SIEMENS

SIMATIC HMI

WinCC V8.1 Update 3 WinCC: Configurations and communication (Part 2: Communication channels)

System Manual

Alarm channel 1
Allen Bradley - Ethernet IP 2
Mitsubishi Ethernet 3
Modbus TCPIP 4
Omron Ethernet-IP 5
OPC Channel 6
OPC UA WinCC Channel 7
PROFIBUS DP 8
S5 Ethernet Layer 4 9
S5 PROFIBUS FDL 10
S5 Programmers Port AS511 11
S5 Serial 3964R 12
SIMATIC S7 Protocol Suite 13
SIMATIC S7-1200, S7-1500 Channel 14
SIMATIC TI Ethernet Layer 4 15
SIMATIC TI Serial 16
SIMOTION 17
SinumerikNC 18
System Info 19
WinCC Unified Channel 20

06/2025
A5E54073177-AC

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

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

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, 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 complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens Aktiengesellschaft. 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.

Siemens Aktiengesellschaft
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E54073177-AC
Ⓟ 06/2025 Subject to change

Copyright © Siemens 2024 - 2025.
All rights reserved

Table of contents

1 Alarm channel 13

1.1 Channel "Alarm Channel" 13

1.2 Overview of supported data types 14

1.3 Configuring the channel 15
1.3.1 How to configure the channel 15
1.3.2 How to configure the connection 15
1.3.3 How to create a filter tag 16

2 Allen Bradley - Ethernet IP 19

2.1 WinCC Channel "Allen Bradley - Ethernet IP" 19

2.2 Channel Unit Assignment 20

2.3 Supported Data Types 21

2.4 Configuring the Channel 22
2.4.1 Configuring the Channel "Allen Bradley - Ethernet IP" 22
2.4.2 How to configure a connection for the "Allen Bradley - Ethernet IP" channel 23
2.4.3 Examples: Communication path 24
2.4.4 Configuring the tags 26
2.4.4.1 Configuring the tags 26
2.4.4.2 Addressing 28
2.4.4.3 Addressing syntax 29
2.4.4.4 Addressing Types 30
2.4.4.5 Examples for Addressing 31
2.4.4.6 How to configure a tag for the Allen Bradley E/IP ControlLogix channel unit 32
2.4.4.7 How to configure a tag with bit by bit access for Allen Bradley E/IP PLC5 or SLC50x 33
2.4.4.8 How to configure a tag with byte by byte access for Allen Bradley E/IP PLC5 or SLC50x 34
2.4.4.9 How to configure a tag with word by word access for Allen Bradley E/IP PLC5 or SLC50x 35
2.4.4.10 How to configure a text tag for Allen Bradley E/IP PLC5 or SLC50x 36

3 Mitsubishi Ethernet 39

3.1 WinCC channel "Mitsubishi Ethernet" 39

3.2 Supported data types 40

3.3 Configuring the Channel 41
3.3.1 Configuring the "Mitsubishi Ethernet" channel 41
3.3.2 How to configure a "Mitsubishi FX3U Series" channel unit connection 42
3.3.3 How to configure a "Mitsubishi Q Series" channel unit connection 44
3.3.4 How to configure a "Mitsubishi iQ-R series" channel unit connection 45
3.3.5 How to configure a connection of the "Mitsubishi iQ-F series" channel unit 47
3.3.6 Configuring the tags 48
3.3.6.1 Configuring the tags 48
3.3.6.2 How to configure a tag 50

4 Modbus TCPIP 53
4.1 "Modbus TCP/IP" channel 53
4.2 Supported Data Types 54
4.3 Configuring the Channel 55
4.3.1 Configuring the "Modbus TCPIP" Channel 55
4.3.2 How to configure a connection 56
4.3.3 Configuring the tags 57
4.3.3.1 Configuring the tags 57
4.3.3.2 How to Configure a Tag with Bit by Bit Access 59
4.3.3.3 How to Configure a Tag with Word by Word Access 60
4.3.3.4 How to Configure a Text Tag 61
5 Omron Ethernet-IP 63
5.1 WinCC channel "Omron Ethernet IP" 63
5.2 Supported data types 64
5.3 Configuring the channel 65
5.3.1 Configuration of the "Omron Ethernet IP" channel 65
5.3.2 How to configure a connection of the "Omron Ethernet-IP CJ1 Series" channel unit 66
5.3.3 Configuring the tags 67
5.3.3.1 Configuring the tags 67
5.3.3.2 How to configure a tag 68
5.3.3.3 Address ranges for Omron Ethernet/IP 70
6 OPC Channel 75
6.1 WinCC OPC Channel 75
6.2 OPC Item Manager 77
6.3 Overview of the Supported WinCC Data Types 81
6.4 WinCC OPC DA Client 82
6.4.1 Functionality of the WinCC OPC DA Client 82
6.4.2 How to Access a WinCC Tag with the OPC Item Manager 83
6.4.2.1 How to Access a WinCC Tag with the OPC Item Manager 83
6.4.2.2 Configuring the OPC Channel on the WinCC OPC DA Client 84
6.4.2.3 Configuring Access with the OPC Item Manager 84
6.4.3 Accessing a WinCC Tag without the OPC Item Manager 88
6.4.4 Using Structures on a WinCC OPC DA Client 90
6.4.4.1 How to Use Structures on the WinCC OPC DA Client 90
6.4.4.2 Configuring Structures and Structure Tags on the WinCC OPC DA Server 90
6.4.4.3 How to Configure Structures on the WinCC OPC DA Client 92
6.4.5 Error Handling in the Event of Disturbed OPC DA Communication 93
6.4.5.1 Error Handling in the Event of Disturbed OPC Communication 93
6.4.5.2 WinCC as OPC DA Server 94
6.4.5.3 WinCC as OPC DA Client 96
7 OPC UA WinCC Channel 99
7.1 WinCC channel "OPC UA WinCC Channel" 99
7.2 Overview of the supported data types 101
7.3 Overview of the supported OPC UA functionalities 102

7.4 Configuring the OPC UA channel ... 104
7.4.1 General sequence ... 104
7.4.2 Inserting an OPC UA WinCC channel in the WinCC project ... 104
7.4.3 Creating a connection to the OPC UA server ... 105
7.4.4 Creating a connection to the OPC UA server via a NAT router ... 110
7.4.5 Setting up authentication via certificates. ... 112
7.4.6 How to configure a connection to the OPC UA server ... 120
7.4.7 Configuring the OPC UA tags ... 123
7.4.7.1 OPC UA tags ... 123
7.4.7.2 The "Symbols" view ... 126
7.4.7.3 To import an OPC UA node as a WinCC tag ... 127
7.4.8 Using OPC UA types in WinCC ... 129
7.4.8.1 Importing OPC UA types as WinCC structure types ... 129
7.4.8.2 How to use automatically generated structure types ... 131
7.4.8.3 How to use manually generated structure types ... 133
7.4.9 Using OPC UA alarms in WinCC ... 135
7.4.9.1 The "Monitored objects" view ... 135
7.4.9.2 Filters for the OPC UA alarms ... 138
7.4.9.3 How to import Event Notifiers as WinCC messages ... 143

7.5 OPC UA Arrays in the OPC UA WinCC Channel ... 147
7.5.1 Arrays in WinCC ... 147
7.5.2 How to create and configure OPC UA Arrays manually ... 150

7.6 OPC UA methods in the OPC UA WinCC Channel ... 153
7.6.1 OPC UA methods in WinCC ... 153
7.6.2 How to use OPC UA methods in client projects ... 155

7.7 Error handling ... 156
7.7.1 Error Handling in the Event of Disturbed OPC Communication ... 156

8 PROFIBUS DP ... 157

8.1 WinCC Channel "PROFIBUS DP" ... 157

8.2 Properties of the WinCC driver Profibus DP ... 158

8.3 Integrating the "Profibus DP" driver ... 159

8.4 Configuring the "Profibus DP" driver ... 160

9 S5 Ethernet Layer 4 ... 163

9.1 WinCC Channel "SIMATIC S5 Ethernet Layer 4" ... 163

9.2 Data type of the tags ... 165

9.3 Configuring the Channel ... 166
9.3.1 Configuring the channel "SIMATIC S5 Ethernet Layer 4" ... 166
9.3.2 How to configure the connection ... 166
9.3.3 Configuring the tags ... 168
9.3.3.1 Configuring the tags ... 168
9.3.3.2 Addresses of tags ... 169
9.3.3.3 How to Configure a Tag with Bit by Bit Access ... 170
9.3.3.4 How to Configure a Tag with Byte by Byte Access ... 171
9.3.3.5 How to Configure a Tag with Word by Word Access ... 172
9.3.3.6 How to configure a raw data tag ... 174
9.3.4 System parameters ... 175

9.3.4.1 System parameters of the channel unit ... 175
9.3.4.2 How to Change the Device Name ... 176
9.3.4.3 How to change the transport parameter ... 178

9.4 Appendix ... 179
9.4.1 Appendix ... 179
9.4.2 Internal error codes and constants ... 179
9.4.2.1 Internal error codes and constants ... 179
9.4.2.2 Error codes during connection disturbances ... 179
9.4.2.3 iNA960 messages ... 182
9.4.2.4 SCI messages ... 183

10 S5 PROFIBUS FDL ... 187

10.1 WinCC channel "SIMATIC S5 Profibus FDL" ... 187

10.2 Supported data types and data ranges ... 189

10.3 Features of the WinCC channel "SIMATIC S5 Profibus FDL" ... 191

10.4 Configuring the Channel ... 193
10.4.1 How to configure the channel "SIMATIC S5 Profibus FDL" ... 193
10.4.2 Channel unit "FDL (CP5412/A2-1)" ... 193
10.4.3 How to configure a connection ... 194
10.4.4 Configuring the tags ... 196
10.4.4.1 Configuring the tags ... 196
10.4.4.2 How to Configure a Tag with Bit by Bit Access ... 196
10.4.4.3 How to Configure a Tag with Byte by Byte Access ... 197
10.4.4.4 How to configure a tag with word by word access ... 198
10.4.5 System parameters ... 199
10.4.5.1 System parameters of the channel unit ... 199
10.4.5.2 How to Change the Device Name ... 200
10.4.5.3 How to change the write/read monitoring time of process values ... 201

10.5 Special Functions ... 202
10.5.1 Special functions of the "SIMATIC S5 Profibus FDL" Channel ... 202
10.5.2 Raw data tags of the "SIMATIC S5 Profibus FDL" channel ... 202
10.5.2.1 Raw data tags of the "SIMATIC S5 Profibus FDL" channel ... 202
10.5.2.2 How to configure raw data tags ... 203
10.5.3 Configuring the communication types ... 204
10.5.3.1 Configuring the communication types ... 204
10.5.3.2 How to configure an active data transfer ... 204
10.5.3.3 How to configure a passive data transfer ... 206

10.6 Example of configuring the "SIMATIC S5 Profibus FDL" channel ... 208
10.6.1 Example of configuring the "SIMATIC S5 Profibus FDL" channel ... 208
10.6.2 How to configure the data handling blocks in the AS ... 208
10.6.3 How to configure an I/O Field ... 211

11 S5 Programmers Port AS511 ... 213

11.1 WinCC channel "SIMATIC S5 Programmers Port AS511" ... 213

11.2 Data type of the tags ... 214

11.3 Configuring the Channel ... 215
11.3.1 Configuring the "SIMATIC S5 Programmers Port AS511" channel ... 215
11.3.2 How to configure the connection ... 215

11.3.3 Configuring the tags ... 217
11.3.3.1 Configuring the tags ... 217
11.3.3.2 How to configure the address of a tag ... 217
11.3.3.3 How to configure a tag with bit-wise access ... 220
11.3.3.4 How to Configure a Tag with Byte by Byte Access ... 221

12 S5 Serial 3964R ... 223

12.1 WinCC channel "SIMATIC S5 Serial 3964R" ... 223

12.2 Data type of the tags ... 224

12.3 Configuring the Channel ... 225
12.3.1 Configuring the "SIMATIC S5 Serial 3964R" channel ... 225
12.3.2 How to configure the connection ... 225
12.3.3 Configuring the tags ... 227
12.3.3.1 Configuring the tags ... 227
12.3.3.2 How to configure the address of the tag ... 228
12.3.3.3 How to configure a tag with bit-wise access ... 229
12.3.3.4 How to Configure a Tag with Byte by Byte Access ... 231

13 SIMATIC S7 Protocol Suite ... 233

13.1 WinCC Channel "SIMATIC S7 Protocol Suite" ... 233

13.2 Channel unit selection ... 235

13.3 Overview of the supported data types ... 239

13.4 Configuring the Channel ... 240
13.4.1 "SIMATIC S7 Protocol Suite" Channel - Configuration ... 240
13.4.2 How to configure the "SIMATIC S7 Protocol Suite" channel ... 240
13.4.3 How to download AS symbols offline ... 241
13.4.4 How to configure AS structures ... 245
13.4.5 How to export AS project data ... 248
13.4.6 Channel units ... 250
13.4.6.1 Channel units of the "SIMATIC S7 Protocol Suite" channel ... 250
13.4.6.2 "Industrial Ethernet (I+II)" channel units" ... 250
13.4.6.3 "MPI" channel unit ... 253
13.4.6.4 "Named Connections" channel unit ... 255
13.4.6.5 "PROFIBUS (I+II)" channel units ... 258
13.4.6.6 "Slot PLC" channel unit ... 261
13.4.6.7 "Soft PLC" channel unit ... 263
13.4.6.8 "TCP/IP" channel unit ... 264
13.4.7 Configuring the tags ... 267
13.4.7.1 Configuring the tags ... 267
13.4.7.2 How to Configure a Tag with Bit by Bit Access ... 267
13.4.7.3 How to Configure a Tag with Byte by Byte Access ... 268
13.4.7.4 How to Configure a Tag with Word by Word Access ... 270
13.4.7.5 How to Configure a Text Tag ... 271
13.4.8 System parameters ... 272
13.4.8.1 System Parameters of the Channel Unit ... 272
13.4.8.2 Cyclic read services in PLC ... 273
13.4.8.3 How to Configure the System Parameters ... 274
13.4.8.4 How to Change the Logical Device Name ... 276

13.5 Special functions ... 279

13.5.1 Special functions of the "SIMATIC S7 Protocol Suite" Channel ... 279
13.5.2 Data exchange with the S7 function block AR_SEND ... 279
13.5.2.1 Data exchange with the S7 function block AR_SEND ... 279
13.5.2.2 Data Block - Structure and Parameters ... 280
13.5.2.3 Overview of the properties of the AR_SEND variants ... 285
13.5.2.4 AR_SEND variant for an archive tag ... 288
13.5.2.5 Example 1 for data block structure: An archive tag; each process value has a time stamp... 289
13.5.2.6 Example 2 for data block structure: One archive tag; equally spaced time stamp ... 290
13.5.2.7 Example 3 for data block structure: An archive tag; each process value has its own time stamp ... 290
13.5.2.8 Example 4 for data block structure: An archive tag; each process value with relative time stamp (time difference) ... 291
13.5.2.9 AR_SEND variant for multiple archive tags ... 292
13.5.2.10 Example 5 for data block structure: Multiple archive tags; each process value has its own time stamp ... 296
13.5.2.11 Example 6 for data block structure: Multiple archive tags; equally spaced time stamp ... 298
13.5.2.12 Example 7 for data block structure: Multiple archive tags; each process value has its own time stamp ... 300
13.5.2.13 Example 8 for data block structure: Multiple archive tags; process values with relative time stamp (time difference) ... 302
13.5.2.14 AR_SEND variant for multiple archive tags (optimized) ... 304
13.5.2.15 Example 9 for data block structure: multiple archive tags;optimized ... 305
13.5.2.16 How to configure the AR _SEND variant for an archive tag ... 306
13.5.2.17 How to configure the AR _SEND variant for multiple archive tags ... 308
13.5.3 Raw data tags of the Channel "SIMATIC S7 Protocol Suite" ... 310
13.5.3.1 Raw data tags of the Channel "SIMATIC S7 Protocol Suite" ... 310
13.5.3.2 Raw data tag as byte array ... 311
13.5.3.3 How to Configure a Raw Data Tag as Byte Array ... 312
13.5.3.4 Raw data tag for BSEND/BRCV functions of S7 communication ... 314
13.5.3.5 How to Configure a Raw Data Tag for ""BSEND/BRCV" functions ... 317
13.5.4 Software Redundancy ... 319
13.5.4.1 Software Redundancy ... 319
13.5.4.2 Software Redundancy - Connection-specific internal tags ... 320
13.5.4.3 How To Configure a Software Redundancy ... 324
13.5.4.4 How to Clear a Software Redundancy in WinCC ... 327
13.5.4.5 How to Check the WinCC Startup Parameters ... 327
13.5.4.6 How To Load WinCC's system messages into Alarm Logging ... 328
13.5.4.7 Error codes during connection disturbances ... 328

14 SIMATIC S7-1200, S7-1500 Channel ... 329

14.1 "SIMATIC S7-1200, S7-1500 Channel" channel ... 329

14.2 Overview of the supported data types ... 330

14.3 Configuring the channel ... 331
14.3.1 Configuration of the "SIMATIC S7-1200, S7-1500 Channel" channel ... 331
14.3.2 Configuring raw data tags ... 333
14.3.3 How to configure a connection ... 334
14.3.4 "Secure Communication" ... 336
14.3.5 Changing connections ("Change Connection") ... 336
14.3.6 How to configure a tag without optimized block access ... 340
14.3.7 How to configure a tag with optimized block access ... 341
14.3.8 How to download AS symbols offline ... 345

14.3.8.1 Providing an export from the TIA Portal project ... 345
14.3.8.2 Import with the Load Assistant ... 346
14.3.8.3 Configuring an import manually ... 352
14.3.8.4 Editing imported AS symbols ... 354
14.3.9 How to configure AS structures ... 355
14.3.10 How to update the AS structures ... 358
14.3.11 How to export AS project data ... 359
14.3.12 Redundant System S7-1500R/H ... 360
14.3.13 Configuring redundant control systems ... 362
14.3.14 Software redundancy for S7-1500R/H ... 365

15 SIMATIC TI Ethernet Layer 4 ... 371

15.1 WinCC channel "SIMATIC TI Ethernet Layer 4" ... 371

15.2 Data type of the tags ... 372

15.3 Configuring the Channel ... 373
15.3.1 Configuring the channel "SIMATIC TI Ethernet Layer 4" ... 373
15.3.2 How to configure the connection ... 373
15.3.3 Configuring the tags ... 375
15.3.3.1 Configuring the tags ... 375
15.3.3.2 How to configure the address of a tag ... 375
15.3.3.3 How to configure a tag with bit-wise access ... 377
15.3.3.4 How to Configure a Tag with Byte by Byte Access ... 378
15.3.4 System parameters ... 379
15.3.4.1 System parameters of the channel unit ... 379
15.3.4.2 How to Change the Device Name ... 380
15.3.4.3 How to change the transport parameter ... 381

16 SIMATIC TI Serial ... 385

16.1 WinCC channel "SIMATIC TI Serial" ... 385

16.2 Data type of the tags ... 386

16.3 Configuring the Channel ... 387
16.3.1 Configuring the "SIMATIC TI Serial" channel ... 387
16.3.2 How to configure the connection ... 387
16.3.3 Configuring the tags ... 389
16.3.3.1 Configuring the tags ... 389
16.3.3.2 How to configure the address of a tag ... 389
16.3.3.3 How to configure a tag with bit-wise access ... 391
16.3.3.4 How to Configure a Tag with Byte by Byte Access ... 392

17 SIMOTION ... 395

17.1 WinCC channel "SIMOTION" ... 395

17.2 Overview of the supported data types ... 396

17.3 Configuring the channel ... 397
17.3.1 Configuration of the "SIMOTION" channel ... 397
17.3.2 How to export a SIMOTION SCOUT project ... 397
17.3.3 How to create a WinCC project with Simotion Mapper ... 398
17.3.4 How to change a WinCC project with Simotion Mapper ... 399
17.3.5 How to change the connection parameters ... 401
17.3.6 How to change the tag address ... 402

17.3.7 System parameter configuration ... 403
17.3.7.1 System Parameters of the Channel Unit ... 403
17.3.7.2 How to Configure the System Parameters ... 403
17.3.7.3 How to Change the Logical Device Name ... 405

17.4 Diagnosis "SIMOTION" channel ... 407
17.4.1 Diagnosis possibilities of the "SIMOTION" channel ... 407
17.4.2 Description of Log File Entries ... 407

18 SinumerikNC ... 411

18.1 Supported Data Types ... 412

18.2 Configuring the channel ... 413
18.2.1 How to configure a connection to a SINUMERIK 840D sl ... 413
18.2.2 Configuring the tags ... 414
18.2.2.1 How to configure a tag ... 415
18.2.2.2 Configuring GUD tags ... 416
18.2.2.3 Configuring PLC tags ... 417
18.2.3 How to configure alarms on the SINUMERIK 840D sl ... 419
18.2.4 How to configure SINUMERIK S7-300 alarms ... 420

18.3 Executing system functions ... 423
18.3.1 Supported system functions ... 423
18.3.2 LogonNC ... 423
18.3.3 LogoffNC ... 423
18.3.4 ChangeNCPassword ... 424
18.3.5 ConfigureNCMachineData ... 424
18.3.6 ResetNC ... 424
18.3.7 AcknowledgeNCCancelAlarms ... 425
18.3.8 SetNCUserFrame ... 425

19 System Info ... 427

19.1 "System Info" Channel ... 427

19.2 WinCC System Info Channel ... 428

19.3 Overview of the Supported System Information ... 430

19.4 Differences to Other Software Components ... 435

19.5 Configuring the Channel ... 436
19.5.1 How to Configure the System Info Channel ... 436

19.6 Examples of Evaluating and Displaying System Information ... 437
19.6.1 How To Call Up and Evaluate System Information ... 437
19.6.2 How to Configure a Tag in the System Info Channel ... 437
19.6.3 How to Display the Time in an I/O Field ... 438
19.6.4 How to Display the Free Disk Capacity in a Bar Graph ... 440
19.6.5 How to Display the CPU Load in a Trend Window ... 441
19.6.6 How to Configure a Message Regarding Free Disk Capacity ... 442
19.6.7 How to Display a Message regarding the Available Disk Capacity ... 444
19.6.8 How to Display the Printer Status in a Status Display ... 445
19.6.9 How to Check the WinCC Startup Parameters ... 447
19.6.10 How to Insert a Bar Graph ... 447
19.6.11 How to Insert an I/O Field ... 448
19.6.12 How to Start Runtime ... 448

19.7 Special Functions 449
19.7.1 Use in Multi-User and Client Systems 449
19.7.1.1 Use in Multi-User and Client Systems 449
19.7.2 Example of monitoring system information from multiple servers 449
19.7.2.1 Monitoring the system information of several servers on a WinCC client 449
19.7.2.2 How to Configure the First Server 450
19.7.2.3 How to Configure the Second Server 451
19.7.2.4 How to Import the Tags to the WinCC Client 452
19.7.2.5 How to Configure the Process Picture on the WinCC Client 453
19.7.2.6 How to Activate the Project 454

20 WinCC Unified Channel 455

20.1 Channel "WinCC Unified Channel" 455

20.2 Supported Data Types 456

20.3 Configuring the channel 457
20.3.1 Requirements 457
20.3.2 How to configure the channel 457
20.3.3 How to configure a connection 459
20.3.4 Configuring the tags 460
20.3.4.1 How to configure tags 461

Index 463

Alarm channel

1

1.1 Channel "Alarm Channel"

Tags for message archiving

In the communication channel "Alarm Channel" you configure tags for the message system.

The tag value contains the number of pending messages each which correspond to specific filter criteria.

Channel units

The "Alarm Channel" has an "Alarm Channel" channel unit.

Online configuration

Online configuration of the channel and the associated connections is not possible. However, you can create, delete, and change filter tags in a connection that has already been created.

Application

You use these tags, for example for message-dependent dynamizations of objects in the Graphics Designer.

Example:

- A tank is displayed in different colors, depending on the number of the messages which fulfil these two filter criteria:
  - Point of error "Kessel1"
  - Keyword "Temperature" in the message text

Note

Licensing

The process tags required for the "Alarm Channel" need no licenses.

Thus, the tags are not entered in the license count.

1.2 Overview of supported data types

The number of messages that match the filter criteria is written in tags.

The following data types are supported:

- Binary tag
- Unsigned 8-bit value
- Unsigned 16-bit value
- Unsigned 32-bit value
- Text tag, 8-bit character set
- Text tag, 16-bit character set

1.3 Configuring the channel

1.3.1 How to configure the channel

Requirement

- A WinCC project has been created.
- Tag Management is open.

Procedure

1. Open the Tag Management shortcut menu in the navigation area.
2. Select "Add new driver > Alarm Channel".

Result

The "Alarm Channel" is added to Tag Management.

The "Alarm Channel" channel unit is created under the channel. You can configure your connections under this channel unit.

1.3.2 How to configure the connection

In the connection, you can create the filter tags and select the servers to which the filter should be applied.

The number of filtered messages of the individual servers is added as tag value.

The status of the servers is mapped via the following quality codes:

Code	Quality	Status
0x80	GNC_OK	GOOD All servers can be reached.
0x40	UC_NS	UNCERTAIN At least one server can be accessed.
0x08	BAD_NC	BAD None of the servers can be accessed.

Requirement

- A WinCC project has been created.
- Tag Management is open.
- The "Alarm Channel" is added to Tag Management.

Procedure

1. In the Tag Management navigation area, select the channel unit "Alarm Channel" in the tree of the "Alarm Channel" communication driver.
2. Select the entry "New Connection" in the shortcut menu of the channel unit.
3. Enter the name of the connection.
4. Select the entry "Connection parameters" from the shortcut menu of the connection. The "Connection Parameters" dialog opens.
5. Under "Server selection" select one or more servers.
   Enable "All servers" to filter the messages of all servers in the WinCC project.
   The server names are displayed in the "Properties - Connection" area in the "Connection parameters" field.
6. Click "OK" to close the dialog.

Result

The connection is created in the channel unit. You can create the filter tags in the connection.

1.3.3 How to create a filter tag

An Alarm Channel tag contains the number of pending messages which correspond to the configured filter criteria.

Use the SQL statements for filtering messages in WinCC AlarmControl as filter criteria.

Define the filters for the configured message blocks via the selection dialog. The configured message blocks are displayed in the drop-down list.

Requirement

- A WinCC project has been created.
- Tag Management is open.
- The "Alarm Channel" is added to Tag Management.
- A connection has been created in the channel unit.

Procedure

1. In the Tag Management navigation area, select the connection under the channel unit "Alarm Channel".
2. In the data area, enter a tag name in the "Name" column.
3. Select the data type in the "Properties - tag" area or in the data area.

Note

Binary tags have the value "TRUE" if more than one message with the filter criteria is pending.

4. If required, add more information on tags in the "Comment" field.
5. To open the selection dialog, click the "MsgFilterSQL" field and on the button ....
6. Select the parameters of the SQL statement in the selection dialog.
   Alternatively you copy an SQL statement into the field "MsgFilterSQL".
7. Click "OK" to close the dialog.

Result

You have created a filter tag. In Runtime, the number of messages to which the filter criterion applies is written to the filter tag.

Allen Bradley - Ethernet IP

2

2.1 WinCC Channel "Allen Bradley - Ethernet IP"

Introduction

The channel "Allen Bradley - Ethernet IP" is used for linking to Allen-Bradley automation systems. Communication is via the Ethernet IP protocol.

Depending on the communication hardware used, the system supports connections via the following channel units:

- Allen Bradley E/IP PLC5
- Allen Bradley E/IP SLC50x
- Allen Bradley E/IP ControlLogix
- Allen Bradley E/IP GuardLogix

2.2 Channel Unit Assignment

Introduction

The channel unit must be selected for the channel in order to create a connection from WinCC to an existing or planned network.

Channel Unit Assignment

The following table shows an allocation of the channel units of channel "Allen Bradley - Ethernet IP" to the network and automation system (AS).

Channel unit of the channel	Communication network	AS
Allen Bradley E/IP PLC5	Ethernet IP	PLC-5 with Ethernet Port
Allen Bradley E/IP SLC50x	Ethernet IP	SLC 500 with Ethernet Port, e.g. SLC 5/05
Allen Bradley E/IP ControlLogix	Ethernet IP	ControlLogix 5500
Allen Bradley E/IP GuardLogix	Ethernet IP	GuardLogix 5580

2.3 Supported Data Types

Introduction

Define the required tags for a logical connection. The following data types are supported by the "Allen Bradley - Ethernet IP" channel:

- Binary tag
- Signed 8-bit value
- Unsigned 8-bit value
- Signed 16-bit value
- Unsigned 16-bit value
- Signed 32-bit value
- Unsigned 32-bit value
- Floating-point number 32-bit IEEE 754
- Text tag, 8-bit font (maximum length: 82 characters)
- Text tag, 16-bit font (maximum length: 82 characters)

2.4 Configuring the Channel

2.4.1 Configuring the Channel "Allen Bradley - Ethernet IP"

Introduction

WinCC needs a logical connection for communication of WinCC with the automation system (AS).

This section shows how the "Allen Bradley - Ethernet IP" channel is configured.

When using TCP/IP, you must specify the IP address of the AS for the logical link:

- The IP address consists of four numerical values, separated by dots.
- The numerical values must be within the range of 0-255.

Note

Timeout behavior

Interrupted connections are not detected immediately when the TCP/IP protocol is used.

The check-back message can take up to a minute.

Connectable controllers

Connections can be implemented for the following Allen-Bradley PLCs:

- Allen-Bradley CompactLogix 5500
- Allen-Bradley CompactLogix 5300
- Allen-Bradley CompactLogix 5580
- PLC-5 with Ethernet Port
- SLC 500 with Ethernet Port, e.g. SLC 5/05
- MicroLogix

Approved communication types

The following types of communication are system-tested and approved for the "Allen Bradley - Ethernet IP" channel:

- Point-to-point connection:
- Multiple-point connection from the WinCC station with an optional amount of controllers.

Online configuration

Online configuration of the "Allen Bradley - Ethernet IP" channel is not supported.

See also

How to configure a connection for the "Allen Bradley - Ethernet IP" channel (Page 23)

Configuring the tags (Page 26)

2.4.2 How to configure a connection for the "Allen Bradley - Ethernet IP" channel

Introduction

The "Allen Bradley - Ethernet IP" channel can be configured for four channel units:

- Allen Bradley E/IP ControlLogix
- Allen Bradley E/IP GuardLogix
- Allen Bradley E/IP PLC5
- Allen Bradley E/IP SLC50x

The configuration is the same for all three channel units and consists of the following tasks:

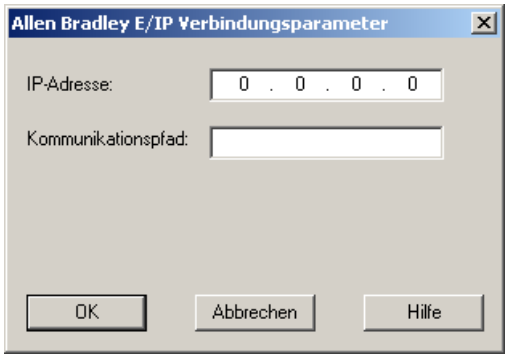
1. Configuring a connection
2. Configuring tags

Requirements

- The communication driver for channel "Allen Bradley - Ethernet IP" is installed and integrated into the project.

Procedure

1. Select the desired channel unit in the tag management.
2. Select the entry "New Connection" in the shortcut menu of the channel unit.
   A new connection is created.
3. Select the new connection.
   Enter the connection name under "General" in the "Properties - Connection" area.
4. Select the "Connection parameters" connection in the shortcut menu.
   The "Allen Bradley E/IP connection parameters" dialog opens.



5. Enter the IP address of the Ethernet/IP module of the controller.
   Port 44818 is permanently set by default for Ethernet IP devices.
6. Define the CIP path from the Ethernet module to the controller in the "Communication path" field.
   You can configure a direct connection as well as a connection via routing.
   Create a logical connection between the Ethernet module and the PLC, even if they are located in different CIP networks.
7. Close the dialog by clicking the "OK" button.

See also

Examples: Communication path (Page 24)

Configuring the tags (Page 26)

2.4.3 Examples: Communication path

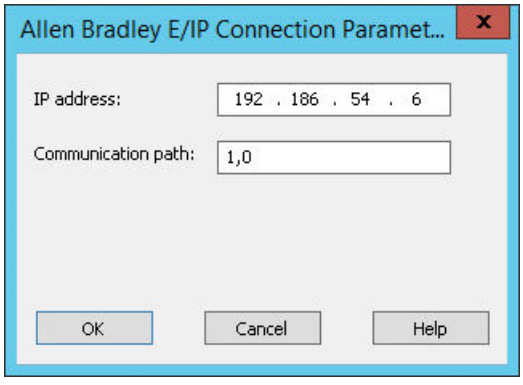
Example 1: Direct connection

Connection with a PLC in the same Allen-Bradley rack.

Communication path:

- 1,0

Number	Meaning
1	Stands for a backplane connection.
0	Stands for a CPU slot number.



Example 2: Connection via routing

Connection with a PLC in other Allen-Bradley racks.

Two Allen-Bradley racks are networked on Ethernet.

Two identical channel units are configured in the same way as two different channel units, e.g.:

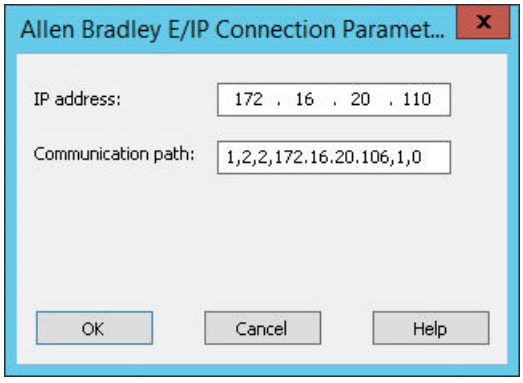
- WinCC - Control Logix (1) - Control Logix (2)
- WinCC - Control Logix (1) - SLC50x (1)

Although different protocols are used for routing to SLC50x or to Control Logix, for example, the address structure remains the same.

Communication path:

- 1,2,2,172.16.20.106,1,0

Number	Meaning
1	Backplane connection of the first module "Control Logix (1)".
2	Stands for the CPU slot number of the second Ethernet module.
2	Stands for an Ethernet connection.
172.16.20.106	IP address of a remote AB rack in the network, e.g. the third Ethernet module of the "Control Logix (2)" module
1	Backplane connection of the second module, e.g. "Control Logix (2)" or "SLC50x (1)".
0	Slot number of the CPU



Example 3: Connecting multiple modules via routing

Connection with a PLC in other Allen-Bradley racks.

More than two Allen-Bradley racks are connected to Ethernet.

Scenario:

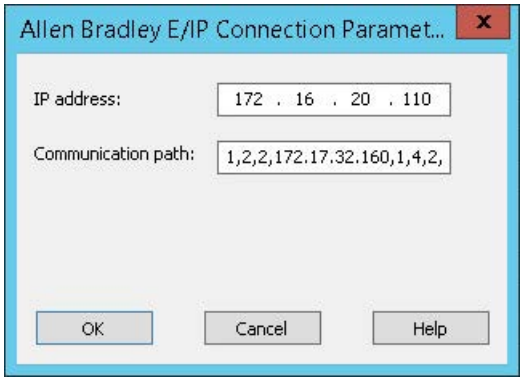
- WinCC - Control Logix (1) - Control Logix (2) - Control Logix (3)
- WinCC - Control Logix (1) - Control Logix (2) - SLC50x (1)

Although different protocols are used for routing to SLC50x or to Control Logix, for example, the address structure remains the same.

Communication path:

- 1,2,2,172.17.32.160,1,4,2,172.17.32.156,1,0

Number	Meaning
1	Backplane connection of the first module "Control Logix (1)".
2	Stands for the CPU slot number of the second Ethernet module.
2	Stands for an Ethernet connection.
172.17.32.160	IP address of a remote AB rack in the network, e.g. the first Ethernet module of the "Control Logix (2)" module
1	Backplane connection of the second module "Control Logix (2)".
4	Stands for the slot number of the third Ethernet module.
2	Stands for an Ethernet connection.
172.17.32.156	IP address of another AB rack in the network, e.g. the "Control Logix (3)" or "SLC50x (1)" module.
1	Backplane connection of the third module, e.g. "Control Logix (3)" or "SLC50x (1)".
0	Slot number of the CPU



See also

How to configure a connection for the "Allen Bradley - Ethernet IP" channel (Page 23)

2.4.4 Configuring the tags

2.4.4.1 Configuring the tags

Introduction

For a connection between WinCC and the automation system (AS) via channel "Allen Bradley - Ethernet IP", tags of different data types can be created in WinCC. The permitted data types are listed in this section.

Tag updating

If the tags are retrieved simultaneously in a picture from a PLC, the "Allen Bradley - Ethernet IP" channel attempts to optimize the update. This can only be accomplished under the following conditions however:

- The tags are in the same address range.
- The tags are as close to one another as possible within the address range.

If you do not observe these recommendations, it can lead to noticeable differences in the picture refresh with large amounts of tags. The acquisition cycles may not be maintained under certain circumstances.

The best performance for the connection is achieved if you observe the following rules when configuring the tags:

- Update of maximum 2000 tags simultaneously.
- Combine the tags in the least possible space, best in only one address range.

Valid data types

The selection of data types listed below can be used to configure tags.

Basic data types

Data type	Bit address space
Bool	-
SInt	0-7
USInt	0-7
Int	0-15
UInt	0-15
DInt	0-31
UDInt	0-31
Real	-
String	-

Arrays

Address	Valid data types
Array	SInt, USInt, Int, UInt, DInt, UDInt, Real

See also

Addressing (Page 28)

How to configure a tag for the Allen Bradley E/IP ControlLogix channel unit (Page 32)

How to configure a tag with bit by bit access for Allen Bradley E/IP PLC5 or SLC50x (Page 33)

How to configure a tag with byte by byte access for Allen Bradley E/IP PLC5 or SLC50x (Page 34)

How to configure a tag with word by word access for Allen Bradley E/IP PLC5 or SLC50x (Page 35)

How to configure a text tag for Allen Bradley E/IP PLC5 or SLC50x (Page 36)

How to configure a connection for the "Allen Bradley - Ethernet IP" channel (Page 23)

2.4.4.2 Addressing

Addressing

A tag is uniquely referenced in WinCC by means of an address in the controller. The address must correspond with the tag name in the PLC. The tag address is defined by a string with a length of up to 128 characters.

Using characters for addressing

Valid characters for tag addressing:

- Letters (a to z, A to Z)
- Numbers (0 to 9)
- Underscore (_)

The tag address consists of tag name and other character strings used to specify the tag in the PLC.

Tag name properties:

- The tag name may begin but not end with an underscore character.
- Strings with successive underscore and space characters are invalid.
- The address may not exceed a length of 128 characters.

Note

The characters reserved for tag addressing may not be used in program/tag names or at any other address instance.

The reserved characters are listed below:

Reserved character	Function
.	Element delimiter
:	Definition of a program tag
,	Delimiter for addressing multi-dimensional arrays
/	Reserved for bit addressing.
[ ]	Addressing of array elements or arrays

Controller and program tags

The "Allen-Bradley E/IP ControlLogix" allows addressing of PLC tags (global project tags) and/or program tags (global program tags). Program tags are declared via the program names in the controller and the actual tag names. Controller variables are addressed by their names.

Note

Addressing errors

Addressing errors are generated when the tag name and data type are inconsistent.

The tag name defined in the address field in WinCC must correspond with the tag name in the controller. The data type of tags in WinCC and in the controller must correspond.

Note

You cannot address module-specific Tags, such as data at input and output modules, directly. Use an Alias tag in the controller instead.

Example: Local:3:O.data cannot be addressed in WinCC

If, for Local:3:O in the controller, the alias "MyOut" is defined, you can address with WinCC via MyOut.Data.

See also

Addressing syntax (Page 29)

Addressing Types (Page 30)

Examples for Addressing (Page 31)

2.4.4.3 Addressing syntax

Notation of addresses

The following tables define the possibilities for writing individual addressing.

Table 2-1 Access to arrays, basic data types and structure elements

Data types	Type	Address
Basic data types	PLC tag	Tag name
	Program tag	Programname:tagname
Arrays	PLC tag	Array tag
	Program tag	Program name: array tag
Bits	PLC tag	Tagname/bitnumber
	Program tag	Programname:tagname/bitnumber
Structure elements	PLC tag	Structure tag. Structure element
	Program tag	Program name: structure tag. structure element

Note

Bit addressing with the data types Bool, Real and String is not permitted and will cause an addressing fault.

