NET S7 OPC server is configured. Data from the automation device is made available to the WinCC client via the SIMATIC NET S7 OPC server.

The current value of the tag is displayed in an I/O field on the WinCC OPC client. As soon as the value of the tags on the SIMATIC NET S7 OPC server changes, the changed value is shown on the process picture. Conversely, a value entered in the I/O field is sent to the automation device.



Requirements

- A computer with WinCC, SIMATIC NET software.
- A configured SIMATIC NET S7 OPC Server. For additional information regarding the setup of SIMATIC NET S7 OPC servers refer to the SIMATIC NET documentation.

Configuration steps

The following configurations are required to establish a WinCC - SIMATIC NET S7 OPC server connection:

1. Adding Tags to a SIMATIC NET S7 OPC Server
2. Configuring Access to the Tags on a SIMATIC NET S7 OPC Server

Communication Manual

The communication manual contains additional information and extensive examples for the channel configuration. This manual is available for download on the Internet:

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

Search by order number:

- A5E00391327

Adding Tags to the SIMATIC NET S7 OPC Server

Introduction

In order for the OPC Item Manager to display the tags, they must be added to the address space of the SIMATIC NET S7 OPC server. The "OPC Scout" program is used for the configuration. OPC Scout is set up by the SIMATIC NET installer. For this example, the marker word "0" in the automation device is addressed.

Table of Parameters Used

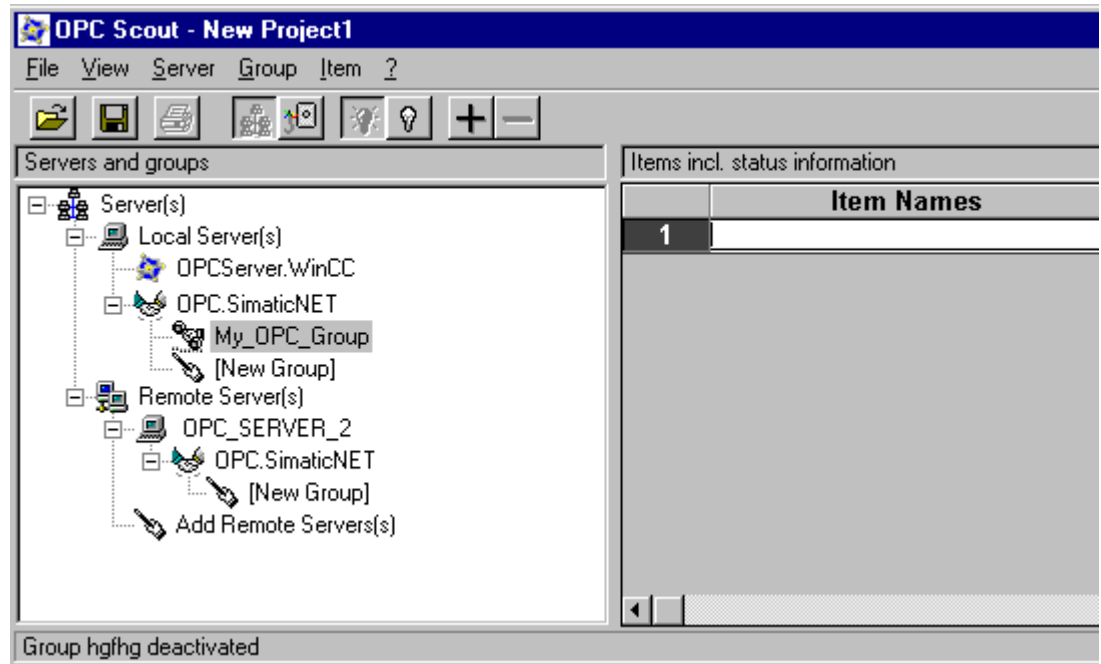
Parameter	Value
Data type	W
Range byte	0
No. values	1
Item alias	MW0

Requirements

- Configure an S7 connection in the SIMATIC NET software. For more information, refer to the SIMATIC NET documentation.

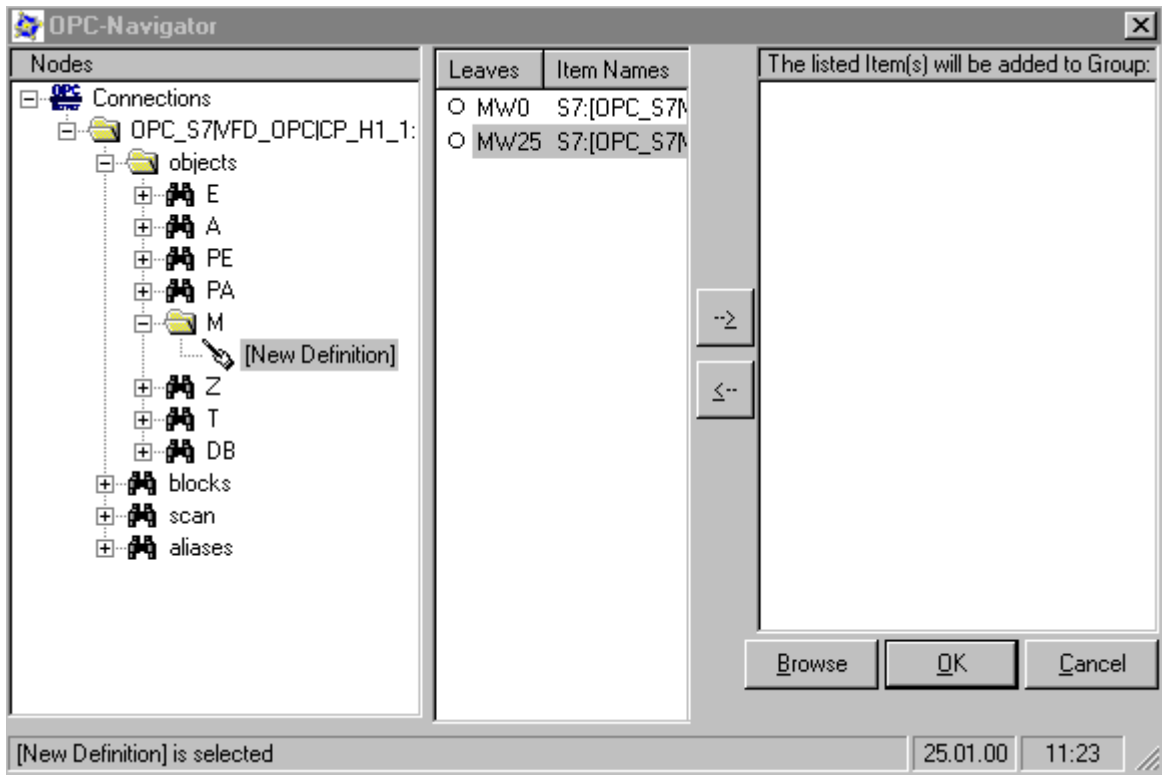
Procedure

1. Open the "OPC Scout" via Start → "Programs" → "SimaticNet" → "OPCServer" → "OPCScout" .

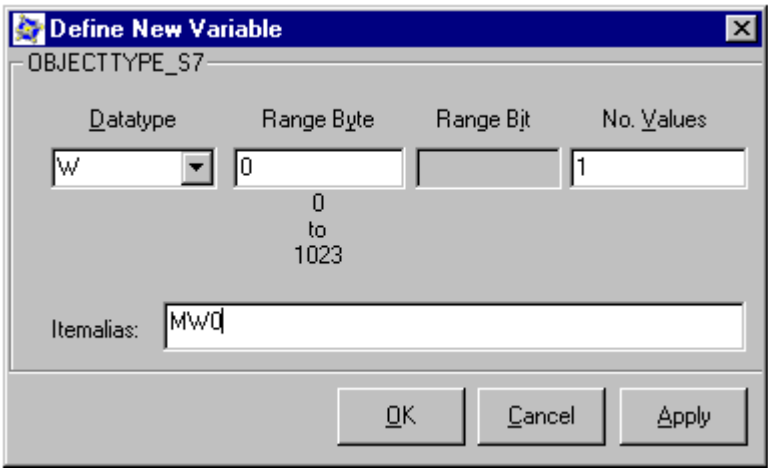


2. Select "OPC.SimaticNet" under "Local Server(s)". If the SIMATIC S7 OPC server is not run on the same computer, select "Add Remote Server(s)" in the "Server(s)" shortcut menu. Enter the name of the computer used as the OPC server in the "Add Remote Server(s)" dialog, then click "OK" to close the dialog.
3. Select "Connect" in the "OPC.SimaticNet" shortcut menu. The "Add Group" dialog is displayed. Enter a name for the group. Click "OK" to close the dialog.

4. Select "Add Item" from the shortcut menu of the added group. The "OPC Navigator" is opened.



5. Select "M" (marker) under "Objects" in the "OPC Navigator". Double-click "(New Definition)" to open the "Define New Tag" dialog.
6. Enter the parameters from the table in the "Define New Tag" dialog.



- Click "OK" to close the "Define New Tag" dialog.
7. Mark the tag "MW0" in the "Leaves" area of the OPC Navigator. Click the "-->" button. Click "OK" in the OPC Navigator.

See also

Configuring Access to the Tags of the SIMATIC NET S7 OPC Server (Page 517)

Configuring Access to the Tags of the SIMATIC NET S7 OPC Server

Introduction

In this section, a WinCC tag is configured in the WinCC project of the WinCC OPC DA client. This tag accesses the tag "MW0" in the address space of the SIMATIC NET S7 OPC server. The tag value is displayed in an I/O field.

Requirements

- Create the tag "MW0" using the OPC Scout.
- Add the "OPC" channel to the WinCC project of the WinCC OPC DA client.

Procedure

1. Select "System Parameters" in the shortcut menu of "OPC Groups(OPCHN Unit#1)". The OPC Item Manager is opened.
2. Choose the name of the computer to be used as the OPC server from the selection dialog. Select "OPC.SIMATICNet" from the list. Click the "Browse Server" button. The "Filter Criteria" dialog is opened.
3. Click the "Next->" button in the "Filter Criteria" dialog. The "OPC.SIMATICNet.." dialog is opened. Select the "MW0" tag. Click the "Add Items" button.
4. If a connection to the SIMATIC NET FMS OPC server already exists, continue with step 5.
If no connection has been configured, a corresponding message is displayed. Click "Yes". The "New Connection" dialog is displayed.



Enter "OPC_SlimaticNET" as the name of the connection. Click "OK".

5. The "Add Tags" dialog is opened. Enter "Client_" in the prefix field and "_xyz" in the suffix field. Select the connection "OPC_SimaticNET". Click "Finish".
6. Click the "<- Back" button in the "OPC.SIMATICNet .." dialog. In the "OPC Item Manager", click "Exit" to close the OPC Item Manager.

7. Start Graphics Designer and open a picture. Add an I/O field to the picture. Select the "I/O field" object from the object list under "Smart Objects". The "I/O Field Configuration" dialog is opened.
8. Enter the name "Client_MW0_xyz" in the "Tag" field. Set the update to "2s". Set the field type to "I/O field".
9. Close the dialog and save the picture. Enable the WinCC project by clicking the "Activate" button in the Graphics Designer.
10. The I/O field on the WinCC OPC DA client displays the current value of the S7 tags. The value is updated every two seconds. Enter a value in the I/O field. The changed value is passed to the automation device.

See also

Adding Tags to the SIMATIC NET S7 OPC Server (Page 514)

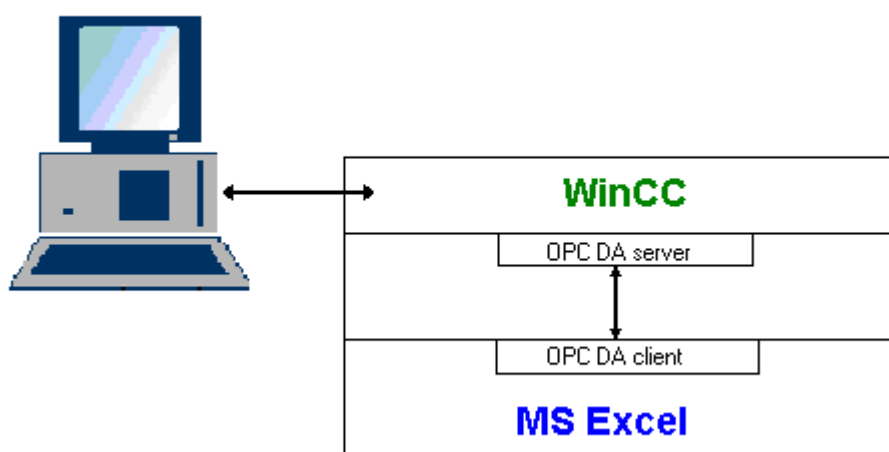
4.7.4.4 WinCC - Microsoft Excel Connection

Example of the WinCC - Microsoft Excel Connection

Introduction

In this example, an OPC DA client is created in Microsoft Excel using the Visual Basic Editor. The OPC DA client reads a WinCC tag in the WinCC project of the WinCC OPC DA server and writes the value into a cell. If a new value is entered in the cell, the value is passed to the WinCC OPC DA server.

A computer on which both WinCC and Microsoft Excel are installed is used for the connection.



Configuration steps

The following configurations must be made in Microsoft Excel:

1. Creating an OPC DA client in Visual Basic Editor of Microsoft Excel
2. Configuring access to a WinCC tag in Microsoft Excel

See also

How to Configure the Access to a WinCC Tag in Microsoft Excel (Page 523)

Creating an OPC DA Client in Microsoft Excel (Page 519)

Creating an OPC DA Client in Microsoft Excel

Introduction

To use Microsoft Excel as an OPC DA client, a special script must be created in the Visual Basic Editor of Microsoft Excel.

Requirements

Basic knowledge of Visual Basic Editor in Microsoft Excel.

Procedure

1. Open Microsoft Excel with a new workbook.
2. In the "Tools" menu of the Visual Basic Editor, → click "Macro". The Visual Basic Editor for Microsoft Excel is opened.
3. In the "Tools" menu of the Visual Basic Editor, select "References...". The "References - VBAProject" dialog is displayed. Locate entry "Siemens OPC DAAutomation 2.0" in the list of available references. Select the corresponding check box. Click "OK".
4. Copy the script shown below. This script is only available in the online help.
5. Open a new code window by double-clicking "Sheet1" in the project window of the Visual Basic Editor.
6. Paste the script into the code window.
7. Select "Save" from the "File" menu. Select "Close and Return to Microsoft Excel" from the "File" menu.