Description of the syntax

Syntax description:

```
(Programname:)tagname([x(,y)(,z)]){.tagname([x(,y)(,z)])}(/
bitnumber)
```

- The "( )" defines an optional, single instance of an expression.
- The "{ }" defines an optional expression with multiple single instances.

The address string length may not exceed 128 characters.

See also

Addressing (Page 28)

2.4.4.4 Addressing Types

Array elements

Elements of one-dimensional, two-dimensional and three-dimensional arrays in the PLC are indexed by setting an index and the corresponding notation in the tag editor. Array addressing starts at element "0", with arrays of all basic types being valid for element addressing. Read/write operations are only carried out at the addressed element, and not for the entire array.

Bits and bit tags

Bit access is allowed to all basic data types with the exception of Bool, Real and String. Bit addressing is also allowed at array/structure elements. Data type Bool is defined in WinCC for addressing bits and bit tags in the basic data types.

One-place bit numbers will be address with "/x" or "/0x" (x = bit number). Bit numbers are defined by up to two digits.

Note

With the "Bool" data type in the data types SInt, Int and DInt, after changing the specified bit the complete tag is then written in the PLC again. In the meantime, no check is made as to whether other bits in the tag have since changed. Therefore, the PLC may have only read access to the specified tag.

Structures

User-defined data types are created by means of structures. These structures group tags of different data types. Structures may consist of basic types, arrays and of other structures. In WinCC, only basic data types are addressed as structure elements and not entire structures.

Structure elements

Structure elements are addressed by means of the name of the structure and of the required structure element. This addressing is separated by point. In addition to basic data types, the structure elements may represent arrays or other structures. Only one-dimensional arrays may be used as a structure element.

Note

The nesting depth of structures is only limited by the maximum length of 128 characters for the address.

See also

Addressing (Page 28)

2.4.4.5 Examples for Addressing

Example of a table for addressing

The following table shows basic addressing variations for control variables. Other addressing variants are possible by means of combination.

Type	Type	Address
General	PLC tag	Tag name
	Program tag	Program:tagname
Array	Access to an element of a 2-dimensional array	Arraytag[Dim1,Dim2]
	Element of a structure array (1-dimensional)	Arraytag[Dim1].structureelement
	Bit in element of a basic type array (2-dimensional)	Arraytag[Dim1,Dim2]/Bit
Structure	Array in structure	Structuretag.arraytag
	Bit in element of an array in a sub-structure	Structuretag.structure2.arraytag [element]/bit

Note

Program tags are addressed by leading the address with the program name derived from the PLC with colon delimiter.

Example: Programname:arraytag[Dim1,Dim2]

Access to array elements

Type	Address
PLC tag	Arraytag[Dim1]
	Arraytag[Dim1,Dim2]
	Arraytag[Dim1,Dim2,Dim3]
Program tag	Programname:arraytag[Dim1]
	Programname:arraytag[Dim1,Dim2]
	Programname:arraytag[Dim1,Dim2,Dim3]

See also

Addressing (Page 28)

2.4.4.6 How to configure a tag for the Allen Bradley E/IP ControlLogix channel unit

Introduction

This section shows how you configure a tag for channel unit "Allen Bradley E/IP ControlLogix" in the automation system (AS) address range.

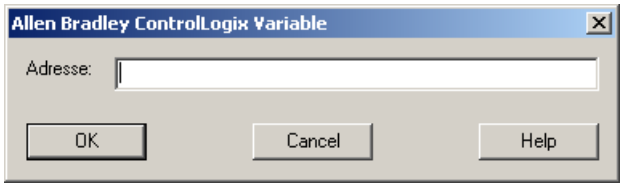
Requirements

- The channel "Allen Bradley - Ethernet IP" must be integrated in the project.
- A connection must be created in the "Allen Bradley E/IP ControlLogix" channel unit.

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column. Enter the name for the tag.
4. Select the desired data type in the "Data type" field.

5. Open the "Allen Bradley ControlLogix Tag" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



6. Enter the address of the tags in AS in the "Address" field.

Note

Addressing errors

Addressing errors are generated when the tag name and data type are inconsistent.

The tag name defined in the address field in WinCC must correspond with the tag name in the controller. The data type of tags in WinCC must correspond with the data types in the controller.

7. Close the dialog by clicking the "OK" button.

See also

Addressing (Page 28)

2.4.4.7 How to configure a tag with bit by bit access for Allen Bradley E/IP PLC5 or SLC50x

Introduction

This section shows you how to configure a tag for bit by bit access for the address area in the automation system (AS).

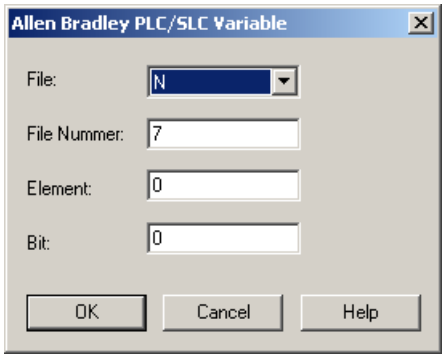
Requirements

- The channel "Allen Bradley - Ethernet IP" must be integrated in the project.
- A connection must be created in the "Allen Bradley E/IP PLC5" or "Allen Bradley E/IP SLC50x" channel unit.

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. Select the desired data type in the "Data type" field.

5. Open the "Allen Bradley PLC/SLC Tag" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



6. Select an address range in the "File" field. Entries N, R, C, T, B, S, I, O, D, A, ST are available.
7. Enter the "File Number" if it is different from the suggested file number.
8. Enter the "Element".
9. Depending on the setting in the "File" field, define the "Bit" to address or select a value for "Bit (octal)" or "Sub".
10. Close the dialog by clicking the "OK" button.

See also

Addressing (Page 28)

2.4.4.8 How to configure a tag with byte by byte access for Allen Bradley E/IP PLC5 or SLC50x

Introduction

This section shows you how to configure a tag for byte by byte access for the address area in the automation system (AS).

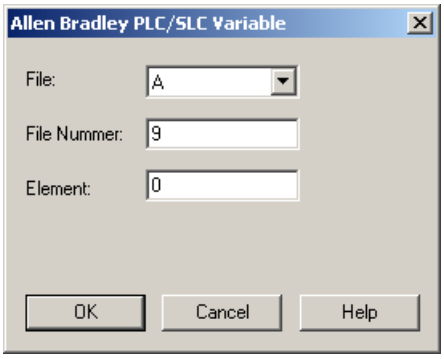
Requirements

- The channel "Allen Bradley - Ethernet IP" must be integrated in the project.
- A connection must be created in the "Allen Bradley E/IP PLC5" or "Allen Bradley E/IP SLC50x" channel unit.

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.

4. Select the desired data type in the "Data type" field.
5. Open the "Allen Bradley PLC/SLC Tag" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



6. Select the A or ST address range in the "File" field.
7. Enter the "File Number" if it is different from the suggested file number.
8. Enter the "Element".
9. Close the dialog by clicking the "OK" button.

See also

Addressing (Page 28)

2.4.4.9 How to configure a tag with word by word access for Allen Bradley E/IP PLC5 or SLC50x

Introduction

This section shows you how to configure a tag for word by word access to the address area in the automation system (AS).

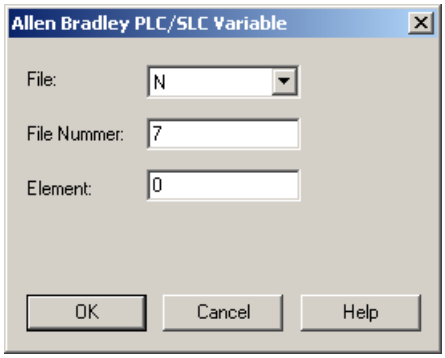
Requirements

- The channel "Allen Bradley - Ethernet IP" must be integrated in the project.
- A connection must be created in the "Allen Bradley E/IP PLC5" or "Allen Bradley E/IP SLC50x" channel unit.

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. Select the desired data type in the "Data type" field.

5. Open the "Allen Bradley PLC/SLC Tag" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



6. Select the address range in the "File" field. Entries N, R, C, T, B, S, I, O, D, A and ST are available.
7. Enter the "File Number" if it is different from the suggested file number. The File number cannot be changed for the setting "S".
8. Enter the "Element".
9. Select one of the values for the "Sub" field if it is displayed. This depends on the setting made in the "File field.
10. Close the dialog by clicking the "OK" button.

See also

Addressing (Page 28)

2.4.4.10 How to configure a text tag for Allen Bradley E/IP PLC5 or SLC50x

Introduction

This section shows you how to configure a tag for word by word access to the address area in the automation system (AS).

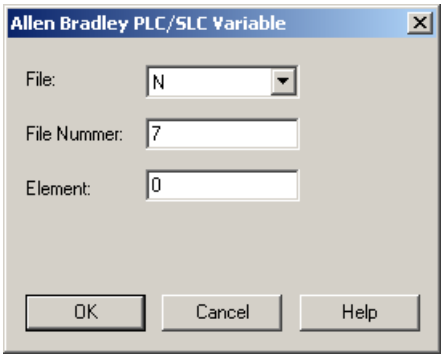
Requirements

- The channel "Allen Bradley - Ethernet IP" must be integrated in the project.
- A connection must be created in the "Allen Bradley E/IP PLC5" or "Allen Bradley E/IP SLC50x" channel unit.

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.

4. Select the desired data type in the "Data type" field.
5. Open the "Allen Bradley PLC/SLC Tag" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



6. Select the A or ST address range in the "File" field.
7. Enter the "File Number" if it is different from the suggested file number. The File number cannot be changed for the setting "S".
8. Enter the "Element".
9. Select one of the values for the "Sub" field if it is displayed. This depends on the setting made in the "File field.
10. Close the dialog by clicking the "OK" button.

See also

Addressing (Page 28)

Mitsubishi Ethernet

3

3.1 WinCC channel "Mitsubishi Ethernet"

Introduction

The "Mitsubishi Ethernet" channel is intended for communication between a WinCC station and Mitsubishi controllers.

The communication takes place via the MELSEC communication protocol (MC protocol).

Channel units

The WinCC channel "Mitsubishi Ethernet" has the following channel units:

- Mitsubishi FX3U series
- Mitsubishi Q series
- Mitsubishi iQ-R series
- Mitsubishi iQ-F series

3.2 Supported data types

Introduction

Define the required tags for a logical connection between WinCC and a connected controller.

The following data types are supported by the "Mitsubishi Ethernet" channel:

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

3.3 Configuring the Channel

3.3.1 Configuring the "Mitsubishi Ethernet" channel

Introduction

WinCC needs a logical connection for communication of WinCC with the automation system (AS).

This section illustrates how to configure the "Mitsubishi Ethernet" channel.

Configuring a channel

The following steps are required for configuring the "Mitsubishi Ethernet" channel:

1. Configuring a connection.
2. Configuring tags.

Note

Connection configuration in the controller

When you use Mitsubishi controllers, you will also have to configure the connections in the controller.

To do this, use the corresponding documentation of the manufacturer.

Online configuration

The "Mitsubishi Ethernet" channel supports the online configuration of tags and connections.

Supported Mitsubishi controllers

You can configure logical connections for the following Mitsubishi controllers:

- MELSEC FX3U series
- MELSEC system Q
- MELSEC system iQ-R
- MELSEC system iQ-F

When you configure the connections and tags, the procedure is identical for both controller families. The configuration differs only in relation to the usable address types of the specific controller family.

Routing of information is only supported by models of the MELSEC system Q and MELSEC system iQ-R series.

Protocol

You can establish a connection to an AS with TCP/IP or UDP/IP as transport protocol.

Configure logical connection for the "Mitsubishi Ethernet" channel:

- Enter IP address and IP port number of the AS
  The IP address consists of four numerical values, separated by dots. The numerical values must be within the range 0 to 255.
- Select UDP or TCP as transport protocol

Note

Timeout Behavior

Interrupted connections are not detected immediately when using the TCP/IP protocol.

The feedback can take longer and is dependent on the operating system.

See also

How to configure a "Mitsubishi FX3U Series" channel unit connection (Page 42)

How to configure a "Mitsubishi Q Series" channel unit connection (Page 44)

How to configure a "Mitsubishi iQ-R series" channel unit connection (Page 45)

Configuring the tags (Page 48)

3.3.2 How to configure a "Mitsubishi FX3U Series" channel unit connection

Introduction

This section shows you how to configure the connection for the "Mitsubishi FX3U Series" channel unit.

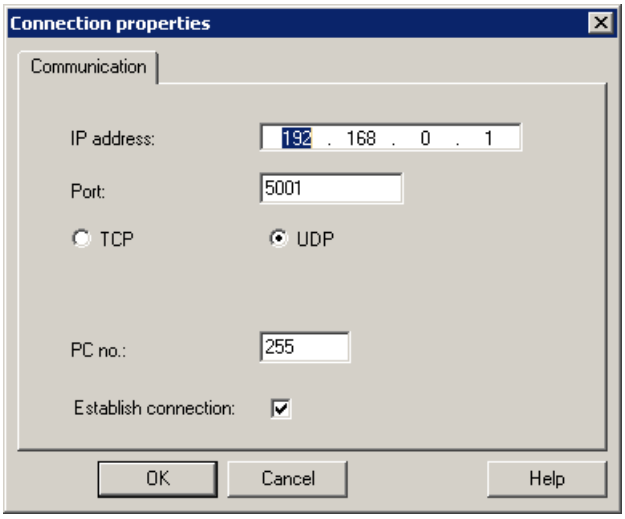
Requirements

- The communication driver for the "Mitsubishi Ethernet" channel is installed and integrated into the project.

Procedure

1. In the navigation area of the tag management, select the channel unit "Mitsubishi FX3U Series" in the tree of the "Mitsubishi Ethernet" communication driver.
2. Select the entry "New Connection" in the shortcut menu of the channel unit.
3. Enter the name of the connection.

4. Select the entry "Connection parameters" from the shortcut menu of the connection.
   The "Connection properties" dialog opens.



5. Enter the IP address of the controller.
6. Enter the port you want to use for the TCP/IP connection.
   The valid range of values is from 0 to 65535.
7. Select the protocol to be used, "TCP" or "UDP".

   Note

   Recommendation: UDP

   We recommend that you use the default protocol "UDP". The timeout behavior is better with this protocol.

   Only use TCP if you are not able to use UDP.

8. Enter the PC number.
   If you do not want to enter a PC number, you must enter the value 255 or 0.
9. To establish the connection, select "Establish connection".
10. In each case, close the dialog with "OK".

See also

Configuring the "Mitsubishi Ethernet" channel (Page 41)

How to configure a tag (Page 50)

3.3.3 How to configure a "Mitsubishi Q Series" channel unit connection

Introduction

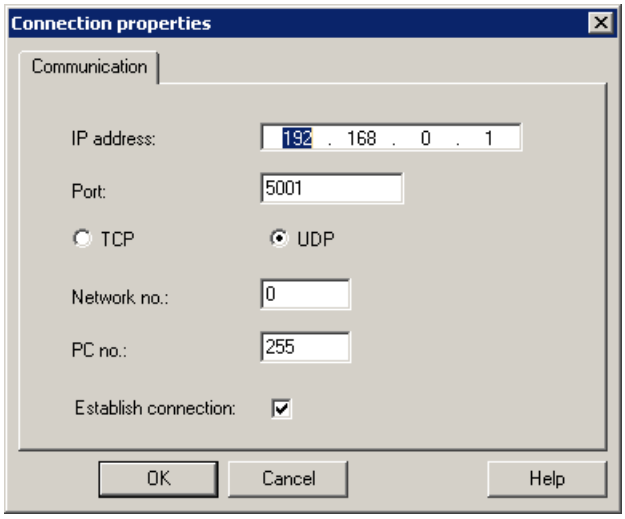
This section shows you how to configure the connection for the "Mitsubishi Q Series" channel unit.

Requirements

- The communication driver for the "Mitsubishi Ethernet" channel is installed and integrated into the project.

Procedure

1. In the navigation area of the tag management, select the channel unit "Mitsubishi Q series" in the tree of the "Mitsubishi Ethernet" communication driver.
2. Select the entry "New Connection" in the shortcut menu of the channel unit.
3. Enter the name of the connection.
4. Select the entry "Connection parameters" from the shortcut menu of the connection.
   The "Connection properties" dialog opens.



5. Enter the IP address of the controller.
6. Enter the port you want to use for the TCP/IP connection.
   The valid range of values is from 0 to 65535.

7. Select the protocol to be used, "TCP" or "UDP".

   Note

   Recommendation: UDP

   We recommend that you use the default protocol "UDP". The timeout behavior is better with this protocol.

   Only use TCP if you are not able to use UDP.

8. Enter the network number.
   The default setting is 0.
9. Enter the PC number.
   If you do not want to enter a PC number, you must enter the value 255 or 0.
10. To establish the connection, select "Establish connection".
11. Close each dialog box by clicking "OK."

See also

Configuring the "Mitsubishi Ethernet" channel (Page 41)

How to configure a tag (Page 50)

3.3.4 How to configure a "Mitsubishi iQ-R series" channel unit connection

Introduction

This section shows you how to configure a connection for the "Mitsubishi iQ-R series" channel unit.

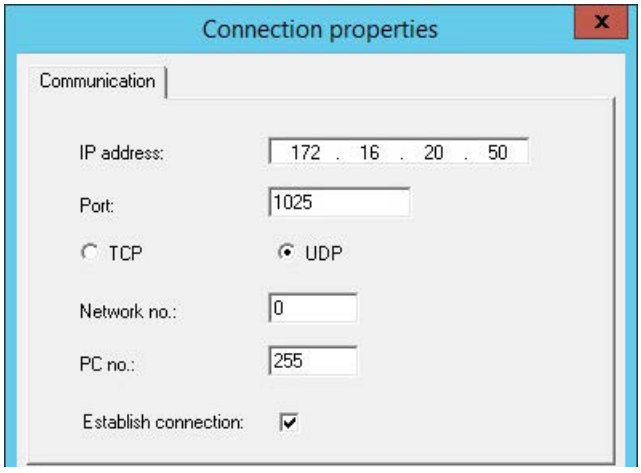
Requirements

- The communication driver for the "Mitsubishi Ethernet" channel is installed and integrated into the project.

Procedure

1. In the navigation area of the tag management, select the channel unit "Mitsubishi iQ-R series" in the tree of the "Mitsubishi Ethernet" communication driver.
2. Select the entry "New Connection" in the shortcut menu of the channel unit.
3. Enter the name of the connection.

4. Select the entry "Connection parameters" from the shortcut menu of the connection.
   The "Connection properties" dialog opens.



5. Enter the IP address of the controller.
6. Enter the port you want to use for the TCP/IP connection.
   The valid range of values is from 0 to 65535. Port 1025 is set by default.
7. Select the protocol to be used, "UDP" or "TCP".

   Note

   Recommendation: UDP

   We recommend that you use the default protocol "UDP". The timeout behavior is better with this protocol.

   Only use TCP if you are not able to use UDP.

8. Enter the network number.
   The default setting is 0.
9. Enter the PC number.
   If you do not want to enter a PC number, you must enter the value 255 or 0.
10. To establish the connection, select "Establish connection".
11. Close each dialog box by clicking "OK."

See also

Configuring the "Mitsubishi Ethernet" channel (Page 41)

How to configure a tag (Page 50)

3.3.5 How to configure a connection of the "Mitsubishi iQ-F series" channel unit

Introduction

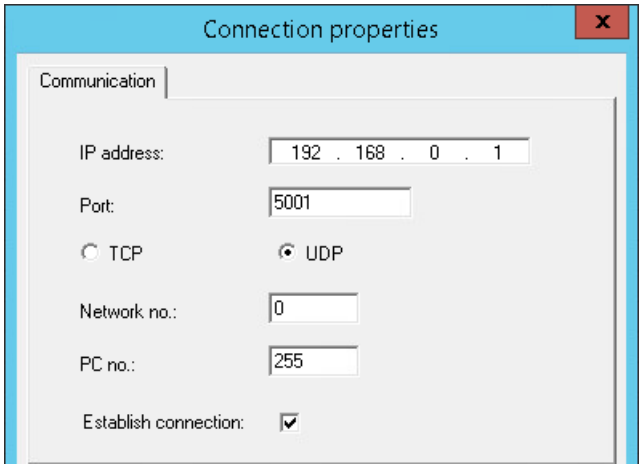
This section shows you how to configure a connection for the "Mitsubishi iQ-F series" channel unit.

Requirements

- The communication driver for the "Mitsubishi Ethernet" channel is installed and integrated into the project.

Procedure

1. In the navigation area of the tag management, select the channel unit "Mitsubishi iQ-F Series" in the tree of the "Mitsubishi Ethernet" communication driver.
2. Select the entry "New Connection" in the shortcut menu of the channel unit.
3. Enter the name of the connection.
4. Select the entry "Connection parameters" from the shortcut menu of the connection. The "Connection properties" dialog opens.



5. Enter the IP address of the controller.
6. Enter the port you want to use for the TCP/IP connection.
   The valid range of values is from 0 to 65535. Port 1025 is set by default.
7. Select the protocol to be used, "UDP" or "TCP".

Note

Recommendation: UDP

We recommend that you use the default protocol "UDP". The timeout behavior is better with this protocol.

Only use TCP if you are not able to use UDP.

8. Enter the network number.
   The default setting is 0.
9. Enter the PC number.
   If you do not want to enter a PC number, you must enter the value 255 or 0.
10. To establish the connection, select "Establish connection".
11. Close each dialog box by clicking "OK."

See also

Configuring the "Mitsubishi Ethernet" channel (Page 41)

How to configure a tag (Page 50)

3.3.6 Configuring the tags

3.3.6.1 Configuring the tags

Introduction

Tags of different data types can be created in WinCC for a connection between WinCC and the automation system via the "Mitsubishi Ethernet" channel.

The following sections describe how to configure the tags.

Address types, address ranges and data types

The table lists the address types, address ranges and data types that can be used when configuring tags and structured tags.

In addition, the automation system (AS) supported by the respective address type is specified:

- FX: MELSEC FX3U series
- Q: MELSEC system Q
  Channel unit "":
- iQR: MELSEC system iQ-R
- iQF: MELSEC system iQ-F

Note

WinCC data type depends on address type

If you want to use a specific WinCC data type, you have to consider the selected address type; for example, address type "D" does not support the WinCC data type "Binary".

Mitsubishi Q series

As of V8.0, WinCC supports extended memory addressing for up to 4 184 063 addresses (4 MB) via ZR register.

Address type	Code	Display	Data type	AS that are supported
Relays				
Link relay	B	Hex.	Bit	Q, iQR, iQF
Input relay (direct)	DX	Hex.	Bit	Q, iQR
Output relay (direct)	DY	Hex.	Bit	Q, iQR
Latch relay	L	Decimal	Bit	Q, iQR, iQF
Special link relay	SB	Hex.	Bit	Q, iQR, iQF
Edge relay	V	Decimal	Bit	Q, iQR
Input relay	X	Hex. (FX: Octal)	Bit	FX, Q, iQR, iQF
Output relay	Y	Hex. (FX: Octal)	Bit	FX, Q, iQR, iQF
Flags				
Error flag (Annunciator)	F	Decimal	Bit	Q, iQR, iQF
Flag (Internal relay)	M	Decimal	Bit	FX, Q, iQR, iQF
Step flag (Step relay)	S	Decimal	Bit	Q, iQF
Diagnostic flag (Special relay)	SM	Decimal	Bit	Q, iQR, iQF
Registers				
Data register	D	Decimal	Word	FX, Q, iQR, iQF
Extension register (File register)	R	Decimal	16-bit	FX, Q, iQF
Extension register	R	Decimal	Word	iQR
Refresh data register	RD	Decimal	Word	iQR
Diagnostic register (Special register)	SD	Decimal	Word	Q, iQR, iQF
Special link register	SW	Hex.	Word	Q, iQR, iQF
Link register	W	Hex.	Word	Q, iQR, iQF
Index register	Z	Decimal	Word	Q, iQR, iQF
Long index register	LZ	Decimal	Double word	iQF
File register (Serial Number Access)	ZR	Hex.	Word	Q, iQR
Counters				
Counter / Coil	CC	Decimal	Bit	Q, iQR, iQF
Counter / current value	CN	Decimal	Word	FX, Q, iQR, iQF
Counter / Contact	CS	Decimal	Bit	FX, Q, iQR, iQF
Long counter (coil)	LCC	Decimal	Bit	iQR, iQF

Address type	Code	Display	Data type	AS that are supported
Long counter (current value)	LCN	Decimal	Double word	iQR, iQF
Long counter (contact)	LCS	Decimal	Bit	iQR, iQF
Timers				
Long retentive timer (current value)	LSTN	Decimal	Double word	iQR
Long timer (Current Value)	LTN	Decimal	Double word	iQR
Retentive timer (coil)	SC	Decimal	Bit	Q, iQF
Retentive timer (current value)	SN	Decimal	Word	Q, iQF
Retentive timer (contact)	SS	Decimal	Bit	Q, iQF
Retentive timer (coil)	STC	Decimal	Bit	iQR
Retentive timer (current value)	STN	Decimal	Word	iQR
Retentive timer (contact)	STS	Decimal	Bit	iQR
Timer (OUT coil)	TC	Decimal	Bit	Q, iQR, iQF
Timer (current value)	TN	Decimal	Word	FX, Q, iQR, iQF
Timer (contact)	TS	Decimal	Bit	FX, Q, iQR, iQF

See also

How to configure a tag (Page 50)

3.3.6.2 How to configure a tag

Introduction

This section shows you how to configure a tag access for the address range in the automation system (AS).

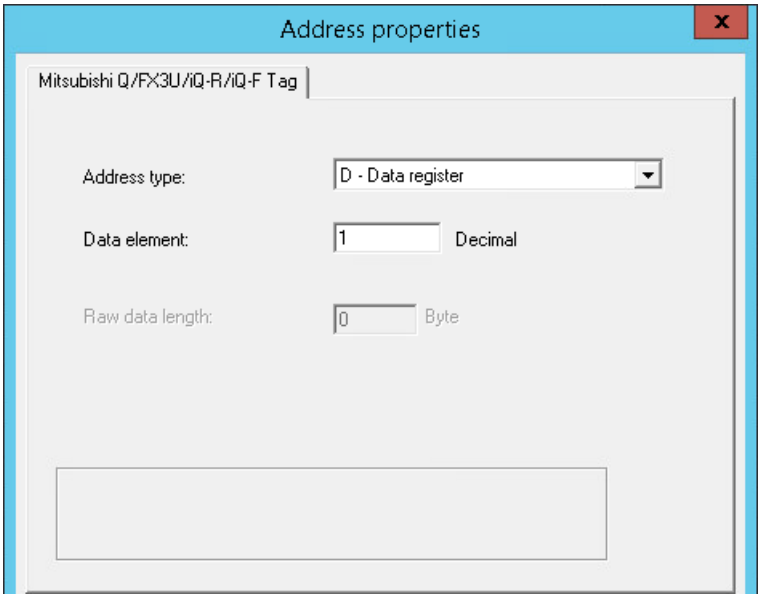
Requirements

- The "Mitsubishi Ethernet" channel is integrated in the project.
- A connection is created in one of the channel units:
  - Mitsubishi FX3U series
  - Mitsubishi Q series
  - Mitsubishi iQ-R series
  - Mitsubishi iQ-F series

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.

3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. Select the desired data type in the "Data type" field.
5. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then the ... button.



6. Set the address type.
7. Enter the corresponding data element.
   The value depends on the configuration of the controller.
8. If necessary, enter the raw data length.
9. Close the "Address properties" dialog by clicking "OK".
   The address of the tag is displayed in the "Address" field of the "Tag properties" dialog.
   The address is adjusted to the data format of the AS.

See also

How to configure a "Mitsubishi FX3U Series" channel unit connection (Page 42)

How to configure a "Mitsubishi Q Series" channel unit connection (Page 44)

How to configure a "Mitsubishi iQ-R series" channel unit connection (Page 45)

Configuring the tags (Page 48)

Modbus TCPIP

4

4.1 "Modbus TCP/IP" channel

Introduction

The "Modbus TCPIP" channel is for communication between a WinCC station and PLCs that support Modbus via Ethernet. The communication is handled with the Modbus TCP/IP protocol.

Channel units

The "Modbus TCPIP" channel comes with the "Modbus TCP/IP Unit #1" channel unit.

See also

Supported Data Types (Page 54)

Configuring the "Modbus TCPIP" Channel (Page 55)

Configuring the tags (Page 57)

4.2 Supported Data Types

Introduction

Define the required tags for a logical connection with a connected controller. The following data types are supported by the "Modbus TCPIP" channel:

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

4.3 Configuring the Channel

4.3.1 Configuring the "Modbus TCPIP" Channel

Introduction

WinCC needs a logical connection for communication of WinCC with the automation system (AS). This section describes the communication with the "Modbus TCP/IP Unit #1" channel unit. All connection-specific parameters are defined during the setup.

If using TCP/IP, you must specify the IP address of the AS for the logic link. The IP address consists of four numerical values, separated by dots. The numerical values must be within the range of 0-255.

Note

Timeout Behavior

Interrupted connections are not detected immediately when using the TCP/IP protocol. The check-back message can take up to a minute.

Enabled Communication Methods with Modbus TCPIP

The following types of communication have been system-tested and approved:

- Point-to-point connection
- Multiple point connection of the WinCC station with an optional amount of controllers.

Note

Integrating the WinCC station via a bridge in a Modbus network is not possible because the WinCC station works as a Modbus Master.

Online configuration

The Online configuration is not supported.

See also

How to configure a connection (Page 56)

Configuring the tags (Page 57)

4.3.2 How to configure a connection

Introduction

The following steps are required for configuring the channel "Modbus TCPIP":

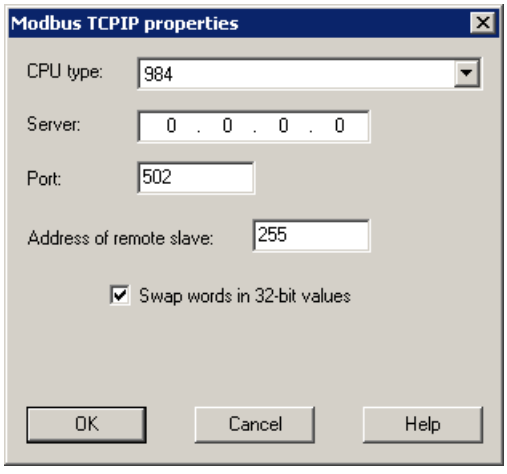
1. Configuring a connection
2. Configuring tags

Requirements

- The communication driver for channel "Modbus TCPIP" is installed and integrated into the project.

Procedure

1. In the navigation area of the tag management, select the channel unit "Modbus TCPIP Unit#1" in the tree of the "Modbus TCP/IP" communication driver.
2. Select the entry "New Connection" in the shortcut menu of the channel unit.
3. Enter the name of the connection.
4. Select the entry "Connection parameters" from the shortcut menu of the connection. The "Modbus TCPIP properties" dialog opens.



5. Select the connected Modicon controller under "CPU Type". The following CPUs are available for selection:
   - 984
     Use this CPU type for the CPU 984 (except for CPU 984A, 984B and 984X).
   - Modicon Compact, Modicon Quantum, Modicon Momentum
   - Modicon Premium, Modicon Micro
6. Enter the IP address of the controller in the "Server" field.
7. Enter the port to be used for the TCP/IP connection in the "Port" field. The default port for Modbus TCP/IP connections is 502.

8. If you are using a bridge, enter the address of the remote controller in the "Address of remote slave" field.
   If no bridge is used, you must enter the default value 255 or 0 as the address.

   Note

   Integrating the WinCC station via a bridge in a Modbus network is not possible because the WinCC station works as a Modbus Master.

9. To swap the order of the 16-bit register in 32-bit process values, select "Swap words in 32-bit values".
10. Close the dialog by clicking "OK".

4.3.3 Configuring the tags

4.3.3.1 Configuring the tags

Introduction

For a connection between WinCC and the automation system (AS) via channel "Modbus TCPIP", tags of different data types can be created in WinCC.

The following sections describe how to configure the tags. The addressing of the data range in the AS and the data type of the WinCC tags are different.

Tag Updating with the Modbus TCP/IP protocol

If the tags are retrieved simultaneously in a picture from a controller, the Modbus TCP/IP channel attempts to optimize the update. This can only be accomplished under the following conditions however:

- The tags are in the same address range.
- The tags are as close to one another as possible within the address range.

If you do not observe these recommendations, it can lead to noticeable differences in the picture refresh with large amounts of tags. The acquisition cycles may not be maintained under certain circumstances.

The best performance for the connection is achieved if you observe the following rules when configuring the tags:

- Update of maximum 2000 tags simultaneously.
- Combine the tags in the least possible space, best in only one address range.

Data Types and Address Ranges in the Controller

The table lists the data types and address ranges that can be used when configuring tags and structured tags.

Name	Area with CPU Premium/Micro	Area with CPU 984, Compact, Quantum, Momentum	data type
Coil (discrete output)	%M [1]	0x	Bit
Discrete input	(%I) – not realized by Premium/Micro	1x	Bit
Input register	(%IW) – not realized by Premium/Micro	3x	Bit, +/- Int, Int
Holding register (output)	%MW	4x	Bit [2], +/- Int, Int, +/- Double, Double, Float, ASCII
Extended memory (Only available with the "Quantum/Momentum" CPU)	--	6x	Bit [2], +/- Int, Int, +/- Double, Double, Float, ASCII

[1] Due to a system characteristics of the external controller the last x bits on the end of the address area cannot be accessed.

[2] In the case of write accesses note:
With the "bit" data type in the "4x", "6x" and "%MW" areas, after changing the specified bit the entire word is written back to the controller. There is no check to determine whether any other bits in the word have changed. As a result, the controller only has read access to the specified word.

The standard bit counting method (16 LSB - 1 MSB) used with controllers of the 984, Compact, Quantum and Momentum series will only be used for these CPUs in the "Tags" editor for the data type "bit". Bit positions have the following allocations:

	Left byte								Right byte							
Counting with tags	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

When entering bit numbers in other locations in WinCC, the bit allocation of WinCC applies (0 LSB - 15 MSB):

How the bit positions are counted	Left byte								Right byte							
In the WinCC you configure:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This bit counting method also applies to Modicon Premium and Modicon Micro controllers

Format for "Signed"

The placeholder "+/-" stands for the data types "Signed Int" and "Signed Double".

See also

How to Configure a Tag with Bit by Bit Access (Page 59)
How to Configure a Tag with Word by Word Access (Page 60)
How to Configure a Text Tag (Page 61)

4.3.3.2 How to Configure a Tag with Bit by Bit Access

Introduction

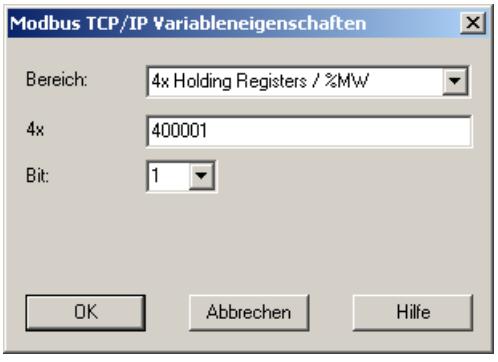
This section shows you how to configure a tag for bit by bit access for the address area in the automation system (AS).

Requirements

- The channel "Modbus TCPIP" must be integrated in the project.
- A connection must be created in the "Modbus TCP/IP Unit #1" channel unit.

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. Set the "Binary tag" data type in the "Data Type" field.
5. Open the "Modbus TCP/IP Tag Properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



6. Enter the address of the tags in the respective address field, "4x" for instance. The value depends on the configuration of the controller.
7. Enter the address of the bit in the "Bit" field if necessary. Whether an entry is possible depends on your selection in the "Area from" field.

8. Select a value for "File" if you have set the value "6x Extended Memory" in the "Area" field.
9. Close the dialog by clicking "OK".

Note

After closing the "Modbus TCP/IP Tag Properties" dialog, the internal address of the tags in the controller is shown in field "Address" of the "Tag Properties" dialog. This address can differ from the entered address because it is adapted to the AS data format.

See also

How to Configure a Text Tag (Page 61)

4.3.3.3 How to Configure a Tag with Word by Word Access

Introduction

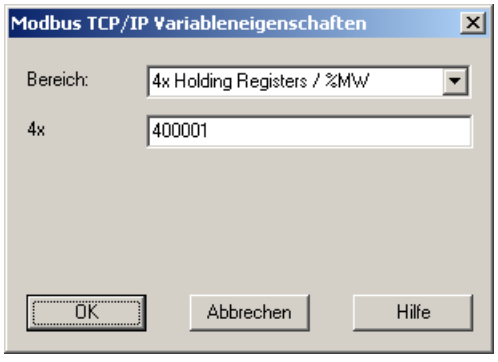
This section shows you how to configure a tag for word by word access to the address area in the automation system (AS).

Requirements

- The channel "Modbus TCPIP" must be integrated in the project.
- A connection must be created in the "Modbus TCP/IP Unit #1" channel unit.

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. Set the data type to "Unsigned 16-bit value" in the "Data Type" field.
5. Open the "Modbus TCP/IP tag properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



6. Enter the address of the tags in the respective address field, "4x" for instance. The value depends on the configuration of the controller.
7. Close both of the dialogs by clicking the "OK" button.

Note

After closing the "Modbus TCP/IP Tag Properties" dialog, the internal address of the tags in the controller is shown in field "Address" of the "Tag Properties" dialog. This address can differ from the entered address because it is adapted to the AS data format.

4.3.3.4 How to Configure a Text Tag

Introduction

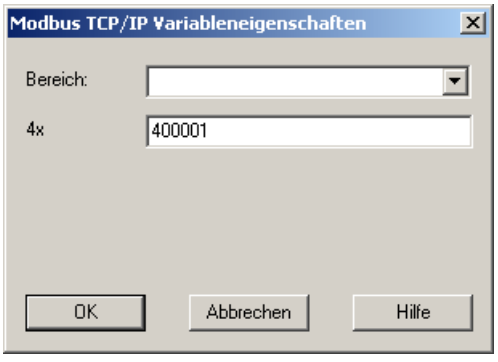
This section shows you how to configure a tag for word by word access to the address area in the automation system (AS).

Requirements

- The channel "Modbus TCPIP" must be integrated in the project.
- A connection must be created in the "Modbus TCP/IP Unit #1" channel unit.

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. In the "Data Type" field, set "Text tag 8-bit character set" as the data type.
5. Open the "Modbus TCP/IP tag properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



6. Enter the address of the tags in address field "4x". The value depends on the configuration of the controller.
7. Close both of the dialogs by clicking the "OK" button.

Omron Ethernet-IP

5

5.1 WinCC channel "Omron Ethernet IP"

Introduction

The "Omron Ethernet IP" channel is used for communication between a WinCC station and Omron controllers.

Channel units

The WinCC channel "Omron Ethernet IP" has the following channel units:

- Omron Ethernet-IP CJ1 Series
- Omron Ethernet-IP CJ2 Series
- Omron Ethernet-IP CS1 Series

5.2 Supported data types

Introduction

Define the required tags for a logical connection between WinCC and a connected controller.

The following data types are supported by the "Omron Ethernet IP" channel:

- Binary Tag
- Signed 8-bit value
- Unsigned 8-bit value
- Signed 16-bit value
- Unsigned 16-bit value
- Signed 32-bit value
- Unsigned 32-bit value
- Floating-point number 32-bit IEEE 754
- Floating-point number 64-bit IEEE 754
- Text tag 8-bit character set
- Text tag 16-bit character set
- Raw data type
- Date/time

5.3 Configuring the channel

5.3.1 Configuration of the "Omron Ethernet IP" channel

Introduction

WinCC needs a logical connection for communication of WinCC with the automation system (AS).

This section shows how the "Omron Ethernet IP" channel is configured.

Configuring a channel

The following steps are required for configuring the "Omron Ethernet IP" channel:

1. Configuring a connection.
2. Configuring tags.

Note

Connection configuration in the controller

When using Omron controllers, you must also configure the connections in the controller.

To do this, use the corresponding documentation of the manufacturer.

Online configuration

The "Omron Ethernet IP" channel supports the online configuration of tags and connections.

Supported Omron controllers

You can configure logical connections for the following Omron controllers:

- CJ1M
- CJ2H, CJ2M
- CS1G, CS1H

When you configure the connections and tags, the procedure is identical for all controller families. The configuration differs only in relation to the usable address types of the specific controller family.

The number of Ethernet connections supported by Omron CS and CJ controllers depends on the model and version.

Examples:

Series	Model	Supported Ethernet connections
Omron CJ	CJ1W-ETN21	4
	CJ2H/CJ2M/CJ2N	8
	CJ2M-CPU3# and CJ2H-CPU6#	4
Omron CS	CS1W-ETN21	4
	CS1W-ETN22	8

Note

Note that these specifications are only general examples and do not take into account every model within the series.

Protocol

You can establish a connection to an AS with TCP/IP as the transport protocol.

Configuring the logical connection for the "Omron Ethernet-IP" channel:

- Enter the IP address.
  The IP address consists of four numerical values, separated by dots. The numerical values must be within the range 0 to 255.

Note

Timeout Behavior

Interrupted connections are not detected immediately when using the TCP/IP protocol.

The feedback can take longer and is dependent on the operating system.

5.3.2 How to configure a connection of the "Omron Ethernet-IP CJ1 Series" channel unit

Introduction

This section describes how to configure a connection for the "Omron Ethernet-IP CJ1 Series" channel unit.

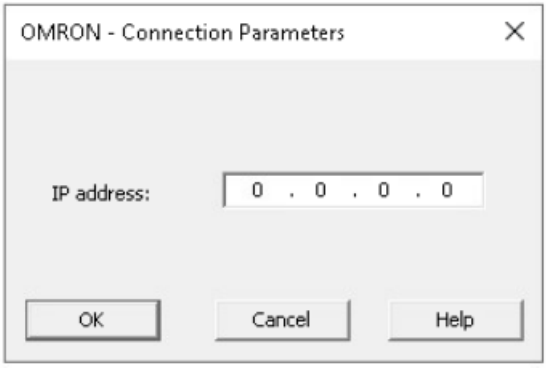
The configuration is carried out identically for the three available channel units.

Requirements

- The communication driver for the "Omron Ethernet-IP" channel is installed and integrated into the project.

Procedure

1. In the navigation area of the Tag Management, select the channel unit "Omron Ethernet-IP CJ1 Series" in the tree of the "Omron Ethernet-IP" communication driver.
2. Select the entry "New Connection" in the shortcut menu of the channel unit.
3. Enter the name of the connection.
4. Select the entry "Connection parameters" from the shortcut menu of the connection. The "Connection Parameters" dialog opens.



5. Enter the IP address of the controller.
6. Close the dialog with "OK".

5.3.3 Configuring the tags

5.3.3.1 Configuring the tags

Introduction

When WinCC is coupled with the automation system (AS) via the "Omron Ethernet-IP" channel, tags of different data types can be created in WinCC.

The following sections describe how to configure the tags.

Extended data storage range

The EM area is divided into 13 benches (0 to C), which contain 32,768 words each.

The EM area addresses range from E0_00000 to EC_32767. This data area is used for general data storage.

Data types and operands

This table lists the data types that can be used to configure tags:

Data type	Operands	Length
Bool	I/O, HR, AR, DM, EM, T, C, TCF, CCF	1 byte
DInt	I/O, HR, AR, DM, EM, T, C	4 bytes
DWord	I/O, HR, AR, DM, EM, T, C	4 bytes
Int	I/O, HR, AR, DM, EM, T, C	2 bytes
LInt	I/O, HR, AR, DM, EM	8 bytes
LReal	I/O, HR, AR, DM, EM	8 bytes
LWord	I/O, HR, AR, DM, EM	8 bytes
Real	I/O, HR, AR, DM, EM	4 bytes
String	I/O, HR, AR, DM, EM	1 to 80 characters
UDInt	I/O, HR, AR, DM, EM, T, C	4 bytes
UDIntBCD	I/O, HR, AR, DM, EM, T, C	4 bytes
UInt	I/O, HR, AR, DM, EM, T, C	2 bytes
UIntBCD	I/O, HR, AR, DM, EM, T, C	2 bytes
ULInt	I/O, HR, AR, DM, EM	8 bytes
ULIntBCD	I/O, HR, AR, DM, EM	8 bytes
Word	I/O, HR, AR, DM, EM, T, C	2 bytes

Note

- The array data type is supported for all of the above data types, except Bool and String.
- The PLC CJ1 supports all operands except EM.

See also

Address ranges for Omron Ethernet/IP (Page 70)

5.3.3.2 How to configure a tag

Introduction

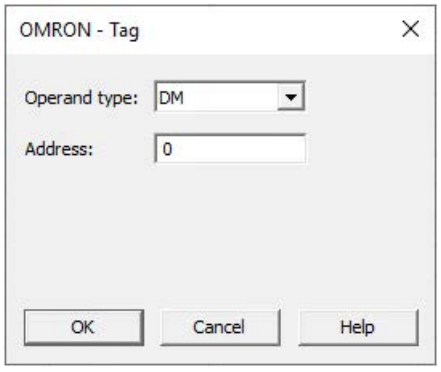
This section shows you how to configure a tag access for the address range in the automation system (AS).

Requirements

- The "Omron Ethernet IP" channel in integrated into the project.
- A connection is created in one of the channel units:
  - Omron Ethernet-IP CJ1 Series
  - Omron Ethernet-IP CJ2 Series
  - Omron Ethernet-IP CS1 Series

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. Select the desired data type in the "Data type" field.
5. Open the "OMRON - Tag" dialog.
   For this purpose, click in the "Address" field and then the ... button.



6. Set the operand type.
7. Enter the corresponding address.
   The value depends on the configuration of the controller.
8. Close the "OMRON - Tag" dialog by clicking "OK".
   The address of the tags is displayed in the "Address" field of the "Tags" dialog.
   The address is adjusted to the data format of the AS.

See also

Address ranges for Omron Ethernet/IP (Page 70)

5.3.3.3 Address ranges for Omron Ethernet/IP

Address ranges for CJ2

Address areas	Bool	DInt	DWord	Int	LInt	LReal	LWord	Real
I/O	I/O 0.0 - I/O 6143.15	I/O 0 - I/O 6142	I/O 0 - I/O 6142	I/O 0 - I/O 6143	I/O 0 - I/O 6140	I/O 0 - I/O 6140	I/O 0 - I/O 6140	I/O 0 - I/O 6142
HR	HR 0.0 - HR 511.15	HR 0 - HR 510	HR 0 - HR 510	HR 0 - HR 511	HR 0 - HR 508	HR 0 - HR 508	HR 0 - HR 508	HR 0 - HR 510
AR	AR 0.0 - AR 1471.15	AR 0 - AR 1470	AR 0 - AR 1470	AR 0 - AR 1471	AR 0 - AR 1468	AR 0 - AR 1468	AR 0 - AR 1468	AR 0 - AR 1470
DM	DM 0.0 - DM 32767.15	DM 0 - DM 32766	DM 0 - DM 32766	DM 0 - DM 32767	DM 0 - DM 32764	DM 0 - DM 32764	DM 0 - DM 32764	DM 0 - DM 32766
EM	EM 0.0:0 - EM 32767.15:25	EM 0:0 - EM 32766:25	EM 0:0 - EM 32766:25	EM 0:0 - EM 32767:25	EM 0:0 - EM 32764:25	EM 0:0 - EM 32764:25	EM 0:0 - EM 32764:25	EM 0:0 - EM 32766:25
T	T 0 - T 4095	T 0 - T 4094	T 0 - T 4094	T 0 - T 4095	NA	NA	NA	NA
C	C 0 - C 4095	C 0 - C 4094	C 0 - C 4094	C 0 - C 4095	NA	NA	NA	NA
TCF	TCF 0 - TCF 4095	NA	NA	NA	NA	NA	NA	NA
CCF	CCF 0 - CCF 4095	NA	NA	NA	NA	NA	NA	NA

Address areas	String	UDInt	UDIntBCD	UInt	UIntBCD	ULInt	ULIntBCD	Word
I/O	I/O 0 - I/O 6143	I/O 0 - I/O 6142	I/O 0 - I/O 6142	I/O 0 - I/O 6143	I/O 0 - I/O 6143	I/O 0 - I/O 6140	I/O 0 - I/O 6140	I/O 0 - I/O 6143
HR	HR 0 - HR 511	HR 0 - HR 510	HR 0 - HR 510	HR 0 - HR 511	HR 0 - HR 511	HR 0 - HR 508	HR 0 - HR 508	HR 0 - HR 511

AR	AR 0 - AR 1471	AR 0 - AR 1470	AR 0 - AR 1470	AR 0 - AR 1471	AR 0 - AR 1471	AR 0 - AR 1468	AR 0 - AR 1468	AR 0 - AR 1471
DM	DM 0 - DM 32767	DM 0 - DM 32766	DM 0 - DM 32766	DM 0 - DM 32767	DM 0 - DM 32767	DM 0 - DM 32764	DM 0 - DM 32764	DM 0 - DM 32767
EM	EM 0:0 - EM 32767:25	EM 0:0 - EM 32766:25	EM 0:0 - EM 32766:25	EM 0:0 - EM 32767:25	EM 0:0 - EM 32767:25	EM 0:0 - EM 32764:25	EM 0:0 - EM 32764:25	EM 0:0 - EM 32767:25
T	NA	T 0 - T 4094	T 0 - T 4094	T 0 - T 4095	T 0 - T 4095	NA	NA	T 0 - T 4095
C	NA	C 0 - C 4094	C 0 - C 4094	C 0 - C 4095	C 0 - C 4095	NA	NA	C 0 - C 4095
TCF	NA	NA	NA	NA	NA	NA	NA	NA
CCF	NA	NA	NA	NA	NA	NA	NA	NA

Address ranges for CS1

Address areas	**Bool**	**DInt**	**DWord**	**Int**	**LInt**	**LReal**	**LWord**	**Real**
I/O	I/O 0.0 - I/O 6143.15	I/O 0 - I/O 6142	I/O 0 - I/O 6142	I/O 0 - I/O 6143	I/O 0 - I/O 6140	I/O 0 - I/O 6140	I/O 0 - I/O 6140	I/O 0 - I/O 6142
HR	HR 0.0 - HR 511.15	HR 0 - HR 510	HR 0 - HR 510	HR 0 - HR 511	HR 0 - HR 508	HR 0 - HR 508	HR 0 - HR 508	HR 0 - HR 510
AR	AR 0.0 - AR 959.15	AR 0 - AR 958	AR 0 - AR 958	AR 0 - AR 959	AR 0 - AR 956	AR 0 - AR 956	AR 0 - AR 956	AR 0 - AR 958
DM	DM 0.0 - DM 32767.15	DM 0 - DM 32766	DM 0 - DM 32766	DM 0 - DM 32767	DM 0 - DM 32764	DM 0 - DM 32764	DM 0 - DM 32764	DM 0 - DM 32766
EM	EM 0.0:0 - EM 32767.15:25	EM 0:0 - EM 32766:25	EM 0:0 - EM 32766:25	EM 0:0 - EM 32767:25	EM 0:0 - EM 32764:25	EM 0:0 - EM 32764:25	EM 0:0 - EM 32764:25	EM 0:0 - EM 32766:25
T	T 0 - T 4095	T 0 - T 4094	T 0 - T 4094	T 0 - T 4095	NA	NA	NA	NA
C	C 0 - C 4095	C 0 - C 4094	C 0 - C 4094	C 0 - C 4095	NA	NA	NA	NA

TCF	TCF 0 - TCF 4095	NA	NA	NA	NA	NA	NA	NA
CCF	CCF 0 - CCF 4095	NA	NA	NA	NA	NA	NA	NA

Address areas	**String**	**UDInt**	**UDIntBCD**	**UInt**	**UIntBCD**	**ULInt**	**ULIntBCD**	**Word**
I/O	I/O 0 - I/O 6143	I/O 0 - I/O 6142	I/O 0 - I/O 6142	I/O 0 - I/O 6143	I/O 0 - I/O 6143	I/O 0 - I/O 6140	I/O 0 - I/O 6140	I/O 0 - I/O 6143
HR	HR 0 - HR 511	HR 0 - HR 510	HR 0 - HR 510	HR 0 - HR 511	HR 0 - HR 511	HR 0 - HR 508	HR 0 - HR 508	HR 0 - HR 511
AR	AR 0 - AR 959	AR 0 - AR 958	AR 0 - AR 958	AR 0 - AR 959	AR 0 - AR 959	AR 0 - AR 956	AR 0 - AR 956	AR 0 - AR 959
DM	DM 0 - DM 32767	DM 0 - DM 32766	DM 0 - DM 32766	DM 0 - DM 32767	DM 0 - DM 32767	DM 0 - DM 32764	DM 0 - DM 32764	DM 0 - DM 32767
EM	EM 0:0 - EM 32767:25	EM 0:0 - EM 32766:25	EM 0:0 - EM 32766:25	EM 0:0 - EM 32767:25	EM 0:0 - EM 32767:25	EM 0:0 - EM 32764:25	EM 0:0 - EM 32764:25	EM 0:0 - EM 32767:25
T	NA	T 0 - T 4094	T 0 - T 4094	T 0 - T 4095	T 0 - T 4095	NA	NA	T 0 - T 4095
C	NA	C 0 - C 4094	C 0 - C 4094	C 0 - C 4095	C 0 - C 4095	NA	NA	C 0 - C 4095
TCF	NA	NA	NA	NA	NA	NA	NA	NA
CCF	NA	NA	NA	NA	NA	NA	NA	NA

Address ranges for CJ1

Address areas	**Bool**	**DInt**	**DWord**	**Int**	**LInt**	**LReal**	**LWord**	**Real**
I/O	I/O 0.0 - I/O 6143.15	I/O 0 - I/O 6142	I/O 0 - I/O 6142	I/O 0 - I/O 6143	I/O 0 - I/O 6140	I/O 0 - I/O 6140	I/O 0 - I/O 6140	I/O 0 - I/O 6142
HR	HR 0.0 - HR 511.15	HR 0 - HR 510	HR 0 - HR 510	HR 0 - HR 511	HR 0 - HR 508	HR 0 - HR 508	HR 0 - HR 508	HR 0 - HR 510

AR	AR 0.0 - AR 959.15	AR 0 - AR 958	AR 0 - AR 958	AR 0 - AR 959	AR 0 - AR 956	AR 0 - AR 956	AR 0 - AR 956	AR 0 - AR 958
DM	DM 0.0 - DM 32767.15	DM 0 - DM 32766	DM 0 - DM 32766	DM 0 - DM 32767	DM 0 - DM 32764	DM 0 - DM 32764	DM 0 - DM 32764	DM 0 - DM 32766
T	T 0 - T 4095	T 0 - T 4094	T 0 - T 4094	T 0 - T 4095	NA	NA	NA	NA
C	C 0 - C 4095	C 0 - C 4094	C 0 - C 4094	C 0 - C 4095	NA	NA	NA	NA
TCF	TCF 0 - TCF 4095	NA	NA	NA	NA	NA	NA	NA
CCF	CCF 0 - CCF 4095	NA	NA	NA	NA	NA	NA	NA

Address areas	**String**	**UDInt**	**UDIntBCD**	**UInt**	**UIntBCD**	**ULInt**	**ULIntBCD**	**Word**
I/O	I/O 0 - I/O 6143	I/O 0 - I/O 6142	I/O 0 - I/O 6142	I/O 0 - I/O 6143	I/O 0 - I/O 6143	I/O 0 - I/O 6140	I/O 0 - I/O 6140	I/O 0 - I/O 6143
HR	HR 0 - HR 511	HR 0 - HR 510	HR 0 - HR 510	HR 0 - HR 511	HR 0 - HR 511	HR 0 - HR 508	HR 0 - HR 508	HR 0 - HR 511
AR	AR 0 - AR 959	AR 0 - AR 958	AR 0 - AR 958	AR 0 - AR 959	AR 0 - AR 959	AR 0 - AR 956	AR 0 - AR 956	AR 0 - AR 959
DM	DM 0 - DM 32767	DM 0 - DM 32766	DM 0 - DM 32766	DM 0 - DM 32767	DM 0 - DM 32767	DM 0 - DM 32764	DM 0 - DM 32764	DM 0 - DM 32767
T	NA	T 0 - T 4094	T 0 - T 4094	T 0 - T 4095	T 0 - T 4095	NA	NA	T 0 - T 4095
C	NA	C 0 - C 4094	C 0 - C 4094	C 0 - C 4095	C 0 - C 4095	NA	NA	C 0 - C 4095
TCF	NA	NA	NA	NA	NA	NA	NA	NA
CCF	NA	NA	NA	NA	NA	NA	NA	NA

OPC Channel

6

6.1 WinCC OPC Channel

Introduction

WinCC can be used as both an OPC server and as an OPC client. The OPC channel is the OPC client application of WinCC.

The OPC communication driver can be used as OPC DA client, OPC XML client, and OPC UA client. The documentation for the OPC UA client is available under "OPC UA channel".

The following OPC components are installed automatically:

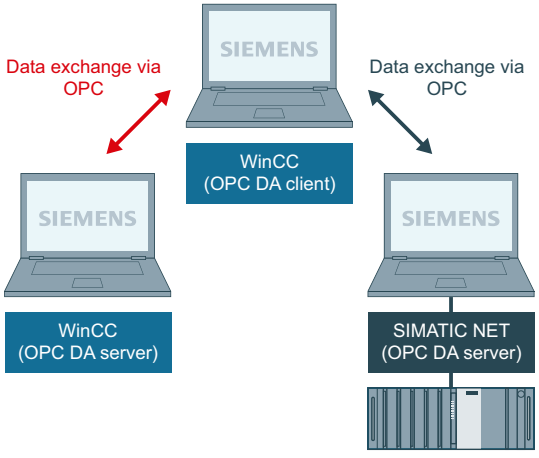
- OPC communication driver
- OPC Item Manager

Possible Applications

WinCC as an OPC DA client

If WinCC is used as an OPC DA client, the OPC channel must be added to the WinCC project. A connection for data exchange is created in the WinCC project of the WinCC OPC DA client; this is used to handle access to the WinCC tags of the OPC DA server.

To simplify the process, the OPC Item Manager is used. A WinCC OPC DA client can access multiple OPC DA servers. This requires that a connection be created for each OPC server. In this way, the WinCC OPC DA client can be used as a central operation and monitoring station.



Note

The WinCC OPC channel establishes connections only to OPC servers which have the status "OPC_STATUS_RUNNING".

Note

"OPC" channel

Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

6.2 OPC Item Manager

Introduction

A connection and a WinCC tag are configured in the WinCC project of the WinCC OPC client to enable access to tags of an OPC server. The OPC Item Manager simplifies this process for you. The OPC Item Manager is automatically installed with WinCC.

Note

"OPC" channel

Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

Requirements

The following requirements must be met in order to use the OPC Item Manager for configuration:

- The OPC server is an OPC DA server.
- A tag is already configured on the OPC server.
- If WinCC is to be used as the OPC server the WinCC project of the WinCC OPC server must be enabled. If this is not the case, the OPC Item Manager cannot access the WinCC OPC server.
- It must be possible to access the computer of the OPC servers via the IP address or HTTP.
- The OPC server must support the browser functionality. If that is not the case, access to the tag of the OPC server must be configured manually.

Note

If you change language in the WinCC Explorer while the OPC Item Manager is open, no tags are displayed when you click the "Browse Server" button. Exit the OPC Item Manager before changing language.

Tasks of the OPC Item Manager

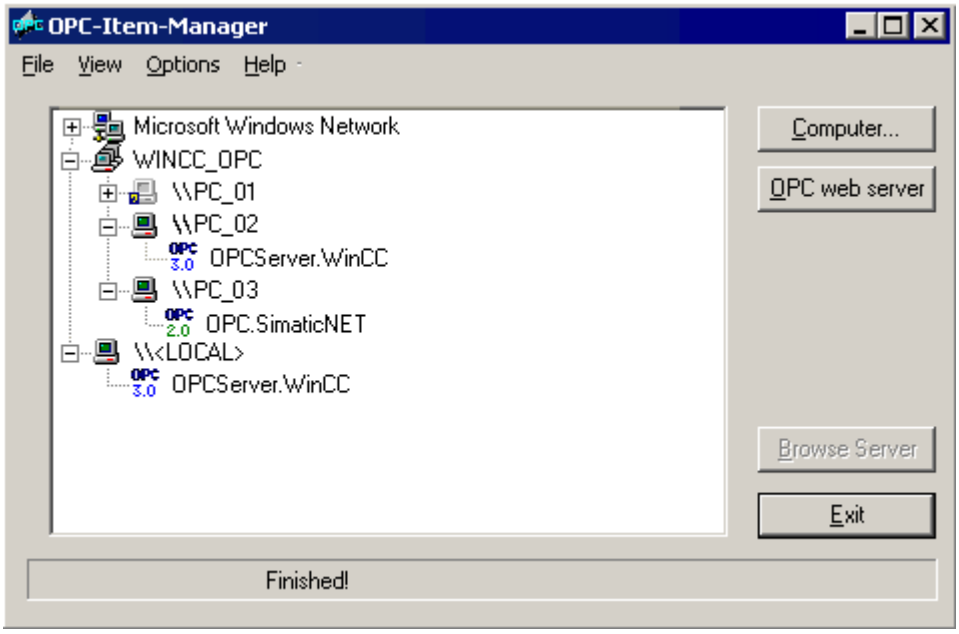
The OPC Item Manager assumes the following tasks:

- Select OPC server
- Creating a connection
- Tag selection
- Adding a tag

Selecting the OPC server

OPC DA server

The OPC Item Manager can be used to determine the name of the OPC DA server in the network. These OPC DA servers can run on the same computer or on different computers in the connected network environment. For further details, refer to "WinCC OPC DA client ".



Icons of the OPC Item Manager	Description
	A networked computer has not yet been searched for installed OPC DA servers.
	The computer was not found in the network or the computer could not be accessed.
	A networked computer has been searched for installed OPC DA servers.
	A networked computer contains the OPC DA server designated with the OPC symbol. The number indicates which OPC DA specification of the WinCC OPC DA client is used.
\\<LOCAL>	Refers to the computer running the OPC Item Manager.

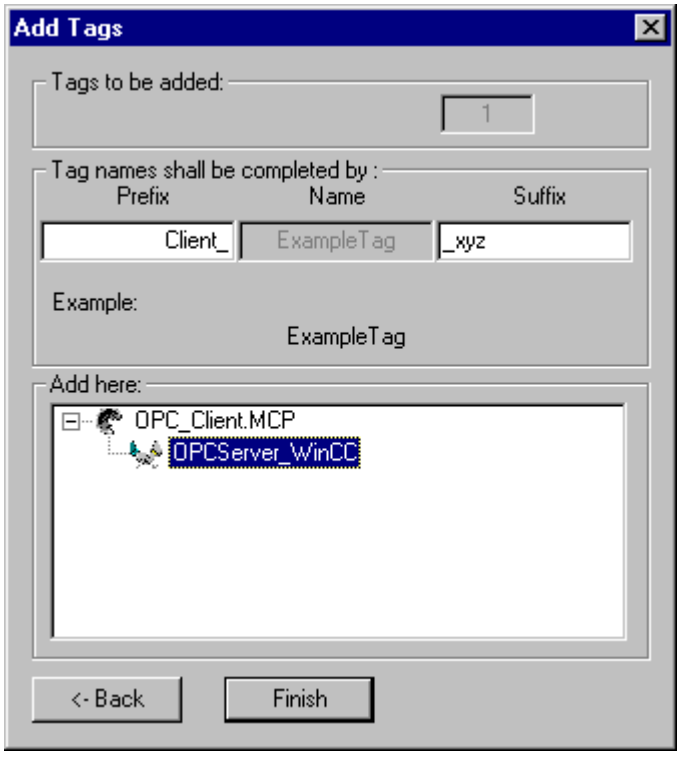
Creating a connection

The OPC Item Manager configures all required settings when creating a connection. If a connection to the OPC server has already been created, this function is not available.

Tag selection

You may use the tag selection dialog to select one or more tags on the OPC server which the WinCC OPC client is to access. Filter criteria can be used to limit the choices in the tag selection dialog.

Adding a tag



The names of the WinCC tags that access the tags of the OPC server can be set in the "Add Tags" dialog.

The WinCC tag name consists of the "prefix", "name" and "suffix". The "Name" field is preconfigured with the "ExampleTag" text. "ExampleTag" stands for the WinCC tag name of the WinCC OPC server.

You can assign a prefix or suffix to distinguish the WinCC tag name on the WinCC OPC client from the WinCC tag name on the WinCC OPC server. When configuring project monitoring, a prefix or suffix must be assigned.

The tag name may be assigned only once in a given WinCC project.

Example

The WinCC tag name on the WinCC OPC DA server is called "OPC_Server_Tag". The "Client_" value is entered in the prefix field and "_xyz" in the suffix field. In the WinCC project of the WinCC OPC DA client, the WinCC tag "Client_OPC_Server_Tag_xyz" is created.

If the tag name on the OPC server contains special characters, they are replaced by an underscore ("_"), because not all special characters occurring in tag names are supported by the OPC Item Manager.

Click "Finish" to add the WinCC tags to the WinCC project of the WinCC OPC DA client. The OPC Item Manager automatically sets the data type, the name and the address parameters for the WinCC tag.

See also

How to Access a WinCC Tag with the OPC Item Manager (Page 83)

6.3 Overview of the Supported WinCC Data Types

The list below shows the data types that are supported by the WinCC OPC DA client and WinCC OPC DA server:

- Binary tag
- Signed 8-bit value
- Unsigned 8-bit value
- Signed 16-bit value
- Unsigned 16-bit value
- Signed 32-bit value
- Unsigned 32-bit value
- Floating-point number 32-bit IEEE 754
- Floating-point number 64-bit IEEE 754
- Text tag, 8-bit character set
- Text tag, 16-bit character set
- Raw data type
- Structure types
- Text reference
- Date/Time

Note

Structure types

For structure types, only the structure elements are supported, not the structure itself. However, the structure can be configured later. For more information, refer to the topic "Using structures on the WinCC OPC DA client."

Text reference

If a text tag is created with the OPC Item Manager, it is assigned a length of 160 characters. This length can be changed to any length.

See also

How to Use Structures on the WinCC OPC DA Client (Page 90)

6.4 WinCC OPC DA Client

6.4.1 Functionality of the WinCC OPC DA Client

Introduction

The OPC channel does not require a separate communication module. The OPC channel is an application which employs the OPC software interface to use an OPC DA server to access process data.

If WinCC is to be used as an OPC DA client, the OPC channel must be added to the WinCC project.

If a communication is established to a WinCC OPC DA server, the values of the WinCC tags are exchanged. To do this, a connection is set up in the WinCC project of the WinCC OPC DA client; it is used to handle access to the WinCC OPC DA server.

For the WinCC OPC DA client to access multiple OPC DA servers, a connection for each of the OPC DA servers must be set up in the WinCC project. For more information about troubleshooting channels and tags, refer to "Troubleshooting".

Note

The WinCC OPC channel establishes connections only to OPC servers which have the status "OPC_STATUS_RUNNING".

Note

"OPC" channel

Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

Connection Monitoring

Three mechanisms are integrated for connection monitoring in the WinCC OPC-DA client. It is thus possible to take the best possible measures in the event of a network error or malfunction of an OPC DA server.

1. If the processing period for a DCOM activation exceeds warning value of 5 seconds, the tag is assigned the value "Addressing Error". If the processing period exceeds the cancellation value of 10 seconds, the connection to the OPC DA server is interrupted. This is displayed in the "Connection Status" dialog of the WinCC Explorers.

The OPC DA specification 3.00 is provided with the "Keep-Alive" feature. If the OPC DA server supports the OPC DA specifications 3.00, this feature is used. The feature causes the OPC DA server to automatically trigger cyclic updating (call OnDataChange) even if the tag values have not changed. If this regular updating is disabled, the WinCC OPC DA client terminates the connection.

The same behavior applies in the case of an OPC DA server which supports the OPC DA specifications 2.05a. In order to check the connection to the OPC DA server, the WinCC OPC DA client requests the status cyclically every 10 seconds. If this regular updating is disabled, the WinCC OPC DA client terminates the connection.

Generally, the WinCC OPC DA client terminates the connection to the OPC DA server when the connection is not capable of functioning. The WinCC OPC DA client attempts to re-establish the connection again, automatically, every 10 seconds.

See also

How to Use Structures on the WinCC OPC DA Client (Page 90)

Accessing a WinCC Tag without the OPC Item Manager (Page 88)

Configuring Access with the OPC Item Manager (Page 84)

Overview of the Supported WinCC Data Types (Page 81)

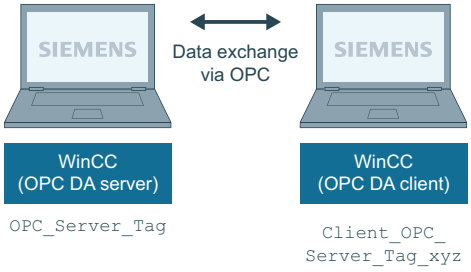
OPC Item Manager (Page 77)

6.4.2 How to Access a WinCC Tag with the OPC Item Manager

6.4.2.1 How to Access a WinCC Tag with the OPC Item Manager

Introduction

When an OPC connection is made between WinCC and WinCC, data exchange occurs using WinCC tags. The WinCC OPC DA client uses an OPC connection to read the WinCC tag "OPC_Server_Tag" on the WinCC OPC DA server. To simplify the process, the OPC Item Manager is used.



Requirements

- Two computers with WinCC projects.
- Both computers must be accessible via their IP addresses.

Configuration Steps

The following configurations are required in the WinCC project of the WinCC OPC DA client:

- Creation of a connection.
- Configuration of the "XMLClient_OPC_Var1_xyz" WinCC tag on the WinCC OPCXML client which accesses the WinCC tag of the WinCC OPC DA server.

See also

Configuring Access with the OPC Item Manager (Page 84)

Configuring the OPC Channel on the WinCC OPC DA Client (Page 84)

6.4.2.2 Configuring the OPC Channel on the WinCC OPC DA Client

Introduction

To use OPC for data exchange, the OPC channel must be set up in the WinCC project.

Procedure

1. Click the "Tag Management" icon in the navigation window of the WinCC Explorer on the WinCC OPC DA client.
2. Select "Add New Driver" from the "Tag Management" shortcut menu. The "Add New Driver" dialog is opened.
3. Select the "OPC.chn" driver and click the "Open" button. The channel is created and the communication driver is displayed in the tag management.

See also

Configuring Access with the OPC Item Manager (Page 84)

6.4.2.3 Configuring Access with the OPC Item Manager

Introduction

This section explains how to use the OPC Item Manager to configure access to the WinCC tag "OPC_Server_Tag" of the WinCC OPC DA server.

Requirements

- 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.
- Add the "OPC" channel to the WinCC project of the WinCC OPC DA client.

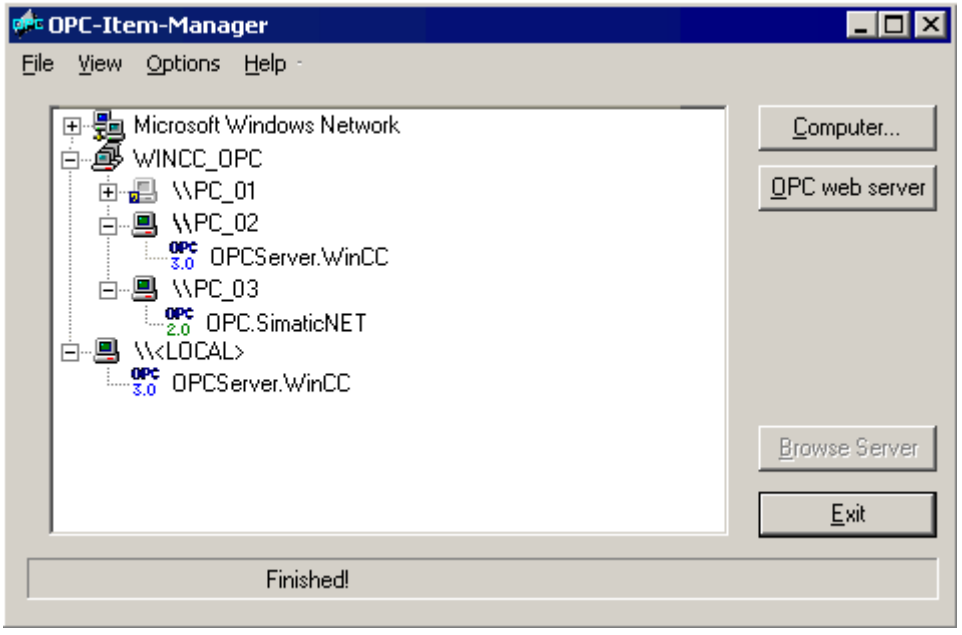
Note

"OPC" channel

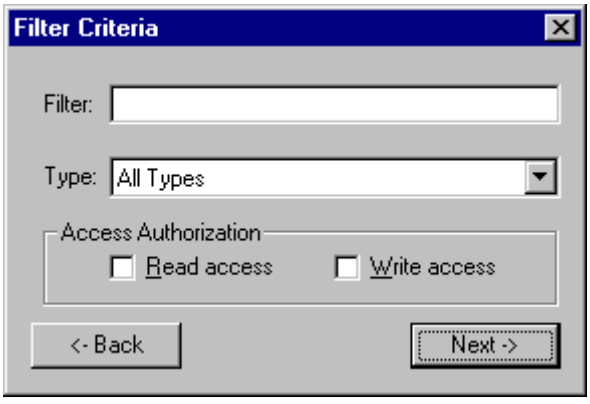
Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

Procedure

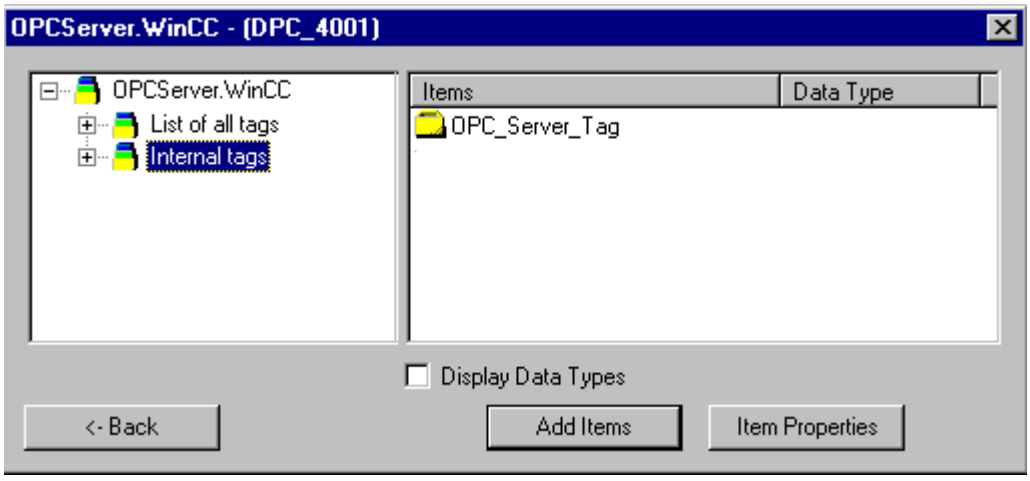
1. In the shortcut menu of the channel unit "OPC Groups(OPCHN Unit#1)" on the WinCC OPC DA client, select "System Parameters". The "OPC Item Manager" opens.



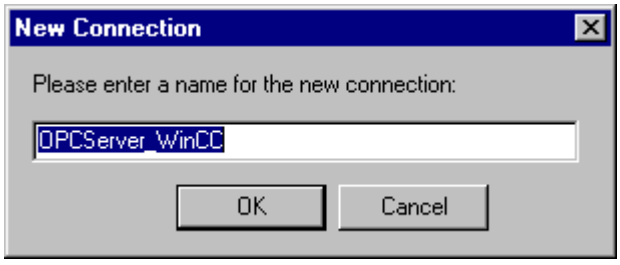
2. Choose the name of the computer to be used as the WinCC OPC DA server from the selection dialog.
   Select "OPCServer.WinCC" from the list displayed.
3. Click the "Browse Server" button.
   The "Filter criteria" dialog is opened.



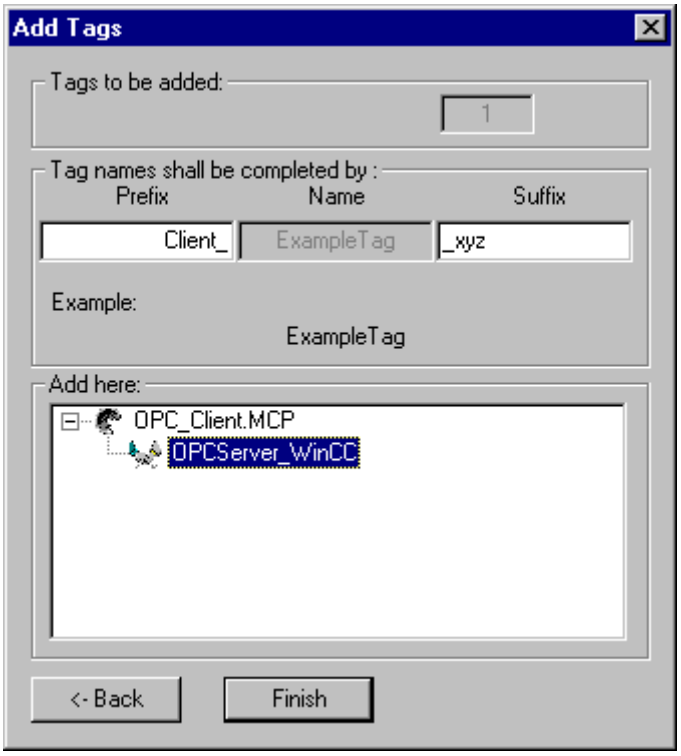
4. Click "Next".
   The "OPCServer.WinCC ..." dialog is opened.



5. Select the WinCC tag "OPC_Server_Tag".
   Click the "Add Items" button.
6. If a connection to the WinCC OPC DA server already exists, continue with step 6.
   If a connection has not been created, a message will be displayed.
   Click on the "Yes" button. The "New Connection" dialog is opened.



7. Enter "OPCServer_WinCC" as the name of the connection. Click "OK".
   The "Add Tags" dialog opens.



8. Enter the text "Client_" in the prefix field and the text "_xyz" in the suffix field.
9. Select connection "OPCServer_WinCC".
   Click "Finish".
10. Click the "Back" button in the "OPCServer.WinCC ..." dialog.
    Click "Exit" to close the OPC Item Manager.

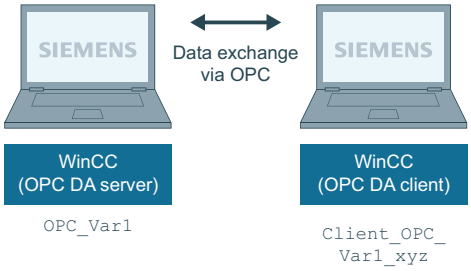
See also

Configuring the OPC Channel on the WinCC OPC DA Client (Page 84)

6.4.3 Accessing a WinCC Tag without the OPC Item Manager

Introduction

OPC servers that do not support browser functionality require access to be configured manually. Configuration of WinCC tags on the WinCC OPC DA client is shown using an example of a WinCC-WinCC OPC connection.



Note

To access a WinCC tag without the OPC Item Manager, the ItemID must be set manually. When addressing WinCC tags, the symbolic computer name (server prefix) can also be specified. The ItemID has the following syntax: Server prefix::WinCC tag. If the WinCC tag of the local WinCC project is addressed, the server prefix is omitted.

The following configurations are required in the WinCC project of the WinCC OPC DA client:

1. Selection of the "OPC_Var1" WinCC tag to be accessed.
2. Creation of a connection.
3. Configuration of the "Client_OPC_Var1_xyz" WinCC tag that accesses the WinCC tag of the WinCC OPC DA server.

Requirements

- Two computers with WinCC projects.
- Both computers must be accessible via their IP addresses.
- Configure an internal tag named "OPC_Var1" with 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.
- Add the OPC channel to the WinCC project of the WinCC OPC DA client.