Script

```
Option Explicit  
Option Base 1
```

```

Const ServerName = "OPCServer.WinCC"

Dim WithEvents MyOPCServer As OpcServer
Dim WithEvents MyOPCGroup As OPCGroup
Dim MyOPCGroupColl As OPCGroups
Dim MyOPCItemColl As OPCItems
Dim MyOPCItems As OPCItems
Dim MyOPCItem As OPCItem

Dim ClientHandles(1) As Long
Dim ServerHandles() As Long
Dim Values(1) As Variant
Dim Errors() As Long
Dim ItemIDs(1) As String
Dim GroupName As String
Dim NodeName As String

'-----
--
' Sub StartClient()
' Purpose: Connect to OPC_server, create group and add item
'-----
--
Sub StartClient()
    ' On Error GoTo ErrorHandler
    '----- We freely can choose a ClientHandle and GroupName
    ClientHandles(1) = 1
    GroupName = "MyGroup"
    '----- Get the ItemID from cell "A1"
    NodeName = Range("A1").Value
    ItemIDs(1) = Range("A2").Value
    '----- Get an instance of the OPC-Server
    Set MyOPCServer = New OpcServer
    MyOPCServer.Connect ServerName, NodeName

    Set MyOPCGroupColl = MyOPCServer.OPCGroups

```



```
'----- Set the default active state for adding groups
MyOPCGroupColl.DefaultGroupIsActive = True
'----- Add our group to the Collection
Set MyOPCGroup = MyOPCGroupColl.Add(GroupName)

Set MyOPCItemColl = MyOPCGroup.OPCItems
'----- Add one item, ServerHandles are returned
MyOPCItemColl.AddItem 1, ItemIDs, ClientHandles, ServerHandles,
Errors
'----- A group that is subscribed receives asynchronous
notifications
MyOPCGroup.IsSubscribed = True
Exit Sub

ErrorHandler:
MsgBox "Error: " & Err.Description, vbCritical, "ERROR"
End Sub

'-----
--
' Sub StopClient()
' Purpose: Release the objects and disconnect from the server
'-----
--
Sub StopClient()
'----- Release the Group and Server objects
MyOPCGroupColl.RemoveAll
'----- Disconnect from the server and clean up
MyOPCServer.Disconnect
Set MyOPCItemColl = Nothing
Set MyOPCGroup = Nothing
Set MyOPCGroupColl = Nothing
Set MyOPCServer = Nothing
End Sub

'-----
--
```

```
' Sub MyOPCGroup_DataChange()
' Purpose: This event is fired when a value, quality or timestamp in
' our Group has changed
'-----
--
'----- If OPC-DA Automation 2.1 is installed, use:
Private Sub MyOPCGroup_DataChange(ByVal TransactionID As Long, ByVal
NumItems As Long, ClientHandles() As Long, ItemValues() As Variant,
Qualities() As Long, TimeStamps() As Date)
'----- Set the spreadsheet cell values to the values read
Range("B2").Value = CStr(ItemValues(1))
Range("C2").Value = Hex(Qualities(1))
Range("D2").Value = CStr(TimeStamps(1))
End Sub

'-----
--
' Sub worksheet_change()
' Purpose: This event is fired when our worksheet changes, so we can
' write a new value
'-----
--
Private Sub worksheet_change(ByVal Selection As Range)
'----- Only if cell "B3" changes, write this value
If Selection <> Range("B3") Then Exit Sub
Values(1) = Selection.Cells.Value
'----- Write the new value in synchronous mode
MyOPCGroup.SyncWrite 1, ServerHandles, Values, Errors
End Sub
```

See also

[How to Configure a WinCC Project on a WinCC OPC DA Server \(Page 507\)](#)

How to Configure the Access to a WinCC Tag in Microsoft Excel

Introduction

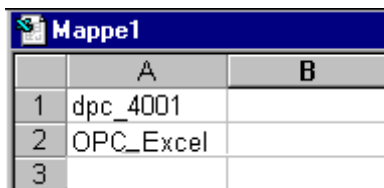
The Excel OPC DA client reads a WinCC tag of the WinCC OPC DA server and writes the value of the tag into a cell. In the WinCC project of the WinCC OPC DA server, the value of the tag is displayed in an I/O field. If the tag value in a cell is changed, this alters the value in the I/O field of the WinCC OPC DA server.

Requirements

- Configure an internal tag named "OPC_Excel" with data type "signed 16-bit value" in the WinCC project of the WinCC OPC DA server.
- Write the value of the "OPC_Excel" tag to an I/O field on the WinCC project of the WinCC OPC DA server.
- Enable the WinCC project of the WinCC OPC DA server.

Procedure

1. In Microsoft Excel, enter the name of the computer used as the OPC server in cell A1. In cell A2, enter the tag name "OPC_Excel".



	A	B
1	dpc_4001	
2	OPC_Excel	
3		

2. In the "Tools" menu in Excel, select "Macro" → "Macros". The "Macro" dialog is opened. Select the entry "Sheet1.StartClient" from the list of macros. Click "Run" to start the OPC client.
3. The value of the tag is written into cell B2, the quality code into C2 and the timestamp into D2.
4. Enter a new value in cell B3. The changed value is displayed in the I/O field on the WinCC OPC server.
5. In the "Tools" menu in Excel, select "Macro" → "Macros". The "Macro" dialog is opened. Select the entry "Sheet1.StopClient" from the list of macros. Click "Run" to stop the OPC client.

4.8 WinCC OPC HDA server

4.8.1 Functionality of the WinCC OPC HDA server

Introduction

The WinCC OPC HDA server is a DCOM application making data needed from the archive system available to the OPC HDA client. Access the data using Item Handles. Read or write access is enabled. The data can also be analyzed.

The WinCC OPC HDA server supports the OPC Historical Data Access 1.20 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 most important information. For more information, refer to the "OPC Historical Data Access 1.20" specification.

Installation

The WinCC OPC HDA server can be selected during the installation of WinCC. It is possible to select whether access is made to the WinCC archive system with or without write function. After installation, the WinCC OPC DA server is immediately available for use without any additional configuration.

In the case of installation without write access, the data in the WinCC archive system can only be read and analyzed. In the case of write access, data in the WinCC archive system can be analyzed, added, deleted and updated.

The WinCC OPC HDA server can be implemented on a WinCC server or a WinCC client.

Licensing

In order to operate the WinCC OPC HDA server, the following licenses must be installed on each WinCC computer implemented as an OPC HDA server:

- WinCC Basic System
- WinCC Option Connectivity Pack

OPC HDA Client

All OPC HDA clients that conform to the OPC Historical Data Access 1.20 specification can access the WinCC OPC HDA server. Proprietary OPC HDA clients can also be used. By creating proprietary OPC HDA clients, most user-specific requirements can be met.

Examples of how an OPC HDA client can be used include:

- Analysis and evaluation of archived data
- Statistical process control of archives from different OPC HDA servers

To request for historical values using OPC HDA client, you need to take care of the following during configuration:

- Select a query cycle in such a way that the client can receive the requested data before the next query is sent. Too short cycles can lead to high time delays while receiving data.
- CPU load of the WinCC server depends on the number of tags per query.

Write access to cyclic archive with configured swapping out

In runtime, the data is modified in the cyclic archives on the WinCC server.

Changes are accepted into the swapped-out archive only when the data is changed almost immediately after being created.

If the concerned archive segment of the circulation archive has already been swapped out, then the change is not done subsequently in the swapped-out archive. Even the modified data is deleted when you delete the archive segment on the WinCC server.

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

Compatibility (Page 495)

Quality codes (Page 530)

Data Structure of a WinCC OPC HDA Server (Page 525)

4.8.2 Data Structure of a WinCC OPC HDA Server

4.8.2.1 Data Structure of a WinCC OPC HDA Server

Introduction

The data on the WinCC OPC HDA server are structured. The available data structures are listed below. This is not a detailed description, but rather a summary of the most important information. For more information, refer to the "OPC Historical Data Access 1.20" specification.

Data structure

	Description
Attributes	Provide additional quality characteristics for the raw data. Attributes include data type, specifications re. archiving, etc. For more information, see the overview of supported attributes.
Assemblies	Summarize raw data of a specified time interval. Aggregates include average value, minimum, maximum, etc. For more information, see overview of supported aggregates.

	Description
StartTime/ EndTime	Set the beginning and end point for the time interval.
Bounding values	Values recorded at the beginning and end. If no bounding values are available, the values closest to these times are used as bounding values.
Raw data	Data from the WinCC archive system of a particular time interval. These data include a time stamp and quality rating.
Item handle	Unique assignment to a WinCC archive tag.
ItemID	Unique identifier of the WinCC archive tag. The ItemID can be used to get an item handle.

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

Overview of the supported functions (Page 528)

Time Format of a WinCC OPC HDA Server (Page 528)

Overview of the supported attributes (Page 526)

Overview of the supported assemblies (Page 527)

4.8.2.2 Overview of the supported attributes

Introduction

The following table contains the attributes supported by the WinCC OPC HDA server. For more information, refer to the "OPC Historical Data Access 1.20" specification.

Attributes

Attribute	Attribute ID	Description
ItemID	OPCHDA_ITEMID	Indicates the WinCC archive tag to be accessed.
Item data type	OPCHDA_DATA_TYPE	Indicates the data type of the WinCC archive tag.
Description	OPCHDA_DESCRIPTION	Returns a description of the WinCC archive tag. The description is defined in the WinCC Tag Logging.
Engineering units	OPCHDA_ENG_UNITS	Sets the display of measurement units. The labeling is defined in the WinCC Tag Logging.

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

Data Structure of a WinCC OPC HDA Server (Page 525)

4.8.2.3 Overview of the supported assemblies

Introduction

The following table lists the aggregates supported by the WinCC OPC HDA server. For more information, refer to the "OPC Historical Data Access 1.20" specification.

Assemblies

Assembly	Description
OPCHDA_COUNT	Returns the raw data count for the specified time interval.
OPCHDA_START	Returns the initial value of the raw data at the beginning of the time interval.
OPCHDA_END	Returns the final value of the raw data at the end of the time interval.
OPCHDA_AVERAGE	Returns the average value of the raw data for the specified time interval.
OPCHDA_TIMEAVERAGE	Returns the time-weighted average of the raw data for the specified time interval.
OPCHDA_TOTAL	Returns the sum total value for the specified time interval.
OPCHDA_STDEV	Returns the standard deviation of the raw data for the specified time interval.
OPCHDA_MINIMUMACTUALTIME	Returns the minimum value of the raw data and its time stamp for the specified time interval.
OPCHDA_MINIMUM	Returns the minimum value of the raw data for the specified interval.
OPCHDA_MAXIMUMACTUALTIME	Returns the maximum value of the raw data and its time stamp for the specified time interval.
OPCHDA_MAXIMUM	Returns the maximum value of the raw data for the specified interval.
OPCHDA_DELTA	Returns the difference between the first and last value in the raw data for the specified time interval.
OPCHDA_REGSLOPE	Returns the slope of the regression line of the raw data for the specified time interval.
OPCHDA_REGCONST	Returns the regression value of the raw data at the starting point.
OPCHDA_REGDEV	Returns the standard deviation of the regression of the raw data in the specified time interval.
OPCHDA_VARIANCE	Returns the variance of the raw data for the specified time interval.
OPCHDA_RANGE	Returns the difference between OPCHDA_MAXIMUM and OPCHDA_MINIMUM of the raw data for the specified time interval.
OPCHDA_DURATIONGOOD	Returns the period of time in which the quality of the raw data was good. The period is indicated in seconds.
OPCHDA_DURATIONBAD	Returns the period of time in which the quality of the raw data was bad. The period is indicated in seconds.
OPCHDA_PERCENTGOOD	Returns the percentage of the raw data of good quality.
OPCHDA_PERCENTBAD	Returns the percentage of the raw data of bad quality.
OPCHDA_WORSTQUALITY	Returns the worst quality of the raw data for the specified time interval.

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

Data Structure of a WinCC OPC HDA Server (Page 525)

Functionality of the WinCC OPC HDA server (Page 524)

4.8.2.4 Overview of the supported functions

Introduction

The following tables list the functions supported by the WinCC OPC HDA server. These functions can be used by the OPC HDA client for data exchange. For more information, refer to the "OPC Historical Data Access 1.20" specification.

Read

Function	Description
ReadRaw	Returns the raw data, its quality and time stamp for the specified time interval.
ReadProcessed	Returns the calculated value, the quality of the value and the time stamp for the specified time interval. The calculated value is determined by the selected aggregate.
ReadAtTime	Returns the raw data, its quality and time stamp for a particular time interval. If no value is available, the value for this point is interpolated.
ReadAttribute	Returns the item attributes and time stamp for the specified time interval.

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

Functionality of the WinCC OPC HDA server (Page 524)

4.8.2.5 Time Format of a WinCC OPC HDA Server

Introduction

The time interval is specified on the WinCC OPC HDA server by setting the starting and ending times. The specified time interval determines the observation period for the historical data. When specifying the times, certain formats must be maintained.

The following options are available for the specification of times:

- Absolute based on UTC
- Relative to the local time of the server

Absolute Value According to UTC

By default, the WinCC OPC HDA server uses the coordinated world time (UTC) as its time base. This time corresponds to the Greenwich Mean Time (Central European Time minus an hour).

Time format

YYYY/MM/DD hh:mm:ss.msmsms

Parameters

YYYY = year

MM = month

DD = day

hh = hours

mm = minutes

ss = seconds

ms = milliseconds

Input example

2002/06/10 09:27:30.000

Specification of Time Relative to Local Time

For this option, the time is entered relative to the local time of the server. The local time zone is set on the computer's "Date/Time" control panel.

Time format

keyword +/-offset1 +/-offset(n)

The offset is the deviation from the local time of the server.

Keywords

NOW = current local time on the server

SECOND = current second

MINUTE = current minute

HOURL = current hour

DAY = current day

WEEK = current week

MONTH = current month

YEAR = current year

Offset

+/-S = deviation in seconds

+/-M = deviation in minutes

+/-H = deviation in hours

+/-D = deviation in days

+/-W = deviation in weeks

+/-MO = deviation in months

+/-Y = deviation in years

Example:

DAY - 1D = previous day

DAY-1D + 7H30 = previous day at 7:30

MO-1D+5H = last day of the previous month at 5:00.

NOW-1H15M = one hour and 15 minutes ago

YEAR+3MO= April of this year

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

Functionality of the WinCC OPC HDA server (Page 524)

4.8.3 Quality codes

Introduction

Quality codes are used to evaluate the status and quality of the raw data. The table below shows the quality codes for OPC HDA. For more information, refer to the "OPC Historical Data Access 1.20" specification.

Quality Codes of the WinCC OPC HDA Server

Code	OPC	Description	Quality
0x00040000	OPCHDA_RAW	Indicates the quality of raw data transmission.	GOOD BAD UNCERTAIN
0x00080000	OPCHDA_CALCULATED	Indicates the quality of calculated data transmission.	GOOD BAD UNCERTAIN
0x00100000	OPCHDA_NOBOUND	No bounding values were found at the starting or ending point.	BAD
0x00200000	OPCHDA_NODATA	No raw data were found for the specified time interval.	BAD
0x00400000	OPCHDA_DATALOST	The raw data in the selected interval were not completely archived.	BAD

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

4.8.4 Supported Write-Accesses

Introduction

The following table shows the write accesses supported by the WinCC OPC HDA server.

Table element:

	Description
Cyclic archive	The process values to be archived are stored in a cyclic archive. The cyclic archive consists of a configurable number of data buffers. The size and a period of time (e.g. in days) for the data buffer are defined. If all data buffers are full, the process data in the first data buffer is overwritten.
Cyclic archive after swapping	In order to protect process data in the data buffers from being overwritten process, it can be swapped (exported).
	Supported by WinCC.
	Not supported by WinCC.

Write Accesses

Adding process values later

Cyclic archive	Cyclic archive after swapping	Supported by WinCC	Description
Yes	No		When the time period is contained in the cyclic archive, a process value can be added later.
Yes	Yes		The data buffer of the corresponding time period is swapped to an archive backup. Process values cannot be added later to an archive backup.
No	No		The cyclic archive is not available. The process value cannot be stored.
No	Yes		The cyclic archive is not available. The process value cannot be stored.

Adding process values in Runtime

Cyclic archive	Cyclic archive after swapping	Supported by WinCC	Description
Yes	No		The process value is added in the data buffer currently valid for the cyclic archive.

Inserting future process values

Cyclic archive	Cyclic archive after swapping	Supported by WinCC	Description
YES	No		During write access, no values can be added in the future.
No	No		With write access, no values can be added in the future.

Deleting process values

Cyclic archive	Cyclic archive after swapping	Supported by WinCC	Description
Yes	No		When the time period is contained in the cyclic archive, a process value can be deleted.
Yes	Yes		The data buffer of the corresponding time period is swapped to an archive backup. Process values can be deleted from an archive backup.

Cyclic archive	Cyclic archive after swapping	Supported by WinCC	Description
No	No		The cyclic archive is not available. The process value cannot be stored.
No	Yes		The cyclic archive is not available. The process value cannot be stored.

Editing process values

Cyclic archive	Cyclic archive after swapping	Supported by WinCC	Description
Yes	No		When the time period is contained in the cyclic archive, a process value can be edited.
Yes	Yes		The data buffer of the corresponding time period is swapped to an archive backup. Process values cannot be edited in an archive backup.
No	No		The cyclic archive is not available. The process value cannot be stored.
No	Yes		The cyclic archive is not available. The process value cannot be stored.

4.8.5 Example of an OPC HDA Connection

4.8.5.1 Example of an OPC HDA Connection

Introduction

In the example below, a connection between WinCC and the OPC HDA client is configured. Data from the WinCC archive system are made available via the WinCC OPC HDA server. The OPC HDA client accesses the data via item handles. To simplify the configuration process, the OPC HDA browser is used.

The OPC HDA client from the OPC Foundation is used. All OPC HDA clients conforming to the OPC Historical Data Access 1.20 specification can access the WinCC OPC HDA server.

Requirements

- Create an internal tag named "OPC_HDA" with data type "unsigned 16-bit value" in the WinCC project of the WinCC OPC HDA server.
- Create a process value archive called "HDA_ProcessValueArchive" in the WinCC archive system.

- Create an WinCC archive tag called "OPC_HDA_Tag" in the "HDA_ProcessValueArchive" process value archive. Link the WinCC archive tag to the internal tag "OPC_HDA".
- In the Runtime list, launch Tag Logging Runtime and disable Graphics Runtime.
- Launch the WinCC project of the WinCC OPC HDA server.

Configuration steps

The following configurations are required to connect WinCC to the OPC HDA client:

1. Configuring access to a WinCC archive tag using the HDA server browser
2. Reading values from the WinCC archive tags

See also

How to Configure Access to a WinCC Archive Tag Using the HDA Server Browser
(Page 536)

HDA server browser (Page 534)

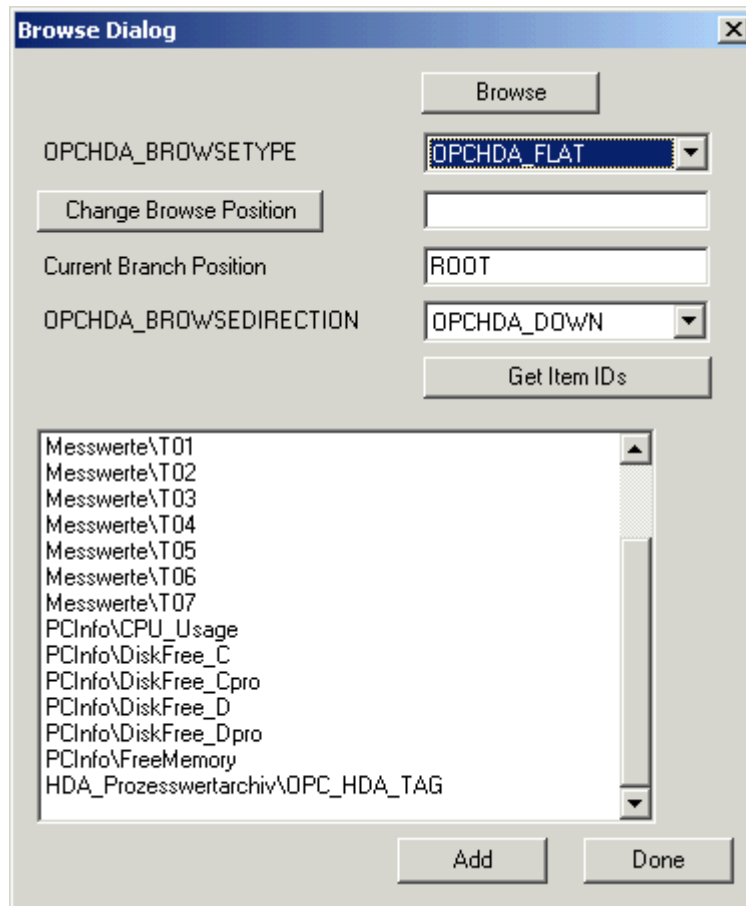
Reading Values of WinCC Archive Tags (Page 537)

4.8.5.2 HDA server browser

Introduction

The OPC HDA client accesses the tag values via item handles. For ease of configuration, the WinCC OPC HDA server supports the browser functionality. The OPC HDA client can use the

HDA server browser to search the address space of the WinCC OPC HDA server. The data are listed hierarchically by process value archive.



Note

Access to a WinCC archive tag without the HDA server browser requires manual configuration of the item ID.

When addressing WinCC archive tags, the computer name (server prefix) is included in the path. The ItemID has the following syntax: Server-prefix::process_value_archive \WinCC_archive_tag.

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

How to Configure Access to a WinCC Archive Tag Using the HDA Server Browser
(Page 536)

4.8.5.3 How to Configure Access to a WinCC Archive Tag Using the HDA Server Browser

Introduction

In this section, the OPC HDA client is used to access a WinCC archive tag. The OPC HDA client from the OPC Foundation is used. The HDA server browser is used to configure access.

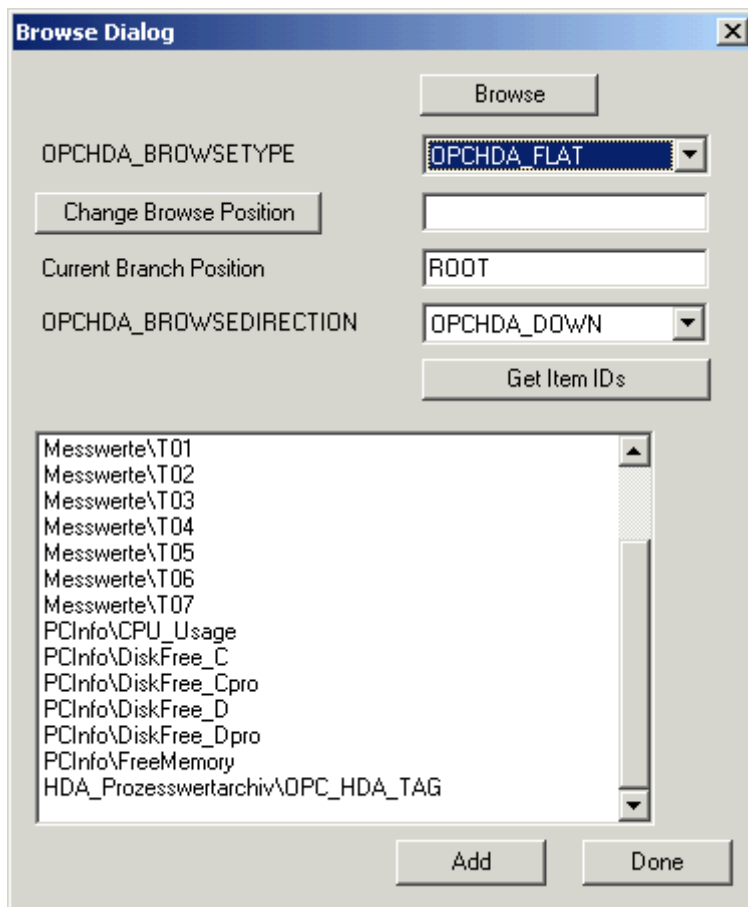
Note

The OPC HDA client described here is the demo client from the OPC Foundation. The source code for it is found on the Internet at <http://www.opcfoundation.org>.

Procedure

1. Copy the " SampleClientHDA.exe " file to a directory of your choice. This application is only available in the online help.
2. Double-click the "SampleClientHDA.exe" file. The HDA client program is started.
3. In the "Server Name" area, select entry "OPCServerHDA.WinCC.1". Click "Connect". Confirm the next dialog.

- Click "Browse" in the HDA client. The "Browse Dialog" dialog is opened. Select "OPCHDA_FLAT" in the "OPCHDA_BROWSETYPE" field.



- In the selection window, select entry "HDA_ProcessValueArchive_HDA_TAG". Click "Add" and then "Done" to close the dialog.

For more information, refer to <http://www.opcfoundation.org>.

See also

Reading Values of WinCC Archive Tags (Page 537)
www.opcfoundation.org (<http://www.opcfoundation.org>)

4.8.5.4 Reading Values of WinCC Archive Tags

Introduction

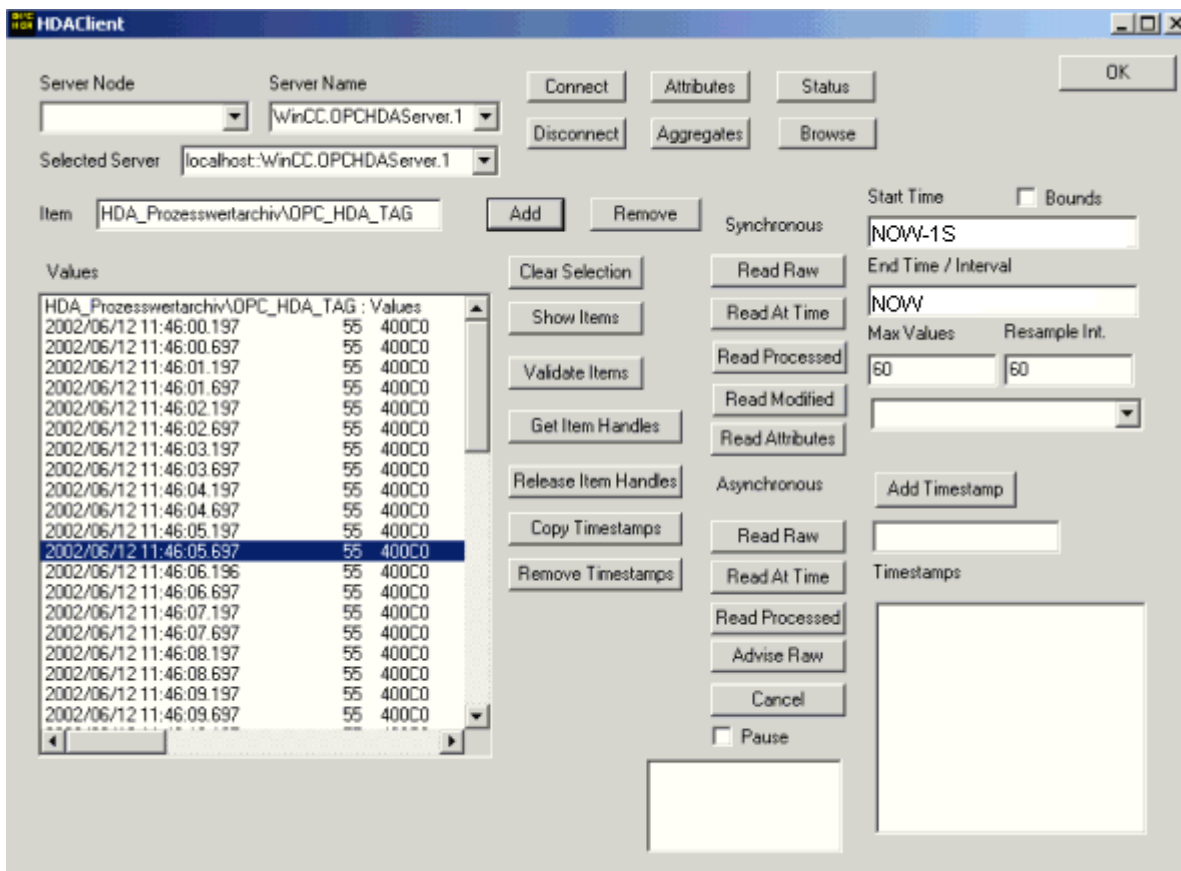
This section explains how you can access and read WinCC archive tags.

Requirement

- The OPC HDA client must be running.

Procedure

1. Click "Show Items" in the HDA client.
2. Click "Get Item Handles" in the HDA client.
3. Double-click "HDA_ProcessValueArchive_HDA_Tag" in the selection field "Value" selection field.
4. Enter "NOW-10S" in the "Start Time" field. Enter "NOW" in the "End Time" field.



5. Click "Read Raw". The values, their quality codes and time stamps are shown in the "Values" selection field.

4.8.6 Special features of the OPC HDA server in WinCC for acyclic logging

Introduction

Tag logging is performed in WinCC cyclically or acyclically. The WinCC OPC HDA server works differently depending on the logging method for tags:

- For all cyclically logged values, the OPC HDA server operates in conformity to the HDA specification of the OPC foundation. The OPC aggregates are linearly interpolated.
- Acyclically logged tags are not included in the HDA specification of the OPC Foundation. The OPC aggregates are interpolated incrementally. Especially when a tag experiences no change for a long period of time, no data is available during a time period. The following should be taken into consideration to nevertheless obtain valid data.

NOTICE

The OPC HDA server is not OPC-compliant for acyclically logged tags. The HDA specification of the OPC Foundation does not recognize acyclically logged tags and, therefore, no archive server can handle acyclically logged tags. The supported aggregates are calculated in conformity to the OPC HDA specification. No non-explicitly called functions are supported.

Configuration of acyclically logged tags

For the configuration of acyclically logged tags, the "Archive after segment change" setting needs to be enabled for the tags. This enters the most recent valid value in the the new log when a segment changes.

Supported aggregates of the WinCC OPC HDA server for acyclically logged tags

The OPC HDA server supports the following aggregates:

- OPCHDA_MINIMUM
- OPCHDA_MAXIMUM
- OPCHDA_AVERAGE
- OPCHDA_END
- OPCHDA_INTERPOLATIVE
- OPCHDA_TIMEAVERAGE
- OPCHDA_TOTAL
- OPCHDA_DURATIONGOOD
- OPCHDA_PERCENTGOOD

Supported functions of the WinCC OPC HDA server for acyclically logged tags

- ReadRaw with "boundings" only. ReadRaw for a tag must always be performed with "boundings", in order to find the last real stored value for an area without logged value change.
- ReadProcessed
- DeleteRaw
- DeleteAtTime
- Insert
- InsertReplace
- Replace

Calculating the aggregates for acyclically logged tags

Calculation of the aggregates is based on the extended "RawData" data record, which contains virtual data points for the calculation in addition to real stored values. The WinCC OPC HDA server prepares the contained "RawData" corresponding to the requirements of the "ReadProcessed". The virtual data points needed for the calculation are formed from the bordering real data points. The following significant points are included for the virtual data points:

- Value for the "StartTime"
- Value for the "EndTime"
- Value for interval limits

Example

The values for "00:59:00", "01:02:00" and "01:03:00" are stored for an acyclical tag logging tags. An OPC HDA client postulates with "ReadProcessed" an aggregate with the following parameters:

- StartTime = 01:00:00
- EndTime = 01:04:00
- Interval = 00:02:00

Note

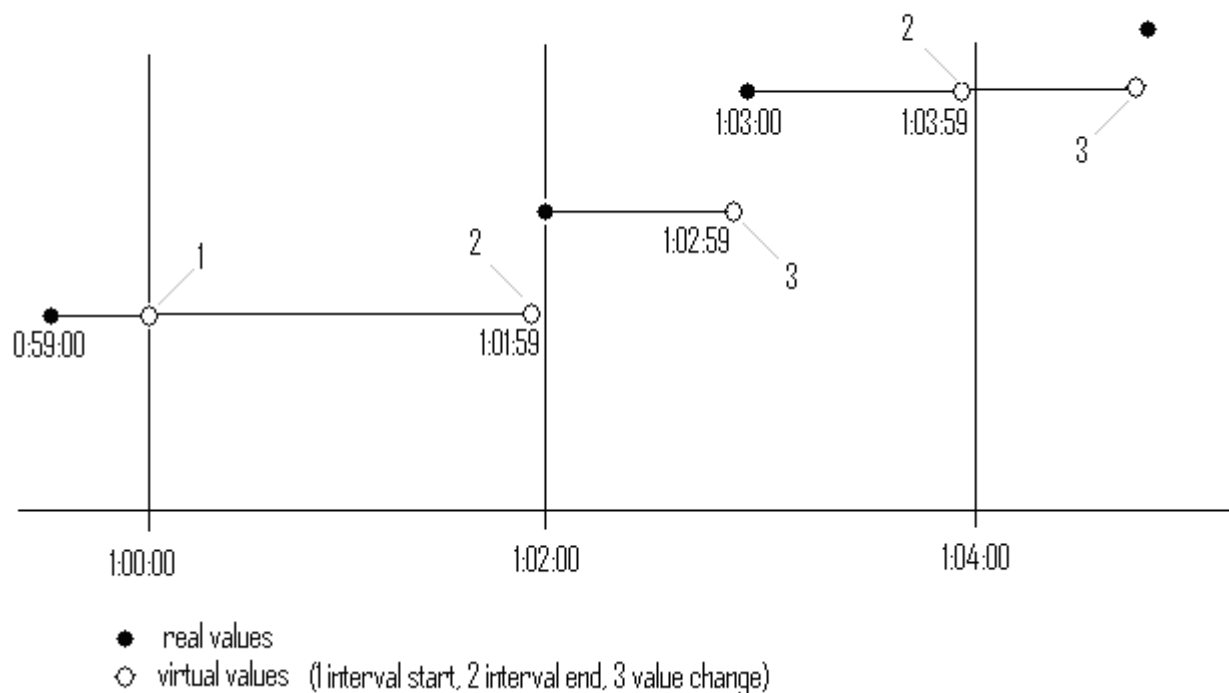
The time period is always 1 μ s less than the time stamp at the limit for the calculation when generating virtual values at limits ("EndTime"/"Interval").