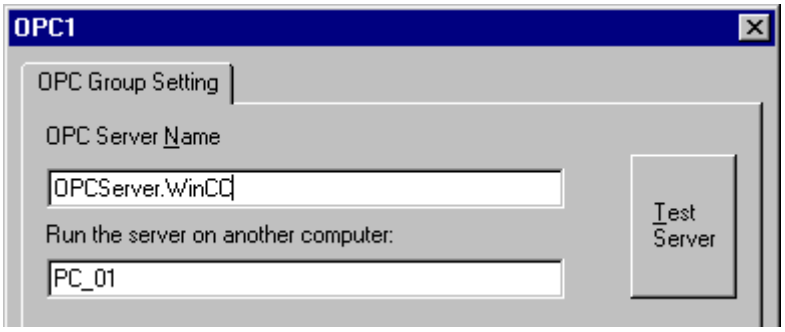
Note

"OPC" channel

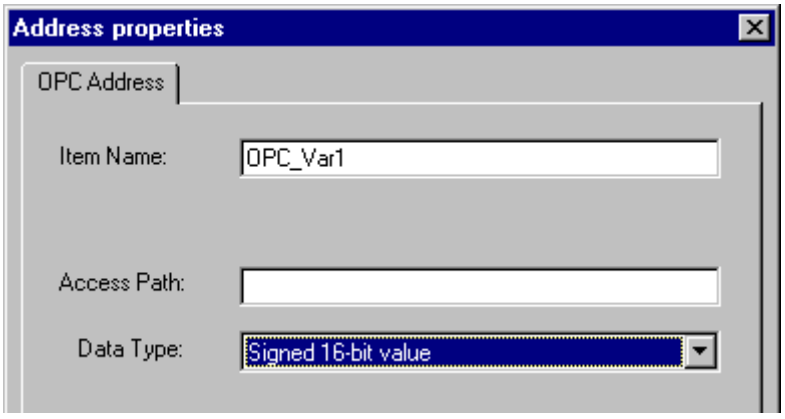
Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

Procedure

1. Select "New Connection" from the shortcut menu of the channel unit "OPC Groups(OPCHN Unit#1)" on the WinCC OPC DA client. The "Connection Properties" dialog is opened. Enter a name for the connection in the corresponding field.
2. Click the "Properties" button. A dialog with the connection name in its title is displayed.



   For connections to WinCC V 6, the entry in the "OPC Server Name" field must be "OPCServer.WinCC".
3. Enter the name of the computer to be used as the OPC DA server in the "Start Server on this Computer" field. Click "Test Server", to check the connection to the WinCC OPC DA server.
4. Select "New Tag" from the shortcut menu of the connection. The "Tag Properties" dialog opens.
5. Enter the name "Client_OPC_Var1_xyz" in the "Tag" field. Set the data type to "signed 16-bit".
6. In the "Tag Properties" dialog, click the "Select" button. The "Address Properties" dialog opens.



   Enter the name of the WinCC tag of the WinCC OPC DA server in the "Item Name" field. Leave the entry in the "Access Path" field unchanged. Set the data type to "signed 16-bit".
7. Click "OK" to close all open dialogs.

6.4.4 Using Structures on a WinCC OPC DA Client

6.4.4.1 How to Use Structures on the WinCC OPC DA Client

Introduction

Structures are used to organize tags and tag types that form a logical unit. This allows them to be referenced using a single logical name.

Structures are not supported by the OPC DA specification. As a result, structures cannot be set up using the OPC Item Manager, only the individual tags in a structure. If you wish to use structures on the WinCC OPC DA client nonetheless, the data structure must be configured subsequently in the WinCC project of the WinCC OPC DA client in order to supply it with the relevant item names of the server tags.

Requirements

- Two computers with WinCC projects.
- Both computers must be accessible via their IP addresses.

Configuration steps

The following configuration steps are necessary to use structures on the WinCC OPC DA client:

- Configuring structures and structure tags on the WinCC OPC DA server
- Using structures on the WinCC OPC DA client in the WinCC project

See also

How to Configure Structures on the WinCC OPC DA Client (Page 92)

Configuring Structures and Structure Tags on the WinCC OPC DA Server (Page 90)

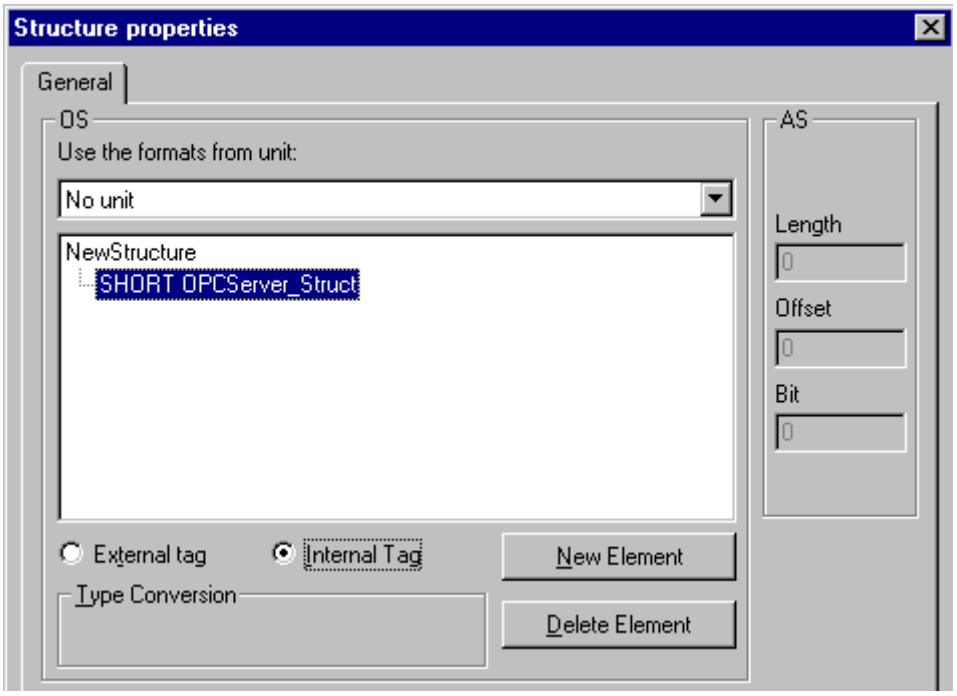
6.4.4.2 Configuring Structures and Structure Tags on the WinCC OPC DA Server

Introduction

In this section a structure and a structure tag is created in the WinCC project of the OPC DA server. This configuration is required for the OPC DA client to access the structure tag.

Procedure

1. Select "New Structure Type" from the structure types shortcut menu on the WinCC OPC DA server. The "Structure Properties" dialog is displayed.
2. Click "New Element" and create the internal tag "OPCServer_Struct" of data type SHORT.



   Click "OK" to close the dialog.
3. In the navigation window, click the plus sign in front of the icon for tag management. Select "New Tag" from the internal tag shortcut menu. Create a WinCC tag named "Var" with this structure type.
4. The data frame of the WinCC Explorer shows the single tag "Var" and the structure tag "Var.OPCServer_Struct".
5. Activate the WinCC project.

See also

How to Configure Structures on the WinCC OPC DA Client (Page 92)

6.4.4.3 How to Configure Structures on the WinCC OPC DA Client

Introduction

Structures are not supported by the OPC DA specification. As a result, structures cannot be set up using the OPC Item Manager. In this section, the structure already present in the WinCC project of the WinCC OPC DA server is configured for the WinCC project of the WinCC OPC DA client. A WinCC tag that accesses the existing structure tag on the WinCC OPC DA server is configured on the WinCC OPC DA client.

Requirements

- Create a structure and a structure tag named "Var.OPCServer_Struct" in the WinCC project of the WinCC OPC DA server.
- Enable the WinCC project of the WinCC OPC DA server.
- Add the OPC channel to the WinCC project of the WinCC OPC DA client.

Note

"OPC" channel

Unicode is not supported for connection names. Make sure that you name all connections in the project in the same language. Open the Control Panel of your computer to set the code page of this language for use in programs that do not support Unicode.

Procedure

1. Select "New Structure Type" from the structure types shortcut menu on the WinCC OPC DA client. The "Structure Properties" dialog is displayed.
2. Click the "New Element" button and set up an external tag. Name the element exactly as it is in the WinCC project of the OPC-DA server. Click "OK" to close the "Structure Properties" dialog.
3. If a connection to the OPC DA server already exists, continue with step 6.
   If no connection has been created, select "New Connection" from the shortcut menu of the channel unit "OPC". The "Connection Properties" dialog is opened. Enter a name for the connection in the corresponding field.
4. Click the "Properties" button. A dialog with the connection name in its title is displayed. For connections to WinCC V 6, the entry in the field "OPC Server Name" must be "OPCServer.WinCC".
5. Enter the name of the computer to be used as the WinCC OPC DA server in the field "Start Server on this Computer". Click "Test Server", to check the connection to the WinCC OPC DA server. Click "OK" to close the dialog.
6. Select "New Tag" from the shortcut menu of the connection. The "Tag Properties" dialog opens. Select the newly created structure type as the data type.

7. In the "Tag Properties" dialog, click the "Select" button. The "Address properties" dialog opens. In the "Item Name" field, enter the name "Var.OPCServer_Struct" for the structure tag of the WinCC OPC DA server. Leave the entry in the "Access Path" field unchanged.
8. Click "OK" to close all open dialogs.

See also

Configuring the OPC Channel on the WinCC OPC DA Client (Page 84)

Configuring Structures and Structure Tags on the WinCC OPC DA Server (Page 90)

6.4.5 Error Handling in the Event of Disturbed OPC DA Communication

6.4.5.1 Error Handling in the Event of Disturbed OPC Communication

Introduction

The procedure for communication testing is independent of how WinCC is used.

WinCC Used as the OPC DA Server

Use the channel diagnostics on the WinCC OPC DA client to determine whether a connection to the OPC DA server can be established. For more information regarding channel problem analysis, refer to "Troubleshooting".

WinCC Used as the OPC DA Client

Use the channel diagnostics on the WinCC OPC DA client to determine whether a connection to the OPC DA server can be established. For more information regarding channel problem analysis, refer to "Troubleshooting".

See also

WinCC is used as the OPC DA client, and the connection is not established. (Page 97)

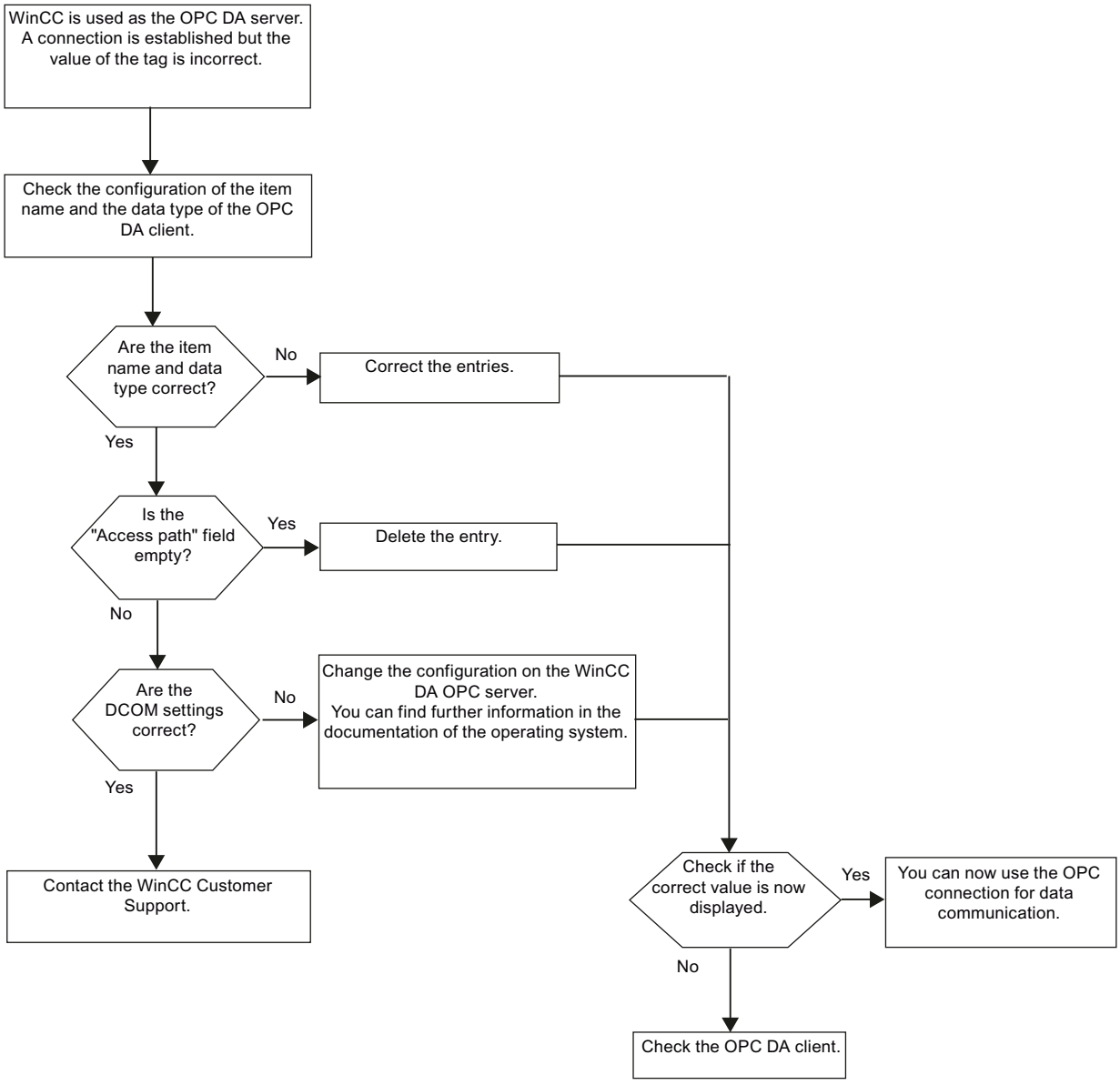
WinCC is used as the OPC DA client, and the connection is established. (Page 96)

WinCC is used as the OPC DA server, and the connection is not established. (Page 95)

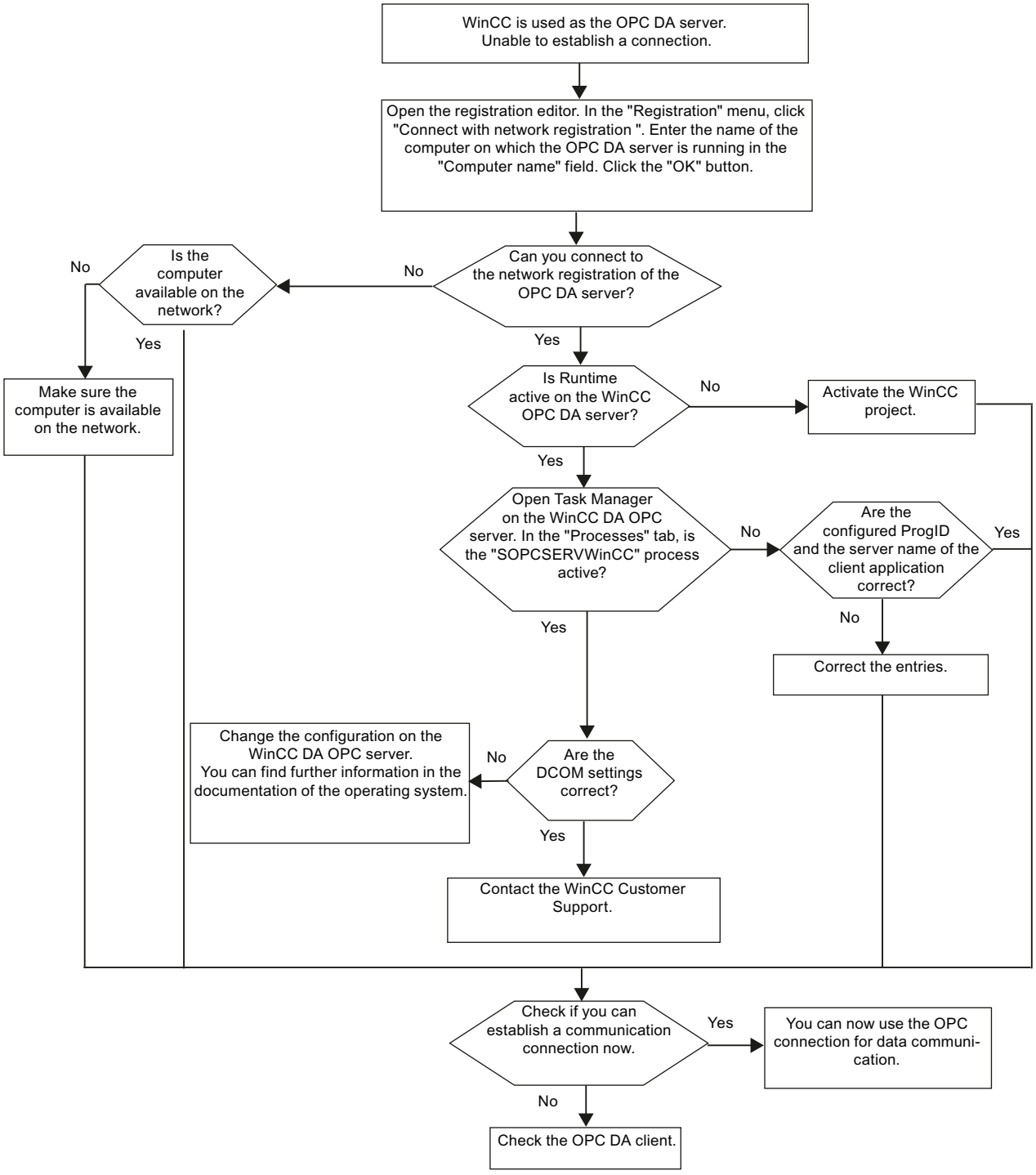
WinCC is used as the OPC DA server, and the connection is established successfully. (Page 94)

6.4.5.2 WinCC as OPC DA Server

WinCC is used as the OPC DA server, and the connection is established successfully.

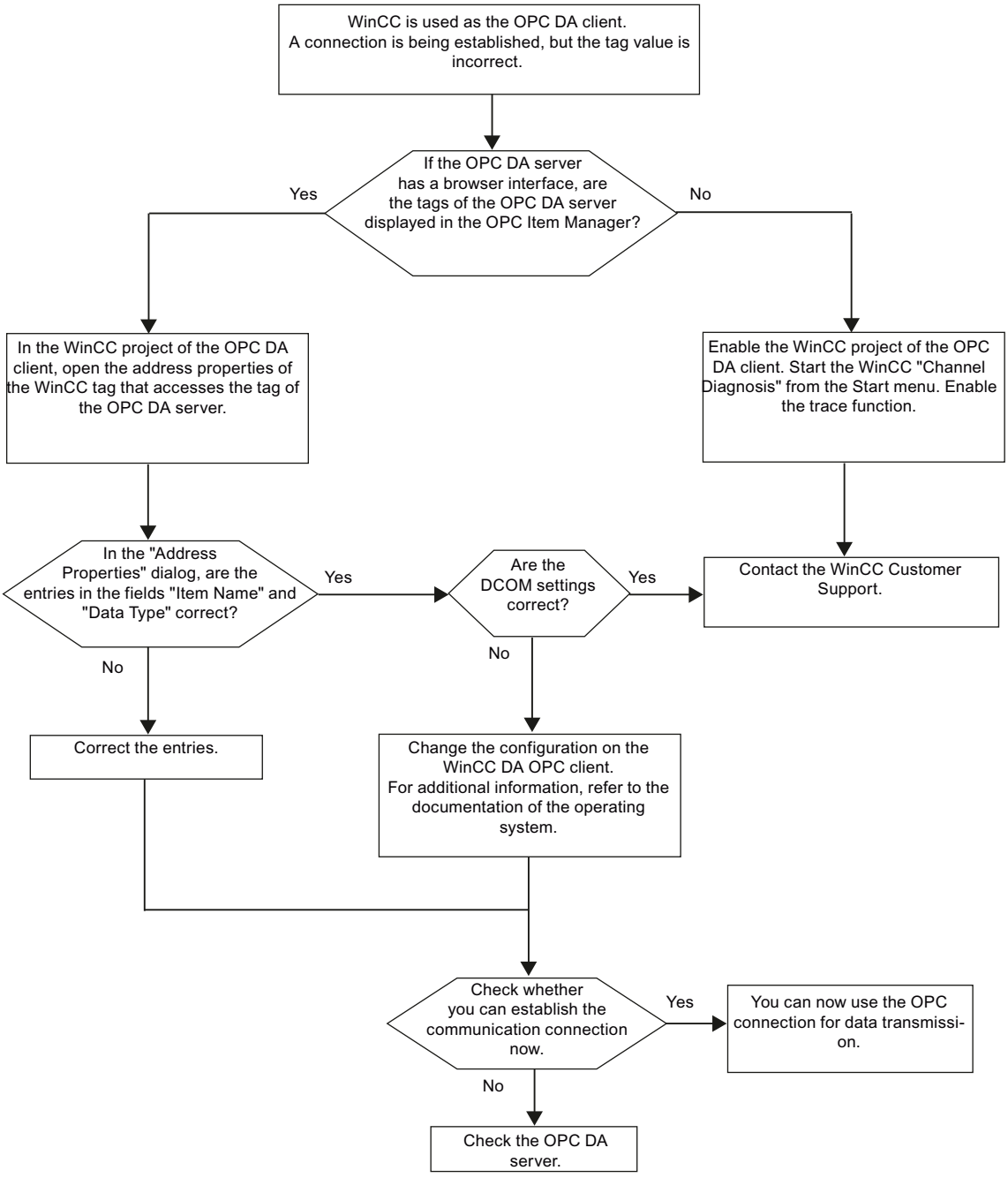


WinCC is used as the OPC DA server, and the connection is not established.

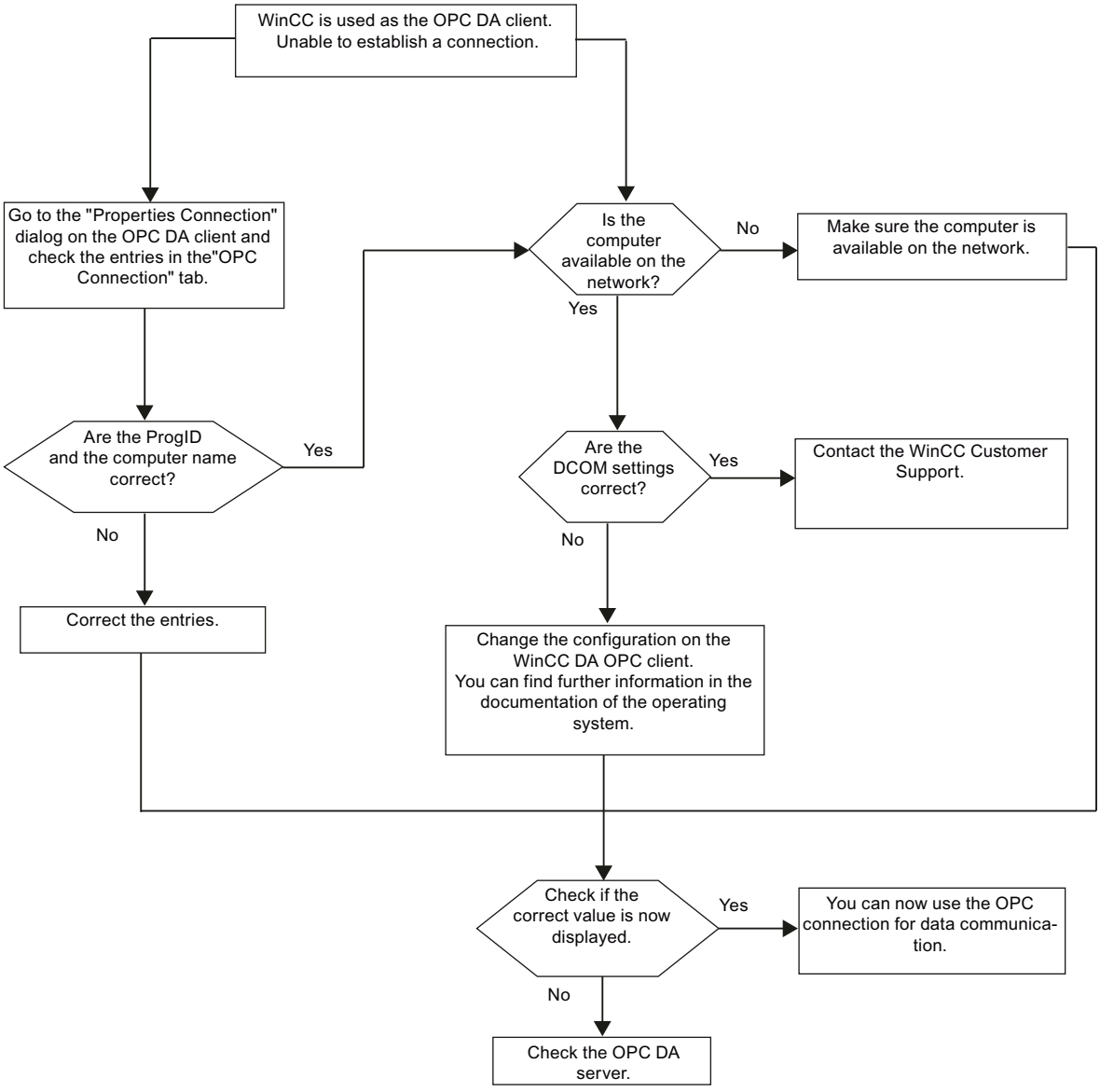


6.4.5.3 WinCC as OPC DA Client

WinCC is used as the OPC DA client, and the connection is established.



WinCC is used as the OPC DA client, and the connection is not established.



OPC UA WinCC Channel

7

7.1 WinCC channel "OPC UA WinCC Channel"

Introduction

WinCC can be used as both an OPC UA server and an OPC UA client. The channel "OPC UA WinCC Channel" is the OPC UA client application of WinCC.

With OPC UA (Unified Architecture), WinCC supports the platform-independent follow-up technology of OPC. You can find details in the OPC UA specification and in the WinCC Information System under "Interfaces > OPC - Open Connectivity > WinCC OPC UA Server". Basic knowledge of OPC UA is required for the configuration.

You can find the documentation on the OPC DA client in the WinCC Information System under "Communication > OPC channel".

Functionality of the WinCC OPC UA client

The WinCC OPC UA client enables data access to any OPC UA server according to the OPC Unified Architecture specification.

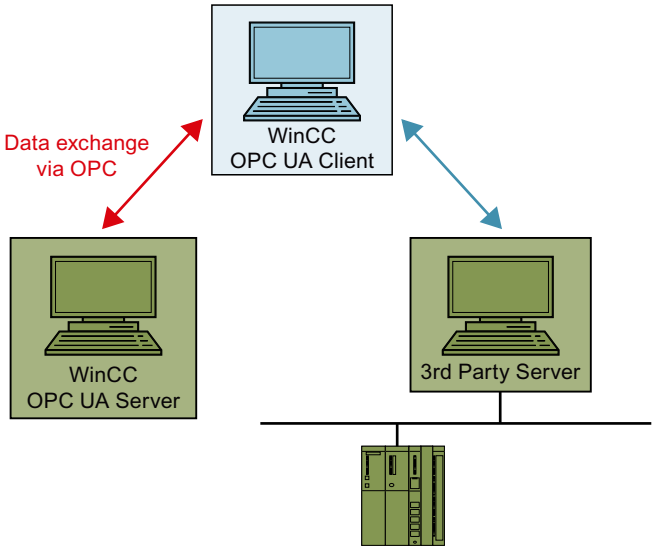
OPC Unified Architecture (OPC UA) provides, for example, additional mechanisms such as authentication and encryption to secure communication between involved partners.

Communication via OPC UA

To use WinCC as WinCC OPC UA client, insert the OPC UA communication driver "OPC UA WinCC Channel" in the WinCC project. You do not need a separate communication module.

A WinCC OPC UA client can access multiple OPC UA servers. You can configure a connection to each OPC UA server for this. You can use the WinCC OPC UA client as central operator control and monitoring station.

You perform the configuration in the tag management of the WinCC Configuration Studio.


Data exchange
via OPC
WinCC
OPC UA Client
WinCC
OPC UA Server
3rd Party Server

7.2 Overview of the supported data types

Introduction

Configure access to the tags of the OPC UA server in the WinCC project of the WinCC OPC UA client for data communication.

To do this, import an OPC UA node as a WinCC tag in the Tag Management.

Supported data types

The WinCC OPC UA client and WinCC OPC UA server support the following WinCC data types.

- Binary tag
- Signed 8-bit value
- Unsigned 8-bit value
- Signed 16-bit value
- Unsigned 16-bit value
- Signed 32-bit value
- Unsigned 32-bit value
- Floating-point number 32-bit IEEE 754
- Floating-point number 64-bit IEEE 754
- Text tag, 8-bit font
- Text tag, 16-bit character set
- Raw data type
- Structure types
- Date/time

7.3 Overview of the supported OPC UA functionalities.

The "OPC UA WinCC Channel" supports the following functionalities of OPC UA:

- Data Access
- Event Access
- Alarms and Conditions
- Methods

Supported OPC UA Services

The following table summarizes the functionalities supported by the OPC-UA-Client:

OPC UA Service Sets	Services	Comment
Discovery Service Set	FindServers GetEndpoints	
Secure Channel Service Session Service Set	All	
View Service Set	Browse BrowseNext	
Attribute Service Set	Read Write	Only available for supported data types
Subscription Service Set	CreateSubscription Publish RePublish DeleteSubscription	
MonitoredItem Service Set	CreateMonitoredItems DeleteMonitoredItems	Available for the attribute "Value" of tags For EventNotifier available upon accessing A & C
Method Service Set	Call	The following methods are supported: • Acknowledge • ConditionRefresh Other methods are only available for supported data types

Supported OPC UA profiles and Conformance Units

The OPC UA server supports the following OPC UA profiles 1.04 with restrictions:

Note

Optional Conformance Units are not supported:

Group	Supported profiles/Conformance Units	Comment
Security	None, Basic128Rsa15, Basic256, Basic256Sha256, Aes128-Sha256-RsaOaep, Aes256-Sha256-RsaPss, User Token – Anonymous Client Facet, User Token – User Name Password Client Facet	None, Basic128Rsa15 and Basic256 have been totally discontinued and are supported for downward compatibility
General	Standard UA Client Profile, Base Client Behaviour Facet, Core Client Facet, AddressSpace Lookup Client Facet, Discovery Client Facet, Multi-Server Client Connection Facet	
Data Access	Attribute Read Client Facet, Attribute Write Client Facet, DataChange Subscriber Client Facet, DataAccess Client Facet	Only attribute "Value"
Event Access	Event Subscriber Client Facet, Base Event Processing Client Facet, Notifier and Source Hierarchy Client Facet	
Alarms & Conditions	A & C Basic Client, A & C Alarm Client , A & C Refresh Client, A & C Acknowledge Client	
Methods	Method Client Facet	Only available for supported data types
Transport	UA-TCP UA-SC UA Binary	

See also

Overview of the supported data types (Page 101)

7.4 Configuring the OPC UA channel

7.4.1 General sequence

Introduction

The OPC UA link between WinCC and an OPC UA server enables the WinCC OPC UA client to access the tags and methods of the OPC UA server via a secure connection. The data of the OPC UA server is mapped to WinCC tags for this purpose.

You perform the configuration in the tag management.

Requirements

- The OPC UA server is active.
- The computers must be connected via TCP.
- The communication must not be blocked by a firewall.
- The port numbers of the OPC UA server must be activated.

Configuration steps

- Inserting an OPC UA WinCC channel in the WinCC project (Page 104)
- Creating a connection to the OPC UA server (Page 105)
- Setting up authentication via certificates. (Page 112)
- Configuring the OPC UA tags (Page 123)
- Using OPC UA alarms in WinCC (Page 135)

7.4.2 Inserting an OPC UA WinCC channel in the WinCC project

Requirement

- A WinCC project is created.
- Tag management is open.

Procedure

1. Open the shortcut menu of tag management in the navigation area.
2. Select "Add new driver > OPC UA WinCC Channel".

Result

The channel OPC UA WinCC Channel is added to tag management.

The channel unit "OPC UA Connections" is created under the channel. Under this channel unit, you configure the connections to one or more OPC UA servers.

7.4.3 Creating a connection to the OPC UA server

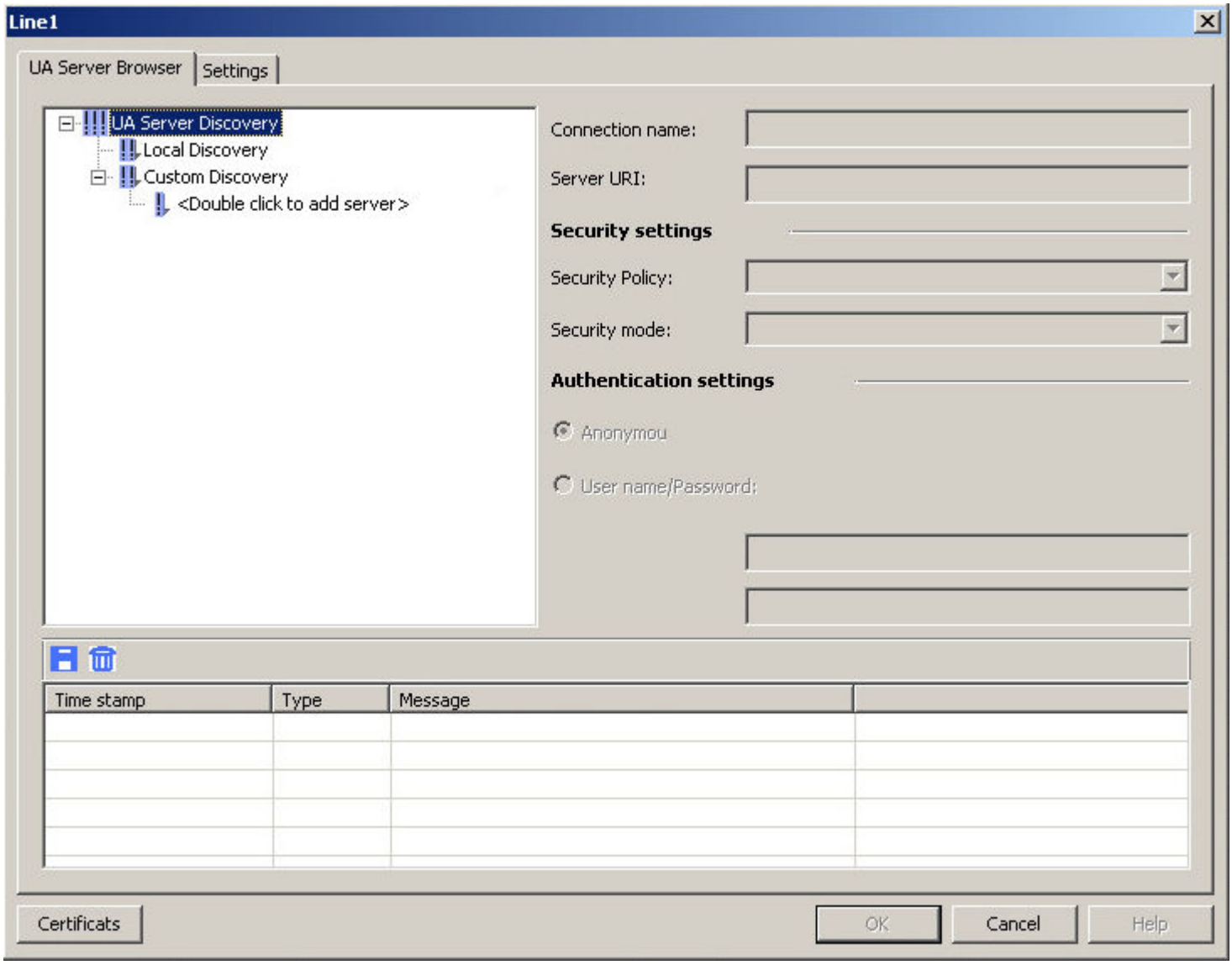
To connect to an OPC UA server, you need information about the server and the security settings.

You can create one connection in the WinCC project for each OPC UA server

You open the "Connection Parameters" dialog with the entry in the shortcut menu of the connection in the tag management.

Connection parameters

You configure the required settings for communication with the OPC UA server in the "UA Server Browser" tab.

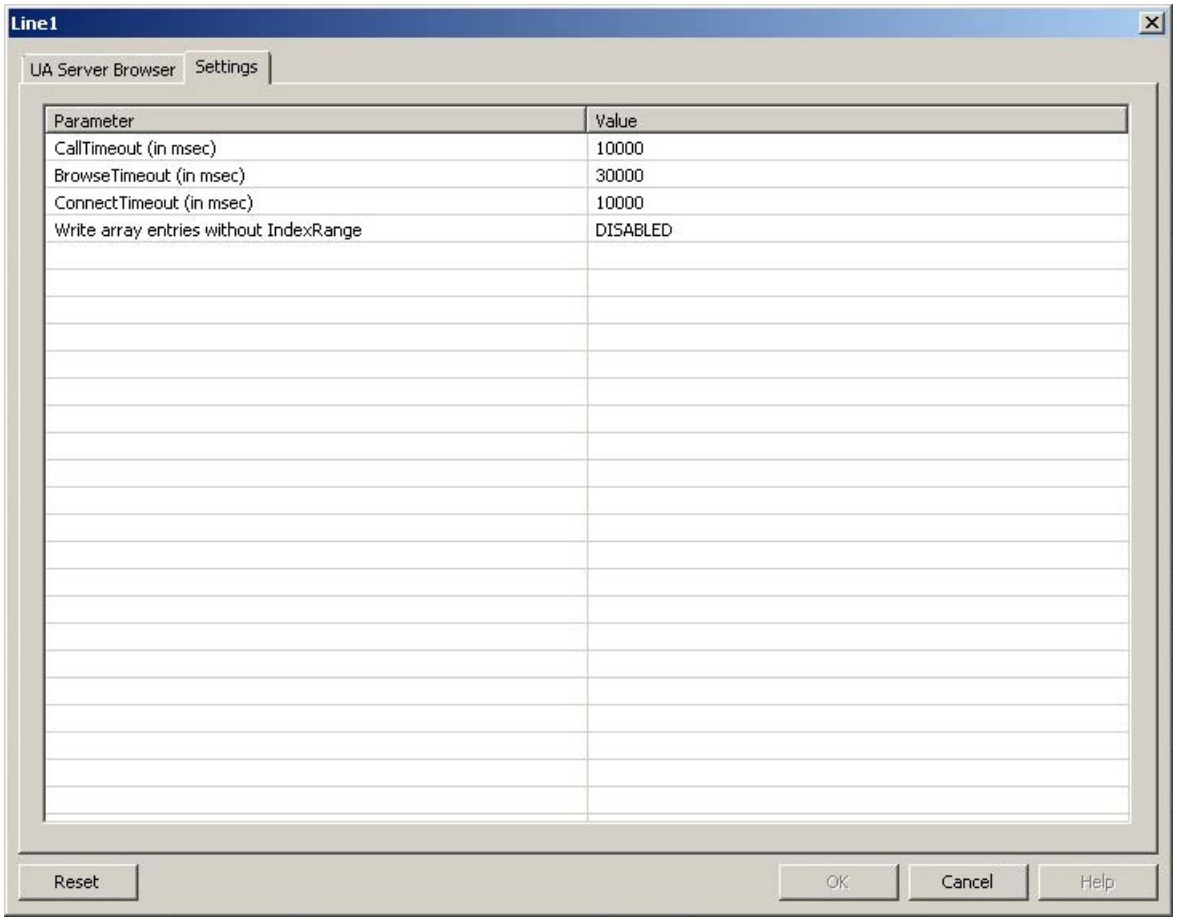


Field / setting	Contents
UA Server Discovery	The UA Discovery Server provides a list of available OPC UA servers.
Local Discovery	Local Discovery Lists all OPC UA Servers on the local computer, which are registered for the Local Discovery Server (LDS). Requirement: The LDS is installed on the local computer.

Field / setting	Contents
Custom Discovery	With Custom Discovery, the user can manually enter an OPC UA Server via its connection name. This is especially necessary when the OPC UA Server is on a remote computer. If the OPC UA Server is not registered at a Discovery Server, enter the Discovery address of the OPC UA Server in the following format: • <opc.tcp://Discovery server address:Port number> Use the "Browse" command from the shortcut menu of the server list to refresh the server display.
Connection name	Connection name of the OPC UA server. The name of the OPC UA server is not affected by this field.
Server URI	Address of the OPC UA server.
Security Policy	Select one of the security policies that the OPC UA server offers. • None • Basic128Rsa15 • Basic256 • Basic256Sha256 • Aes128_Sha256_RsaOaep • Aes256_Sha256_RsaPss • Automatically select the most secure option
Security mode	Select the desired security mode. • Sign • SignAndEncrypt
Authentication Settings	Select whether a user ID is required for the connection or anonymous access is to be allowed. If the user identification is set, enter the user name and password with which the WinCC OPC UA client accesses the OPC UA server. The authorizations are checked by the OPC UA server. With a WinCC OPC UA server, the authorizations are configured via the Windows user administration of the PC on which the OPC UA server is running.
Certificates	Opens the "PKI" folder in the installation directory.