A delta of 1 seconds is used in the following table to provide a better overview. The following graphic illustrates the example.

The OPC server uses the following "RawData" for the calculation of the aggregate:

Number	Time stamp	Real stored values	Generated virtual values
1	00:59:00	1,00	
2	01:00:00		1,00

Number	Time stamp	Real stored values	Generated virtual values
3	01:01:59		1,00
4	01:02:00	2,00	
5	01:02:59		2,00
6	01:03:00	3,00	
7	01:03:59		3,00



4.9 WinCC OPC A&E Server

4.9.1 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 1.10. 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 most important information. For more information, refer to the "OPC Alarms & Events 1.10" 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 any additional configuration.

Starting with WinCC V6.2, the WinCC OPC A&E server can be implemented on a WinCC server and a WinCC client.

Licensing

In order to operate the WinCC OPC A&E server, the following licenses must be installed on each WinCC server implemented as an OPC A&E server:

- WinCC Basic System
- WinCC Option Connectivity Pack

Server types

The WinCC OPC A&E server supports conditional events and simple events. In addition, there are tracking events.

Condition-related event server

With a condition-related event server, the event is associated with a condition. The condition might for example be the exceeding of a tag's bounding value. A message is generated 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 the launching and exiting programs.

Note

Note the following while using redundant systems:

Simple events that switch to internal tags are sent twice while comparing tags.

The first message is triggered by master, the second from standby.

Tracking event server

If a change in a process occurs, the OPC A&E client receives a message. Such a change might for example be a regulator adjustment.

OPC A&E Client

All OPC A&E clients conforming to the OPC Alarms & Events 1.10 specification can access the WinCC OPC A&E server. The OPC A&E client might also be a proprietary client. By creating proprietary OPC clients, most user-specific requirements can be met. An OPC A&E client can, for example, be used for the analysis and common archiving of alarms from multiple OPC A&E servers.

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

Compatibility (Page 495)

Quality Codes for OPC A&E (Page 549)

Mapping of the WinCC Message System on OPC A&E (Page 543)

4.9.2 Mapping of the WinCC Message System on OPC A&E

4.9.2.1 Mapping of the WinCC Message System on OPC A&E

Introduction

During the configuration of the WinCC message system, settings are made to determine which process events generate a message. This message is shown as an alarm in OPC A&E. The table below lists the most important parameters of the alarm. It also describes how the information is made available by the WinCC message system. For more information, refer to "Alarm Structure".

Overview

OPC	WinCC message system
Source	Indicates the source of the message. The source has the format "<server prefix>::@LOCALMACHINE::".
Time	Issues a time stamp for received, sent and acknowledged messages. Issues a time stamp in UTC (Universal Time Coordinated).
Type	Indicates whether the event is a simple, tracking or condition-related event. WinCC - POC A&E server supports simple, condition-related and tracking events.
Severity	Indicates the priority of the WinCC message.

OPC	WinCC message system
EventCategory	Returns the category of the message. For more information on this topic, refer to "Displaying Message Classes and Types".
Message	Indicates the message text of the corresponding message number.
ConditionName	Indicates the message number.
ChangeMask	Indicates the changed status of the message. For more information, refer to "Acknowledgement Theory".
NewState	Returns the message status. For more information, refer to "Acknowledgement Theory".
ConditionQuality	Returns the quality of the message. For more information, refer to "Quality Codes".
AckRequired	Indicates whether the message requires acknowledgement (receipt).
ActiveTime	Returns the time stamp for received messages.
EventAttribute	Lists the attributes required for the respective message. For more information, refer to "Attributes of the WinCC Message System".
Quality	Returns the quality code of the message.
Cookie	Returns the cookie from the OPC A&E server. The cookie corresponds to the message number in the WinCC alarm system

See also

Acknowledgement theory (Page 547)

Attributes of the WinCC Message System (Page 545)

Mapping the WinCC message classes and message types (Page 544)

4.9.2.2 Mapping the WinCC message classes and message types

Introduction

The WinCC message system informs the user of disturbances and operating conditions in the process. A WinCC message always belongs to a specific message class and message type that is related to the event category.

The mapping of the WinCC message system on OPC is configured via the CcAeProvider.ini file.

Event Category

An event category is created on the WinCC OPC A&E server for every combination of a message class and type.

An event category is determined by a category ID and a descriptive "Category Description". The category ID is composed of the WinCC internal IDs for the message class and the message type; the category description is composed of the message class and message type.

NOTICE

If the OPC A&E server is run on a WinCC-Client of a connectivity station, the OS servers linked to it must have an identical configuration of message classes and message types. If this is not the case, the entered OPC client must access the OS server directly.

The names of the message classes and message types can be ascertained via the alarm attributes CLASSNAME and TYPENAME.

4.9.2.3 Mapping the WinCC message priority

Introduction

The priority of WinCC messages is displayed by the OPC server to the attribute "Severity".

When configuring alarms in the WinCC messaging system, you can configure a priority between 0 and 16. The OPC A&E specification defines a value range from 1 to 1000 for the severity where 1 stands for the lowest and 1000 for the highest severity.

Therefore, the values of the WinCC priority are suitably displayed to the OPC severity. In the standard mapping, the WinCC priority 0 becomes OPC severity 1. All other priority values are interpolated in a linear manner up to severity 1000. Other priority mapping rules can be configured in the CcAeProvider.ini file.

4.9.2.4 Attributes of the WinCC Message System

Introduction

The following table lists the OPC attributes of the WinCC message system. The attributes are configured in the WinCC message system. Some attributes are intended for internal use in WinCC only and are therefore not relevant to an OPC A&E client. These attributes are not listed.

Attributes

OPC attributes	WinCC message system	Data type
CLASSNAME	Returns the message class name.	VT_BSTR
TYPENAME	Returns the message type name.	VT_BSTR
FORECOLOR	Returns the text color for the display of received, sent and acknowledged messages.	VT_I4
BACKCOLOR	Returns the background color for the display of received, sent and acknowledged messages.	VT_I4
FLASHCOLOR	Returns the flashing color.	VT_I4
FLAGS	Indicates whether the message requires acknowledgment (receipt).	VT_I4
TEXT01	Returns the content of UserTextBlock01.	VT_BSTR

OPC attributes	WinCC message system	Data type
TEXT02	Returns the content of UserTextBlock02.	VT_BSTR
TEXT03	Returns the content of UserTextBlock03.	VT_BSTR
TEXT04	Returns the content of UserTextBlock04.	VT_BSTR
TEXT05	Returns the content of UserTextBlock05.	VT_BSTR
TEXT06	Returns the content of UserTextBlock06.	VT_BSTR
TEXT07	Returns the content of UserTextBlock07.	VT_BSTR
TEXT08	Returns the content of UserTextBlock08.	VT_BSTR
TEXT09	Returns the content of UserTextBlock09.	VT_BSTR
TEXT10	Returns the content of UserTextBlock10.	VT_BSTR
PROCESSVALUE01	Returns the content of ProcessValueBlock01.	VT_VARIANT
PROCESSVALUE02	Returns the content of ProcessValueBlock02.	VT_VARIANT
PROCESSVALUE03	Returns the content of ProcessValueBlock03.	VT_VARIANT
PROCESSVALUE04	Returns the content of ProcessValueBlock04.	VT_VARIANT
PROCESSVALUE05	Returns the content of ProcessValueBlock05.	VT_VARIANT
PROCESSVALUE06	Returns the content of ProcessValueBlock06.	VT_VARIANT
PROCESSVALUE07	Returns the content of ProcessValueBlock07.	VT_VARIANT
PROCESSVALUE08	Returns the content of ProcessValueBlock08.	VT_VARIANT
PROCESSVALUE09	Returns the content of ProcessValueBlock09.	VT_VARIANT
PROCESSVALUE10	Returns the content of ProcessValueBlock10.	VT_VARIANT
STATETEXT	Returns the status message.	VT_BSTR
INFOTEXT	Returns the information text for the message.	VT_BSTR
LOOPIN_ALARM	States if LoopInAlarm has been configured.	VT_I4
CLASSID	Returns the message class ID.	VT_I4
TYPEID	Returns the message type ID.	VT_I4
MODIFYSTATE	Outputs the value of the status tag of the message.	VT_I4
AGNR	Returns the number of the automation device that generated the message.	VT_I2
CPUNR	Returns the number of the CPU that generated the message.	VT_I2
DURATION	Indicates the period of time between message received, sent and acknowledged.	VT_I4
COUNTER	Outputs the number of messages after the start of Runtime.	VT_I4
QUITSTATETEXT	Indicates whether the message has been acknowledged.	VT_BSTR
QUITCOUNT	Outputs the number of active, unacknowledged messages.	VT_I4
PARAMETER	Outputs the message parameter. (image of the message configuration).	VT_BSTR

OPC attributes	WinCC message system	Data type
BLOCKINFO	Returns the current content of the message block.	VT_BSTR
ALARMCOUNT	Outputs the number of messages pending.	VT_I4
LOCKCOUNT	Outputs the number of locked messages.	VT_I4
PRIORITY	Indicates the configured priority of the message.	VT_I4
APPLICATION	Outputs the application which triggered the message.	VT_BSTR
COMPUTER	Outputs the name of the computer which processed the message.	VT_BSTR
USER	Outputs the name of the user who processed the message.	VT_BSTR
COMMENT	Outputs the message comment.	VT_BSTR

4.9.2.5 Acknowledgement theory

Introduction

For WinCC, the acknowledgment philosophy is how a message is displayed and processed from "came in" to "went out". On the WinCC OPC A&E server, this message status is managed in parameters "ChangeMask" and "NewState".

Conditional, Simple and Tracking Events

Typically, messages from the WinCC system are sent to the client as conditional events. In order for a message to be treated as a simple event, the following conditions must be met during configuration of the message class:

- "Acknowledgment Came In" is not activated.
- "Message Without Status Went Out" is activated.

Depending on the mapping configuration, the messages of the message class "System without Acknowledgement" and of the message type "Operations message" are transferred as OPC Tracking Events.

ChangeMask

The "ChangeMask" parameter keeps track of where the message status was changed.

Parameter values:

- OPC_CHANGE_ACTIVE_STATE
- OPC_CHANGE_ENABLE_STATE
- OPC_CHANGE_ACK_STATE

NewState

The "NewState" parameter indicates the message status after a change.

Parameter values:

- OPC_CONDITION_ACTIVE
- OPC_CONDITION_ENABLED
- OPC_CONDITION_ACKED

Overview

WinCC	NewState	ChangeState
Received message	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Sent message with receipt	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Sent message without receipt	OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Acknowledged messages (message pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Acknowledged messages (message no longer pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Locked message	-----	OPC_CHANGE_ENABLED_STATE
Unlocked message	OPC_CONDITION_ENABLED	OPC_CHANGE_ENABLED_STATE
Received, acknowledged message	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Received, sent message with receipt	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Received, sent message without receipt	OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Message acknowledged by the system (message pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Message acknowledged by the system (message no longer pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Emergency-acknowledged message (message pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Emergency-acknowledged message (message no longer pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

4.9.3 Quality Codes for OPC A&E

Introduction

Quality codes are used to evaluate the status and quality of a message. The table below lists the quality codes for OPC A&E.

Quality codes

Code	Quality	Status
0xC0	OPC_GOOD	OK
0x40	OPC_UNCERTAIN	Returned in cases of uncertainty, e.g. in the event of delayed acknowledgement (receipt).
0x00	OPC_BAD	Returned if the connection to the source is interrupted.

4.9.4 Example of an OPC A&E Connection

4.9.4.1 Example of an OPC A&E Connection

Introduction

In the example below, a connection between WinCC and an OPC A&E client is configured. Data from the WinCC message system are made available via the WinCC OPC A&E server.

The OPC A&E client is kept informed of status changes of WinCC messages by means of a subscription.

All OPC A&E clients conforming to the OPC Alarms&Events 1.1 specification can access the WinCC OPC A&E server.

Configuration Step

The following configurations are required for connection between WinCC and the OPC A&E client:

1. Configuring access to the WinCC message system

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

How to Configure Access to the WinCC Message System (Page 550)

4.9.4.2 How to Configure Access to the WinCC Message System

Introduction

In this section, the OPC A&E client of the OPC foundation accesses the WinCC message system.

Note

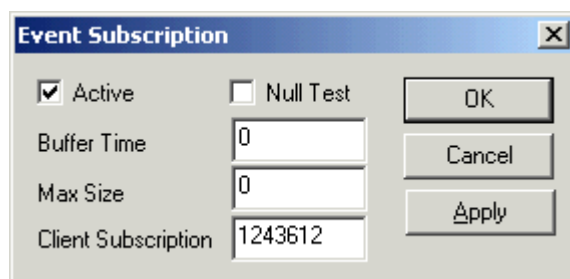
The OPC A&E client described here is the demo client from the OPC Foundation. The source code for it is found on the Internet at <http://www.opcfoundation.org>.

Requirement

- Create several internal tags of the "binary" data type in the WinCC project of the WinCC OPC A&E server.
- Configure the WinCC message system in the WinCC project of the WinCC OPC A&E server. Link the messages to the internal tags.
- Configure a picture with the Graphics Designer. Add the WinCC alarm control and an I/O field to the picture. Link the message tags to the graphic objects.
- Enable the "Alarm Logging Runtime" in the start list.
- Enable the WinCC project of the WinCC OPC A&E server.

Procedure

1. Copy the "SampleClientAE.exe" "binary" file to a directory of your choice. This application is only available in the online help.
2. Select "OPC" > "Connect..." in the menu bar. Select "OPC.WinCC-AlarmsEvent" in the "OPC Alarm Server" dialog. Click "OK" to close the dialog.
3. Select "OPC" > "Event Subscription..." from the menu bar. The "Event Subscription" dialog is opened.
4. Select the check box labeled "Active" in the dialog. Enter "1000" in the "Buffer Time" and "Max Size" fields. Click "OK" to close the "Event Subscription" dialog.



- The messages from the WinCC message system are displayed in the OPC Event Sample Client.



The screenshot shows the 'Unbenannt - OPC Event Sample Client' window. It contains a table with the following columns: Mask, NewState, Source, Event Time, Message, Event Type, E.Cat, Severity, Condition, Sub, Quality, AckReq, and Active Time. The table lists several events related to a 'Mischer' (mixer) and 'Kessel' (vessel) status, with timestamps from 13.06.2002 20:02:09 to 13.06.2002 20:02:59.

Mask	NewState	Source	Event Time	Message	Event Type	E.Cat	Severity	Condition	Sub	Quality	AckReq	Active Time
0x1	ACT ENA	localhost	13.06.2002 20:02:59	Mischer ist angelaufen	COND	69540	200	1	1	0xc0	0	13.06.2002 20:02:59
0x1	ACT ENA	localhost	13.06.2002 20:02:49	Kessel 1/2 leer	COND	69537	200	2	2	0xc0	0	13.06.2002 20:02:49
0x1	ENA	localhost	13.06.2002 20:02:49	Mischer ist angelaufen	COND	69540	200	1	1	0xc0	0	13.06.2002 20:02:19
0x1	ENA	localhost	13.06.2002 20:02:39	Kessel 1/2 leer	COND	69537	200	2	2	0xc0	0	13.06.2002 20:02:09
0x1	ACT ENA	localhost	13.06.2002 20:02:19	Mischer ist angelaufen	COND	69540	200	1	1	0xc0	0	13.06.2002 20:02:19
0x1	ACT ENA	localhost	13.06.2002 20:02:09	Kessel 1/2 leer	COND	69537	200	2	2	0xc0	0	13.06.2002 20:02:09

- Select "OPC" > "Filter" from the menu bar. The "Filter" dialog is opened. Select a category from the "Event Category" field. Click "OK" to close the "Filter" dialog.
- The messages meeting the filter criteria are displayed in the OPC Event Sample Client.

"Buffer Time" and "Max Size" Parameters

According to OPC specification, the "Buffer Time" and "Max Size" parameters are configured in WinCC as follows:

OPC Client demands return value	WinCC uses
Buffer time < 100 OPC_S_INVALIDBUFFERTIME	Revised buffer time = 100
100 <= buffer time <= 600000 S_OK	Revised buffer time = buffer time
Buffer time > 600000 OPC_S_INVALIDBUFFERTIME	Revised buffer time = 600000
Max size = 0 OPC_S_INVALIDMAXSIZE	Revised max size = 1000
0 < max size < 10 OPC_S_INVALIDMAXSIZE	Revised max size = 10
10 <= max size <= 1000 S_OK	Revised max size = max size
Max Size = 1000 OPC_S_INVALIDMAXSIZE	Revised max size = 1000

Parameters may be set while creating a subscription. However, you cannot change an existing subscription using SetState() after the fact.

For more information, refer to <http://www.opcfoundation.org>.

See also

www.opcfoundation.org (<http://www.opcfoundation.org>)

4.9.5 OPC A&E server with hierarchical access

4.9.5.1 Functionality of the OPC A&E server

Introduction

The OPC-A&E server uses DCOM services for transferring messages between OPC-capable applications. The OPC A&E server supports the specification OPC Alarm&Event 1.10.

The following chapter explains the mapping of the WinCC message system on OPC A&E with hierarchical access and the attributes supported by the OPC A&E server. This documentation includes an overview of the specific information. For more information, refer to the "OPC Alarms & Events 1.10" specification.

Principle of operation

The OPC-A&E client receives WinCC messages via subscription. You can use the subscription filter to reduce the number of events that will be transferred with a subscription. The OPC-A&E client can be set for every event category that displays message attributes.

Installation

The WinCC OPC A&E server can be selected during the installation of WinCC. After installation, the WinCC OPC A&E server can be used immediately without any additional configuration.

As of WinCC V6.2, the WinCC OPC A&E server can be implemented on a WinCC server and a WinCC client.

Licensing

In order to operate the OPC A&E server, one of the following licenses must be installed on each computer running an OPC A&E server:

- WinCC Basic System
- WinCC Connectivity Pack Add-on

Event types

The OPC-A&E server with hierarchical access supports conditional events, simple events and tracking events.

Condition related events

With a condition related event, the event is associated with a condition. The condition can be the limit violation of a tag, for example. This limit violation generates a message that is shown as an alarm with OPC A&E.

Simple events

Simple events are messages that inform the OPC A&E client about events. Simple events include the launching and closing of programs.

Note

Note the following when using redundant systems:

Simple events interconnected to internal tags are sent twice when tags are updated.

The first message is triggered by the master, the second by the standby.

Tracking events

A tracking event is sent with a operator input message to the OPC A&E client. An operator input message is triggered by manual intervention in the process.

OPC A&E client

All OPC A&E clients conforming to the OPC Alarms & Events 1.10 specification can access the OPC A&E server. The OPC A&E client may also be a proprietary client. By creating proprietary OPC clients, most user-specific requirements can be met. An OPC A&E client, for example, may be used for analysis and joint archiving of alarms from different OPC A&E servers. The acknowledgement of archived messages is not possible; only current alarms and events can be acknowledged.

Note

Documentation on OPC

You can find additional information on OPC in the Chapter "Interfaces > OPC - OLE for Process Control".

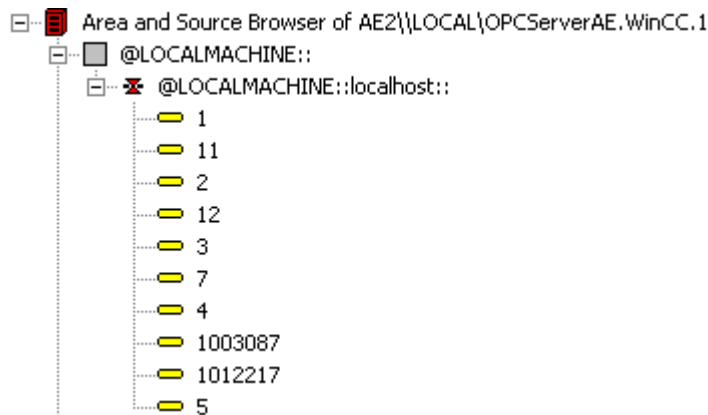
4.9.5.2 OPC A&E Server of WinCC V6.2 SP2 or higher

Differences between OPC A&E and OPC A&E with hierarchical access

Displaying messages with OPC A&E

The OPC A&E server supports "conditional events" and "simple events" for accessing the message system. With "conditional events", the message numbers are shown for each source. Since an WinCC server can hold many more message numbers, it is difficult to maintain an overview of the messages.

The following figure shows an example of the display in an OPC browser:



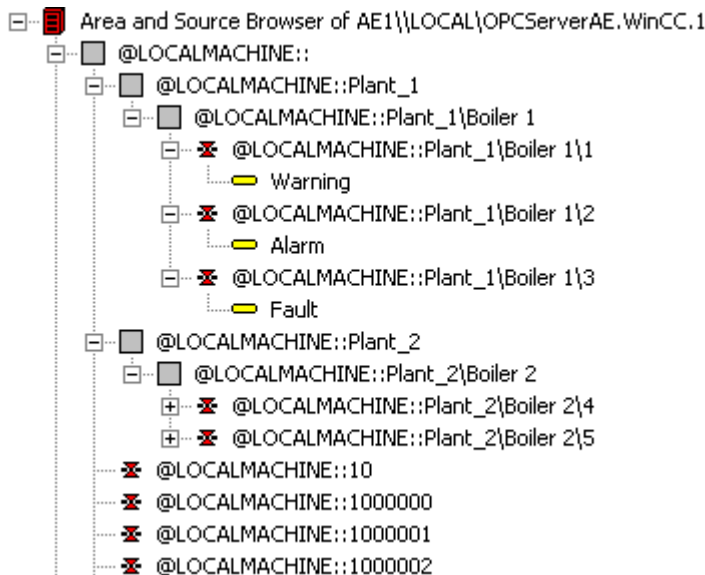
Displaying the messages with OPC A&E and hierarchical access

The OPC A&E server with hierarchical access supports the event types, conditional events, simple events and tracking events.

The user text block 2 determines the source of the messages for "conditional events". With the default setting, user text block 2 corresponds to the fault location. In order to present messages hierarchically, they must be combined in user-defined group messages in alarm logging messages. The structure of group messages is determined by the areas in OPC A&E.

Tracking events occur when operator input messages are triggered in the system.

The following figure shows an example of the display of conditional events in an OPC browser. The "Condition" is shown in addition to "Area" and "Source":



Recommendation

Use an OPC A&E server with hierarchical access when creating a new project.

After a project upgrade to WinCC V6.2 SP2 or higher, OPC A&E Server can be used as before, or be converted for hierarchical access. The conversion can be undone again without any loss of data. For further information, refer to "How to upgrade OPC A&E to WinCC V6.2 SP2 or higher".

How to upgrade OPC A&E to WinCC V6.2 SP2 or higher

Introduction

OPC A&E for WinCC V6.2 SP2 or higher was enhanced with functions for hierarchical access to the message system. The OPC-A&E server with non-hierarchical access remains the standard.

Upgrading from OPC A&E

If you are using the OPC A&E with hierarchical access and want to use all functions, you may need to adapt the OPC A&E client currently used.

Starting with the WinCC version of your project, possible upgrade scenarios are described for OPC A&E:

- Upgrading the project from WinCC V6.2 to WinCC V6.2 SP2
- The project is created in WinCC V6.2 SP2 or higher.

Upgrading from WinCC V6.2 to WinCC V6.2 SP2

You can upgrade a project with WinCC V6.2 to WinCC V6.2 SP2 in regard to OPC A&E as follows:

Retaining previous OPC A&E without hierarchical access

If you want to continue to work with the previously used OPC A&E server, the following scenarios are possible:

- If you have not changed the standard "CcAeProvider.ini" file, you do not need to make any other settings.
- If you have changed the standard "CcAeProvider.ini" file and want to keep these changes, proceed as follows:
 - Save the "CcAeProvider.ini" file from the WinCC installation path in the "OPC\AlarmEvent\bin" folder.
If you are working on a distributed system or a system integrated in STEP 7, save the file from the project on the WinCC client or on the OS.
 - Following the upgrade, copy the file into the WinCC project directory.
If you are working on a distributed system or a system integrated in STEP 7, copy the file on the WinCC server or on the ES into the project directory of the client projects or OS projects.
- If you have changed the standard "CcAeProvider.ini" file and want to replace it with the standard "CcAeProvider.ini" file supplied with the product, before the upgrade delete these files on the servers and clients or on the ES computers and OS computers. The project folder is located in "wincproj" sub-folder on the ES.

Switching to OPC A&E with hierarchical access

If you want to work with the OPC A&E server with hierarchical access, proceed as follows:

1. After upgrading, copy the "CcAeProvider.ini" file to the project folder. The file is located in the WinCC installation path in the folder "OPC\AlarmEvent\Hierarchical-Access".
2. Update the clients or perform a complete download for the OS servers.

Creating a project in WinCC V6.2 SP2 or higher

New projects created in WinCC V6.2 SP2 or higher still use OPC A&E Server without hierarchical access. You do not have to make any additional settings.

If you want to work with the OPC A&E server with hierarchical access, proceed as follows:

1. After upgrading, copy the "CcAeProvider.ini" file to the project folder of the ES projects. You can find the file in the "OPC\AlarmEvent\Hierarchical-Access" folder below the installation path of WinCC.
2. Update the clients or perform a complete download for the OS servers.

4.9.5.3 Mapping the WinCC Message System on OPC A&E

Mapping the WinCC message system

Introduction

The WinCC message system resulting from the configuration defines which event in the process will generate a message. This message is shown as an event notification in OPC A&E.

Mapping the WinCC message system on OPC A&E with hierarchical access

The OPC source of the WinCC user text block "2" and the OPC message of WinCC user text block "1" are used in WinCC as a default setting for mapping the WinCC message systems.

Overview

The following table shows the most important attributes of the event notifications and the respective information from the WinCC message system.

The events that use the configured attributes are shown in the third column of the table:

- "S" means a simple event
- "C" means a conditional event
- "T" means a tracking event

OPC	WinCC message system	Event type
Area	The structure of the group messages determine the areas in OPC A&E. If there is no group message configured for the message, only the OPC area corresponding to the server prefix is available.	S, C, T
Source	Indicates the source of a message. The source has the format "<server prefix>::Area \user text block 2". The server prefix of a local computer is "@LOCALMACHINE". The server prefix always shows the top Areas in the hierarchy of the server.	S, C, T
Time	Issues a time stamp for received, sent and acknowledged messages. Issues a time stamp in UTC (Universal Time Coordinated).	S, C, T
Type	Indicates whether the event is a simple, tracking or conditional event.	S, C, T
Severity	Returns the priority of the message.	S, C, T
EventCategory	Indicates the message class. "Event Category" is made up of the "CategoryID" and the "Category Description". "CategoryID" corresponds to the internal ID of the message class. "Category Description" corresponds to the name of the message class.	S, C, T
Message	Indicates the message text of the corresponding message number.	S, C, T
Condition	Indicates the message type.	C
Sub-condition	Corresponds with the "Condition" parameter.	C
ChangeMask	Specifies the change of the condition. For more information, refer to "Acknowledgment Theory".	C
NewState	Indicates the current status of the condition. For more information, refer to "Acknowledgment Theory".	C
ConditionQuality	Returns the quality of the message. For more information, refer to "Quality codes".	C
AckRequired	Indicates whether the message requires acknowledgment.	C
EventAttribute	Lists the attributes required for the respective message. For more information, refer to "Attributes of the WinCC message system".	C
Quality	Returns the quality code of the message.	C
Cookie	Does not include any usable information for the client	C
ActorID	Indicates which user acknowledged the message.	T

Note

If text without wild cards are specified as a filter for the area, only the messages of the area are returned. If you want to include sources that are located in areas outside the specified area, you need to use wild cards.

NOTICE

The message classes and message types **must** be configured identically on the connected OS servers, if you run the OPC A&E server as follows:

- On a WinCC Client
- On a Connectivity station

If the OS server is not configured identically, the employed OPC client must access the respective OS server directly.

Mapping the message priority

Introduction

The priority of messages is mapped by the OPC A&E server to the attribute "Severity".

When configuring alarms in the messaging system, you can configure a priority between "0" and "16". The OPC A&E specification defines a value range of "1" to "1000" for the severity. In this case, "1" stands for the lowest and "1000" for the highest severity.

Therefore, the values of the priority are suitably displayed to the OPC severity. In the standard mapping, priority "0" is assigned to OPC severity "1" and priority "16" to OPC severity "1000". All other priority values are interpolated linearly between "0" and "1000".

Attributes of the WinCC Message System

Introduction

The following table lists the OPC attributes of the WinCC message system. The attributes are configured in the WinCC message system. Some attributes are intended for internal use in WinCC only and are therefore not relevant to an OPC A&E client. These attributes are not contained in the table.