Settings

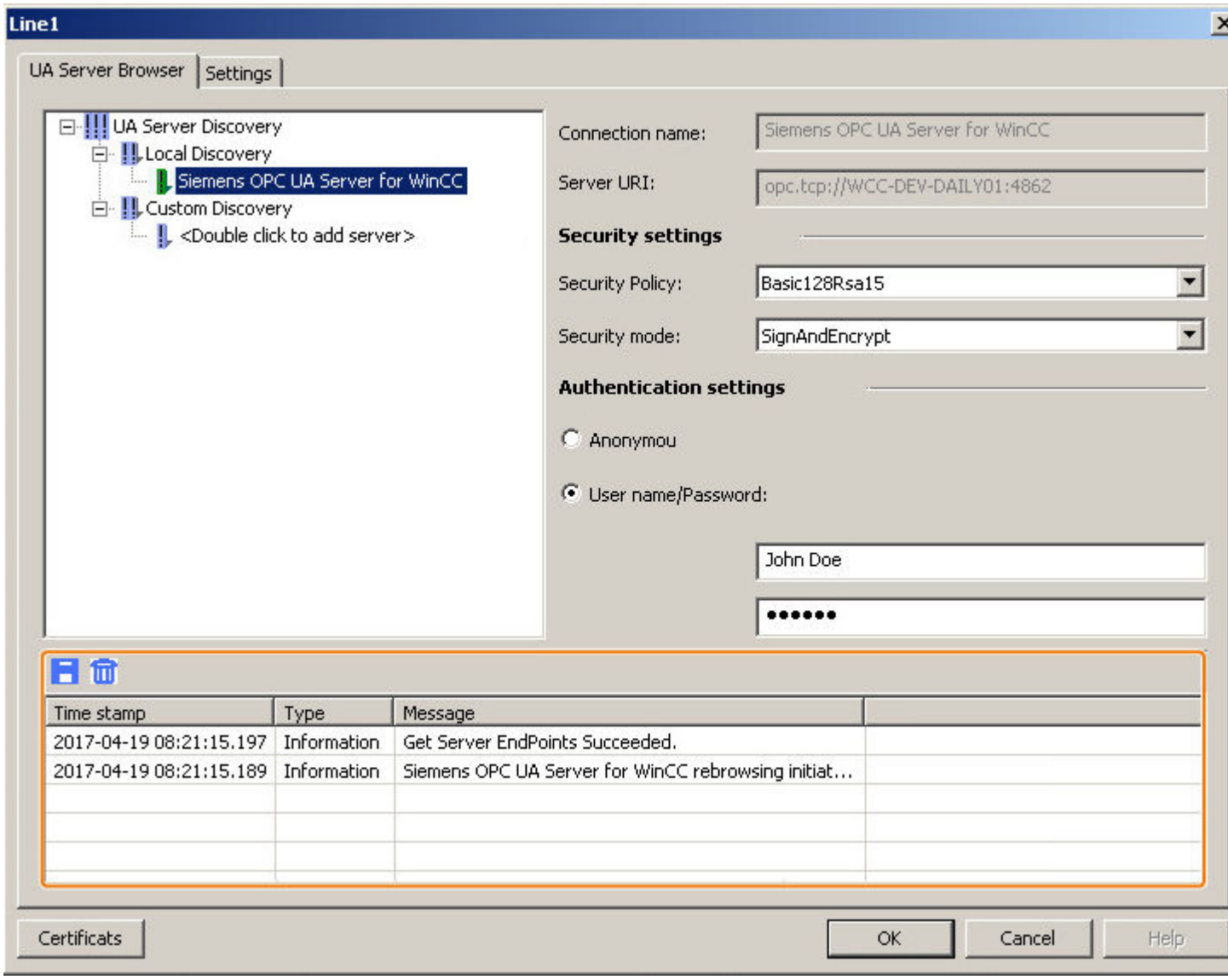
You configure the connection settings for communication with the OPC UA server by means of the "Settings" tab.



Field / setting	Contents
CallTimeout (ms)	Timeout for all OPC UA operations such as read, write, call.
BrowseTimeout (ms)	Timeout for browsing OPC UA objects such as tags. BrowseTimeout overwrites the CallTimeout for browsing because browsing larger object trees sometimes can take several seconds.
ConnectTimeout (ms)	Timeout for the connection establishment. ConnectTimeout overwrites the CallTimeout for connecting, because the server is possibly running on another PC and remote access is possible.
Write Array entries without IndexRange	Activates or deactivates the IndexRange support for remote access.

Console

The "Console" output window contains the status information and error messages with time stamps and type of message.



Deleting messages

You delete all entries with the following button:

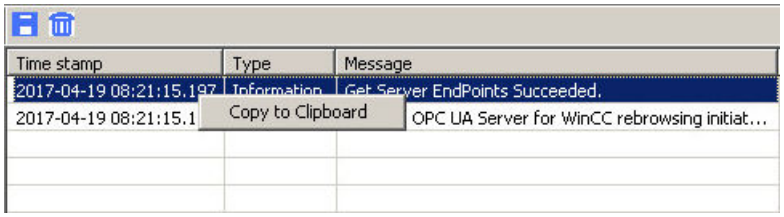
NOTICE
Delete: "Undo" not possible Deleted entries cannot be restored.

Saving messages

You store messages in a log file by clicking the following button:

Select the storage location from the dialog that opens here.

Individual messages can be copied to the clipboard via the shortcut menu of the message.



See also

How to configure a connection to the OPC UA server (Page 120)

Creating a connection to the OPC UA server via a NAT router (Page 110)

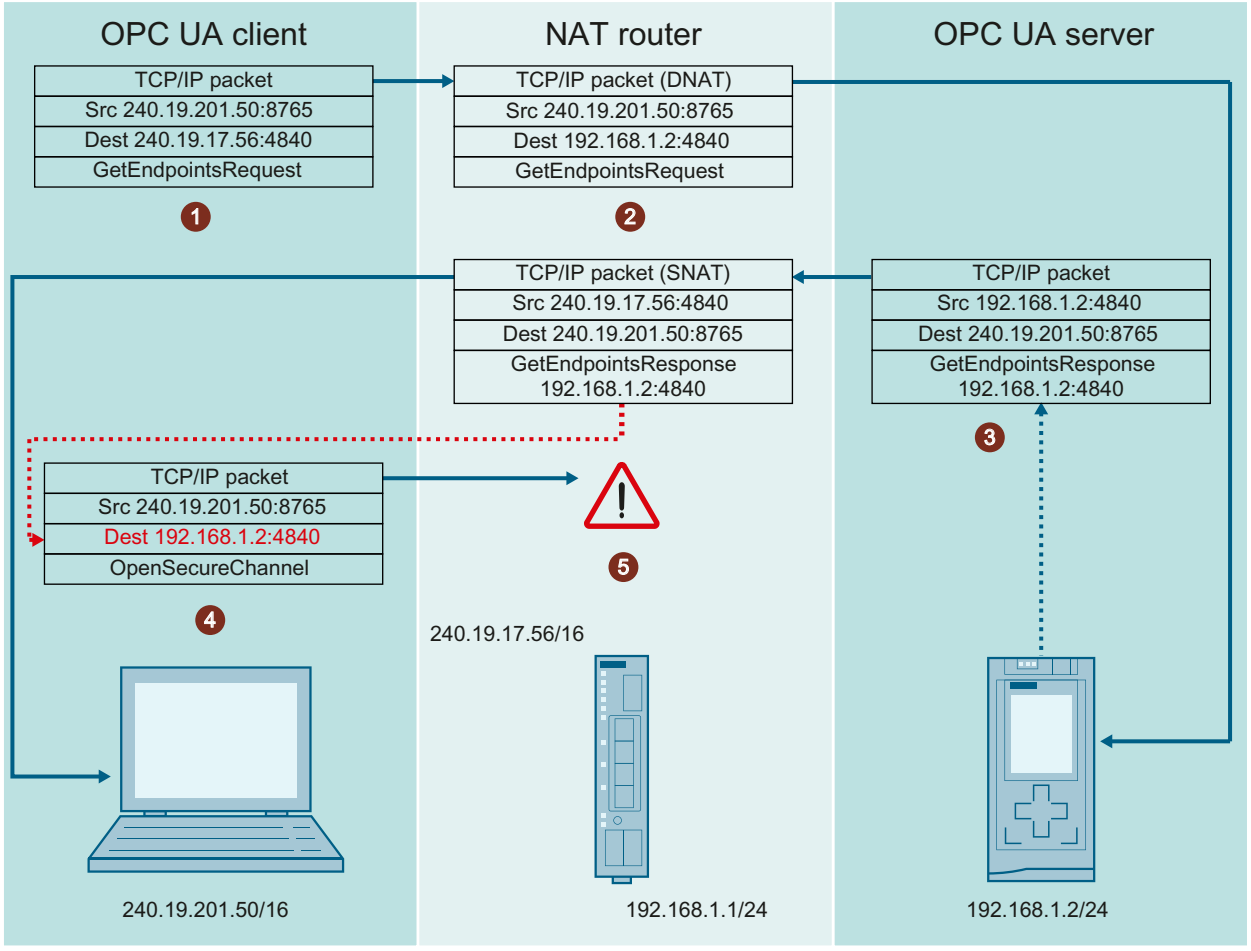
7.4.4 Creating a connection to the OPC UA server via a NAT router

Establishing a connection to the OPC UA server via NAT router can fail with the error messages "BadCommunicationError" or "BadNotConnected".

The IPv4 packets are manipulated by the router in NAT systems. As a result, the source IP ("Source NAT") or the destination IP ("Destination NAT") of a packet is replaced by an IP address configured in the router (depending on the destination port).

Faulty establishment of connection

A faulty connection establishment via a NAT router is explained in the following example:



1. The OPC UA client sends a "GetEndpointsRequest" with the destination IP address (DNAT) 240.19.17.56.
2. The NAT router changes the destination IP of the packet to 192.168.1.2 in order to send the packet to the OPC UA server.
3. The OPC UA server generates a "GetEndpointResponse" that contains its own IP address (192.168.1.2), among other things, and sends the telegram back to the client via the NAT router.
   For the client to be able to assign the telegram, the router needs to adjust the source IP address (SNAT) of the packet to 240.19.17.56.
4. The client starts to establish a connection ("OpenSecureChannel") to the IP address contained in the "GetEndpointResponse" of the server.
   The NAT router cannot manipulate the IP address in the "GetEndpointResponse" of the server.
5. However, the IP address of the response telegram cannot be reached directly from the client. Therefore, the connection cannot be established.

Solution: OPC UA client compares the IP addresses

The OPC UA client compares the IP address of the "GetEndpointRequest" with the IP address of the "GetEndpointResponse" that was sent by the OPC UA server.

If the IP addresses differ, the OPC UA client replaces the IP address of the "GetEndpointResponse" with the IP address of the "GetEndpointRequest".

This allows the OPC UA client to connect with the information of the manipulated "GetEndpointResponse".

Troubleshooting during certificate verification

If the connection attempt terminates with the error message "BadCertificateHostNameInvalid", it is possible to suppress the error in the configuration file of the configuration client and the Runtime client with "ValidationOptions".

The "SuppressHostNameInvalid" flag must be set in "TrustedCertificateStore" and "IssuerCertificateStore".

You can find additional information under "Setting up authentication via certificates. (Page 112)".

See also

Creating a connection to the OPC UA server (Page 105)

7.4.5 Setting up authentication via certificates.

Server certificates and client certificates

Distinguish between client and server certificates when configuring. Secure communication is only possible when client and server recognize each other's certificates.

Certificates are linked to the respective computers. After having moved, copied, or duplicated the WinCC project to a different computer, repeat this procedure so that each computer can verify the other's certificates.

Diagnostics

Use the console and the WinCC channel diagnostics for analysis.

You can find additional information under:

- Interfaces > OPC - Open Connectivity > WinCC OPC UA Server > Security Concept of OPC UA
- Communication > Communication Diagnostics > Diagnostics Channel "OPC"

Working with certificates

A self-signed certificate for the WinCC OPC UA Client is created with the installation.

An OPC UA Client can only connect to the OPC UA Server if the server recognizes this client certificate as trustworthy.

When a new connection is created, the OPC UA Server checks the client certificates. For communication via the WinCC channel "OPC UA WinCC Channel", the server must recognize the certificates for the configuration client and the Runtime client as trustworthy.

The certificates are located in the following folders in the WinCC installation path:

Certificates • Siemens OPC UA Tag Importer for WinCC • Siemens OPC UA Client for WinCC Runtime	opc\UAClient\PKI\OPCUA\certs
Private key	opc\UAClient\PKI\OPCUA\private

Rejected certificates

If the server does not recognize the client certificate as trustworthy, the connection is rejected and marked in red.

A message is generated on the console or entered in the log file, for example:

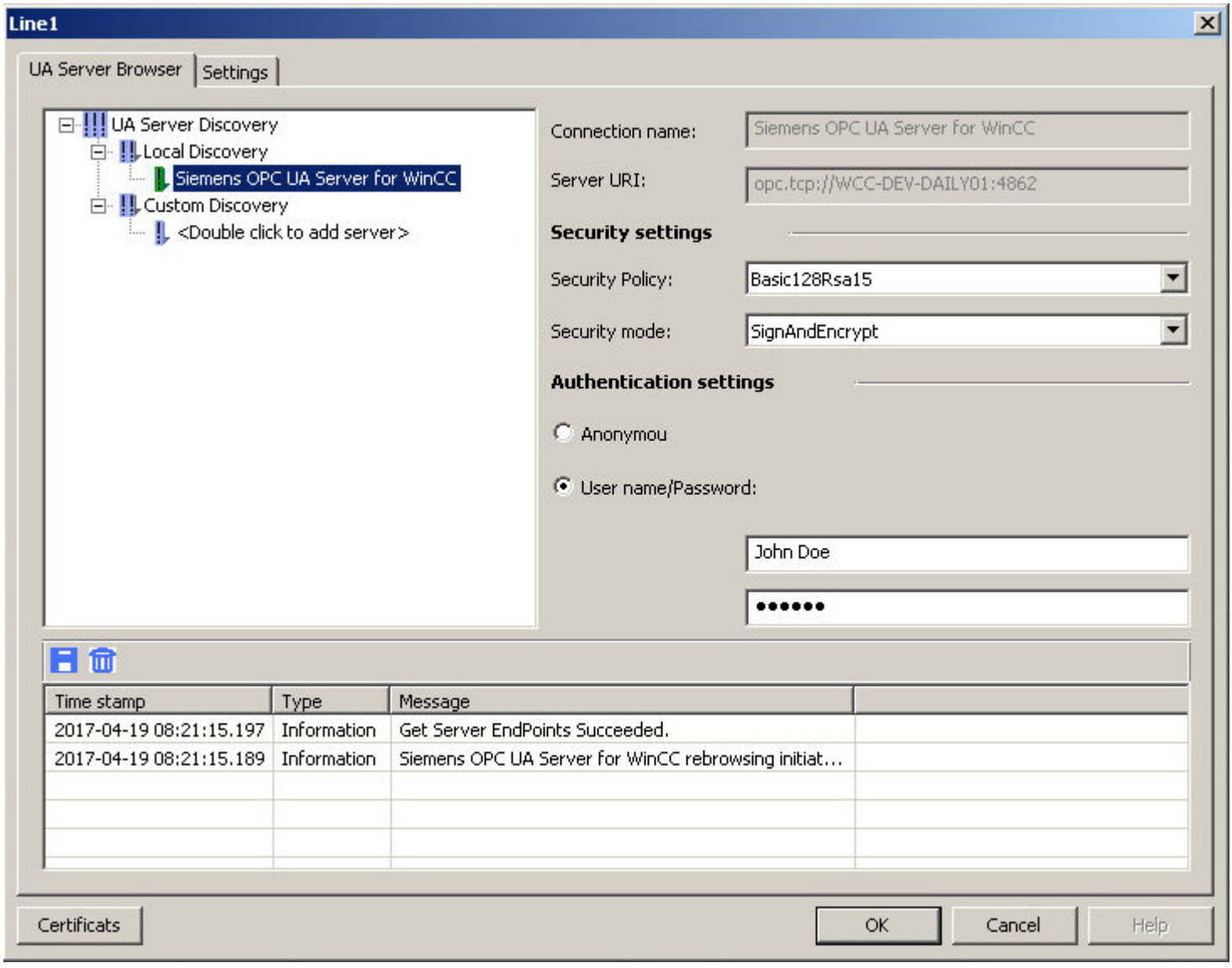
- Discovery of UA Server failed - The Certificate is not trusted.

Each rejected certificate is stored in the "PKI\...\Rejected\Certs" folder.

Trusting certificates

To specify that a certificate is trusted, move the certificate to the "Trusted\Certs" folder.

To do this, open the "PKI" folder by clicking the "Certificates" button in the "UA Server Browser" tab.



Move the certificates of the certification authority which should not be trusted into the "Issuer\certs" folder and, if available, into the "Issuer\crl" folder of the corresponding certificate revocation list.

```
- <DefaultApplicationCertificateStore>
        <!--The maximum size of the TrustList in bytes. 0 means no limit.-->
    <MaxTrustListSize>0</MaxTrustListSize>
        <!--File based certificate store used with OpenSSL. -->
    - <OpenSSLStore>
        <CertificateTrustListLocation>[ApplicationPath]\..\PKI\Trusted\certs\</CertificateTrustListLocation>
        <CertificateRevocationListLocation>[ApplicationPath]\..\PKI\Trusted\crl\</CertificateRevocationListLocation>
        <IssuersCertificatesLocation>[ApplicationPath]\..\PKI\Issuers\certs\</IssuersCertificatesLocation>
        <IssuersRevocationListLocation>[ApplicationPath]\..\PKI\Issuers\crl\</IssuersRevocationListLocation>
    </OpenSSLStore>
        <!--Application instance certificate for the client. -->
    + <ClientCertificate>
        <!--Folder used to store rejected server certificates. Administrators can copy files from here to the trust list. -->
    <RejectedCertificatesDirectory>[ApplicationPath]\..\PKI\Rejected\certs\</RejectedCertificatesDirectory>
</DefaultApplicationCertificateStore>
```

Configuration file

Description	Application	Configuration file
Configuration client Certificate: Siemens OPC UA Tag Importer for WinCC The attempt to establish a connection aborts when no valid client certificate is found. Storage path: • opc\UAClient\UaConfigServer	CCOpcUaImporter.exe	CCOpcUaImporter.xml
Runtime client Certificate: Siemens OPC UA Client for WinCC Runtime Without a valid runtime certificate, no current values are displayed in runtime. Storage path: • opc\UAClient\UaDAS	CcUaDAS.exe	CcUaDAS.xml

In the configuration files of the two clients, the certificate parameters are available in the <ClientCertificate> in the section <CertificateSettings>.

Example: Parameters for the control of the certificate

```
- <ClientCertificate>
  - <OpenSSLStore>
      <ClientCertificate>[ApplicationPath]\..\PKI\OPCUA\certs\</ClientCertificate>
      <ClientPrivateKey>[ApplicationPath]\..\PKI\OPCUA\private\</ClientPrivateKey>
    </OpenSSLStore>
      <!--SecurityCheckOverwrites for the client certificate. -->
  - <SecurityCheckOverwrites>
        <!-- Flag used to disable the client certificate validation error BadCertificateTimeInvalid, default is false. -->
      <DisableErrorCertificateTimeInvalid>false</DisableErrorCertificateTimeInvalid>
        <!-- Flag used to disable the client certificate validation error BadCertificateIssuerTimeInvalid, default is false. -->
      <DisableErrorCertificateIssuerTimeInvalid>false</DisableErrorCertificateIssuerTimeInvalid>
        <!-- Flag used to disable the client certificate validation error BadCertificateRevocationUnknown. Default is true. -->
      <DisableErrorCertificateRevocationUnknown>true</DisableErrorCertificateRevocationUnknown>
        <!-- Flag used to disable the client certificate validation error BadCertificateIssuerRevocationUnknown. Default is true. -->
      <DisableErrorCertificateIssuerRevocationUnknown>true</DisableErrorCertificateIssuerRevocationUnknown>
        <!-- Flag used to disable the check if the hostname of the client matches the hostname or IP addresse in the client certificate. The
        default is false. -->
      <DisableErrorCertificateHostNameInvalid>false</DisableErrorCertificateHostNameInvalid>
        <!-- Flag used to disable the check if the ApplicationUri in the client certificate matches the ApplicationUri of the client. The default
        is false. -->
      <DisableApplicationUriCheck>false</DisableApplicationUriCheck>
    </SecurityCheckOverwrites>
      <!--Enable client certificate creation if certificate is not available; true/false-->
    <GenerateCertificate>true</GenerateCertificate>
  - <CertificateSettings>
        <!--Name of the client application. -->
      <CommonName>Siemens OPC UA Tag Importer for WinCC</CommonName>
        <!--DomainComponent - [NodeName] is the default value to use the hostname of the machine. -->
      <DomainComponent>[NodeName]</DomainComponent>
        <!--Name of the organization using the OPC UA client. -->
      <Organization>Siemens AG</Organization>
        <!--Name of the organization unit using the OPC UA client. -->
      <OrganizationUnit>DF FA AS</OrganizationUnit>
        <!--Two letter code for country where the OPC UA client is running, e.g. DE or US. -->
      <Country>DE</Country>
        <!--The number of years the certificate is valid for. The maximum accepted number is 20, but it is strongly recommended to use a
        shorter time. -->
      <YearsValidFor>5</YearsValidFor>
        <!--Key length of the certificate to create. Valid values are 1024, 2048 for RsaMin and 2048, 3072 and 4096 for RsaSha256. -->
      <KeyLength>2048</KeyLength>
        <!--Defines the algorithm used to sign the certificate. Valid values are RsaMin and RsaSha256. Applications that support the
        Basic128Rsa15 and Basic256 profiles need a Certificate of type RsaMin. Applications that support the Basic256Sha256 profile need a
        Certificate of type RsaSha256. In this version of the SDK it is not possible to support multiple certificates for one Endpoint, thus it is
        not possible to support the RsaMin and the RsaSha256 profile at the same time.-->
      <CertificateType>RsaSha256</CertificateType>
        <!-- An application instance certificate needs to provide one or more DNSNames and/or IPAddresses at which the Endpoint can be
        reached. This information is added to the SubjectAlternativeName of the certificate. If this parameter is not set, the [NodeName] is
        used by default. -->
        <!-- Example for a list with 2 DNSNames plus 2 IPAddresses <IPAddress>2a00:1158:400:407:0:0:0:1b2</IPAddress>
        <IPAddress>213.95.4.190</IPAddress> <DNSName>demo.siemensautomation.com</DNSName> <DNSName>[NodeName]
        </DNSName> -->
      <DNSName>[NodeName]</DNSName>
    </CertificateSettings>
  </ClientCertificate>
```

Description

Parameter	Meaning
CommonName DomainCompo-nent OrganizationUnit Organization Country	Descriptive elements The parameters can be changed and have no effect on the function of the applica-tions.
YearsValidFor	Validity period of the certificate in years After the specified time has expired, the client can no longer be operated with this certificate. • Default value: 5

Parameter	Meaning
KeyLength	Length of the private key with which the certificate is created The length depends on the certificate type. • 1024: Minimum length for secure communication via OPC UA • 2048: Minimum length when Sha256 is used [1)]
CertificateType	Certificate type with which the certificate is signed The possible values are RsaMin and RsaSha256. The use of both the certificate types simultaneously is not supported. Applications that support the Basic128Rsa15 and Basic256 profiles require the certificate type RsaMin. Applications that support the profile Basic256Sha256 require the certificate type RsaSha256.
DNSName	The client certificate must provide at least one DNS name or at least one IP address. If the parameter is not set, "[NodeName]" is used.

1) To establish a secure connection to an OPC UA server with the Security Policy "Basic256Sha256", the server as well as the OPC UA client need a certificate with the following values:

- KeyLength: At least 2048
- SignatureAlgorithm: Sha256

Creating new client certificates

You need administrator rights to create new certificates on the OPC UA client.

When you create new certificates, the trust settings are reset. Certificates that were previously recognized as trustworthy are no longer trusted.

1. Create a backup.
2. Delete the existing certificates and the associated private keys in the corresponding folders.
3. In the configuration files, update the certificate parameters and save the XML files.
4. Open the DOS window "cmd.exe" in Microsoft Windows with administrator rights.
5. To create the certificates, go to the installation path of the respective OPC UA application.
6. Enter the corresponding call:
   - CCOpcUaImporter.exe/CreateCertificate
   - CcUaDAS.exe/CreateCertificate

   The new certificates and private keys are created in the storage paths.
   Specify that the new certificates are trusted.

Creating a client certificate automatically

If no certificate is found in the application certificate store and the "GenerateCertificate" flag is "true" in the configuration file, the client creates its own instance certificate.

The instance certificate is created as soon as the client is started.

```
- <ClientCertificate>
  + <OpenSSLStore>
      <!--SecurityCheckOverwrites for the client certificate. -->
  + <SecurityCheckOverwrites>
      <!--Enable client certificate creation if certificate is not available; true/false-->
    <GenerateCertificate>true</GenerateCertificate>
```

Suppressing certificate check error

Note

The operator must be informed about suppressed errors in the certificate check.

The certificates are strictly checked.

You still have the option to suppress some known errors for both applications of the client.

Own instance certificate of the client

When the client starts, its own instance certificate is checked.

If the certificate is invalid or has expired, the connection to the server cannot be established.

You have the option to install a valid certificate.

Alternatively, you can suppress errors that occur in the configuration file of the client.

To that end, adjust "SecurityCheckOverwrites" in the section "ClientCertificate" to match.

Trusted certificates and certificate authority certificates

The server certificate check can fail due to invalid certificates. You can suppress errors that occur in the configuration file of the client.

To that end, adjust "SecurityCheckOverwrites" in the section "UaClient" to match.

"SecurityCheckOverwrites"

Flag	Description	Comment
DisableErrorCertificateTimeInvalid	Ignores errors that occur when the server certificate validity period expires.	Default value: False
DisableErrorCertificateIssuerTimeInvalid	Ignores errors that occur when the validity period of the certification authority expires.	Default value: False
DisableErrorCertificateRevocationUnknown	Ignores errors that occur owing to the absence of the certificate embargo list of the server certificates.	Default value: False
DisableErrorCertificateIssuerRevocationUnknown	Ignores errors that occur due to missing certificate revocation list of the certificate authority.	Default value: False
DisableErrorCertificateHostNameInvalid	Deactivates the test if the host name of the server is linked to that of the client, to a host name specified in a server certificate, or to an IP address.	Default value: False

Flag	Description	Comment
DisableApplicationUriCheck	Deactivates the check for the absence of concordance between the parameter "ApplicationUri" in the server certificate and the parameter "ApplicationUri" that is returned by "EndpointDescription".	Default value: False
DisableNonceLengthCheck	Deactivates the check whether the server nonce has a minimum length of 32 bytes.	Default value: False
DisableEncryptedPasswordCheck	Deactivates the check whether the password of the user token was encrypted correctly.	The check is not successful if the server nonce was not specified or if "PasswordEncryptionMode" has the value "None". Default value: False
DisableTrustedCertificateForUserTokenRequired	Ignores the check whether the server certificate with which the password was encrypted has to be a trusted one.	The check is not successful if the certificate with which the password was encrypted is not a trusted one. Default value: False
DisableSessionIdCheck	Deactivates the check whether the server returns the value "Null" for "SessionId".	Default value: False
DisableCertificateUsageCheck	Deactivates the check for the certificate structure and the use of the key.	The following parameters are checked: • "SubjectAlternativeName" • "KeyUsage" • "ExtendedKeyUsage" Default value: False
DoServerCertificateVerify	The client is forced to check the server certificate before setting up the connection.	Default value: True

Example: "SecurityCheckOverwrites" of a certificate

```
        <!--SecurityCheckOverwrites for trusted server certificates. -->
- <SecurityCheckOverwrites>
        <!-- Flag used to disable the server certificate validation error BadCertificateTimeInvalid, default is false. -->
    <DisableErrorCertificateTimeInvalid>false</DisableErrorCertificateTimeInvalid>
        <!-- Flag used to disable the server certificate validation error BadCertificateIssuerTimeInvalid, default is false. -->
    <DisableErrorCertificateIssuerTimeInvalid>false</DisableErrorCertificateIssuerTimeInvalid>
        <!-- Flag used to disable the server certificate validation error BadCertificateRevocationUnknown. Default is false. -->
    <DisableErrorCertificateRevocationUnknown>false</DisableErrorCertificateRevocationUnknown>
        <!-- Flag used to disable the server certificate validation error BadCertificateIssuerRevocationUnknown. Default is false. -->
    <DisableErrorCertificateIssuerRevocationUnknown>false</DisableErrorCertificateIssuerRevocationUnknown>
        <!-- Flag used to disable the check if the hostname the client connected to matches one of the hostnames or IP addresses in t
        certificate. The default is false. -->
    <DisableErrorCertificateHostNameInvalid>false</DisableErrorCertificateHostNameInvalid>
        <!-- Flag used to disable the check if the ApplicationUri in the server certificate matches the ApplicationUri the server returns i
        EndpointDescription. The default is false. -->
    <DisableApplicationUriCheck>false</DisableApplicationUriCheck>
        <!-- Flag used to disable the check if the ServerNonce has the correct length of minimum 32 bytes. The default is false. -->
    <DisableNonceLengthCheck>false</DisableNonceLengthCheck>
        <!-- Flag used to disable the check if the password of a UserPassword identity token is encrypted properly. The check will fail i
        ServerNonce is not set or the PasswordEncryptionMode is None. The default is false. -->
    <DisableEncryptedPasswordCheck>false</DisableEncryptedPasswordCheck>
        <!-- Flag used to disable the check if the certificate used to encrypt a password needs to be trusted. The check will fail if the s
        certificate used to encrypt a user token is not trusted. The default is false. -->
    <DisableTrustedCertificateForUserTokenRequired>false</DisableTrustedCertificateForUserTokenRequired>
        <!-- Flag used to disable the check if the server returned Null SessionId. The default is false. -->
    <DisableSessionIdCheck>false</DisableSessionIdCheck>
        <!-- Flag used to disable the checks for the server certificate structure and key usage. These checks include checking for the
        SubjectAlternativeName, the KeyUsage and ExtendedKeyUsage of the certificate.The default is false. -->
    <DisableCertificateUsageCheck>false</DisableCertificateUsageCheck>
        <!-- Flag used to force the client to verify the server certificate before establishing the connection. The default is true. -->
    <DoServerCertificateVerify>true</DoServerCertificateVerify>
  </SecurityCheckOverwrites>
```

See also

How to configure a connection to the OPC UA server (Page 120)

Creating a connection to the OPC UA server via a NAT router (Page 110)

7.4.6 How to configure a connection to the OPC UA server

Introduction

This section shows you how to connect to the OPC UA server.

Requirements

- The OPC UA server is active.
- The communication must not be blocked by a firewall.
  The port numbers of the OPC UA server must be activated.
- It must be possible to access the PC of the OPC server from the WinCC PC via the IP address.

- The OPC UA server trusts the client certificate.
- The channel "OPC UA WinCC Channel" is added in the WinCC project of the WinCC OPC UA client.

Procedure

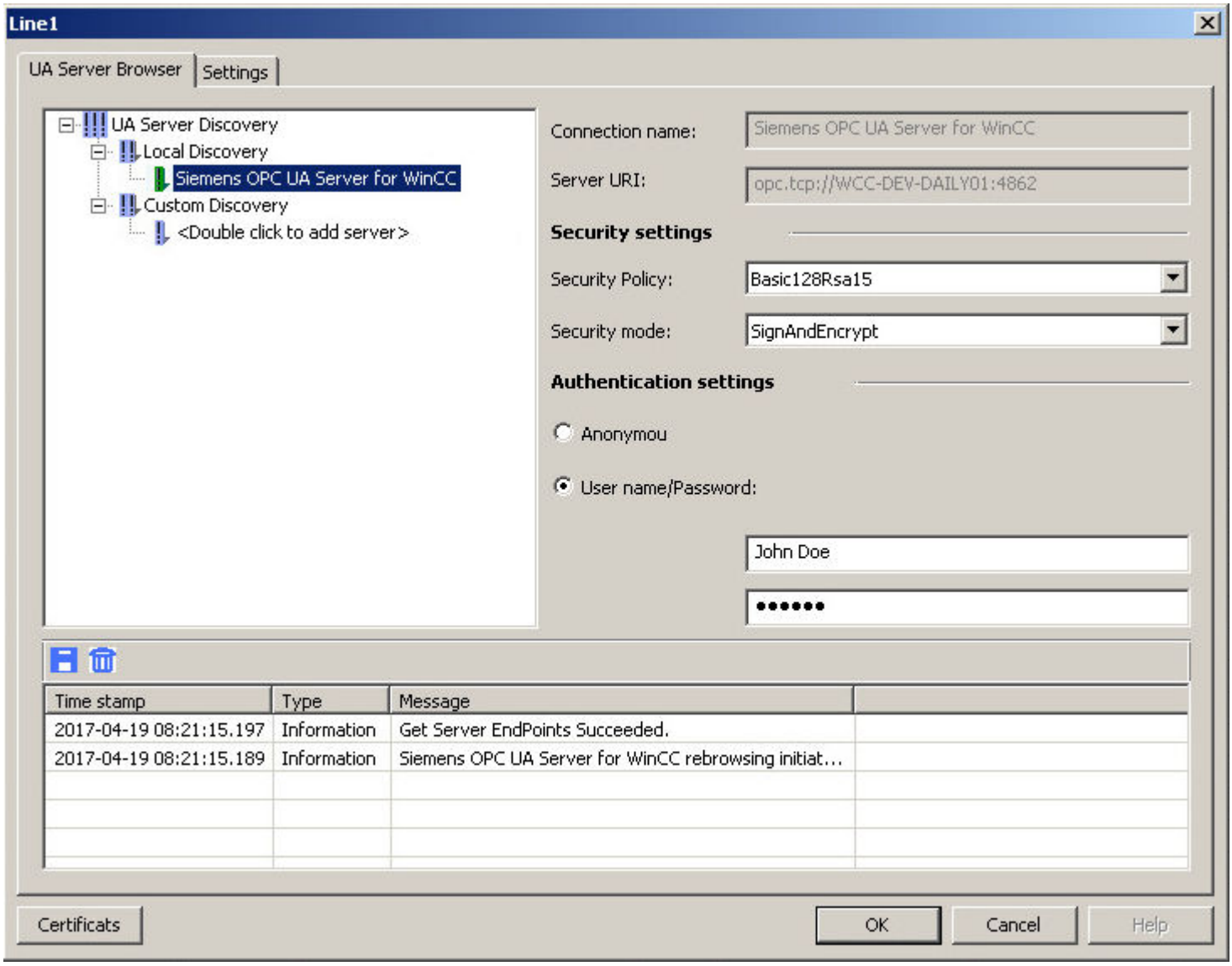
1. Open WinCC Tag Management in the WinCC Configuration Studio.
2. Create a new connection using the shortcut menu of "OPC UA Connections".
3. Name the connection.
   Keep the following restrictions in mind when naming the connection:
   - The first character must be a letter.
   - Only letters contained in the English alphabet are allowed.
   - Spaces are not allowed.
   - The following special characters are not allowed: . ! @ & $ # \ / : * ? " < > |
   - Reserved keywords of Windows devices are not allowed, e.g.: CON, PRN

   Note

   A connection can only be renamed when Runtime is deactivated.

4. Open "Connection Parameters" dialog from the shortcut menu of the created connection.
5. Select a server:
   - To update the display of the local OPC UA server, select "Browse" in the shortcut menu of "Local Discovery".
   - To enter the URL of an OPC UA server, double-click the row under "Custom Discovery". Enter the IP address in the following format:
     - opc.tcp://<OPC-UA-Server_Address:Port_Number>

6. Specify the desired settings and confirm your entries with "OK".
   The connection to the OPC UA server is established and marked green in the "UA Server Browser" tab.



7. To create the system tags for connection establishment and connection status, select the "Create Enable/Disable Tags" entry in the shortcut menu of the connection.
   The following tags are created in the internal tag group "ConnectionStates":
   - @<Connectionname>@ForceConnectionStateEx
   - @<Connectionname>@ConnectionStateEx

Result

If Runtime is activated, the connection is marked with a green check mark in the Tag Management.

Client certificate

If the OPC UA server does not recognize the client certificate, the connection is not established.

The connection is marked in red in the "UA Server Browser" tab. The connection is marked with a red exclamation point in the project tree of the tag management.

Ensure that the OPC UA server accepts the client certificate.

See also

The "Symbols" view (Page 126)

OPC UA tags (Page 123)

Setting up authentication via certificates. (Page 112)

7.4.7 Configuring the OPC UA tags

7.4.7.1 OPC UA tags

Supported OPC UA nodes

When the connection to the OPC UA server has been established, you load the objects and OPC UA nodes of the OPC UA server into the "Symbols" view.

Data types

Tags with the following data types are supported:

OPC UA node	WinCC tag type
Binary Tag	Binary tag
Byte	Signed 8-bit value or unsigned 8-bit value
Int16	Signed 16-bit value
UInt16	Unsigned 16-bit value
Int32	Signed 32-bit value
UInt32	Unsigned 32-bit value
Float	Floating-point number 32-bit IEEE 754 or floating point number 64-bit IEEE 754
String	Text tag, 8-bit character set or text tag, 16-bit character set
ByteString	Raw data tag
DateTime	Date/time
Enumerations	Signed 32-bit value

Importing an OPC UA node as a WinCC tag

The OPC UA nodes that you can import as WinCC tag can be accessed in the data area in the "Access" column using the ☑ icon. The field is deactivated for unsupported data types.

To import OPC UA nodes that do not match any WinCC data type, manually assign a matching WinCC data type to the individual OPC UA nodes.

An OPC UA node can only be imported once.

You can find additional information under "To import an OPC UA node as a WinCC tag (Page 127)".

WinCC tag names

When the OPC UA nodes are imported, the names of the WinCC tags are assigned automatically.

You configure the settings for the tag name in the properties of the respective connection after loading the OPC UA nodes.

If the tag name on the OPC UA server contains special characters, they are replaced by an underscore "_".

Settings for name generation

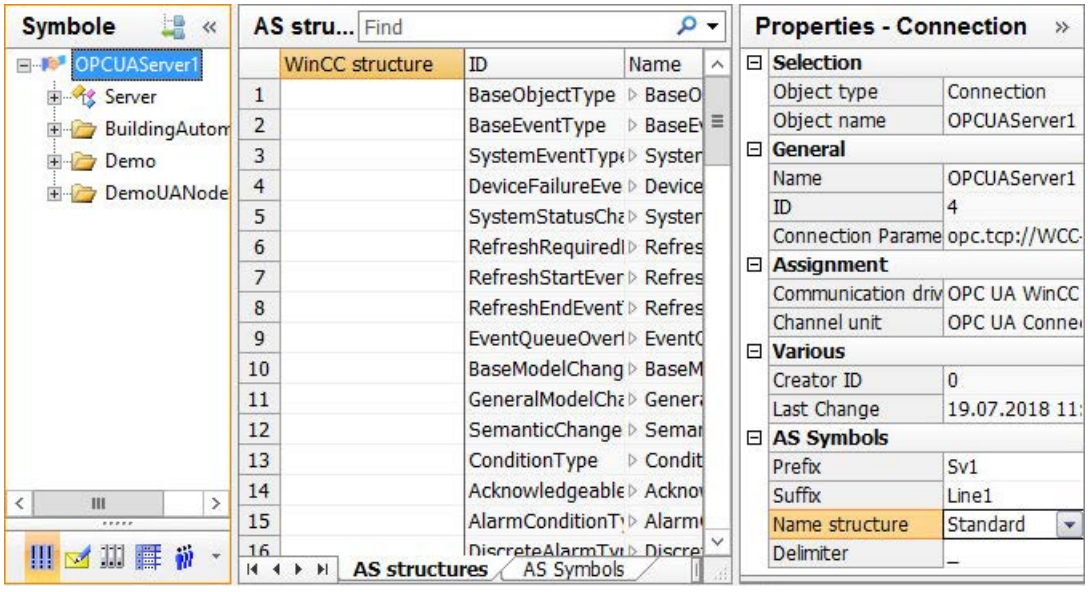
To display the settings in the "AS symbols" property group, click the connection name in the "Symbols" view.

The following settings can be made:

- The path of the OPC UA node is applied as the name.
- The name of the OPC UA node is applied and, if appropriate, supplemented by a prefix or suffix.
- The path of the OPC UA node is applied, and, if appropriate, supplemented with a prefix or suffix.

The "prefix" or "suffix" option adds the specified string to the tag name.

The components of the name are connected through the separator. The underscore is used by default.



Examples

There is "CurrentState" tag on the OPC UA server in the path "Spectrometer/Channel_0/ChannelStateMachine".

"Prefix_" is entered in the "Prefix" field and "Suffix" in the "_Suffix" field.

The following WinCC tag is created in the WinCC project of the WinCC OPC UA client:

Setting	WinCC tag name
Path name without prefix and suffix	Spectrometer/Channel_0/ChannelStateMachine/CurrentState
Name of the OPC UA node	Prefix_CurrentState_Suffix
Path name	Prefix_Spectrometer/Channel_0/ChannelStateMachine/CurrentState_Suffix

Deleting WinCC tags

No active connection to the OPC UA server is necessary to delete the WinCC OPC UA tags.

To delete an imported WinCC tag in the WinCC Tag Management, select the "Delete" entry in the shortcut menu of the WinCC tag or use the <Del> key.

Creating tag groups

To create a tag group below a connection, select "New group" in the shortcut menu of the connection.

To change the name, click on the group name.

When creating the WinCC tags the following action is to be observed:

- The connection is selected in the project navigation window:
  - The WinCC tag is created directly below the connection.
  - The "Tags" data area shows all tags, even if they are assigned to a tag group. The "Group" field may contain the assigned tag group. You can change the group assignment via the drop-down list.
- The tag group is selected in the project navigation window:
  - The WinCC tag is created in the tag group.
  - The "Tags" data area only shows the tags that were created in the tag group.

Migration of WinCC projects

Prior to WinCC V7.4, the WinCC OPC UA connections were created in the OPC channel.

During migration of the WinCC project, the connections and tags of the WinCC OPC UA client are also migrated into the changed structure.

If you have exported WinCC OPC UA tags, note the following order:

1. Import the exported WinCC OPC UA tags.
2. Migrate the WinCC project.

See also

The "Symbols" view (Page 126)

To import an OPC UA node as a WinCC tag (Page 127)

7.4.7.2 The "Symbols" view

Introduction

After the successful connection configuration, you have access to the OPC UA nodes on the OPC UA server.

To create the WinCC tags, you load the OPC UA nodes in the Tag Management into the "Symbols" view.

Display of the symbols

Representation of the data structure

The representation of the data in the structure tree corresponds to the hierarchy on the OPC UA server.

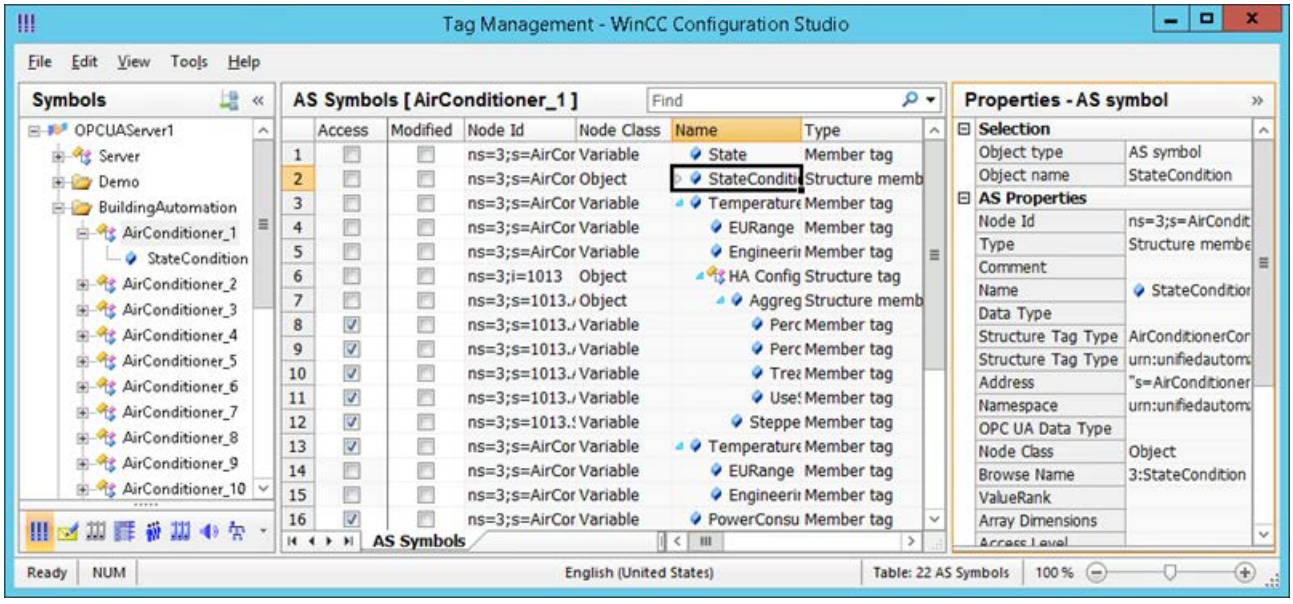
Complex data structures are displayed in the data area.

To show lower-level hierarchies in the data area, click on the symbol for extending in the "Name" column: ▷.

Linking with WinCC tags

On the "AS Symbols" and "AS structures" tabs, you specify which OPC UA nodes are linked with WinCC tags.

You can only change the properties of the OPC UA nodes on the OPC UA server.



"AS structures" tab

If the loaded data also contains structures, the "AS structures" tab is displayed additionally.

The tab is displayed when the connection name is selected in the navigation area.

You can also configure the structure assignment in the default view of the Tag Management.

Click the connection under the communication channel "OPC UA WinCC Channel" to display the "AS structures" tab.

Change View To

You use the following button to switch in Tag Management between the default view and the "Symbols" view: 

The button is available only after the data records have been loaded.

After the WinCC Configuration Studio has been closed, the "Symbols" view with the tabs "AS Symbols" and "AS structures" tab is hidden again.

In the default view, the "AS structures" tab is also only visible again when the OPC UA nodes have been loaded once more.

See also

How to use automatically generated structure types (Page 131)

To import an OPC UA node as a WinCC tag (Page 127)

How to configure a connection to the OPC UA server (Page 120)

OPC UA tags (Page 123)

7.4.7.3 To import an OPC UA node as a WinCC tag

Introduction

This section shows you how to import OPC UA nodes as WinCC tags to the WinCC Tag Management.

The tags for the OPC UA WinCC Channel are created under the channel unit "OPC UA Connections" in the WinCC Configuration Studio.

Allocation of data types

The supported types of OPC UA nodes are automatically assigned to the corresponding WinCC tag types. You can find the assignment table under "OPC UA tags (Page 123)".

For OPC UA nodes that are not supported, the "Access" option is initially disabled. These nodes are not imported into the Tag Management.

To also import OPC UA nodes that do not match any WinCC data type, assign a matching WinCC data type manually.

Requirements

- The connection to the OPC UA Server is established.

Procedure

1. Select the configured connection in the navigation area under "OPC UA Connections".
2. Select the "Browse OPC server" entry from the shortcut menu of the connection.
   The available data of the OPC UA server is loaded and the "Symbols" view opens.
   The loaded data is displayed in the table area on the "AS Symbols" tab.
   If the loaded data also contains structures, the "AS structures" tab is displayed additionally.
3. Select the connection in the navigation area of the "Symbols" view.
4. Select the options for the WinCC tag names in the "Properties - Connection" section in the "AS Symbols" group:
   - Name structure
   - Separator
   - If applicable, prefix and suffix
5. Select the required entry in the navigation area.
   The loaded OPC UA nodes respectively contained are displayed on the "AS Symbols" tab.
6. To create WinCC tags for the required AS symbols, activate the "Access" column in each case.
   To import all supported OPC UA nodes of the selected object in WinCC, select "Select All" from the shortcut menu of the "Access" column.
7. To import OPC UA nodes that are not supported, select a matching WinCC data type in the "Data Type" column.
   The "Access" field of the OPC UA node can now be selected.
   When you enable the "Access" option, the OPC UA node is imported as WinCC tag with the selected data type.

Result

You will see the newly configured WinCC tags in the WinCC Tag Management.

Only change the properties of tags on the OPC UA server, however.

Synchronizing WinCC tags with the OPC UA server

After loading from the controller or a file, the Tag Management compares the properties of the AS symbols with the linked WinCC tags.

If the properties of a symbol do not match, the "Modified" field on the "AS Symbols" tab is activated.

The respective property field is highlighted in red. The tooltip of the field contains additional details.

Updating WinCC tags

To apply the current properties of the OPC UA node, deactivate the "Modified" field.

Alternatively, deactivate the "Access" field and activate it again to recreate the WinCC tag.

See also

How to use automatically generated structure types (Page 131)

The "Symbols" view (Page 126)

7.4.8 Using OPC UA types in WinCC

7.4.8.1 Importing OPC UA types as WinCC structure types

Introduction

This section shows you how to import object types or objects of the OPC UA server to the WinCC Tag Management.

The objective is the easy configuration of OPC UA objects as structure tags in WinCC.

Overview: Basic procedure

1. Assigning object types
2. Configuring objects

The imported objects are created as WinCC structures or structure tags and mapped as follows:

OPC UA	WinCC
OPC UA object type	Structure type
Properties / tags of the OPC UA object type	Structure type elements
OPC UA object	Structure tag
Properties / tags of the OPC UA object	Structure tag elements

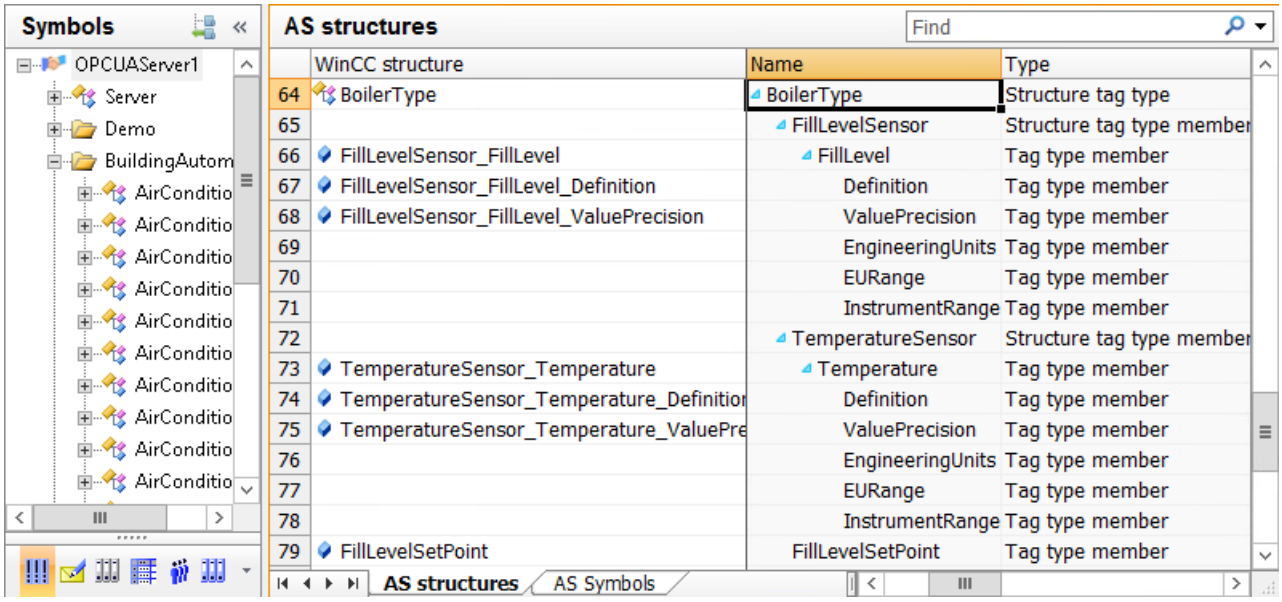
Configuration step 1: Assigning object types

If the connection name is selected in the navigation area of the "Symbols" view, the "AS structures" tab is shown.

You link the OPC UA object types with the WinCC structure types in the "AS structures" tab.

The properties and tags of the OPC UA object type are linked with the structure type elements.

You can have the WinCC structure types and structure type elements created automatically or assign already created WinCC structure types.



Automatic assignment

You are having the WinCC structure types and structure type elements created automatically.

A structure type with the name of the OPC UA structure is created in WinCC Tag Management.

Structure type elements are created for properties and tags of the OPC UA object type that can be mapped in WinCC.

The hierarchy of the OPC UA object types is mapped through the names of the structure type elements, for example, "FillLevelSensor_FillLevel_Definition".

Note

Maximum length of the tag name

Note that the WinCC tag names have a maximum length of 128 characters.

This limit applies to the entire expression for structure tag elements:

- Structure tag name + Period + Structure type element name

Manual assignment

You create structure types and structure type elements in the WinCC Tag Management. Make sure that the data type of a structure type element and the DataType of the corresponding property or tag are always the same.

You link the read OPC UA object types with the created WinCC structure types.

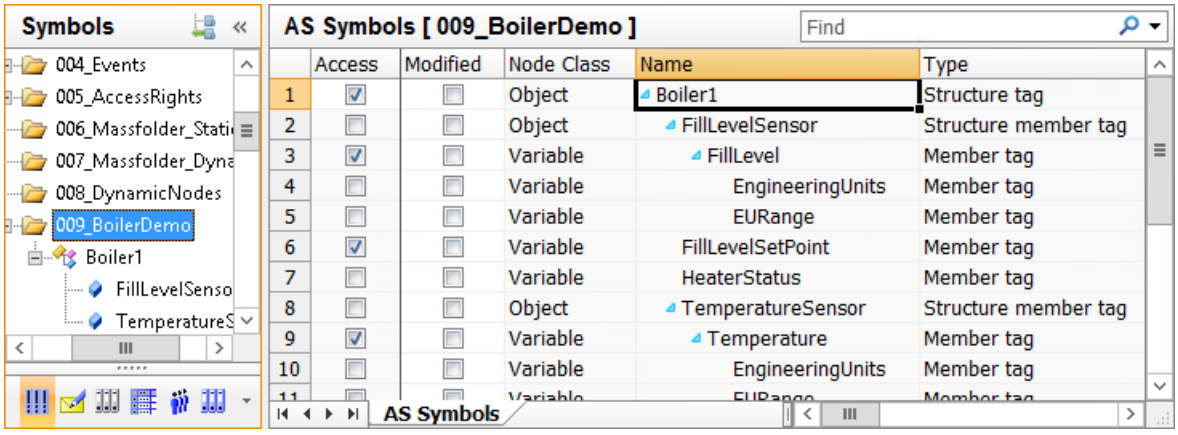
If the structure type elements and properties or tags have the same name and data type, they are assigned automatically.

Alternatively, you assign the structure type elements individually to the properties and tags.

Configuration step 2: Assigning objects

You configure the OPC UA objects as WinCC structure tags in the "AS Symbols" tab.

As soon as you activate the access for an OPC UA object, the structure tags and structure tag elements are created automatically.



7.4.8.2 How to use automatically generated structure types

Introduction

In this approach you are having the WinCC structure types and structure type elements created automatically when importing OPC UA objects.

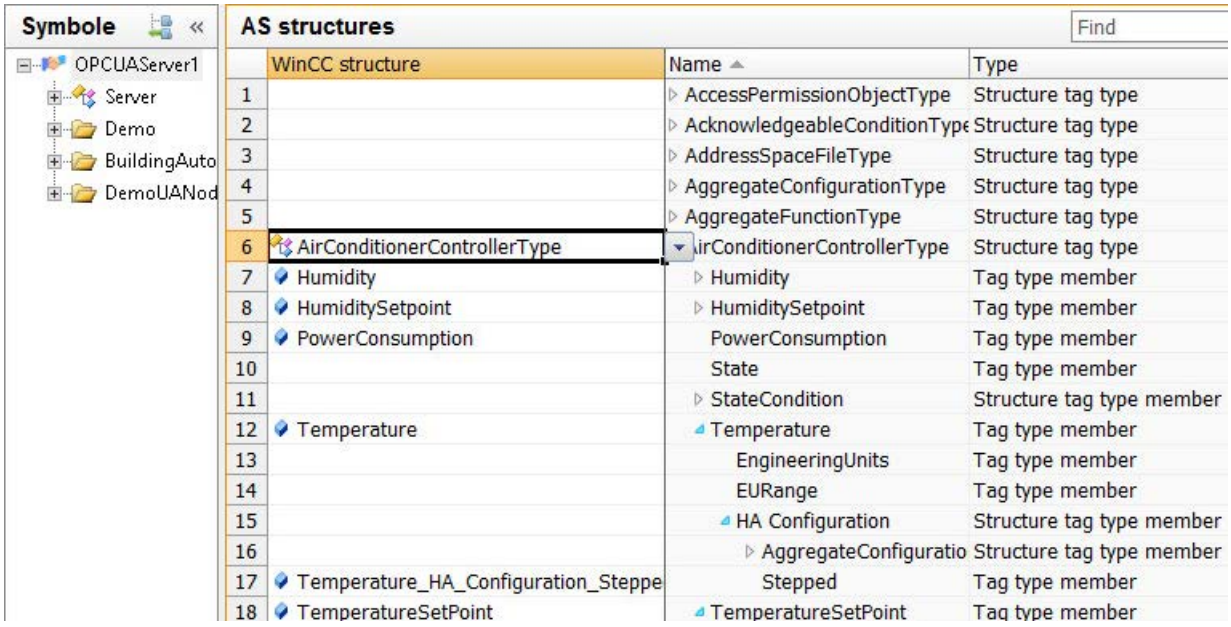
Requirements

- The connection to the OPC UA Server is established.

Procedure

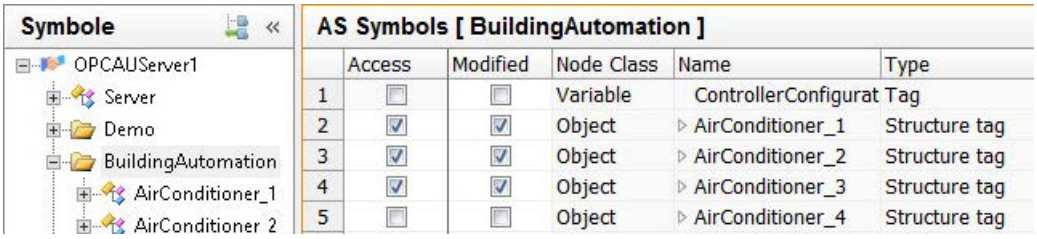
1. Select the "Browse OPC server" entry from the shortcut menu of the OPC UA connection.
   The available data of the OPC UA server is loaded.
   The "AS structures" tab is displayed in the "Symbols" view with the OPC UA object types. The tab is displayed when the connection name is selected in the navigation area.
   To display the elements below the object type, click the arrow in front of the object type name in the "Name" field.
2. To select an OPC UA object type, click the line number.
   More than one can be selected.

3. Select the "Create structure" entry in the shortcut menu of the row.
   - A structure type with the name of the OPC UA object type is created in WinCC Tag Management.
   - One structure type element each is created for all properties and tags of the OPC UA object type that can be mapped.
   - The hierarchy is mapped through the names of the structure type elements.
     Note that the WinCC tag names have a maximum length of 128 characters.
     If necessary, shorten the name of a structure type element before you create the structure tag.



4. To edit the structure type and the structure type elements in the "Tag Management" view, click the following symbol: .
   If required, change the names of the structure types or structure type elements under "Structure tags".
   If necessary, delete any structure type elements you do not need.
   The changes are applied in the structure tags of the OPC UA connection.
5. To display the "Symbols" view again, click on the OPC UA connection and the following symbol: .

6. To display the OPC UA objects, select the desired node in the navigation.
7. In the "AS Symbols" tab, activate the "Access" field of the OPC UA object.



A structure tag is created in the linked structure type for the OPC UA object.
The properties and tags of the OPC UA object are mapped to the structure type elements.

See also

To import an OPC UA node as a WinCC tag (Page 127)

The "Symbols" view (Page 126)

7.4.8.3 How to use manually generated structure types

Introduction

In this approach you are using the structure types and structure type elements that were created in the WinCC Tag Management for importing the OPC UA objects.

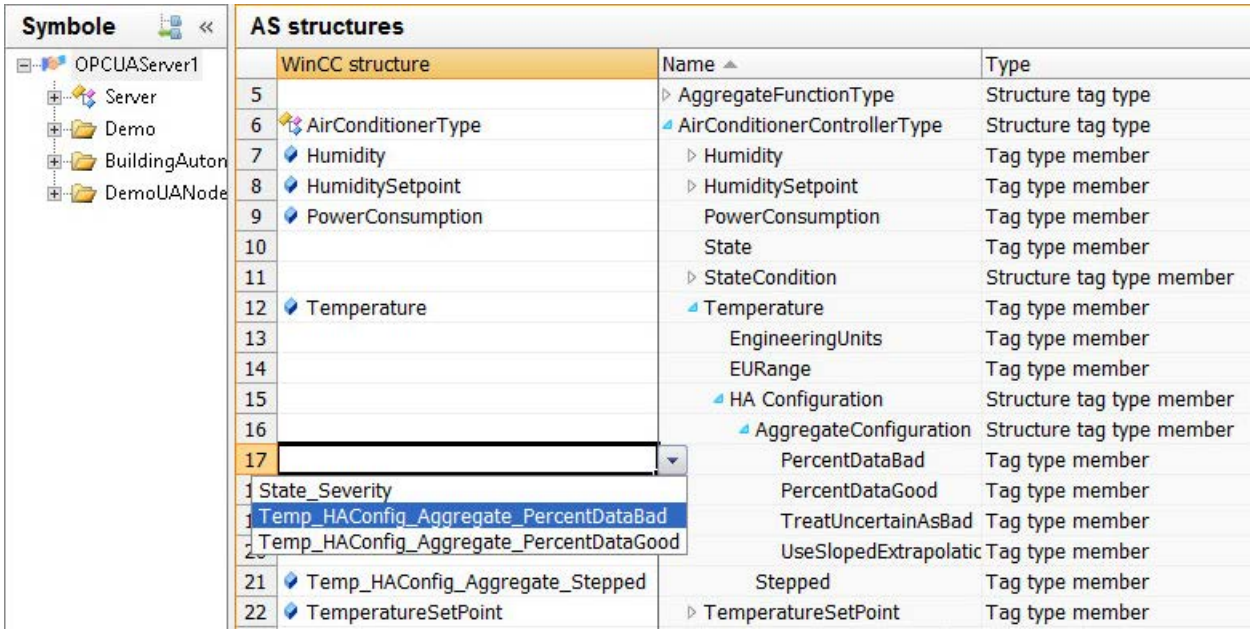
Requirements

- The connection to the OPC UA Server is established.
- A structure type was created in the WinCC Tag Management.
- Structure type elements with the following properties are configured in the structure type:
  - External: Enabled
  - Data type: DataType of the corresponding property or tag of the OPC UA object type

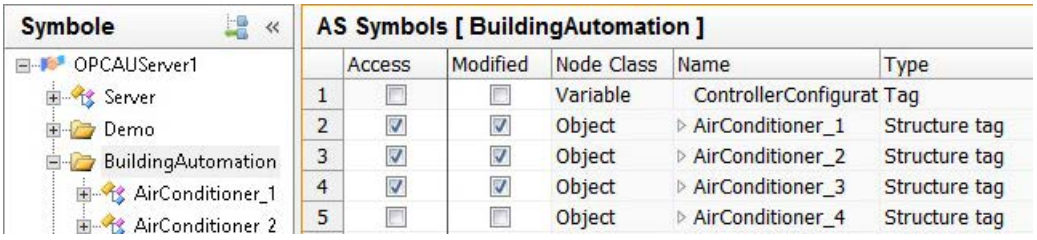
Procedure

1. Select the "Browse OPC server" entry from the shortcut menu of the OPC UA connection.
   The available data of the OPC UA server is loaded.
   The "AS structures" tab is displayed in the "Symbols" view with the OPC UA object types. The tab is displayed when the connection name is selected in the navigation area.
   To display the elements below the object type, click the arrow in front of the object type name in the "Name" field.
2. In the "WinCC structure" field, select the created structure type that you want to assign to the OPC UA object type.
   Structure type elements that have the same name and data type as a property or tag of the object type are assigned automatically.

3. To assign a structure type element with a different name to a property or tag, select the element in the "WinCC structure" field.
   The list contains all the structure type elements which have not been assigned yet and have the same data type as the property or tag.



4. To display the OPC UA objects, select the desired node in the navigation.
5. In the "AS Symbols" tab, activate the "Access" field of the OPC UA object.



   A structure tag is created in the linked structure type for the OPC UA object.
   The properties and tags of the OPC UA object are mapped to the structure type elements.

6. To edit the structure tags in the "Tag Management" view, click the following symbol:
   If required, change the names of the structure types, structure type elements or structure tags under "Structure tags".
   The changes are applied in the structure tags of the OPC UA connection.

7.4.9 Using OPC UA alarms in WinCC

7.4.9.1 The "Monitored objects" view

Event Notifier and alarms

After the successful connection configuration, you have access to the Event Notifiers on the OPC UA server.

The Event Notifiers trigger alarms or events that you can have output as WinCC messages.

To configure WinCC messages for the OPC UA alarms, load the Event Notifiers in Alarm Logging to the "Monitored objects" view.

NOTICE
A local WinCC OPC UA server is not permitted The function is not enabled for a local WinCC OPC UA server. Linking of WinCC messages with Event Notifiers of a local WinCC OPC UA server can lead to a continuous loop of the Alarm Logging in the case of unfavorable configuration.

OPC Event Notifier in WinCC Alarm Logging

As soon as you create a connection under the "OPC UA WinCC Channel" communication channel, the "OPC messages" entry is created in the "Alarm Logging" editor.

The created connections are listed under the entry.

You can load the Event Notifiers of the connected OPC UA server into the "Monitored objects" view.

Triggering WinCC messages

To display the alarms that an Event Notifier triggers in the WinCC project, link the Event Notifier with a WinCC message.

The WinCC message is then triggered by all alarms that are triggered by the Event Notifier as well as its hierarchically subordinate nodes. This means that the number of messages can increase significantly.

You can use filters to determine which OPC UA alarms trigger the WinCC message.

This reduces the number of triggered messages and only accepts the alarms for relevant events.

Note

System performance: Avoid Event Notifier "Server"

When you link a WinCC message with the higher-level Event Notifier "server object", this can result in a large number of messages.

Even if you use a filter that reduces the number of OPC UA events, this procedure can have a negative impact on the performance.

Assigning WinCC messages

You link an Event Notifier and a WinCC message through the message number.

If the message number has already been created in Alarm Logging, this message is linked. Otherwise a message with the specified number is created in Alarm Logging.

You can link the same message with several Event Notifiers.

However, you can always use a WinCC message only for one OPC UA connection. If you have created several OPC UA connections, each connection uses different WinCC messages.

Assigning multiple WinCC messages

An Event Notifier can be linked to several WinCC messages.

When the alarms are triggered, the messages and filters are processed from top to bottom. The message number has no influence on the sequence in which the WinCC messages are triggered.

To change the order of the messages, collapse the filters and select "Move up" or "Move down" in the shortcut menu of the row.

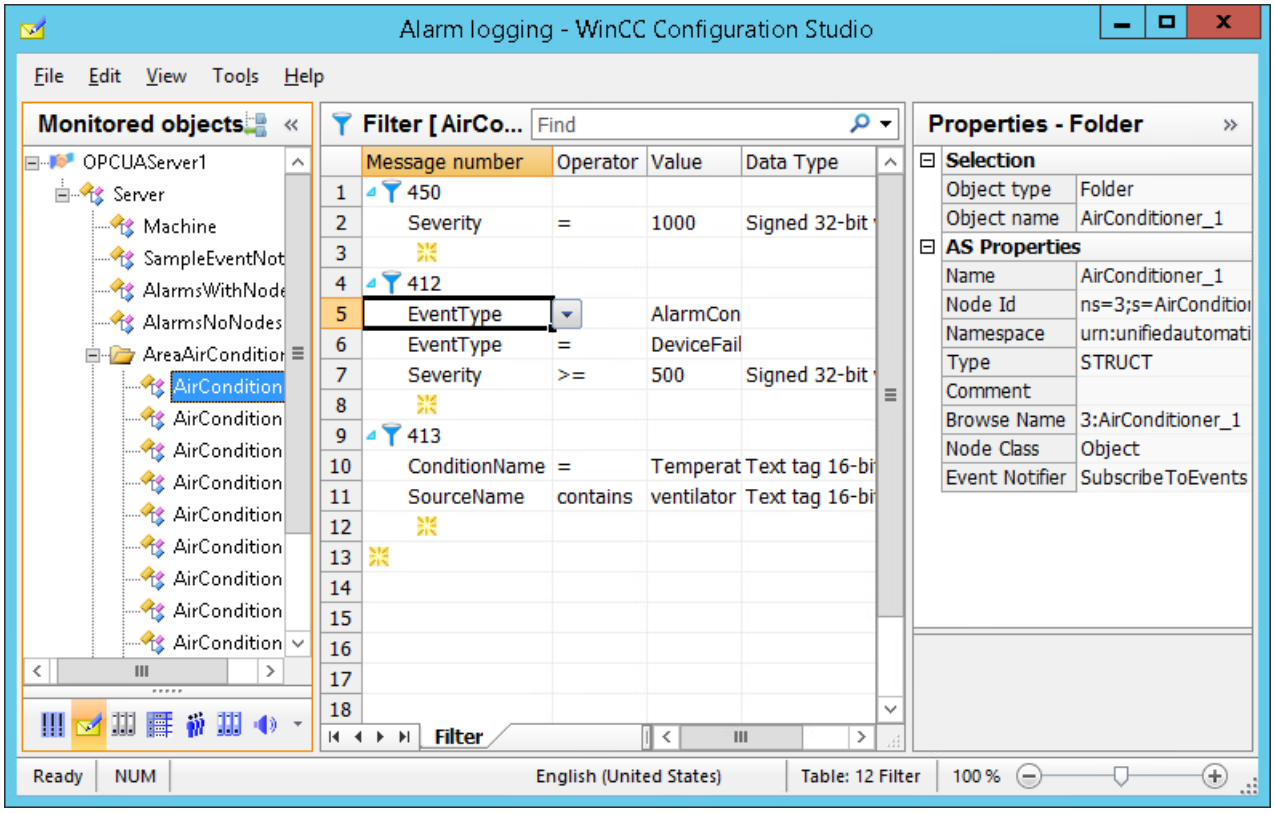
Display of the symbols

The representation of the data in the structure tree corresponds to the hierarchy on the OPC UA server.

On the "Filters" tab, you specify which WinCC messages are linked with an Event Notifier.

For every message, you can define one or more filters for the triggered alarms.

The properties of the selected Event Notifier are displayed in the "Properties - Folders" area. You can only change the properties on the OPC UA server.



Change View To

You use the following button to switch in Alarm Logging between the default view and the "Monitored objects" view:

The button is available only after the data records have been loaded.

After the WinCC Configuration Studio has been closed, the "Monitored objects" view is hidden again.

The "Monitored objects" tab remains visible in the default view "Alarm Logging".

Alarm Logging: "Assignments" tab

Click "OPC messages" to display the "Assignments" tab in the default view of Alarm Logging.

Here you configure the assignment rules for the attributes of the OPC UA alarms.

The configured rules apply to all OPC UA connections.

Configuring assignment rules

If triggered OPC UA alarms are linked with a WinCC message, their attributes are applied in the process value blocks 1 to 10.

For this purpose, configure the process value blocks as "Used" under "Message blocks" in Alarm Logging.

Process value block 1 always contains the message text of the OPC UA alarm.

For process value blocks 2 to 10 respectively, select the desired attribute from the drop-down list box.

The default rule "Default" cannot be changed.

Alarm Logging: "Monitored objects" tab

To display the "Monitored objects" tab in the default view of Alarm Logging, click the connection under "OPC messages".

The Event Notifiers linked with WinCC messages and their filters are displayed.

Assigning assignment rules

On this tab, you select the assignment rules of the Event Notifier. The "Default" rule is assigned by default.

The same assignment rule is used for all alarms of an Event Notifier and its hierarchically subordinate nodes.

Editing filters in the default view

You can also edit the filters on the "Monitored objects" tab.

The changed filters are used in the "Monitored objects" view on the "Filters" tab.

In the default view, however, the filter criteria are not checked for consistency and correct input.

Test the changed filters and correct the filters if necessary in the "Monitored objects" view.

See also

How to import Event Notifiers as WinCC messages (Page 143)

Filters for the OPC UA alarms (Page 138)

7.4.9.2 Filters for the OPC UA alarms

Filtering OPC UA alarms

You can specify one or more filters for each WinCC message that you link with an Event Notifier.

An Event Notifier triggers multiple alarms or events, of which usually only some are required for WinCC messages.

With the filters you reduce the triggered messages to relevant events.

Note

System performance: Avoid Event Notifier "Server"

When you link a WinCC message with the higher-level Event Notifier "server object", this can result in a large number of messages.

Even if you use a filter that reduces the number of OPC UA events, this procedure can have a negative impact on the performance.

WinCC messages: Unique assignment

Make sure you define specific filters that assign OPC UA alarms or events and WinCC messages as clearly as possible.

Configure WinCC messages in such a way that they map the properties of the alarms or events, e.g. acknowledgment philosophy and alarm source (Source).

Configuring filters

You can use filters to define which OPC UA alarms or OPC UA events trigger the WinCC message.

You can link a message with several Event Notifiers, but filter them by different alarms.

You configure the filters in the "Monitored objects" view on the "Filters" tab.

To create a filter, click the arrow in front of the message number. In the row displayed, select the filter criterion, the operator and the value.

The data type for the filter criterion is automatically added and cannot be changed.

Online configuration

If you change filters in Runtime, they are applied immediately.

Filter criteria and operators

With the filter criteria, you can determine which conditions the alarms of the Event Notifier have to fulfill so that they can trigger the linked message.

The operators depend on the selected filter criterion.

Filter criterion	Operators	Description
EventType	=	Drop-down list of types "BaseEventType" value: • Returns all OPC UA alarms or OPC UA events (no filtering).
ConditionName SourceName	= contains	Free text input Take the filter rules from the following table into account.

Filter criterion	Operators		Description
Severity	=	Equal to	Numerical input Value range: • 1 to 1000 Mapping in WinCC messages: • Priority 0 = Severity 1 • Priority 1 to 15 = linear interpolation between 0 and 1000 • Priority 16 = Severity 1000 Example "between": • 100, 200 Corresponds to Severity from 100 to 200 (Including the specified value respectively)
	!=	Not equal to	
	>	Is greater than	
	<	Is less than	
	>=	Greater than or equal to	
	<=	Less than or equal to	
	be-tween	Range from, to	

Filter rules for "contains" operator

The following rules apply when using the "contains" operator:

Charac-ter	Use	Example – Filter input	Example – Result
%	Corresponds to a character string of 0 or more characters. If a filtered text is to contain a "%" character, place the character in square brackets.	`%en%` "%en%" filters for all character strings that contain the letter sequence "en".	"green", "entail", "content"
		`5[%]`	"5%"
_	Corresponds to an individual character in a character string.	`_ould`	"would", "could"
\	Escape character to interpret the following special characters as literal characters: `\, *, +, ?, |, {, [, (,), ^, $,., #,space` Put a special character in front of "\", to prevent the interpretation as part of the operator.	`Plant\\Plant@2\+/Equipment1`	Path "Plant\Plant@2+/Equipment1"
[]	Contains a group of characters, one of which must be present in the character string.	`a[123]b`	"a1b", "a2b", "a3b"
		`ab[x-z]`	"abx", "aby", "abz"
[^]	Contains a group of characters which many not occur in the character string. The group is prefixed with "^".	`a[^123]`	Excluded: • "a1b", "a2b", "a3b" However, the result contains e.g: • "a4b"

Combining filter criteria

You can combine the filter criteria for a filter or use the same filter criteria multiple times:

- Different filter criteria are linked with "AND".
- Same filter criteria are linked with "OR".
- There is no filter hierarchy.
  The order of the entered filter criteria has no influence on the application of the filter.

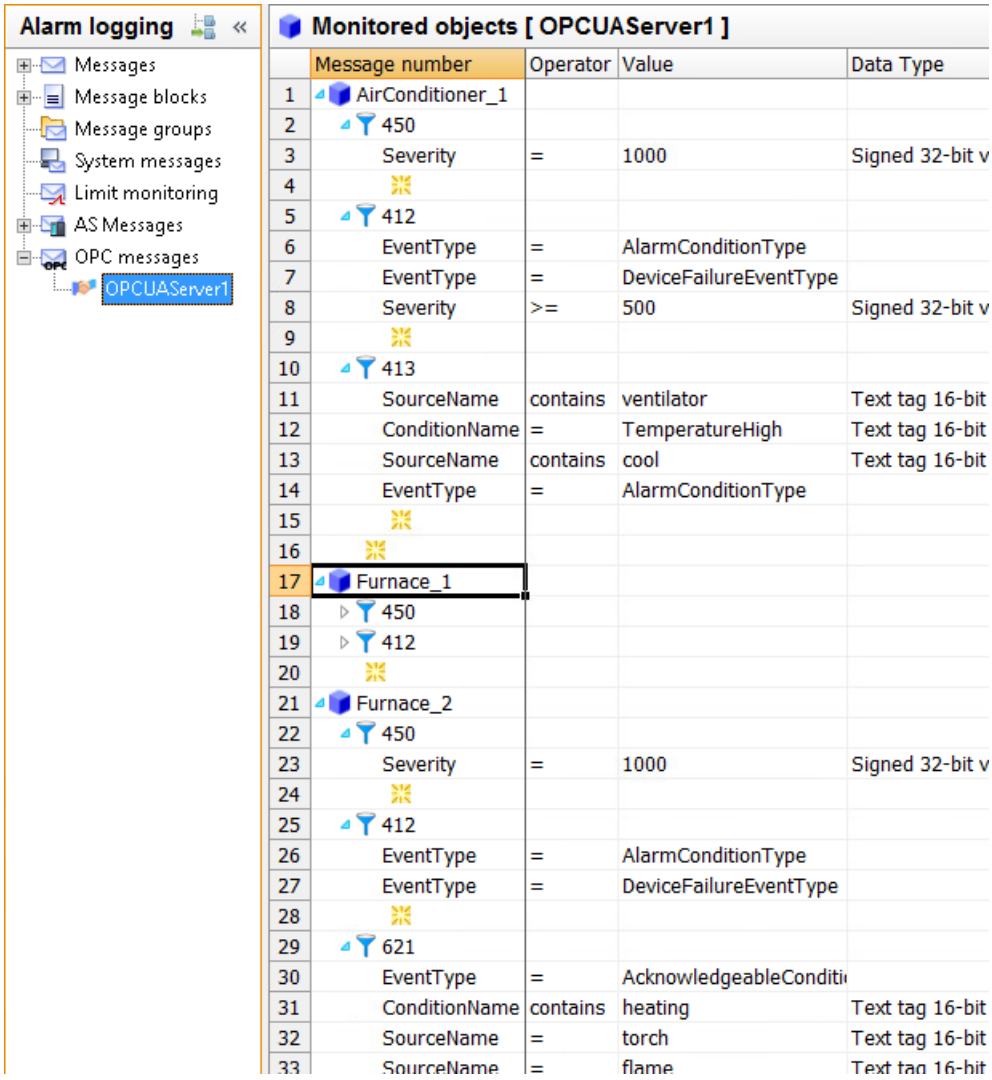
Example: "Alarm Logging" view

In the default view of the Alarm Logging, the Event Notifiers that are linked with a WinCC message are listed for each connection.

Under the node of an Event Notifier, the message numbers are displayed with the filters below them in each case.

You can also use this view to synchronize the filters of multiple Event Notifiers with each other.

In this example, you see the configured Event Notifier of the "OPCUAServer1" connection:



Filter example

- EventType = AlarmConditionType
- EventType = DeviceFailureEventType
- Severity >= 500

The example corresponds to the following condition:

- (EventType=AlarmConditionType OR EventType=DeviceFailureEventType) AND Severity>=500

See also

The "Monitored objects" view (Page 135)

How to import Event Notifiers as WinCC messages (Page 143)

7.4.9.3 How to import Event Notifiers as WinCC messages

Introduction

This section shows you how to connect Event Notifiers of an OPC UA server with WinCC messages.

OPC UA alarms of the Event Notifiers trigger the messages in the WinCC Alarm Logging in Runtime and can be archived as well as displayed in the WinCC AlarmControl.

Overview: Basic procedure

1. Load OPC UA Event Notifier in WinCC Alarm Logging
2. Link Event Notifier with WinCC message numbers
3. Optional: Determine filters for the triggered alarms of the Event Notifier
4. Optional: Define assignment rule
5. Specify assignment rule for each Event Notifier
6. Optional: Configure the properties, display and archiving of WinCC messages

---

Note

System performance: Avoid Event Notifier "Server"

When you link a WinCC message with the higher-level Event Notifier "server object", this can result in a large number of messages.

Even if you use a filter that reduces the number of OPC UA events, this procedure can have a negative impact on the performance.

WinCC messages: Unique assignment

Make sure you define specific filters that assign OPC UA alarms or events and WinCC messages as clearly as possible.

A WinCC message should be configured in such a way that it maps the properties of the alarms or events, e.g. acknowledgment philosophy and message source (Source).