Attributes

OPC attributes	WinCC message system	Data type
CLASSNAME	Outputs the message class name.	VT_BSTR
TYPENAME	Outputs the message type name.	VT_BSTR
FORECOLOR	Outputs the text color for activated, deactivated and acknowledged messages.	VT_I4

OPC attributes	WinCC message system	Data type
BACKCOLOR	Outputs the background color for activated, deactivated and acknowledged messages.	VT_I4
FLASHCOLOR	Outputs the flash color.	VT_I4
FLAGS	Indicates mandatory message acknowledgment	VT_I4
TEXT01	Outputs the content of UserTextBlock01.	VT_BSTR
TEXT02	Outputs the content of UserTextBlock02.	VT_BSTR
TEXT03	Outputs the content of UserTextBlock03.	VT_BSTR
TEXT04	Outputs the content of UserTextBlock04.	VT_BSTR
TEXT05	Outputs the content of UserTextBlock05.	VT_BSTR
TEXT06	Outputs the content of UserTextBlock06.	VT_BSTR
TEXT07	Outputs the content of UserTextBlock07.	VT_BSTR
TEXT08	Outputs the content of UserTextBlock08.	VT_BSTR
TEXT09	Outputs the content of UserTextBlock09.	VT_BSTR
TEXT10	Outputs the content of UserTextBlock10.	VT_BSTR
PROCESSVALUE01	Outputs the content of ProcessValueBlock01.	VT_VARIANT
PROCESSVALUE02	Outputs the content of ProcessValueBlock02.	VT_VARIANT
PROCESSVALUE03	Outputs the content of ProcessValueBlock03.	VT_VARIANT
PROCESSVALUE04	Outputs the content of ProcessValueBlock04.	VT_VARIANT
PROCESSVALUE05	Outputs the content of ProcessValueBlock05.	VT_VARIANT
PROCESSVALUE06	Outputs the content of ProcessValueBlock06.	VT_VARIANT
PROCESSVALUE07	Outputs the content of ProcessValueBlock07.	VT_VARIANT
PROCESSVALUE08	Outputs the content of ProcessValueBlock08.	VT_VARIANT
PROCESSVALUE09	Outputs the content of ProcessValueBlock09.	VT_VARIANT
PROCESSVALUE10	Outputs the content of ProcessValueBlock10.	VT_VARIANT
STATETEXT	Outputs the status message.	VT_BSTR
INFOTEXT	Outputs the message infotext.	VT_BSTR
LOOPINALARM	Indicates whether LoopInAlarm was configured.	VT_I4
CLASSID	Outputs the message class ID.	VT_I4
TYPEID	Outputs the message type ID.	VT_I4
MODIFYSTATE	Outputs the value of the status tag of the message.	VT_I4
AGNR	Outputs the number of the AS that generated the message.	VT_I2
CPUNR	Outputs the number of the CPU that generated the message.	VT_I2
DURATION	Outputs the interval between the activation, deactivation and acknowledgment of a message.	VT_I4
COUNTER	Outputs the number of messages after the start of Runtime.	VT_I4
QUITSTATETEXT	Indicates whether the message has been acknowledged.	VT_BSTR
QUITCOUNT	Outputs the number of active, unacknowledged messages.	VT_I4
PARAMETER	Outputs the message parameter. (image of the message configuration).	VT_BSTR
BLOCKINFO	Outputs the current content of the message block.	VT_BSTR
ALARMCOUNT	Outputs the number of messages pending.	VT_I4
LOCKCOUNT	Outputs the number of locked messages.	VT_I4
PRIORITY	Indicates the message priority configured.	VT_I4
APPLICATION	Outputs the application which triggered the message.	VT_BSTR

OPC attributes	WinCC message system	Data type
COMPUTER	Outputs the name of the computer which processed the message.	VT_BSTR
USER	Outputs the name of the user who processed the message.	VT_BSTR
COMMENT	Outputs the message comment.	VT_BSTR
HIDDEN-COUNT	Outputs the number of hidden messages.	VT_I4
BIG COUNTER	Outputs the number of messages after the start of Runtime.	VT_CY
OS-HIDDEN	Outputs the hidden status of the message.	VT_BOOL
OS-EVENTID	Outputs the message ID configured for the message.	VT_I4

Acknowledgement Theory

Introduction

The acknowledgment policy in WinCC is how a message from "came in" to "went out" is displayed and processed. On the OPC A&E server, this message status is displayed in the "ChangeMask" and "NewState" parameters.

Conditional events, simple events and tracking events

Messages from the system are sent to the client as conditional events with acknowledgment.

In order for a message to be handled as a simple event, the message class of the message must meet the following conditions:

- "Acknowledgment came in" is not activated.
- "Message without status went out" is activated.

In WinCC, messages of message class "System, does not require acknowledgment" with "Operator input message" message type are transferred as tracking events.

NOTICE

Messages with "System, does not require acknowledgment" message class and "Process control system" message type are transferred as simple events with the "System message" event category.

ChangeMask

The "ChangeMask" parameter keeps track of where the message status was changed.

Parameter values:

- OPC_CHANGE_ACTIVE_STATE
- OPC_CHANGE_ENABLE_STATE
- OPC_CHANGE_ACK_STATE

NewState

The "NewState" parameter indicates the message status after a change.

Parameter values:

- OPC_CONDITION_ACTIVE
- OPC_CONDITION_ENABLED
- OPC_CONDITION_ACKED

Overview

WinCC	NewState	ChangeState
Received message	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Went out message with acknowledgment	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Went out message without acknowledgment	OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Acknowledged messages (message pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Acknowledged messages (message no longer pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Locked message	-----	OPC_CHANGE_ENABLED_STATE
Unlocked message	OPC_CONDITION_ENABLED	OPC_CHANGE_ENABLED_STATE
Came in, acknowledged message	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Came in, went out message with acknowledgment	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Came in, went out message without acknowledgment	OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Message acknowledged by the system (message pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Message acknowledged by the system (message no longer pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Emergency-acknowledged message (message pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Emergency-acknowledged message (message no longer pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE

Note

Historical alarms and events are not acknowledged. The OPC A&E historical events interface only has read access.

4.9.5.4 Quality Codes for OPC A&E

Introduction

Quality codes are used to evaluate the status and quality of a message. The table below lists the quality codes for OPC A&E.

Quality codes

Code	Quality	Status
0xC0	OPC_GOOD	OK
0x40	OPC_UNCERTAIN	Returned in cases of uncertainty, for example in the event of delayed acknowledgment (receipt).
0x00	OPC_BAD	Returned if the connection to the source is interrupted.

4.9.6 Reading archived messages

4.9.6.1 Accessing archived events

Introduction

You can access the archived messages via the OPC A&E server using an OPC client. Two methods are supported for accessing archived messages:

- Output archived messages from a time period in the past
- Output archived messages from a time period in the past without mentioning end of period. After the output of archived messages, all other newly generated messages are automatically sent to the OPC client.

NOTICE

After reading archived messages, you cannot use the returned "ActiveTime" of a message for acknowledging the message or tracing transitions of the message. To ensure this, the OPC A&E client must check the "EventType" of a message with the extra flag "OPC_HAE_HISTORICAL_EVENTFLAG". The "ActiveTime" is incorrect on archived messages. You can find information on the additional flag under "Identifying archived messages".

Querying the "Historic Alarms and Events" functionalities

In addition to the standard filters, the following filters are offered with the expanded OPC A&E server of WinCC:

Filter	Filter Values	Description
OPC_HAE_FILTER_BY_TIMEFRAME	0x80000000	Matches "ReadRaw" function for OPC historical data access
OPC_HAE_FILTER_BY_STARTTIME	0x40000000	Matches "AdviseRaw" function for OPC historical data access

Source filter and historical alarm request

To request the archive messages, the OPC client must support the "SetFilter" to a subscription functionality. The OPC server will also send archived messages if you also insert keyword "OPCHAEServer" the array of the "Source Filter" of a subscription. In addition to this keyword, you can use other parameters to define which messages are to be read:

- Method
- Time period
- With or without limits

The lists of sources that are assigned in the filter can include other source names besides the "OPCHAEServer" source. In such a case, the subscription delivers only the historic events of the given sources. The sequence of the source names is inconsequential.

After configuring the source filter, the selected time period can be called up from the client with a "Refresh" call.

4.9.6.2 Syntax for accessing archived messages using OPC

Syntax

```
OPCHAEServer hMode=(read|advise) htStartTime=szTime
[hEndTime=szTime] [bBounds=(TRUE|FALSE)]
```

Parameter

hMode = [read|advise]

This parameter is required. Defines how the archived messages and events are to be read.

Read: Outputs archived messages and events of a definite period from the past (comparable to ReadRaw in case of OPC Historical Data Access).

The following is an example for setting a filter for reading over the last 30 minutes:

```
OPCHAEServer hMode=read htStartTime=NOW-30M bBounds=TRUE
```

Advise: Outputs archived messages and events from a definite period, After receiving all archived messages, new messages are sent in the same way as in the case of an active subscription (comparable to AdviseRaw in case of OPC historical data access).

In the following example, the messages of the last 30 minutes are read (subscription must be active):

```
OPCHAEServer hMode=advise htStartTime=NOW-30M
```

Note

The following notation is supported for parameters "htStartTime" and "htEndTime":

- Relative notations, for example NOW
- Symbolic values, for example NOW, YEAR, MONTH
- Specification of absolute UTC data/time values according to XML notation:
2006-09-01T10:00:00.000Z

Using the symbolic notation corresponds to the syntax from OPC historical data access.

htStartTime =

This parameter is required. Defines the time from when the messages and events are to be read from the archive.

htEndTime =

This parameter is optional. Defines the time up to which the messages and events are to be read from the archive. With "hMode = read", the default setting "NOW" is used.

bBounds = [TRUE|FALSE]

This parameter is optional. Defines how messages close to the start and end time are to be handled. The function is identical to OPC historical data access.

bBounds=FALSE:

- The time stamp of the first transferred message $\geq$ htStartTime
- The time stamp of the last transferred message $\geq$ htEndTime

bBounds=TRUE:

- The time stamp of the first transferred message $\leq$ htStartTime
- The time stamp of the last transferred message $\geq$ hEndTime

Default setting is FALSE.

4.9.6.3 Read methods for archived messages

Introduction

You can use one of the two read modes to read archived messages:

- Read
- Advise

"Read" mode

"Read" mode is used to read archived messages from a defined period in the past. The sequence of read messages is always read from the alarms in chronological sequence in reference to each OS server. By setting the start and end time, you can specify whether the last message is to be read first or last. If the start time is earlier than the end time, the last message is last in the output.

To use the "read" mode, you need to run the following functions on the subscription:

1. SetFilter
2. Refresh

"SetFilter" during "Refresh" will be rejected. Activating the subscription during "Refresh" does not have any effect on refresh.

The historic events continue to be transferred with the Refresh flag.

The newly generated events are transferred according to the standard reaction of an active subscription:

- Taking into account the set filter values with the exception of "historic" source "OPCHAEServer"
- Without the Refresh flag

The client can therefore distinguish between the received events based on the Refresh flag. An event package never contains historic and new events at the same time.

- Event packages with Refresh flag only contain historic events. These events can also be in queue.
- Event packages without the Refresh flag only contain newly generated events.

"Advise" mode

"Advise" mode is used to read archived messages from a defined period onwards in the past. After reading all archived messages, new messages are sent in the same way as for an active subscription. The archived messages are transferred in chronological sequence in reference to each OS server: The archived messages from a start time onwards are transmitted. Thereafter, the newly archived messages transferred.

Note that you should not define an end time for "advise".

An active subscription is used for "advise" mode. If you run the "SetFilter" function on an active subscription, the historical alarms are transferred immediately.

If you run "SetFilter" function on an inactive subscription, the archived messages are only sent after activating the subscription. If you want to use "advise" read mode with an inactive subscription, proceed as follows:

1. SetFilter
2. Set subscription to active using SetState

The transmission is interrupted if you deactivate the subscription.

The transmission is ended if you set the subscription to "inactive". "SetFilter" is rejected when the subscription is active.

A "Refresh" on an active "historic" subscription in the "advise" mode functions in the same way as for a standard subscription:

All queued condition related events are transferred to packages with Refresh flag. The last package also contains an additional flag "Last Refresh".

A "Refresh" call has no influence on reading historical alarms in "advise" mode.

4.9.6.4 Identifying archived messages

General procedure

Archived messages are distinguished using an additional flag in EventType. This flag is linked to the real EventType via a OR link.

Name	EventType	EventType (archived message)
OPC_SIMPLE_EVENT	0x01	0x81
OPC_CONDITION_EVENT	0x04	0x84
OPC_TRACKING_EVENT	0x02	0x82
OPC_HAE_HISTORICAL_EVENTFLAG		0x80

Examples

Example 1

The following source filter is used to output archived messages and events of the last 30 minutes in "read" mode. The oldest message for each OS server is output as the first one. The low limit value is also sent.

```
OPCHAEServer hMode=read htStartTime=NOW-30M bBounds=TRUE
```

Example 2

The following source filter is used to output archived events on September 1, 2006 from 10:00 to 12:00 hours in "read" mode. The newest message for each OS server is output as the first one. The limits for this time period are also sent.

```
OPCHAEServer hMode=read htStartTime=2006-09-01T12:00:00.000Z  
htEndTime=2006-09-01T10:00:00.000Z bBounds=TRUE
```

Example 3

The following source filter is used to output archived messages and events of the last 30 minutes in "advise" mode. After reading the archived messages, newly generated messages are sent in the same way as for an active subscription.

```
OPCHAEServer hmode=advise htStartTime=NOW-30M
```

4.10 Commissioning

4.10.1 OPC Commissioning

Introduction

Data exchange between a WinCC OPC server and OPC client is completed via DCOM.

After installation of WinCC, the DCOM settings of the WinCC OPC server are correctly configured.

If a WinCC OPC server or client communicates with an external OPC system, corresponding adaptations must be performed.

The "Local access" and "Remote access" authorizations must be entered for the user in "DCOM/Workplace/COM Security/Access rights/Edit default" of User Administration.

4.10.2 Configuring Windows

4.10.2.1 This is how you configure Windows accounts for the use of WinCC OPC

Introduction

The OPC client and the OPC server are DCOM applications. A distributed DCOM application can only be run under the same user account. Therefore the OPC server must recognize the OPC client's user account and vice-versa.

If the WinCC OPC servers are used with WinCC OPC clients, the correct configuration is already warranted by the installation.

Declaration of the user account, if an external OPC server or client is used

For further information on the granting of user rights, refer to the Windows XP documentation.

Requirements

Log on as the administrator to both the WinCC OPC server and OPC client workstations to configure the user permissions.

Procedure

1. Go to "Control Panel > Administration > Computer Management > Local Users and Groups".
2. In the "Users" shortcut menu, select "New User".
In the "New User" dialog, enter the user account details of the communication partner. Click "Create" and close the dialog.

3. Click the "Users" icon. Double-click the relevant user. The "Properties" dialog for this user is displayed.
4. Select the "Member Of" tab. Click "Add". The "Select group" dialog is opened.
5. Add the group "users".
If you are on a computer that has WinCC installed, also add the group "SIMATIC HMI".
Click "OK" to close all open dialogs.

4.10.2.2 How to adapt the Windows firewall settings

Introduction

After installation of WinCC, the Windows firewall settings of the WinCC OPC servers are correctly configured.

If OPC clients access OPC servers in different subnets, you must adapt the configuration of the permitted network areas to the OPC servers.

4.10.3 XML

4.10.3.1 Commissioning - OPC XML

Introduction

The OPC XML server of WinCC is realized as a web service. It makes the PC accessible via Internet. It is therefore necessary to define appropriate access rights.

4.10.3.2 Defining the Security Settings with IIS

Introduction

The Internet Information Services make the PC accessible via the Internet. It is therefore necessary to define appropriate access rights.

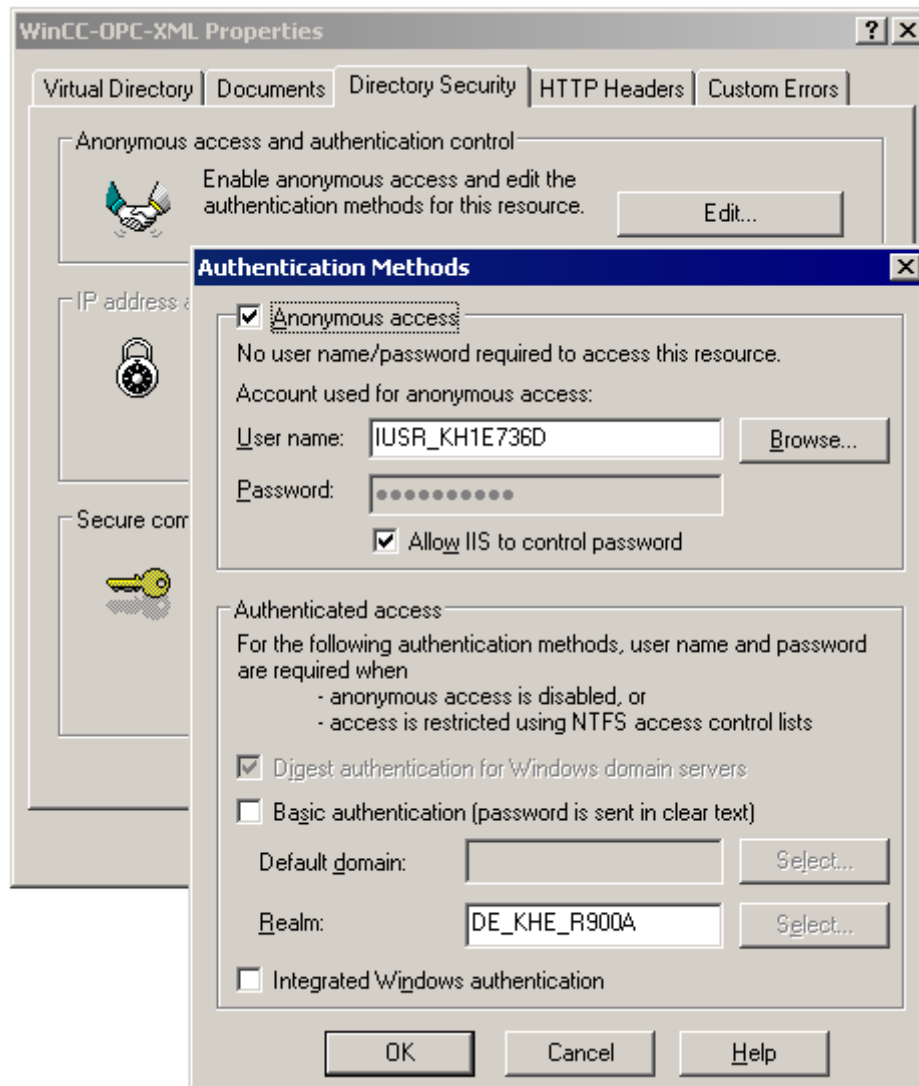
Note

If you have any questions or experience problems with the following settings, contact your Intranet/Internet administrator.