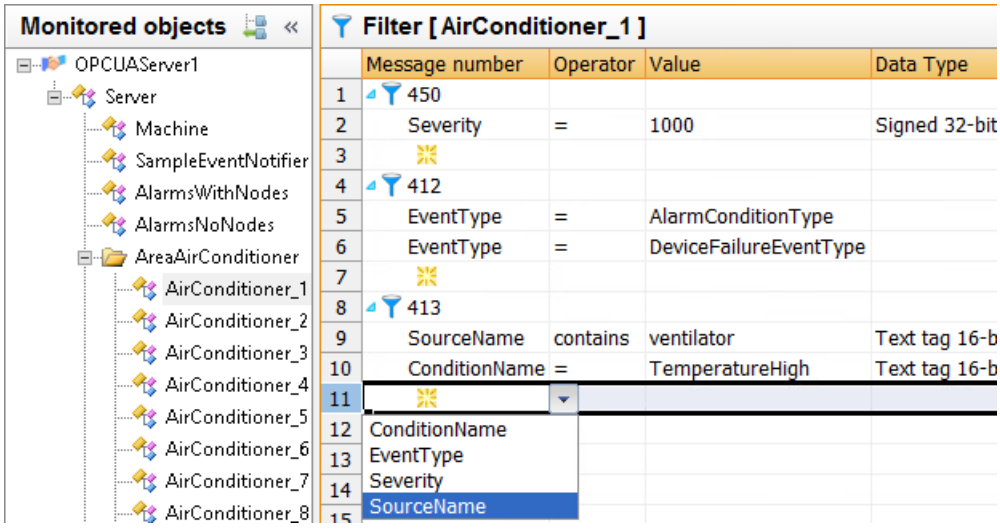
NOTICE
A local WinCC OPC UA server is not permitted The function is not enabled for a local WinCC OPC UA server. Linking of WinCC messages with Event Notifiers of a local WinCC OPC UA server can lead to a continuous loop of the Alarm Logging in the case of unfavorable configuration.

Requirements

- The connection to the OPC UA Server is established.
- The process value blocks are activated for usage.

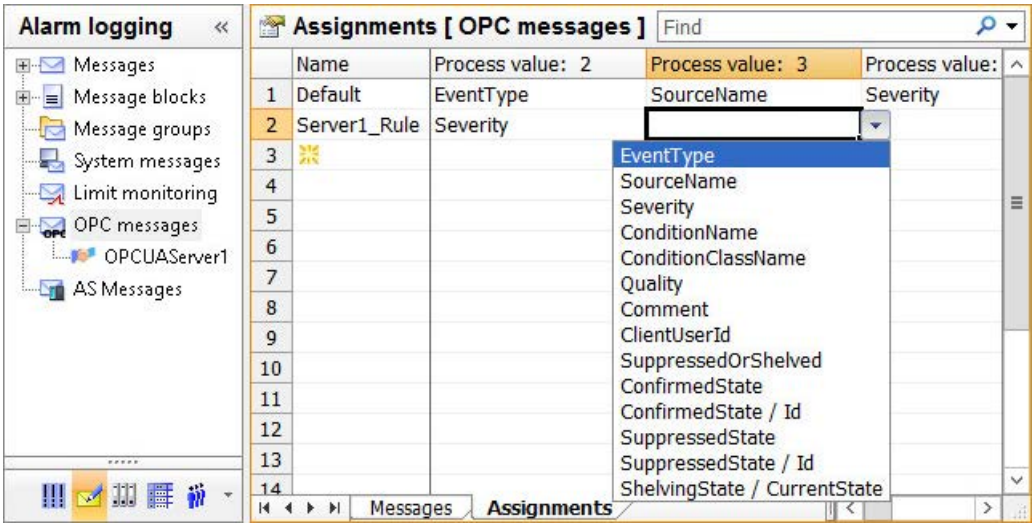
Procedure

1. Select the configured connection in the "Alarm Logging" editor under "OPC messages".
2. Select the "Browse OPC server" entry from the shortcut menu of the OPC UA connection.
   The available data of the OPC UA server is loaded. The "Monitored objects" view opens.
   In the navigation area, the Event Notifiers are displayed under the connection name.
3. Select an Event Notifier in the navigation area.
4. Enter one or more WinCC message numbers in the data area.
   Avoid the link on the highest hierarchy level "Server" since all subordinate Event Notifiers also trigger the linked message. A large number of triggered messages can have a negative effect on the performance.



5. To define a filter, click the arrow in front of the message number.
   In the row displayed, select the filter criterion, the operator and the value.
   The data type for the filter criterion is automatically added and cannot be changed.
6. To edit the messages in the "Alarm Logging" view, click the following symbol:
7. To display the assignment rules, click on "OPC messages" and select the "Assignments" tab.

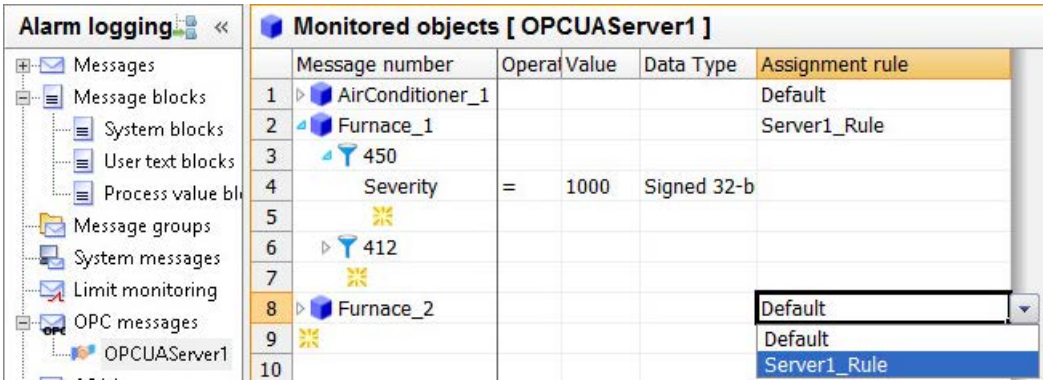
8. Enter a new rule name in the "Name" field under the "Default" rule.



9. Select the desired attribute of the Event Notifier in each case from the drop-down list of the process value fields.
   The attribute is linked with the corresponding process value block.

10. Click the connection name under "OPC messages".
    The messages and filters of the OPC UA connection are displayed.

11. Select the respective assignment rules for the Event Notifiers.
    The "Default" rule is linked by default.



12. To display the WinCC messages, click on "OPC messages".
    The messages and their properties are displayed on the "Messages" tab.

13. Configure the properties of the WinCC messages, for example the message class, type of message, archiving.
    To access the content of a process value block in a user text block, use the "@1%" format.
    You can find additional information on process value blocks in the WinCC Information System under "Working with WinCC > Setting up a message system > Configuring the message system > Working with messages":
    - "How to specify the text of a message"
    - "How to insert process values in user text blocks"

See also

The "Monitored objects" view (Page 135)
Filters for the OPC UA alarms (Page 138)

7.5 OPC UA Arrays in the OPC UA WinCC Channel

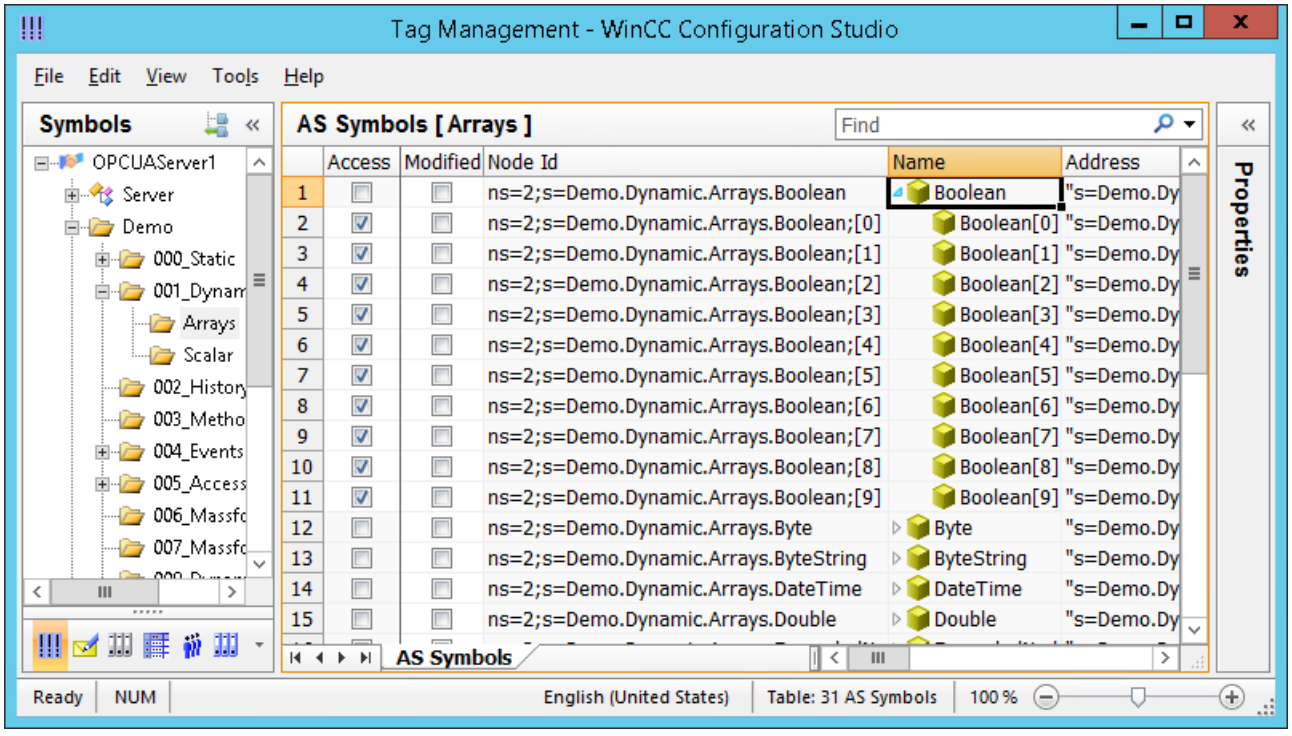
7.5.1 Arrays in WinCC

WinCC supports the configuration of OPC UA Arrays. In this context a WinCC tag can only correspond to a single Array element. This means it is not possible to map a complete OPC UA Array value with only one WinCC tag.

Representation of Arrays in the WinCC Configuration Studio

Arrays are represented as expanding data entries in the “AS symbols” working area in the WinCC Configuration Studio. The Array type and dimensions are displayed in the OPC UA DataType column.

You obtain access to the individual Array elements by clicking the respective arrow symbol in the "Name" column.



In order to map the Array elements as WinCC tags activate the check box in the “Access” column.

Data types

WinCC supports the following OPC UA Array data types with any number of dimensions:

- Boolean
- SByte

- Byte
- Int16
- UInt16
- Int32
- UInt32
- Float
- Double
- String
- DateTime
- ByteString

Note

If a server outputs an Array data type that is not supported, no tag can be configured in the WinCC Configuration Studio.

Write Array entries without IndexRange

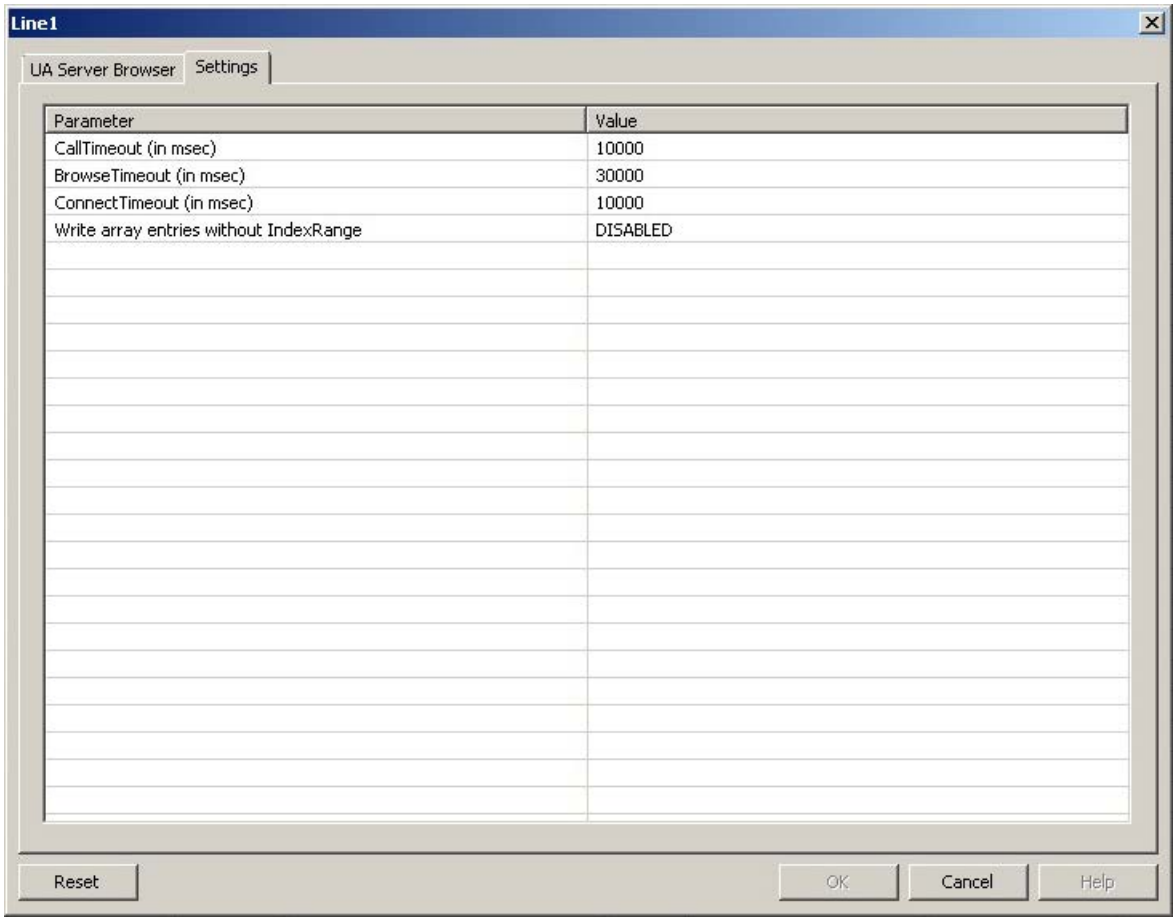
When reading and writing an OPC UA Array the OPC UA Client can access the entire Array or only a section of it by the IndexRange function specified in the OPC-UA norm being used.

The client can, for example, only read or write Elements 3 to 10 of an Array by specifying "3,10" as a index range. Or only the Array element 5 is read and written by specifying "5" as the IndexRange.

OPC UA Servers that agree with the OPC UA specification should support IndexRange when reading Arrays. On the other hand the writing of Arrays with IndexRange could not be supported.

WinCC will always attempt to read or write Array elements with IndexRange. If Array elements cannot be read with IndexRange, WinCC tries to read the complete Array and to extract the Array elements configured in WinCC. If Array elements cannot be written with IndexRange, WinCC uses the fallback strategy “Write Array entries without IndexRange". Since this can result in data inconsistencies, it is deactivated by default.

The “Write Array entries without IndexRange” function is activated in the “Settings” tab of the “Connection parameters” dialog.



NOTICE
Data inconsistency The activation of this function can result in data inconsistency!

Restrictions

Data consistency

In Runtime no guarantee can be given for the data consistency of the OPC UA Array values configured in WinCC.

Array data can be inconsistent in particular in the following cases:

- When reading several Array elements in different read cycles.
- When reading and updating complete arrays (see section “Write Array entries without IndexRange”).

Performance

Performance problems can arise when reading several elements of an Array if the OPC Server does not support IndexRange, because WinCC has to read the complete Array value and extract the configured Array elements.

Dynamic Arrays

The length of an OPC-UA Array can be dynamic and can be modified in the Runtime. For example, WinCC tag values can be of bad quality if an Array value is shorter than expected.

Compatibility

Some OPC UA servers are not fully compatible with the OPC UA standard so that Tag Management may not be able to display array elements. In this case you have to create the Array tags manually and modify the Array address.

You therefore require server-specific knowledge in order to be able to configure Arrays for such Servers.

Further information about manual configuration is available in the Section How to create and configure OPC UA Arrays manually (Page 150).

7.5.2 How to create and configure OPC UA Arrays manually

If an OPC UA Server does not display the Arrays in accordance with the norm, it is possible that WinCC is not able to display the Array elements of an OPC UA Array. In this case you have to create and configure the Array elements manually in the WinCC tag management. You require Server-specific knowledge to this purpose.

Requirement

- The connection to the OPC UA Server is established.
- Tag management is open.

Procedure

1. Create a new tag by copying and inserting an existing tag via the shortcut menu in the "Tags" working area.

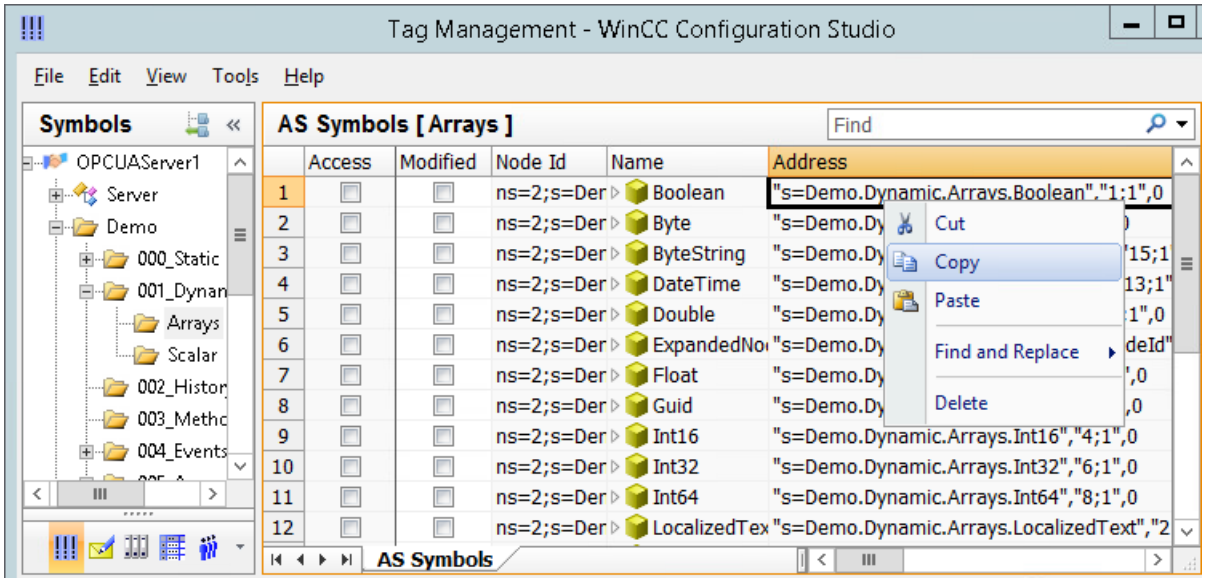
   Note

   Copying and inserting tags

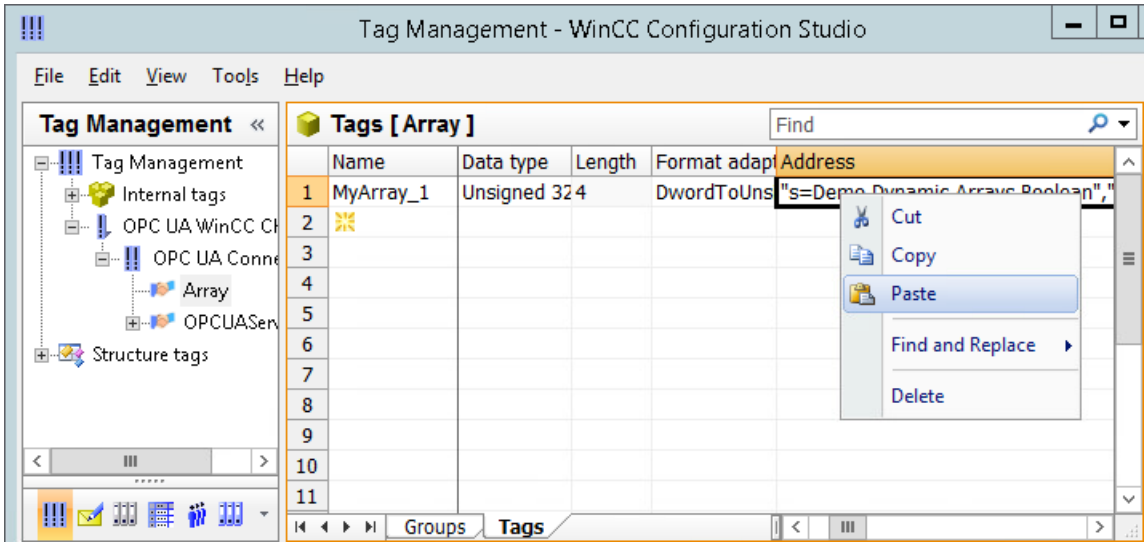
   In order to copy tags with all the properties, the complete row has to be marked and not only the tag name. To this purpose click the preceding number.

2. Rename the newly created tag, if applicable.

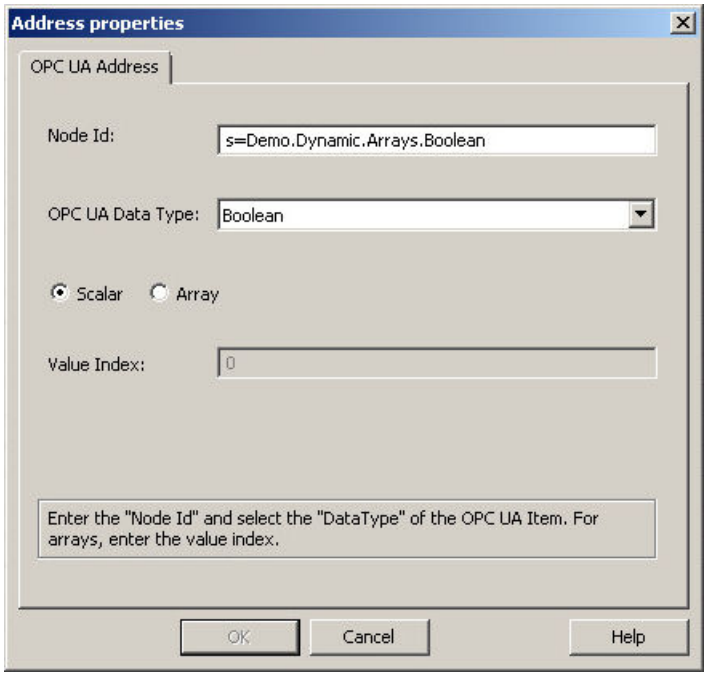
3. Copy the address of the corresponding Array value in the "AS symbols" working area into the Clipboard.



4. Insert the address of the corresponding Array value in the "AS symbols" working area into the "Address" column of the copied tag.



5. Open the "Address properties" dialog in the properties of the tag by using the ... button.



6. Carry out the required settings and confirm with "OK".

7.6 OPC UA methods in the OPC UA WinCC Channel

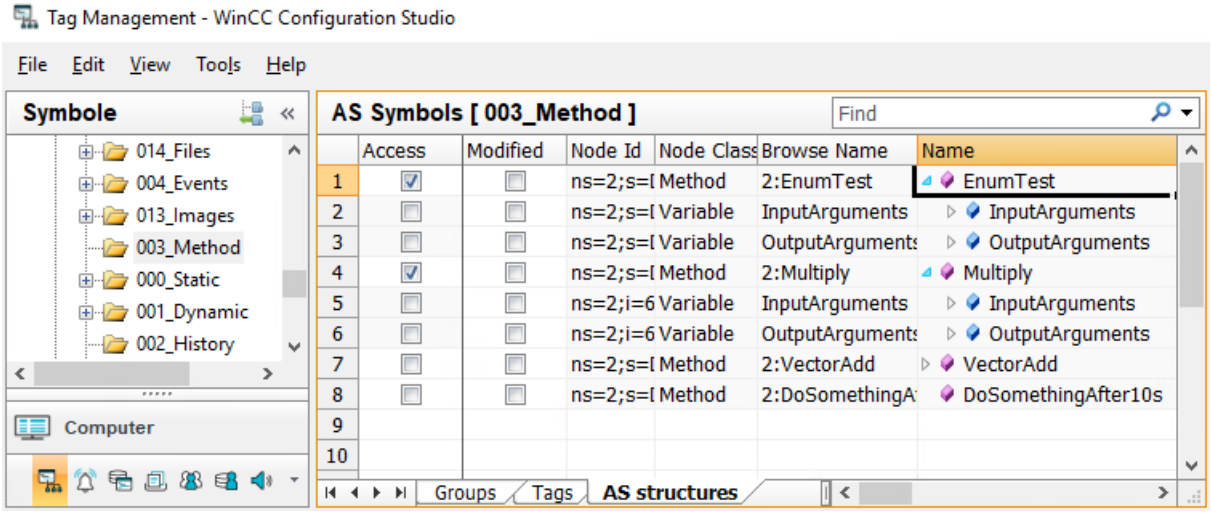
7.6.1 OPC UA methods in WinCC

WinCC supports the usage of OPC UA methods. The OPC UA methods and the associated optional input parameters and optional return parameters are available in the WinCC Configuration Studio. OPC UA methods are executed consecutively.

Representation of OPC UA methods in the WinCC Configuration Studio

Methods are represented in the "AS symbols" working area of WinCC Configuration Studio.

The individual input parameters and return parameters can be accessed by clicking the respective arrow symbol in the "Name" column.



In order to use the methods in the Global Script Editor (VBScript), activate the check box in the "Access" column.

Note

If you carry out changes to the "Access" column, you have to update the channel methods in the VBScript Editor.

Data types of input parameters and return parameters

WinCC supports the following data types for input parameters and return parameters of OPC UA methods:

- Boolean
- SByte
- Byte

- Int16
- UInt16
- Int32
- UInt32
- Float
- Double
- String
- DateTime

Format conversions from non-supported to supported data types are not permissible.

Note

If an unsupported data type is used, the corresponding method cannot be executed.

If there is more than one return parameter, they are returned as an array.

Naming restrictions

Connection name

Keep the following restrictions in mind when naming the connection:

- The first character must be a letter.
- Only letters contained in the English alphabet are allowed.
- Spaces are not allowed.
- The following special characters are not allowed: . ! @ & $ # \ / : * ? " < > |
- Reserved keywords of Windows devices are not allowed, e.g.: CON, PRN

Method name

Observe the following restrictions when naming methods:

- Spaces are not allowed.
- Only letters contained in the English alphabet are allowed.
- The following special characters are not allowed: . ! @ & $ # \ / : * ? " < > |

If the method name contains disallowed characters, they are removed during import.

If the method name consists exclusively of disallowed characters, the name is applied during import.

Input parameters and return parameters

During import, the prefix "ip_" is added to input parameters and the prefix "op_" to return parameters.

Observe the following restrictions when naming input parameters and return parameters:

- Spaces are not allowed.
- Only letters contained in the English alphabet are allowed.
- The following special characters are not allowed: . ! @ & $ # \ / : * ? " < > |

If the method name contains disallowed characters, they are removed during import and the corresponding prefix is added.

If the method name consists exclusively of disallowed characters, the name is applied during import and the corresponding prefix is added.

Note

The combination of connection name and method name must be less than 255 characters.

Only correctly named methods, input parameters, and return parameters can be used in VB script.

7.6.2 How to use OPC UA methods in client projects

You can use OPC UA methods in server and client projects.

If you use OPC UA methods of a server project in client projects, you must make changes in the project folder of the client.

The required data are transferred from the computer of the server project to the computer of the client project.

Requirement

- A server project is created and uses OPC UA methods.
- A client project is created.

Procedure

1. Open the WinCC project folder of the server project in the file explorer.
2. Change to the subfolder "ScriptLib".
3. Copy the files with the matching name of the OPC UA connection with the extension ".bmo" to the "ScriptLib" subfolder of the client project.
4. Change to the "OPC" subfolder of the server project.
5. Copy the file "OPCUaMethods.xml".
6. Create the "OPC" subfolder if it does not already exist in the client project.
7. Insert the file "OPCUaMethods.xml" in the subfolder "OPC" of the client project.

7.7 Error handling

7.7.1 Error Handling in the Event of Disturbed OPC Communication

Introduction

The procedure for communication testing is independent of how WinCC is used.

You can find more information about channel diagnostic under "Communication > Communication Diagnostics".

WinCC used as OPC UA server

Use the channel diagnostics on the WinCC OPC UA client to check whether a connection can be established to the WinCC OPC UA server.

WinCC used as OPC UA client

Use the channel diagnostics on the WinCC OPC UA client to check whether a connection can be established to the WinCC OPC UA server.

PROFIBUS DP

8

8.1 WinCC Channel "PROFIBUS DP"

Contents

The "PROFIBUS DP" channel is used for communication between a WinCC station as PROFIBUS DP-Master and the corresponding periphery assemblies, for e.g. ET200.

Communication uses the PROFIBUS DP protocol.

This chapter informs you about the following topics:

- How to configure data transfer with the "PROFIBUS DP" channel
- How to configure a connection and a tag

Changes in the current version of PROFIBUS DP

Compared to the supplied documentation, the following change occurs in the current version of PROFIBUS DP:

- PROFIBUS DP Master is used as Application OPC Server.

8.2 Properties of the WinCC driver Profibus DP

Properties

The WinCC driver Profibus DP, has the following properties:

- The WinCC PC with the communications processor (= CP) is DP master on the Profibus.
- All DP standard slaves can be addressed.
- Up to four CP cards can be initialized and configured with maximum 123 DP slave stations per CP module.
  The limit values can be changed with newer versions and should therefore be checked before commissioning.

Communication to other Profibus bus partners is possible using other protocols, provided the driver allows this.

Note

Only one DP master

No other DP master which addresses the same slaves may be connected on the Profibus bus.

8.3 Integrating the "Profibus DP" driver

Standards

Based on the Profibus distributed I/O (DP) standards:

- DIN 19245-3, or according to
- pr EN 50170

Requirement

Hardware:

- To use WinCC driver Profibus DP you require a communications processor CP 5613 (A3) or CP 5612 for the connection of the Profibus
  All DP standard slaves can be addressed with this.
- The number of communications processors used depends on the still free interrupts in the PC.

Software:

- To install and configure the communications processor you require the driver and configuration software.
  You can find this on the SIMATIC NET CD.

Procedure

1. Select the "Profibus DP" communication driver in the navigation area of the tag management.
   The channel units are created.
2. In the channel unit shortcut menu, select "System parameters".
   The configuration dialog opens.
3. Specify the CP board number and the monitoring time.
4. Select the entry "New Connection" in the shortcut menu of the channel unit.
5. Enter the name of the connection.
6. Select the entry "Connection parameters" from the shortcut menu of the connection.
   The configuration dialog opens.
7. Select the slave address and confirm with OK.

8.4 Configuring the "Profibus DP" driver

Setting the system parameters

CP board number

Number of the CP card in the PC (from configuration tool).

Value range:

- 1 to 4
- 0 = not installed

Watchdog time

Input of a factor for the monitoring time of WinCC on the communications card. The monitoring time is a multiple of 0.4 seconds in each case.

This function is only valid for the slaves that can be supplied with output data.

- Input 0:
  Monitoring is disabled.
- Input > 0:
  If no further write access occurs the outputs are set to 0 when the time elapses.
  This must be ensured by a suitable WinCC configuration.

Settings the connection parameters

Slave address

Address of the slave that is to be read or written.

Value range:

- 1 to 127

Setting the tag address

To configure the tag address, click in the empty "Address" field in the "Properties - Tag" window.

Open the configuration dialog using the following symbol: ...

Note

Performance of the connection

If power and throughput are impaired, note the following:

The update time for interconnecting a tag affects the connection, as access is only possible to the entire DP device in Profibus.

Properties of the process tags

Field	Meaning
Input	Input range of the slave
Output	Output range of the slave
Length (bits)	Display of the tag size in bits The value is based on the previously selected data type. Exception: Raw data tag
Byte offset	Number of bytes after which the content of the tag is stored Value range: 0 to length -1
Bit offset	Only active with "Binary tag" data type Number of the bit in the above specified byte in which the binary tag is entered Value range: 0 to 7 Larger values are possible provided the length of the buffer is not exceeded.
Changing the byte arrangement	Deactivated: Little Endian (default setting) Activated: Big Endian

Properties of raw data tags

Field	Meaning
Input	Input range of the slave
Output	Output range of the slave
Length (bits)	Not active
Byte offset	Number of bytes after which the content of the tag is stored Value range: 0 to length -1
Bit offset	Not active
Length (bytes)	Enter the required block length in this field. The length unit for this field is byte.
Send/receive block	The defined data block is sent or received after request from WinCC.

S5 Ethernet Layer 4

9

9.1 WinCC Channel "SIMATIC S5 Ethernet Layer 4"

Introduction

The communication driver is used e.g. to connect automation systems SIMATIC S5-115U/H, SIMATIC S5-135U and SIMATIC S5-155U/H with the ISO transport protocol or the TCP/IP protocol.

Depending on the communication protocol that is used, the following communication partners will be implemented:

Communication protocol	WinCC side	SIMATIC S5 side
ISO transport protocol	CP1612 A2 (3Com compatible) CP1613 A2 CP1623	CP1430 TF
TCP/IP (conforming with RFC1006)	CP1612 A2 (3Com compatible) CP1613 A2 CP1623	CP1430 TCP

When using this channel, no local database is required.

Channel units

The communication driver has two channel units "CP1413-x" with which a maximum of two channel units CP1612 A2, CP1613 A2 or CP1623 can be operated. The functionality of the channel unit is identical. They differ only in the logical device names of the two CPs. CP1623 is identical to CP1613 A2, but is operated via PCI Express.

Communication can be established via the TCP/IP protocol with a CP1612 A2, CP1613 A2 or CP1623 using the third channel unit "TCP/IP".

The logical device name can be changed in the system parameters of a channel unit. Here, it is also possible to set the parameters for the protocol used.

The following application capabilities exist:

- Channel unit "S5-Transport (CP 1413-1)" for the communication modules for SIMATIC Industrial Ethernet (CP 1612 A2 / 1613 A2 / 1623).
- Channel unit "S5-Transport (CP 1413-2)" for the communication modules for SIMATIC Industrial Ethernet (CP 1612 A2 / 1613 A2 / 1623).
- Channel unit "S5 Transport (TCP/IP)" for the communication modules for SIMATIC Industrial Ethernet (CP 1612 A2 / 1613 A2 / 1623).

See also

System parameters of the channel unit (Page 175)

Configuring the tags (Page 168)

Configuring the channel "SIMATIC S5 Ethernet Layer 4" (Page 166)

9.2 Data type of the tags

Introduction

Define the required tags for a logical connection. From the WinCC viewpoint, you can access the following data types:

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

9.3 Configuring the Channel

9.3.1 Configuring the channel "SIMATIC S5 Ethernet Layer 4"

Introduction

The following steps are required for configuring the channel "SIMATIC S5 Ethernet Layer 4".

1. Configuring the connection
2. Configuring the tags
3. System parameter configuration

See also

How to configure the connection (Page 166)
System parameters of the channel unit (Page 175)
Configuring the tags (Page 168)

9.3.2 How to configure the connection

Introduction

The connection parameters are almost identical for all protocols used. In the following example, communication is described using the ISO transport protocol with a channel unit "CP1413-x".

When implementing the TCP/IP protocol, the IP address of the AS is entered instead of the Ethernet address. The IP address consists of four numerical values, separated by dots. The numerical values must be within the range of 0-255.

For a logical connection, WinCC establishes one connection in the transport layer for reading ("READ function" area) and one for writing ("WRITE function" area). The address parameters for both functions are defined in the dialog. Only if both connections are established is the logical connection also indicated as being "established".

Allocations for the READ function

WinCC side	SIMATIC S5 side
FETCH-Active (Request "READ-Active")	READ-Passive (Request "READ-Passive")
FETCH-Passive (Request "WRITE-Passive")	WRITE-Active (Request "WRITE-Active")

Note

It is not possible to write binary or byte variables in the data area of the AS, if the data from the AS is sent active, i.e. the READ function is set to "FETCH Passive" in the connections parameters.

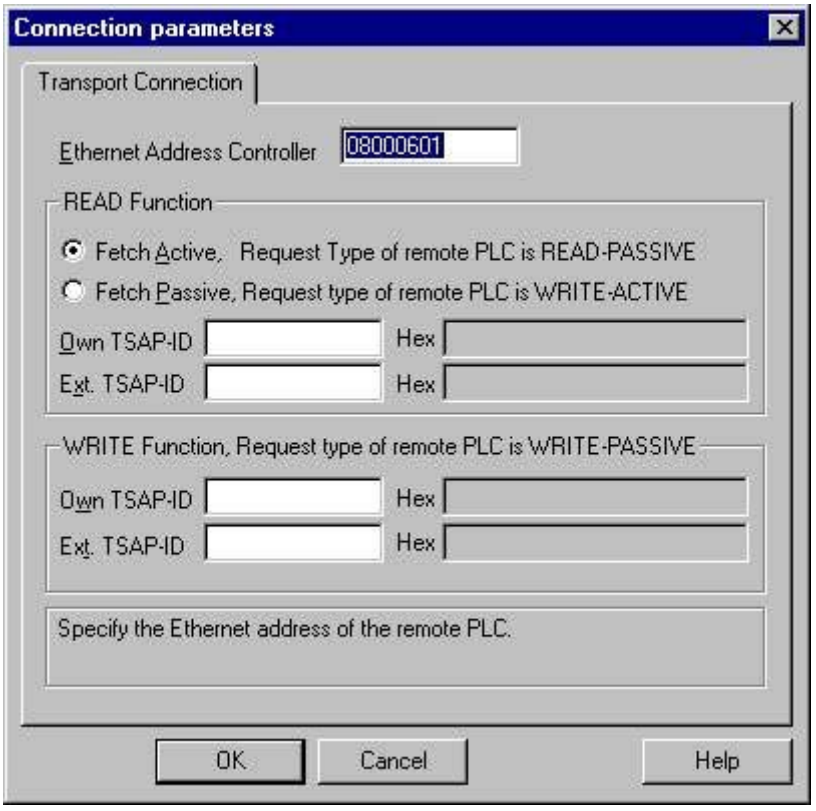
A FETCH Passive connections is only assigned the "OK" status if at least one telegram has been sent from AS to WinCC.

Allocations for the WRITE function

WinCC side	SIMATIC S5 side
Request "WRITE Active"	Request "WRITE Passive"

Procedure

1. Select the entry "Connection parameters" from the shortcut menu of the connection. The "Connection properties" dialog opens.



2. Enter the station address of the SIMATIC S5 on the industrial Ethernet bus in the field "Ethernet Address AG". When the TCP/IP protocol is being implemented, the IP address is entered here in the IP address AG" field.

3. Define the parameters for the READ function in the WinCC system. These are independent of the request used in the SIMATIC S5.
4. Then, enter the value in the allocated field "Own TSAP" that was configured in the "Remote parameter" as "TSAP" while configuring the CP1430 TF.
5. Now, enter the value in the allocated field "Remote TSAP" that was configured in the "Local parameter" as "TSAP" while configuring the CP1430 TF.
6. Define the parameters "Own TSAP" and "Remote TSAP" for the WRITE function accordingly.

Note

In the entries for "TSAP", you must not use any spaces.

See also

System parameters of the channel unit (Page 175)

Configuring the tags (Page 168)

9.3.3 Configuring the tags

9.3.3.1 Configuring the tags

Introduction

For a connection between WinCC and the AS via channel "SIMATIC S5 Ethernet Layer 4", tags of different data types can be created in WinCC. The following describes how to configure a tag of these data types.

- Addresses of tags
- Configuring a tag with bit by bit access
- Configuring a tag with byte by byte access
- Configuring a tag with word by word access
- Configuring a raw data tag

See also

How to configure a raw data tag (Page 174)

How to Configure a Tag with Word by Word Access (Page 172)

How to Configure a Tag with Byte by Byte Access (Page 171)

How to Configure a Tag with Bit by Bit Access (Page 170)

Addresses of tags (Page 169)

9.3.3.2 Addresses of tags

Introduction

The tag address is entered according to the address structure of the SIMATIC S5.

Depending on the tag type, the access to memory areas in the AS is bit by bit, byte by byte or word by word. For this purpose, the addressed memory area is read from the AS for every single write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the AS's memory.

Note

Changes that have been made by the AS in a read data area are overwritten when writing back into the data area.

It is not possible to write binary or byte variables in the data area of the AS, if the data from the AS is sent active, i.e. the READ function is set to "FETCH Passive" in the connections parameters.