Procedure

1. Start the Management Console in Windows 2003 via "Control Panel > Administration > Internet Information Services Manager".
In Windows XP, select "Control Panel" > "Administration" > "Internet Information Services" and activate the Management Console.
2. Select the virtual directory "WinCC-OPC-XML". Choose the "Properties" option from the shortcut menu. The "WinCC OPC XML Properties" dialog is opened.
3. Click the "Directory Security" tab. On this tab, select the relevant web server security features.

4. Click the "Edit" button in the "PLC for Anonymous Access and Authentication". The "Authentication Methods" dialog is displayed.



5. Activate the "Integrated Windows Authentication" option in the "Authenticated access" area. Anonymous access to the web service is possible but should not be activated for security reasons.
6. Close all open dialogs.

See also

How to Test the Installation (Page 572)

4.10.3.3 How to set the correct version of ASP.NET

Introduction

If you want to use the WinCC-OPC-XML-DA server, ensure you have set the correct version of "ASP.NET" for the Web site for the installation through which the WinCC-OPC-XML Web service is linked.

Note

If you have any questions or experience problems with the following settings, contact your Intranet/Internet administrator.

Procedure

1. Start the Management Console in Windows 2003 via "Control Panel > Administration > Internet Information Services Manager".
In Windows XP, select "Control Panel" > "Administration" > "Internet Information Services" and activate the Management Console.
2. Select the virtual directory "WinCC-OPC-XML". Choose the "Properties" option from the shortcut menu. The "WinCC OPC XML Properties" dialog is opened.
3. Click the "ASP.NET" tab. In this tab, configure the settings for the "ASP.NET" of the Web server.
4. Click on text selection box for the "ASP.NET version". If version "2.x" is not yet set, select version "2.x".
5. Close all open dialogs.
6. The Web service needs to be restarted after changes are made.

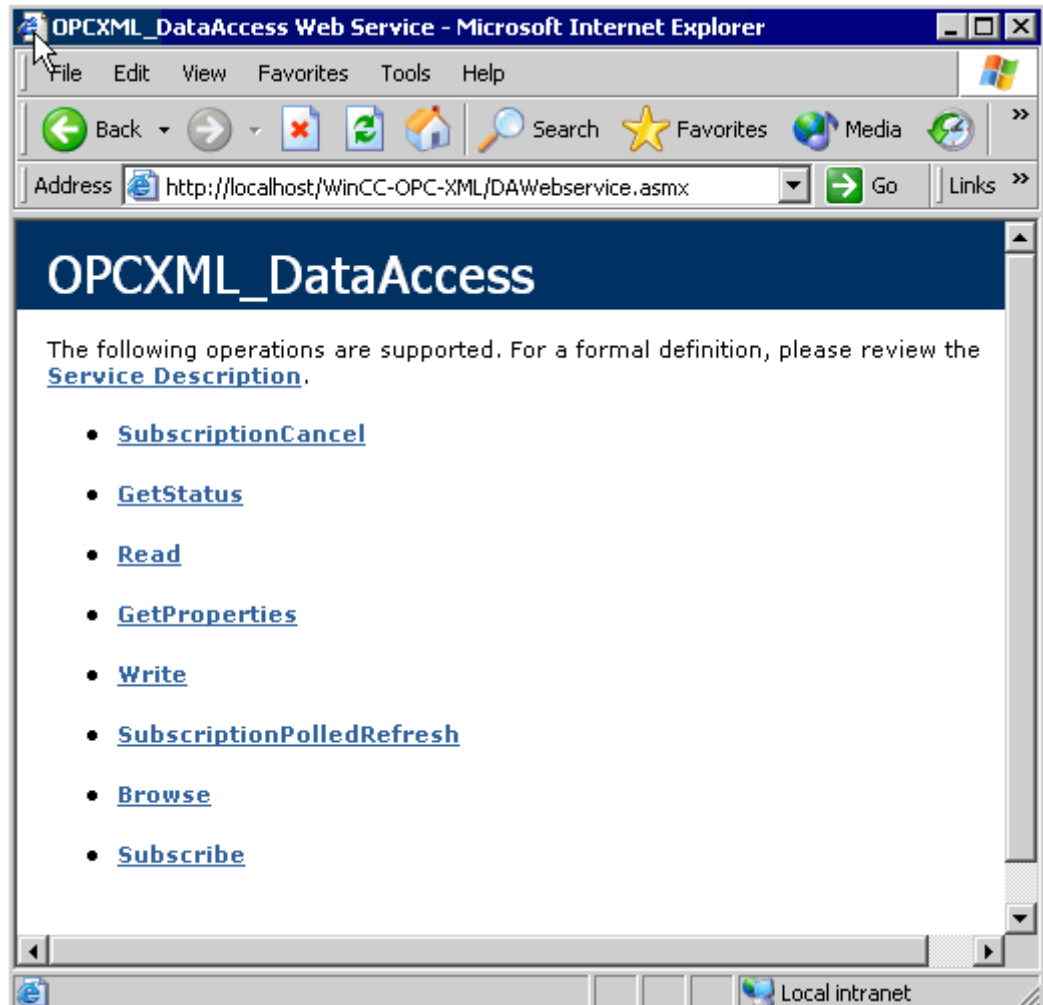
4.10.3.4 How to Test the Installation

Introduction

OPC XML-DA makes the OPC process data available as a web page. The web page can be accessed via the Internet using HTTP. The following section explains how to test the installation.

Procedure

1. Start Internet Explorer on the computer run as the WinCC OPC XML server.
2. Enter the URL "http://localhost/WinCC-OPC-XML/DAWebService.asmx" in the address bar. Confirm your entry with <ENTER>.
3. When the OPC XML DA function requests appear, installation was successful.



See also

Defining the Security Settings with IIS (Page 570)

4.10.4 Trace

Introduction

The "Trace" function can be used to log tag values and function calls for purposes of testing and error analysis.

The entries are stored in a trace file. The trace shows the step-by-step progress of the establishment of the connection, thus making it easier to identify the source of a connection problem.

Settings

Trace output must be set in the registry of the operating system. For more information, refer to the SIMATIC Customer Support.

Index

- A&E server
 - Mapping WinCC message classes and message types, 556
- Connecting WinCC - OPC A&E client
 - Example of OPC A&E , 549
- OPC A&E server , 556
- OPC A&E Server , 544
- User archive
 - Sorting the displayed data, 448
- User Archives:Volumes, 304
- UserArchiveControl
 - Selection of data of the user archive, 446
- .
- .emf, 100
- .emf file, 100

A

- A&E server, 543, 552, 553, 556
 - Conditional event, 552
 - Mapping the WinCC message system, 556
 - Simple event, 552
 - Tracking event, 552
- A&E Server, 541
 - Condition Related Event, 541
 - Hierarchical access, 553
 - Mapping of the WinCC message system, 543
 - Tracking Event, 541
- Access rights, 322, 327
- Acknowledgement theory, 547
- Acknowledgment policy, 560
- Adding a tag
 - Example of OPC DA, 507
- Address strings, 240
- Alarm Control
 - Toolbar, 434
- Alarm logging, 150
 - Worksheets, 156, 158, 162, 164, 166

AlarmControl

- Configuring the data export operation, 438
- Configuring the status bar, 434
- Configuring the toolbar, 434
- Online configuration, 439
- Properties of the table elements, 430
- Sorting by column header, 430, 433
- Table element colors, 431

Amplitude, 21

Appendix, 361

Archive ConfigurationTool, 243

Archive fields

- configuring, 330
- Configuring, 318, 325, 326, 327
- Create, 325

Archive fields:Configuring, 318, 325, 326, 327, 330

Archive fields:Creating, 325

Archive tag, 178

Archive Tag dialog, 193

ASCII file, 12

Auto-Scrolling, 433

B

- Bounding values, 525
- Buffer time, 550

C

- Changing language, 351
- COM interface, 488
 - Description, 488
- COM object, 486
- COM server, 486, 488
 - Description, 488
 - Example of integration, 487
 - Output of data, 486
 - Registry entries, 488
- COM server object, 485
 - Developing, 485
 - Integrating, 485
 - Registring, 485
- Commissioning, 568, 574
- Communication, 320, 354
 - via Raw data tags, 356, 357, 358
- Communication Configurator, 111
- Communication:via raw data tags, 355, 357, 359
- Communication:via WinCC tags, 354

- Compressed archive tag, 182
- Compressed archives, 182
- Configuring: from Archive Fields, 318, 325, 326, 327, 330
- Configuration Tool, 112
 - Diagnosis, 228
 - Dropdown menu, 115
 - Fundamentals, 112
 - Insert Project Folder, 121
 - Interface, 114
 - Menu, 114
 - Operating system, 113
 - Operation, 118
 - Quantity structure, 231
 - Requirements, 113
 - System Requirements, 113
 - Tips, 231
 - Worksheets, 131
- Configuration: of User Archives Table Element, 451
- Configuration: of a User Archive, 317
- configuring
 - from archive fields, 330
 - of a Form View, 465
 - of a User Archives Control, 451, 453
 - of user archives, 317, 329, 350
 - of User Archives Table Element, 451, 453
 - of views, 345, 348
- Configuring
 - from archive fields, 318, 325, 326, 327
 - of a User Archive, 317
 - of user archives, 320, 321, 322, 325
 - of views, 340, 343, 345, 346, 348, 349
- Configuring: of User Archives, 317, 321, 322, 325, 329, 350
- Configuring: views, 340, 343, 345, 346, 348, 349
- Connecting WinCC - OPC A&E client, 549
- Connections, 141
- Connections/Groups, 141
- Control characters, 488
 - for COM server objects, 488
- Control tags, 321
 - Example:, 334
 - Features, 333
- Control Tags: Properties, 333
- Create actions, 366
- Creating, 485
 - COM server object, 485
- Cross reference, 101, 102

D

- Data export in runtime, 438

- Data manager, 136
 - Worksheets, 138, 140, 142, 145, 147
- Data structure, 14
- Data Window, 300
- Decrement, 19, 21
- Default values, 137, 171
 - Data manager, 138
- Defining new tag, 514
 - Example of OPC DA, 514
- Description fo COM interface, 488
- Description of the User Archives Functions , 379
- Developing, 485
 - COM server object, 485
- Diagnosis
 - Error column, 229
 - Reading error, 188
 - WinCC Configuration Tool, 228
 - Write error, 187
- Dialogs, 189
 - Archive Tag dialog, 193
 - Tag dialog, 189
- Dropdown menu, 115
- Dynamic Wizard Editor, 24
 - Editor window, 29
 - Help Editor, 30
 - Installation, 26
 - Output window, 31
 - Structure, 27
 - Toolbar, 28
- Dynamic Wizard function, 32

E

- E-server
 - Functionality of WinCC OPC A&OPC, 498
- Excel, 112, 113
- Export, 11, 12, 13, 14
- Exporting, 11

F

- File names, 11
- File selection, 107
- File structure , 14
- Filter criteria, 461
- Form field
 - Button, 469
 - Deleting, 470
 - Edit, 470
 - Editing Box, 468
 - Text field, 467

- Form Field:Delete, 470
- Form Field:Edit, 470
- Form view, 449, 450, 465, 471
 - Defining, 456
- Form View:Define, 456
- Function test, 18
- Functionality of the OPC A&E server
 - Conditional event, 552
 - Simple event, 552
 - Tracking event, 552
- Functionality of the WinCC OPC A&E Server
 - Simple Event, 541
- Functions, 103
- FunctionTrendControl
 - Configuring the data export operation, 438
 - Configuring the status bar, 434
 - Configuring the toolbar, 434
 - Online configuration, 439

G

- Group Messages, 162
- Groups, 141

H

- Handles of the Standard Functions (User Archives), 366, 368

I

- I/O field configuration, 507
 - Example of OPC DA, 507
- Import, 11, 12, 14
- Importing, 11, 12, 14
- Increment, 19, 21
- Integration of COM server objects, 485
- Interface, 114, 115, 117, 118
 - Menu, 114, 115, 117
- Interface for selection of parameters, 488
- Interface IWinCCProtProvider, 488
- Interface IWinCCProtProviderPicture, 488
- Interface IWinCCProtProviderTable, 488
- Interface IWinCCProtProviderText, 488
- Introduction to User Archives, 299
- Item handle, 525
- Item ID, 525

L

- Limit, 167
- Limit Value Monitoring, 167
- Limit value monitoring/Limit, 167
- Lower limit, 21

M

- Max size, 550
- Memory location, 227
 - Project folder, 228
 - WinCC Project, 227
- Menu, 114, 115, 117
 - Dropdown menu, 115
 - Shortcut Menu, 117
 - toolbar, 114
- Menu commands, 306
- Message blocks, 156
- Message classes, 158
- Message classes on OPC A&E servers, 556
- Message classes/Message types, 158
- Message types, 158
- Message types on OPC A&E servers, 556
- Microsoft Excel, 112, 113
- Modification Configuration of User Archives, 339

N

- Navigation window, 300
- New user, 568

O

- Offset, 21
- Online configuration, 439
- OnlineTableControl
 - Configuring the data export operation, 438
 - Configuring the status bar, 434
 - Configuring the toolbar, 434
 - Marking the selected cells and rows, 430, 433
 - Online configuration, 439
 - Properties of the table elements, 430
 - Sorting by column header, 430, 433
 - Table element colors, 430, 431

- OnlineTrendControl
 - Configuring the data export operation, 438
 - Configuring the status bar, 434
 - Configuring the toolbar, 434
 - Online configuration, 439
- OPC, 493
 - Commissioning, 568
 - Configuring access to a WinCC archive tag using the HDA server browser, 536
 - Configuring access to the WinCC message system, 550
 - Example, 510, 511, 512, 514, 517, 518, 519, 523
 - Example:, 506, 507, 533
 - Functionality of the OPC A&E server, 552
 - Functionality of the WinCC OPC A&E Server, 541
 - Functionality of the WinCC OPC DA server, 502
 - Functionality of the WinCC OPC HDA server, 524
 - Functionality of WinCC OPC XML server, 498
 - HDA server browser, 534
 - New user, 568
 - OPC in WinCC, 496
 - ProgID, 504
 - Setting up user account on the OPC computers, 568
 - Specifications, 494
 - Trace, 574
 - Using Multiple OPC DA Servers, 503
 - WinCC as OPC DA Server, 496
 - WinCC as OPC XML Server, 496
- OPC A&E Client
 - Example of OPC A&E , 549
- OPC A&E server, 552, 553
 - Quality codes, 562
- OPC A&E Server, 541, 549, 550
 - Configuring access to the WinCC message system, 550
 - Quality codes, 549
- OPC A&E servers, 498
- OPC A&Raw Data, 549
- OPC DA server, 502
 - Functionality of the WinCC OPC DA server, 502
 - Using multiple OPC DA servers, 503
- OPC HDA server, 524
 - Assemblies, 527
 - Attributes, 526
 - Bounding values, 525
 - Data structure, 525, 526
 - Item handle, 525
 - Item ID, 525
 - Principle of Operation, 524
 - Quality codes, 530, 549
 - Raw data, 524
 - Supported functions, 528
 - Time Format of a WinCC OPC HDA server, 528
 - Write Accesses, 531
- OPC Item Manager, 504
- OPC XML, 570, 572
 - Security settings in IIS, 570
 - Testing installation, 572
- OPC XML server, 498
 - Installation, 500, 502
 - Installing of .NET Framework 1.1, 501
- OPCScout new project1
 - Example of OPC DA, 514
- Open Connectivity
 - OPC, 493
 - Specifications, 494
- Operate
 - WinCC Configuration Tool, 118
- Operating system
 - WinCC Configuration Tool, 113
- Operating the User Archive Control, 473
- Operating the User Archive Table Element, 473
- Operation
 - Applying a filter, 208
 - Copying data, 206
 - Cutting data, 206
 - Deleting data, 201
 - Sorting data, 207
 - Writing data, 196
- Operation in Runtime, 477
- Oscillation, 19, 21
- Output of user-specific data, 486
- Overshoot, 21

P

- Performance to read and write tags, 364
- Period of oscillation, 21
- Print job properties, 100
- Print jobs, 99
- Process peripherals, 19
- Process tag, 19
- Process value archive tag, 178

Process value archives, 178
 Processing data, 195
 ProgID
 Querying, 504
 Project
 Example of OPC DA, 507
 Project folder, 121
 With connection, 124, 126
 Without connection, 123
 Project Properties, 134
 Project selection, 106
 Properties
 of a User Archive, 325
 Properties of User Archives Table Elements, 480
 Properties:of a User Archive, 325

Q

Quality codes, 530, 549, 562
 Quantity framework, 363
 Quantity structure
 WinCC Configuration Tool, 231

R

Random numbers, 19
 Random numbers , 21
 Raw data, 524
 Raw data tags, 355, 356, 357, 358, 359
 Reading error, 188
 Recipe, 302, 303
 Redundancy and User Archive , 350
 Registering, 485
 COM server object, 485
 Registry entries of COM server, 488
 Relations, 346
 Report page, 99
 Requirements
 WinCC Configuration Tool, 113
 Runtime
 Form view, 471
 Operating the User Archive Control, 473
 Operating the User Archive Table Element, 473, 477
 Operation of the UserArchiveControl, 442
 Table View, 470
 Runtime:Form View, 471
 Runtime:Table View, 470

S

Script, 102
 Script management, 109
 Scripts, 371
 Editing User Archives, 366
 Setpoint value, 21
 Shortcut Menu
 Creating objects, 213, 217, 222
 Deleting data, 203
 Generate archive tags, 213
 Generate limit monitors, 222
 Generate single messages, 217
 Writing data, 200
 SIMATIC S5, 360
 SIMATIC S7, 354, 355, 360
 SIMATIC S5, 354, 355
 Simulation, 18
 Simulator, 18
 Sine, 19
 Sine wave, 21
 Single messages., 164
 Slider, 19, 21
 Sort Criteria, 461
 Spreadsheet, 131
 SQL, 346, 361, 461, 473, 475, 476
 Standard functions, 301
 Standard functions (User Archives), 301, 365
 Editing User Archives, 366
 for configuring user archives, 382
 Handles, 366, 368
 Runtime functions, 394, 400
 Standard functions (User Archives):Example of use, 371
 Start value, 21
 Status bar, 434
 Status Bar, 118, 442
 Stop value, 21
 Structure, 32
 Structure for Programming the "uaAddArchive"
 Handle, 392
 Structure instance elements, 143
 Structure tags, 145
 Structure type elements, 148
 Structure types, 14, 148
 System requirements, 244
 System Requirements
 WinCC Configuration Tool, 113

- T**
- Table View, 449, 450, 470
 - Defining, 456
 - Table View:Define, 456
 - Table window, 300, 316
 - Tables
 - Marking the selected cells and rows, 430, 433
 - Properties of the table elements, 430, 431
 - Sorting by column header, 430, 433
 - Table element colors, 430, 431
 - Tag
 - Configuring HDA server browser, 536
 - Example of OPC DA, 514, 517, 523
 - Example of OPC HDA, 536, 537
 - HDA server browser, 534
 - Tag dialog, 191
 - Tag logging, 243
 - Tag Logging, 171
 - Worksheets, 171, 176, 178, 181
 - Tag Logging Default Values, 171
 - Tag logging editor, 243
 - Tag simulator, 18
 - Tags, 14, 142
 - Target computer, 13, 14
 - Text Library, 184
 - Worksheets, 185
 - Texts, 185
 - Timers, 176
 - Tips
 - Address generation, 235
 - Address strings Simatic S7 ProtocolSuite, 240
 - Changing the Spreadsheet, 233
 - Excel Performance, 235
 - Load tag, 242
 - Packages, 242
 - Permitted characters, 239
 - Row limitation, 235
 - Special characters, 239
 - VBA Macros, 239
 - toolbar, 114
 - Toolbar, 434, 442
 - Operating the User Archive Control, 473
 - Operating the User Archive Table Element, 473
 - TrendRulerControl
 - Configuring the data export operation, 438
 - Configuring the status bar, 434
 - Configuring the toolbar, 434
 - Marking the selected cells and rows, 430, 433
 - Online configuration, 439
 - Properties of the table elements, 430, 433
 - Sorting by column header, 430, 433
 - Table element colors, 430, 431
- U**
- uaAddArchive, 382
 - uaAddField, 383
 - uaArchiveClose, 401
 - uaArchiveDelete, 402
 - uaArchiveExport, 402
 - uaArchiveGetCount, 403
 - uaArchiveGetFieldLength, 404
 - uaArchiveGetFieldName, 404
 - uaArchiveGetFields, 405
 - uaArchiveGetFieldType, 405
 - uaArchiveGetFieldValueDate, 406
 - uaArchiveGetFieldValueDouble, 407
 - uaArchiveGetFieldValueFloat, 407
 - uaArchiveGetFieldValueLong, 408
 - uaArchiveGetFieldValueString, 408
 - uaArchiveGetFilter, 409
 - uaArchiveGetID, 410
 - uaArchiveGetName, 410
 - uaArchiveGetSort, 411
 - uaArchiveImport, 411
 - uaArchiveInsert, 412
 - uaArchiveMoveFirst, 413
 - uaArchiveMoveLast, 413
 - uaArchiveMoveNext, 413
 - uaArchiveMovePrevious, 414
 - uaArchiveOpen, 414
 - uaArchiveReadTagValues, 415
 - uaArchiveReadTagValuesByName, 416
 - uaArchiveRequery, 416
 - uaArchiveSetFieldValueDate, 417
 - uaArchiveSetFieldValueDouble, 418
 - uaArchiveSetFieldValueFloat, 418
 - uaArchiveSetFieldValueLong, 419
 - uaArchiveSetFieldValueString, 420
 - uaArchiveSetFilter, 420
 - uaArchiveSetSort, 421
 - uaArchiveUpdate, 422
 - uaArchiveWriteTagValues, 422
 - uaArchiveWriteTagValuesByName, 423

- uaConfigArchive "Structure for Programming the Handles", 393
- uaConnect, 395
- uaDisconnect, 395
- uaGetArchive, 383
- uaGetField, 384
- uaGetLastError, 379
- uaGetLastHResult, 381
- uaGetLocalEvents, 395
- uaGetNumArchives, 385
- uaGetNumFields, 385
- uaIsActive, 396
- uaOpenArchives, 396
- uaOpenViews, 397
- uaQueryArchive, 397
- uaQueryArchiveByName, 398
- UaQueryConfiguration, 386
- uaReleaseArchive, 399
- uaReleaseConfiguration, 386
- uaRemoveAllArchives, 387
- uaRemoveAllFields, 387
- uaRemoveArchive, 388
- uaRemoveField, 388
- uaSetArchive, 389
- uaSetField, 390
- uaSetLocalEvents, 399
- uaUsers, 400
- Upper limit, 21
- User account, 568
 - Making OPC computers known, 568
- User archive
 - Selection of data in the UserArchiveControl , 446
- User archive:Create actions, 366
- User archives
 - Access rights, 322
 - Application Options, 302
 - Archive fields, 325
 - Change Configuration, 339
 - Communication, 320, 354
 - configuring, 317, 329, 350
 - Configuring, 317, 320, 321, 322, 325
 - Control tags, 321
 - Create, 317, 320, 321, 322, 325
 - Editing With Scripts, 366
 - Editor, 300
 - Performance Features, 304
 - Properties, 325
 - Quantity framework, 304
 - Standard functions, 365
 - Views, 340
- User Archives, 299
 - User Archives Control, 449
 - configuring, 451, 453
 - Define Columns, 458
 - Define Fonts, 463
 - Defining colors, 464
 - Defining output format, 458
 - Deleting, 454
 - Filter criteria, 461
 - Form view, 465, 471
 - Operation in Runtime, 473
 - Sort Criteria, 461
 - Status Bar, 460
 - Table View, 470
 - Toolbar, 459
 - User Archives Table Element, 301, 449
 - configuring, 451, 453
 - Define Access Types, 456
 - Define User Archive, 456
 - Define View, 456
 - Features, 455
 - Form view, 465
 - Form View:Define, 456
 - Object properties, 455
 - Operation in Runtime, 473, 477
 - Table View:Define, 456
 - User Archives Table Element:Configuration, 451
 - User Archives Table Element:Define Access Types, 456
 - User Archives Table Element:Define Colors, 464
 - User Archives Table Element:Define Columns, 458
 - User Archives Table Element:Define Fonts, 463
 - User Archives Table Element:Define Form View, 456
 - User Archives Table Element:Define Output Format, 458
 - User Archives Table Element:Define Table View, 456
 - User Archives Table Element:Define User Archive, 456
 - User Archives Table Element:Define View, 456
 - User Archives Table Element>Delete, 454
 - User Archives Table Element:Filter Criteria, 461
 - User Archives Table Element:Form View, 471
 - User Archives Table Element:Sort Criteria, 461
 - User Archives Table Element>Status Bar, 460
 - User Archives Table Element:Table View, 470
 - User Archives Table Element:Toolbar, 459
 - User archives: Using WinCC Redundancy , 350
 - User archives:Access rights, 322
 - User Archives:Application areas, 302
 - User archives:Archive fields, 325
 - User Archives:Configuring, 317, 321, 322, 325, 329

- User Archives:Configuring , 350
- User archives:control tags, 321
- User archives:Creating, 317, 321, 322, 325
- User Archives:Editor, 300
- User Archives:Performance markers, 304
- User archives:Properties, 325
- User archives:Views, 340
- UserArchiveControl, 425, 427
 - Configuring the data export operation, 438
 - Configuring the status bar , 434
 - Configuring the toolbar, 434
 - Determining the contents:, 427
 - Functional scope, 424
 - Marking the selected cells and rows, 430
 - Online configuration, 439
 - Operation in Runtime, 442
 - Properties of the table elements, 430
 - Sorting by column header, 430, 433
 - Sorting the displayed user archive data, 448
 - Table element colors, 430, 431

V

- Variable table
 - Generate objects, 211
 - Shortcut Menu, 211
- Views, 302, 303
 - configuring, 345, 348
 - Configuring, 340, 343, 345, 346, 348, 349
 - Create, 340
 - Relations, 346
- Views:Configuring, 340, 343, 345, 346, 348, 349
- Views:Creating, 340
- Views:Relations, 346
- Visualization of messages
 - Marking the selected cells and rows, 433
 - Properties of the table elements, 430
 - Sorting by column header , 433
 - Table element colors, 431

W

- WinCC
 - as OPC DA client, 496
 - as OPC-Server in a distributed system, 496
 - OPC in WinCC, 496
- WinCC - Microsoft Excel Connection, 518
 - Example of OPC DA, 518
- WinCC - OPC HDA client connection, 533
 - Example of OPC HDA, 533

- WinCC - SIMATIC NET FMS OPC server connection, 510
 - Example of OPC DA, 510
- WinCC - SIMATIC NET S7 OPC server connection, 512
 - Example of OPC DA, 512
- WinCC - WinCC connection, 506
 - Example of OPC DA, 506
- WinCC AlarmControl
 - Standard functions, 441
- WinCC archive, 243
- WinCC Archive ConfigurationTool, 243
 - Adding tables, 257
 - analog archive tag parameters, 263
 - Archive parameter, 260, 276
 - Archive tag parameters, 263, 270, 280
 - Archive tags, 289
 - Binary archive tag parameters, 263
 - Changing archive tags, 294
 - Changing archives, 289
 - Checking, 284
 - Checking archive data, 284, 290
 - Configuring a process value archive, 259
 - Configuring archive parameters, 260, 276
 - Configuring archive tag parameters, 263, 270, 280
 - Configuring compressed archives, 276
 - Configuring parameters, 260, 263, 270, 276, 280
 - Creating an archive folder, 253
 - Creating archive tags, 294
 - Creating archives, 245, 289, 290
 - Deleting archive tags, 296
 - Deleting archives, 289, 292
 - Dialog for tag selection, 274
 - Error list, 288
 - Error text, 288
 - Inserting tables, 257
 - Operation, 252
 - Process-controlled archive tag parameters, 270
 - Quick start, 245
 - Reading archives, 255
 - Status, 288
 - System requirements, 244
 - Tag logging editor, 260, 263, 270, 276, 280
 - Tag selection, 274
 - Using the error dialog, 286
 - Working with WinCC Archive, 243
 - Working without WinCC, 244

- WinCC Configuration Tool, 112, 243
 - Alarm logging, 150
 - Archive Tag dialog, 193
 - Compressed archive, 182
 - Compressed archive tag, 182
 - Connections/Groups, 141
 - Data manager, 136
 - Diagnosis, 228
 - Dialogs, 189
 - Dropdown menu, 115
 - Fundamentals, 112
 - Group Messages, 162
 - Insert Project Folder, 121
 - Interface, 114, 118
 - Memory location, 227
 - Menu, 114
 - Message blocks, 156
 - Operating system, 113
 - Operation, 118
 - Process value archive tag, 178
 - Process value archives, 178
 - Processing data, 195
 - Project folder, 123, 124, 126
 - Project Properties, 134
 - Quantity structure, 231
 - Requirements, 113
 - Shortcut Menu, 117
 - Single messages., 164
 - Status Bar, 118
 - Structure tags, 145
 - Structure type elements, 148
 - Structure types, 148
 - System Requirements, 113
 - Tag dialog, 193
 - Tag Logging, 171
 - Tag Logging Default Values, 171
 - Tags, 142
 - Text Library, 184
 - Texts, 185
 - Timers, 176
 - Tips, 231
 - Worksheets, 131
- WinCC CrossReferenceAssistant, 101, 102
- WinCC Explorer-OPC_Client.MPC, 504
- WinCC message system
 - Attributes, 545, 558
 - Configuring access to the WinCC message system, 550
 - Mapping WinCC Message Classes and Message Types, 544, 545
 - on OPC A&OPC-A&WinCC message system, 543
 - WinCC Message Classes on OPC A&E Server, 544
- WinCC OPC A&E server
 - Hierarchical access, 553
- WinCC Project, 124, 126
 - Creating, 129
- WinCC report system, 99
- WinCC Tags, 354
- WinCC User Archives Control, 449
- WinCC User Archives Element, 301
- WinCC User Archives Table Element, 301, 449
- Worksheets
 - Alarm logging, 150
 - Data manager, 136
 - Project Properties, 134
 - Reading error, 189
 - Tag Logging, 171
 - Text Library, 184
 - Write error, 188
- Write error, 187