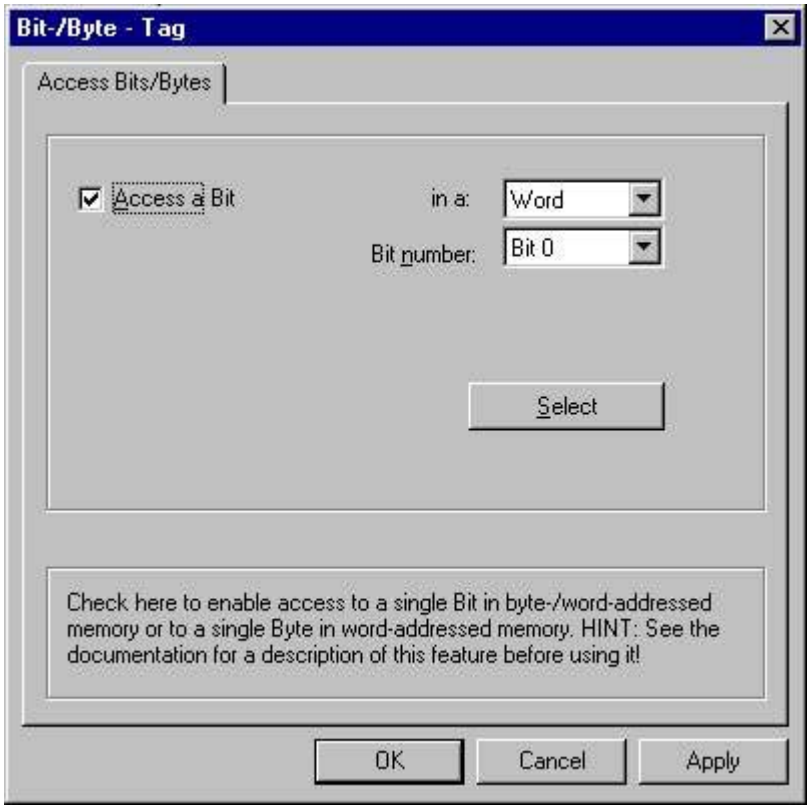
Configuring the address of a tag is done independent of the tag type:

- With tags of type "binary" or 8 bit value", first the "Bits-/Bytes-tag" dialog is opened, in which the bitwise or byte-wise access to the memory area of the AS is defined.
  Afterwards, the address of the tag in the AS memory is defined in the "Address properties" dialog.
- For word-oriented tags, the address of the tag in the AS memory is defined in the "Address properties" dialog.
  The dialog "Bits-/Bytes-tag" is not opened, since the access to the AS memory is word by word.

9.3.3.3 How to Configure a Tag with Bit by Bit Access

Procedure

1. Select the connection and open the dialog window "Bit/Byte tag" in the shortcut menu. For this purpose, click in the "Address" field and then on the ... button.
2. Click the "Select" button. The "Bit/Byte tag" dialog is opened.



3. Use the check box to define whether access should be enabled for reading and writing certain bits in the memory area.
4. Select the addressing methods for the AS memory in the selection field e.g. "Word" or "Byte".
5. Select the number of bits to be changed in the selection field.
6. Use the "Selection" button to open the "Address properties" dialog for defining the tag address in AS.

Note

With the S5, flags, inputs and outputs can be addressed byte by byte; data blocks (DB, DX) are addressed word by word.

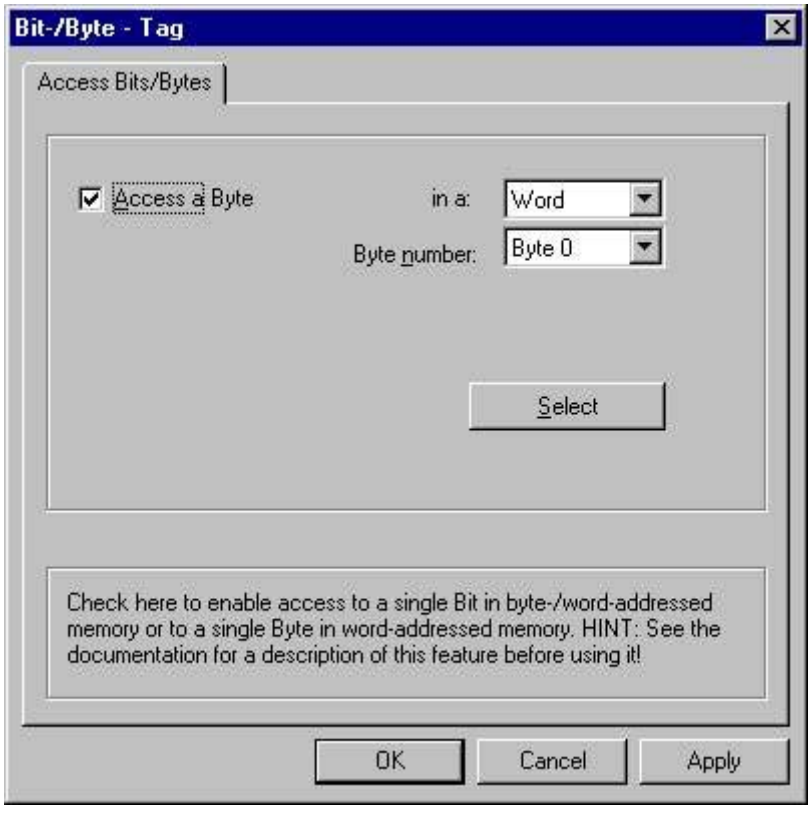
Activating the check box "Access a bit" affects the display of the fields of the "Address properties" dialog.

For word-oriented tags, the described "Bit-/Byte-tag" is not opened because the address of the tags and therefore the access to PLC memory is by word.

9.3.3.4 How to Configure a Tag with Byte by Byte Access

Procedure

1. Select the tag and select the data type "Unsigned 8-bit value" or "Signed 8-bit value" in the "Data type" field.
2. Select the connection and open the dialog window "Bit/Byte tag" in the shortcut menu. For this purpose, click in the "Address" field and then on the ... button.
3. Click the "Select" button. The "Bit/Byte tag" dialog is opened.



4. Use the check box to define whether access should be enabled for reading and writing certain bytes in the memory area.

5. Only "Word" is shown as the AS memory addressing type in the selection field.
6. Select the number of bytes to be changed in the selection field.
7. Use the "Selection" button to open the "Address properties" dialog for defining the tag address in AS.

Note

With the S5, flags, inputs and outputs can be addressed byte by byte; data blocks (DB, DX) are addressed word by word.

Selecting the check box "Access a byte" affects the display of the fields of the "Address properties" dialog.

For word-oriented tags, the described "Bit-/Byte-tag" is not opened because the address of the tags and therefore the access to PLC memory is by word.

9.3.3.5 How to Configure a Tag with Word by Word Access

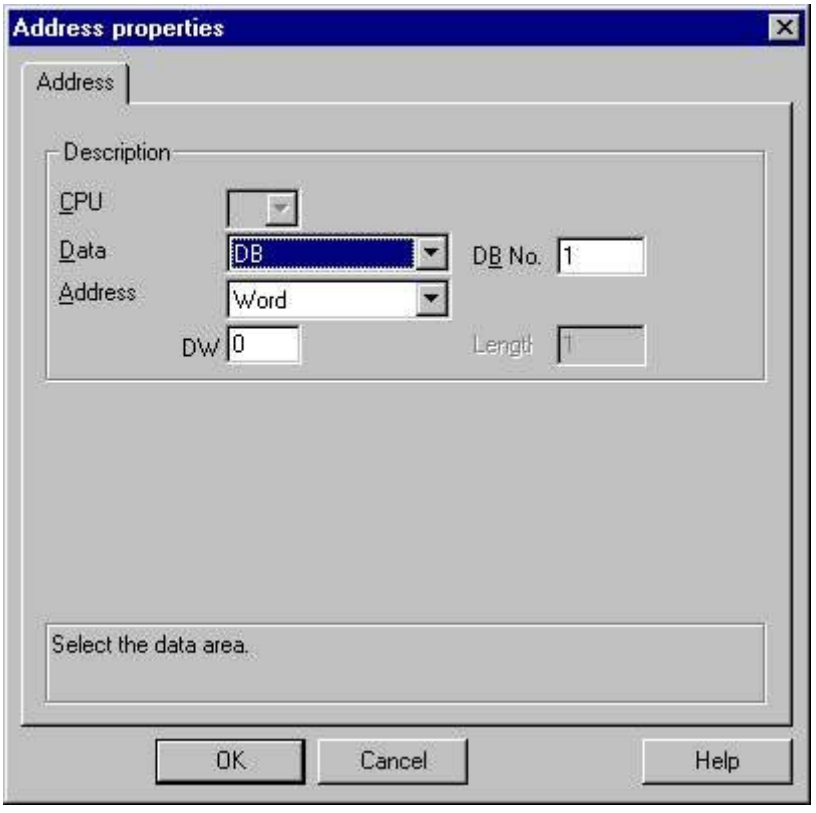
Introduction

The addresses of tags in AS are defined with the dialog that is described here.

- With tags of type "binary" or 8 bit value", first the "Bits-/Bytes-tag" dialog is opened, in which the bitwise or byte-wise access to the memory area of the AS is defined.
- For word-oriented tags, the "Bit-/Byte-tag" dialog is not opened because the address of the tags and therefore the access to AS memory is by word.

Procedure

1. Select the tag and select the required data type for the tags (e.g. signed 16-bit value) in the field "Data type".
2. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



3. Choose whether the tag is located in a data block, the flag area, an input area or an output area in the "Data Area" field of the "Address" tab.
4. If the tag is in a data block, the "DB No." field is also shown. Here, you enter the number of the data block.
5. The type of addressing is entered in the "Addressing" field. Normally, you can use the default definition.
6. Enter the address in the respective field (e.g. "DW ").

Note

For tags of type "binary" or "8 bit value", displaying the fields of this dialog depends on the selection made for "Access to bits/bytes" in the "Bits-/Bytes-tag" dialog.

If the tag of a word-oriented data area is to be written, the start address must be in the left byte and the length of the tags must be an even number.

9.3.3.6 How to configure a raw data tag

Introduction

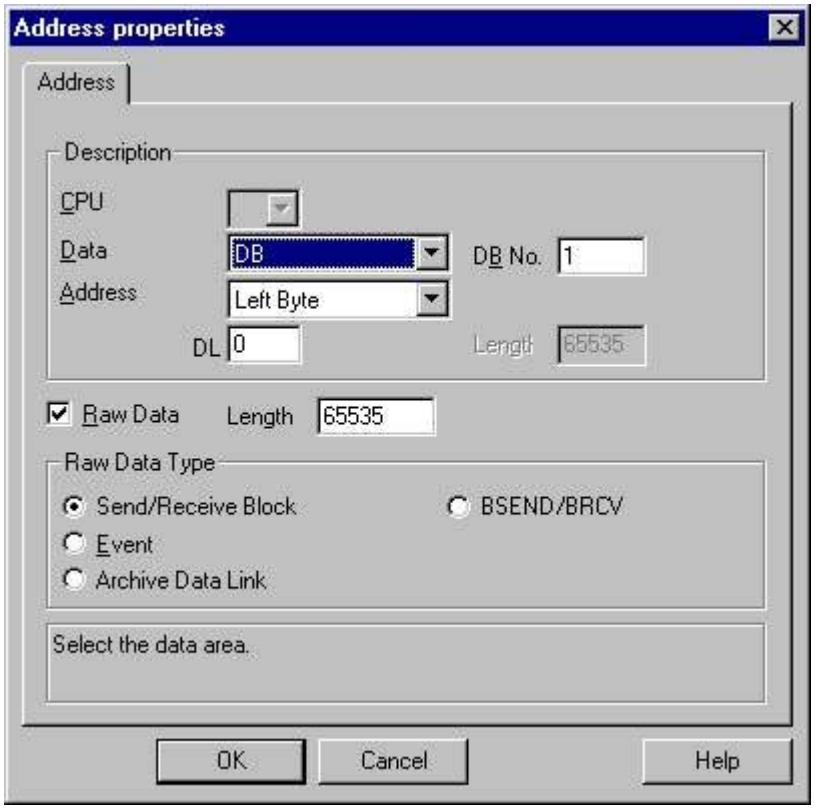
The following is a description of how the address of a raw data tag is defined.

Note

If the tag of a word-oriented data area is to be written, the start address must be in the left byte and the length of the tags must be an even number.

Procedure

1. Select the tag and select the entry "Raw data type" in the field "Data type".
2. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



3. Choose whether the tag is located in a data block, a flag, an input area or an output area in the "Data Area" field.
4. If the tag is in a data block, the "DB No." field is also shown. Here, you enter the number of the data block.
5. The type of addressing is entered in the "Addressing" field. Normally, the default definition can be used.

6. Enter the data address in the field underneath. The label on the field depends on the entry in the "Addressing" field, e.g. "DBW" for Word addressing.
7. Enter the length of the raw data block in bytes in the Length field.
8. Define which type of raw data tag is concerned in the "Raw data type" area.

9.3.4 System parameters

9.3.4.1 System parameters of the channel unit

Introduction

If you require a configuration that deviates from the standard WinCC settings, you can make all the required changes using the "System Parameter" dialog of the channel unit.

The system parameters are almost identical for all protocols used. When implementing the TCP/IP protocol, only the device name given during the installation is different.

The following individual points can be changed:

- the device name
- the transport parameter

Note

The system parameters apply for all CPs in the AS.

Device Name

Communication between WinCC and the automation system takes place via logical device names. These names are assigned during the installation of the communication module and are unit-specific. The device name represents the logical device name. The logical device name is given the name "/CP_H1_1:/ SCP" with the ISO transport protocol and "/TCP_IP:/SCP" with the TCP/IP protocol as a default definition.

Note

When using the TCP/IP protocol, you must check whether the device name in WinCC matches the "Access point of the application" in the "Set PG/PC interface" dialog. The device name must also be changed in "Set PG/PC interface".

Transport Parameter

Specific settings for the channel unit are made in the transport parameters, e.g. PDU size, setup attempts, etc.

See also

How to change the transport parameter (Page 178)

How to Change the Device Name (Page 176)

9.3.4.2 How to Change the Device Name

Introduction

Parameters of the channel unit are set with the system parameters, e.g. the logical device name or the transport parameters.

The system parameters are almost identical for all protocols used.

In the following example, communication is described using the ISO transport protocol with a channel unit "CP1413-x".

When implementing the TCP/IP protocol, only the device name given during the installation is different.

Procedure

1. Select the channel unit and open dialog window "System parameters" with the context menu.
2. Select the "Device Name" Tab.



3. Now, you can select the device name shown in bold print with the mouse and change it with a mouse click in the name field for the device name.

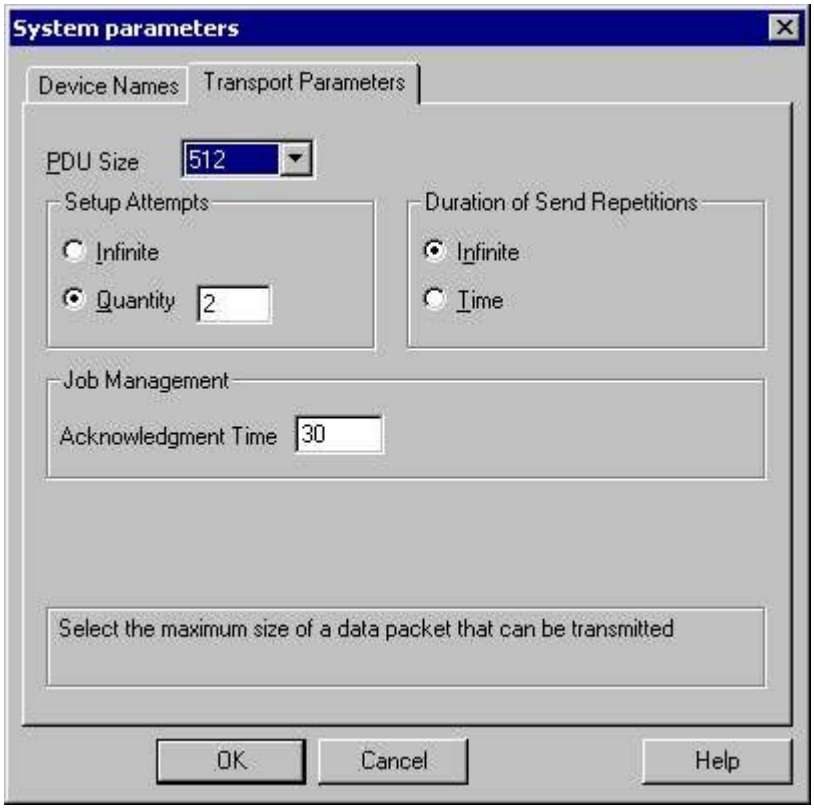
Note

The device name is defined during the installation of the hardware driver. Only if you have defined another name there, which is not recommended, will you have to change the device name here as well.

9.3.4.3 How to change the transport parameter

Procedure

1. Select the channel unit and open the "System parameters" dialog window with the shortcut menu.
2. Select the "Transport Parameters" tab.



3. Set the value for "PDU Size" to the value that was configured on the communication module CP 1430.
4. Define how often a connection establishment should be attempted in the "Setup Attempts" field.
5. Select "Infinite" in the "Duration of Send Repetitions" area.
6. In the "Acknowledgment Time" field, enter the value 30, for example, so that you are informed of the tag status after 30 seconds at the most, if the communication partner has not responded within this time (e.g. AS in "Stop" status).

9.4 Appendix

9.4.1 Appendix

Introduction

Added information on the channel "SIMATIC S5 Ethernet Layer 4" is provided in the appendix.

See also

Internal error codes and constants (Page 179)

9.4.2 Internal error codes and constants

9.4.2.1 Internal error codes and constants

Introduction

The following tables contain the most important error codes and constants. The information is intended for "insiders". Therefore, we have not gone into more detail on the meanings of the codes and constants.

- Error codes during connection disturbances
- iNA960 messages
- SCI messages

See also

SCI messages (Page 183)

iNA960 messages (Page 182)

Error codes during connection disturbances (Page 179)

9.4.2.2 Error codes during connection disturbances

Introduction

The most important error codes are listed in this section. If an error with an error code that is not in the table occurs, please call the WinCC hotline.

Fehler_0002-INVALID_RQ

Faulty request block.

Fehler_0004-NO_RESOURCES

No resources free in CP.

Fehler_0006-UNKNOWN_REFERENCE

Incorrect OPEN reference defined.

Fehler_0008-BUFFER_TOO_SHORT

User buffer too short.

Fehler_0010-BUFFER_TOO_LONG

User buffer too long.

Fehler_0012-ILLEGAL_REQ

Incorrect "negot_options" defined.

Fehler_0014-REM_ABORT

Connection aborted by remote station.

Fehler_0016-LOC_TIMEOUT

Timeout.

Fehler_0018-UNKNOWN_CONN_CLASS

Unknown connection class.

Fehler_0020-DUP_REQ

Connection already established.

Fehler_0022-CONN_REJECT

Connection request rejected by remote.

Fehler_0024-NEGOT_FAILED

Connection abort faulty "negot-option".

Fehler_0026-ILLEGAL_ADDRESS

Faulty transport address.

Fehler_0028-NETWORK_ERROR

Bus or CP disrupted.

Fehler_0030-PROTOCOL_ERR

Protocol error.

Fehler_0032-ILLEGAL_RB_LENGTH

Incorrect request block length.

Fehler_0784-E_NO_HW

No communication hardware found.

- Communication module defective.
- Communication module not installed correctly.
- Wrong port address defined.

Fehler_0786-E_CNF

Driver configured incorrectly or invalid parameter in the registry.

Fehler_0787-E_BAUDRATE

Incorrect baudrate or incorrect interrupt vector defined.

Fehler_0788-E_HSA

Incorrect HSA (Highest Station Address) defined.

Fehler_0789-E_TS

The defined local participant number (TS_ADR) is already assigned.

Fehler_0791-E_INT_NOT_PROV

The defined interrupt vector (IRQ) is not available on the communication module.

Fehler_0792-E_INT_BUSY

The defined interrupt vector (IRQ) is already occupied on the communication module.

Fehler_0800-E_NO_FILE

The selected communication driver cannot be loaded; the file was not found.

- Communication driver not installed correctly.

Fehler_0897-E_LOGDEV

The logical device is not defined in the registry.

- Communication driver not installed correctly.
- Entry damaged or deleted in the registry.
- Check the setting of the logical device name with the "Set PG/PC interface" program.
- Check the setting for the logical device name in the "System parameter - Device" mask.

Fehler_0898-E_L2DRIVER

The entry "L2DRIVER" is missing in the registry.

- Module error or module installed incorrectly.

Fehler_0900-E_L4DRIVER

The entry "L4DRIVER" is missing in the registry.

- Module error or module installed incorrectly.

Fehler_30000-EC_WATCHDOG

Watchdog error.

Fehler_30001-EC_PDUERROR

PDU not expected.

Fehler_30005-EC_ONLERROR

Fault loading the S7-Online-DLL.

9.4.2.3 iNA960 messages

General iNA960 messages

OK_RESP	1	0x01	Request executed with no errors
OK_EOM_RESP	3	0x03	Data block received with no errors

OK_DECIDE_REQ_RESP	5	0x05	Request executed with no errors
OK_CLOSED_RESP	7	0x07	Connection aborted by local user

iNA960 error messages

INVALID_REQ	2	0x02	Faulty request block
NO_RESOURCES	4	0x04	No resources free in CP
UNKNOWN_REFERENCE	6	0x06	Incorrect OPEN reference defined
BUFFER_TOO_SHORT	8	0x08	User buffer too short
BUFFER_TOO_LONG	10	0x0A	User buffer too long
ILLEGAL_REQ	12	0x0C	Incorrect "negot_options" defined
REM_ABORT	14	0x0E	Connection aborted by remote station
LOC_TIMEOUT	16	0x10	Timeout
UNKNOWN_CONN_CLASS	18	0x12	Unknown connection class
DUP_REQ	20	0x14	Connection already established
CONN_REJECT	22	0x16	Connection request rejected by remote
NEGOT_FAILED	24	0x18	Connection abort faulty negot-option
ILLEGAL_ADDRESS	26	0x1A	Faulty transport address
NETWORK_ERROR	28	0x1C	Bus or CP disrupted
PROTOCOL_ERR	30	0x1E	Protocol error
ILLEGAL_RB_LENGTH	32	0x20	Incorrect request block length

9.4.2.4 SCI messages

See description in the "SINEC Communication Interface SCI" manual (A/5-15).

SCI messages

SCP_OK	0	0x00	No error
SCP_INCONS	201	0xC9	Minor device number is not 00
SCP_RESOURCE	202	0xCA	DPRAM request invalid
SCP_CONFIG	203	0xCB	Configuration error (NUM_PROCS)
SCP_NOCONFIG	204	0xCC	SCP driver not configured
SCP_PARAM	206	0xCE	Incorrect mode
SCP_DEVOPEN	207	0xCF	Open already performed
SCP_BOARD	208	0xD0	Board not inserted/recognized
SCP_SOFTWARE	209	0xD1	IRQ error or software not found
SCP_MEM	210	0xD2	Low memory in DPRAM
SCP_MODE	211	0xD3	Download process not yet ended
SCP_LOADER	212	0xD4	No response from loader
SCP_SIGNAL	213	0xD5	Process started asynchronously
SCP_NOMESS	215	0xD7	No message arrived for the process
SCP_USERMEM	216	0xD8	length_of_buffer too small

SCP_WINDOW	217	0xD9	Too many SEND calls
SCP_TIMEOUT	219	0xDB	Timeout on SCP
SCP_ATTACH	220	0xDC	Reset not executed/Channel still active
SCP_ILLEGAL_REQUEST	221	0xDD	Illegal request
SCP_ERECOVERF	223	0xDF	Buffer not retrieved with scp_receive
SCP_ECLOSED	224	0xE0	All buffers assigned for the connection
EUSERMAX	225	0xE1	
SCP_EINTR	226	0xE2	
SCP_BOARD_OPEN	231	0xE7	
SCP_NO_WIN_SERV	233	0xE9	
EPROTECT	234	0xEA	License not found

SCI messages

SCP_DB_FILE_DOES_NOT_EXIST	240	0xF0
SCP_DB_FILE_CLOSE_NOT_OK	241	0xF1
SCP_SEND_NOT_SUCCESSFUL	242	0xF2
SCP_RECEIVE_NOT_SUCCESSFUL	243	0xF3
SCP_NO_DEVICE_AVAILABLE	244	0xF4
SCP_ILLEGAL_SUBSYSTEM	245	0xF5
SCP_ILLEGAL_OPCODE	246	0xF6
SCP_BUFFER_TOO_SHORT	247	0xF7
SCP_BUFFER_1_TOO_SHORT	248	0xF8
SCP_ILLEGAL_PROTOCOL_SEQUENCE	249	0xF9
SCP_ILLEGAL_PDU_ARRIVED	250	0xFA
SCP_REQUEST_ERROR	251	0xFB
SCP_NO_LICENSE	252	0xFC

Additional online DLL messages on the SCP interface

E_TIMER_INIT	768	0x0300	WIN Set-timer request unsuccessful
E_INIT_COM	769	0x0301	
E_NO_HW	784	0x0310	MPI module not found
E_HW_DEFEKT	785	0x0311	Problem with the hardware
E_CNF	786	0x0312	Incorrect configuration parameter
E_BAUDRATE	787	0x0313	Incorrect baudrate/incorrect IntVector
E_HSA	788	0x0314	Incorrect HSA configured
E_TS	789	0x0315	Configured address already assigned
E_OCC	790	0x0316	HW_Device already assigned
E_INT_NOT_PROV	791	0x0317	Interrupt not available
E_INT_BUSY	792	0x0318	Interrupt occupied
E_SAP	793	0x0319	SAP deactivate: SAP not occupied
E_UNPLUGGED	794	0x031a	No remote station found

E_SYNI	795	0x031b	Syni Error occurred
E_AMPRO	796	0x031c	AMPRO 2 reported a system error
E_BUFFSIZE	797	0x031d	No buffer of this size created
E_NO_FILE	800	0x0320	DLL/VxD File not found or entries in registry destroyed
E_NO_ENTRY	801	0x0321	Address does not exist in DLL
E_VERSION	816	0x0330	Version conflict between SMC driver and SMC firmware
E_COMCNF	817	0x0331	Problem with the COM port configuration
E_NO_SMC	818	0x0332	SMC no longer responds
E_COMMBADID	819	0x0333	COM port is not configured
E_COMMOPEN	820	0x0334	COM port is not available
E_SMCBUSY	821	0x0335	Serial driver is currently in use with another configuration
E_SMCMODEM	822	0x0336	No connection exists to a PC/MPI cable
E_SMCNOLEG	823	0x0337	PC/MPI cable rejects request, necessary authorization is missing
E_ONLINE	896	0x0380	Internal error at the IOCTL interface
E_LOGDEV	897	0x0381	Logical device not in registry
E_L2DRIVER	898	0x0382	L2DRIVER entry is missing in the registry
E_L4DRIVER	900	0x0384	L4DRIVER entry is missing in the registry
E_SYSERROR	1023	0x03FF	System error

Channel-specific error codes

EC_WATCHDOG	30000	0x7530	Watchdog error
EC_PDUERROR	30001	0x7531	PDU not expected
EC_ONLERROR	30005	0x7535	Fault loading the S7-Online-DLL

S5 PROFIBUS FDL

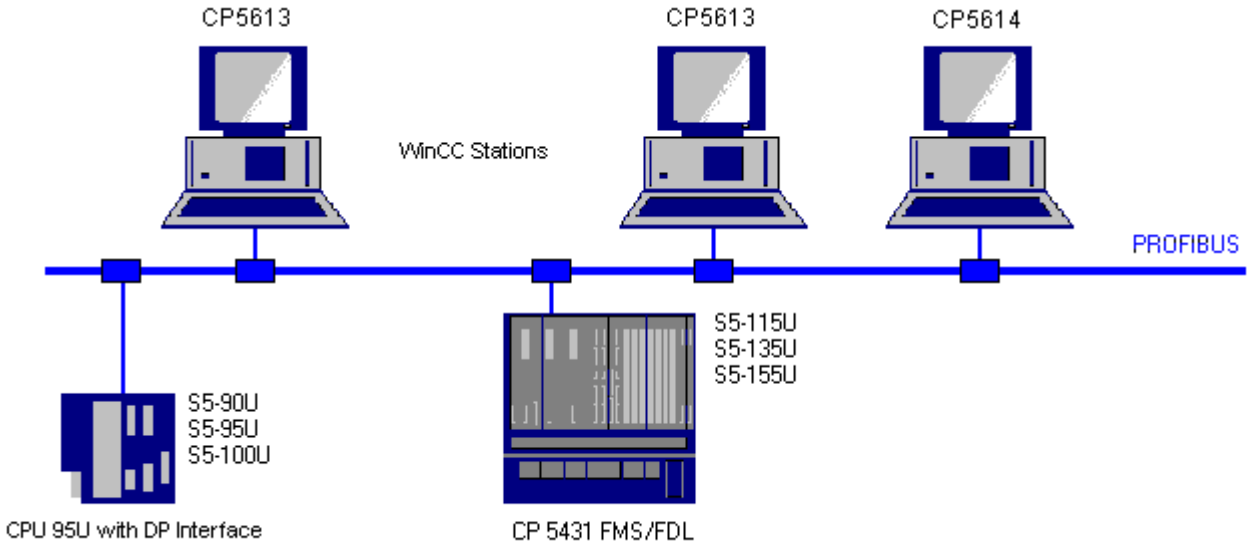
10

10.1 WinCC channel "SIMATIC S5 Profibus FDL"

Introduction

The "SIMATIC S5 Profibus FDL" channel is used for communication between a WinCC Station and a SIMATIC S5 automation system. The PROFIBUS (Process Field Bus) network type and the FDL (Field Data Link) protocol are used in this case.

PROFIBUS is the network for small to medium-sized data volumes. A broad range of automation tasks can be met with a maximum of 127 connectable nodes.



Tags are read/written via PROFIBUS using the FDL protocol using request and response frames. The request frame is sent to the automation device from WinCC. The AS answers with the response frame.

An FDL connection is specified by the local and remote connection end points (Service Access Point).

This section shows you

- how to configure data transmission with the "SIMATIC S5 Profibus FDL" channel.
- how to create a sample project

Channel unit FDL (CP5412/A2-1)

Regardless of the communications processor used, the possibility exists to connect to the SIMATIC S5 via the "FDL (CP5412/A2-1)" channel unit.

This channel unit supports up to a maximum of 24 connections. In order for the channel to function, a channel unit and a connection must be created.

Service Access Point

SAPs are local data interfaces within a PROFIBUS node. The SAPs must be configured in WinCC and on the AS. A unique identification is defined with the Service Access Point. This unique identification is required for communication between the WinCC and the AS.

Active connection

An active connection is also called a Fetch connection. This is a connection in which an active partner fetches data from a communication partner. The communication partner from which the data is fetched is called a passive partner.

Passive connection

A passive connection exists if the active AS sends data to the passive WinCC partner asynchronously without a request frame.

See also

Example of configuring the "SIMATIC S5 Profibus FDL" channel (Page 208)

Special functions of the "SIMATIC S5 Profibus FDL" Channel (Page 202)

Features of the WinCC channel "SIMATIC S5 Profibus FDL" (Page 191)

Supported data types and data ranges (Page 189)

10.2 Supported data types and data ranges

Introduction

Only certain data types and data ranges are supported for communication from SIMATIC S5 via PROFIBUS FDL.

Supported data types

WinCC Data type	SIMATIC S5 data type
Binary tag	BIT
Signed 8-bit value	non-existent in the SIMATIC S5
Unsigned 8-bit value	BYTE
Signed 16-bit value	WORD
Unsigned 16-bit value	WORD
Signed 32-bit value	DWORD
Unsigned 32-bit value	DWORD
Floating-point number 32-bit IEEE 754	DWORD
Floating-point number 64-bit IEEE 754	non-existent in the SIMATIC S5
Text tag, 8-bit character set	ARRAY OF BYTE
Text tag, 16-bit character set	non-existent in the SIMATIC S5
Raw data type	ARRAY OF BYTE

Access to SIMATIC S5 tags

The access to SIMATIC S5 tags is done word by word to data block DB or extended data blocks DX. This allows read and write access.

Access to a SIMATIC S5 tag of data type BIT

SIMATIC S5 tags of data type BIT only allow read access. This restriction applies for active or passive connections.

Access to a SIMATIC S5 tag of data type BYTE

SIMATIC S5 tags of data type BYTE only allow read access.

To configure a byte tag, the "left byte" or "right byte" of a 16 bit data word must be selected for addressing.

Access to a SIMATIC S5 tag of data type ARRAY OF BYTE

SIMATIC S5 tags of data type ARRAY OF BYTE only allow read access.

See also

Raw data tags of the "SIMATIC S5 Profibus FDL" channel (Page 202)

10.3 Features of the WinCC channel "SIMATIC S5 Profibus FDL"

Introduction

The capabilities of communication, from WinCC via the communication driver for PROFIBUS, are listed in the following. All supported data types and the respective capabilities for type conversion are also listed.

Note

Particularities when writing tags

When configuring in WinCC, make sure that every tag is transferred individually when writing more than one tag into data areas of the automation system.

This behavior is especially important when writing multiple tags with the "SetTagMultiWait" function, e.g. in a script. Since this function is only executed on completion of the transmission of all the tags transferred to it, noticeable waiting times can occur with larger tag quantities.

A check must be made to determine whether use of the "Wait" function is required for a larger tag quantity. In this case, use of a raw data tag may also be a good idea, especially if the data is sequential in the AS data area.

Type conversion

A type conversion is required if a certain value range or a conversion, e.g. from decimal to BCD ("Unsigned 8 bit value" converted to "ByteToBCDWord"), is needed. No type conversion is performed by default.

The following table lists the supported WinCC data types and the respective capabilities for type conversion.

WinCC Data type	Type conversion
Binary tag	No
Signed 8-bit value	Not available in the S5
Unsigned 8-bit value	Yes
Signed 16-bit value	Yes
Unsigned 16-bit value	Yes
Signed 32-bit value	Yes
Unsigned 32-bit value	Yes
Floating-point number 32-bit IEEE 754	Yes
Floating-point number 64-bit IEEE 754	Not available in the S5
Text tag, 8-bit font	No
Text tag, 16-bit font	Not available in the S5
Raw data type	No

WinCC side

The communication driver SIMATIC S5 Profibus FDL supports communication using the following communications processors:

Communications processor	Bus type
CP 5613 A3	PCI
CP 5612	PCI

AS side

Programmable logic controllers can generally be connected to a PROFIBUS network in two different ways.

The connection can be done via the integrated interface on the central module or using special communication modules.

System	Module
S5-90U, S5-95U, S5-100U	CPU95U
S5-115U, S5-135U, S5-155U	CP5431 FMS/DP

Note

For connections to the S5-95U with L2-SS, no Fetch connections are possible since WinCC can only be a passive partner.

10.4 Configuring the Channel

10.4.1 How to configure the channel "SIMATIC S5 Profibus FDL"

Introduction

The steps in configuring the channel "SIMATIC S5 Profibus FDL" are described in this and the following sections.

This section shows how the channel "SIMATIC S5 Profibus FDL" is configured.

Procedure

1. In the navigation area of the tag management, select the entry "Add new driver" in the shortcut menu of node "Tag Management".
2. Select the driver "SIMATIC S5 Profibus FDL". The channel is created and the communication driver is displayed in the tag management.

10.4.2 Channel unit "FDL (CP5412/A2-1)"

Introduction

The "SIMATIC S5 Profibus FDL" communication driver contains only the "FDL (CP5412/A2-1)" channel unit.

The communication between WinCC and the SIMATIC S5 programmable logic controller takes place via the "FDL (CP5412/A2-1)" channel unit.

A maximum of 24 connections can be created within the channel unit. Special connection parameters must be set for each configured connection. Each configured tag must be defined by tag parameters.

Note

The name of the "FDL (CP5412/A2-1)" channel unit is bound to the communication driver, "SIMATIC S5 Profibus FDL.CHN", and is independent of the communications processor used.

For example, CP5613 A3 can be used as the communications processor.

Tag parameters

The following tag parameters must be specified for each configured tag:

- Data area (e.g. DB)
- Data block number

- Addressing (e.g. "left byte")
- Start address (e.g. DL 0, if "left byte" has been selected for addressing)

Connection parameters

The following connection parameters must be specified for each configured connection:

- The station address of the AS
- The priority
- Own and external SAPs (Service Access Point) must be specified for the read and write function

Whether the connection is to be an active or passive connection must also be configured for the read function In the case of an active read connection, the values are requested by the WinCC station. In the case of a passive connection, the transfer of values to the WinCC station is initiated by the AS.

10.4.3 How to configure a connection

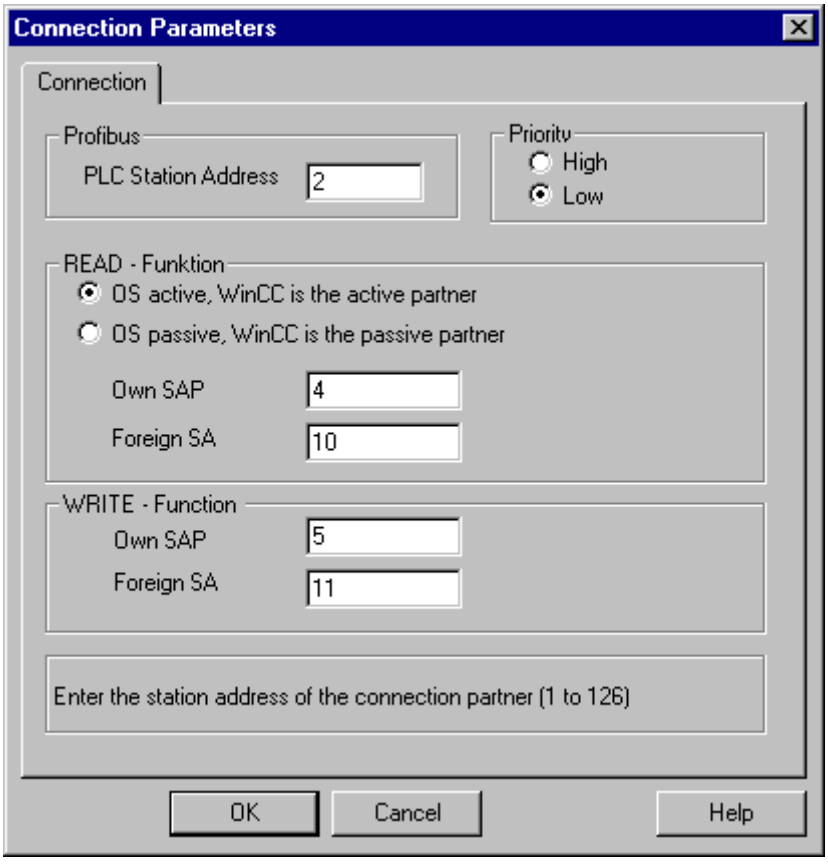
Requirements

- The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.

Procedure

1. Select the channel unit "FDL (CP5412/A2-1)".
2. Select the entry "New Connection" in the shortcut menu of the channel unit.
3. Enter the name of the connection.

4. Select the entry "Connection parameters" from the shortcut menu of the connection. The "Connection properties" dialog opens.



5. Enter the unique address of the AS in the "PLC Station Address" field.
6. The "Priority" option button must always be set to "Low" for an FDL connection.
7. The function of the WinCC station is defined with fields "OS active, WinCC is the active partner" or "OS passive, WinCC is the passive partner". Activate the required option button.
8. Enter the SAP addresses configured for the reading and writing access in the fields "Own SAP" and "Foreign SAP". The SAPs value range is between 2 and 54.
9. Click "OK" to close all open dialogs.
10. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens. Configure the tag.
11. Click "OK" to close all open dialogs.

See also

How to configure a tag with word by word access (Page 198)

How to Configure a Tag with Byte by Byte Access (Page 197)

How to Configure a Tag with Bit by Bit Access (Page 196)

10.4.4 Configuring the tags

10.4.4.1 Configuring the tags

Introduction

For a connection between WinCC and the AS via channel "SIMATIC S5 Profibus FDL", data types binary, byte and word can be defined within WinCC. The following describes how to configure a tag of these data types.

See also

Raw data tags of the "SIMATIC S5 Profibus FDL" channel (Page 202)

How to configure a tag with word by word access (Page 198)

How to Configure a Tag with Byte by Byte Access (Page 197)

How to Configure a Tag with Bit by Bit Access (Page 196)

10.4.4.2 How to Configure a Tag with Bit by Bit Access

Introduction

This section shows you how to configure a tag for bit by bit access for the address area in the AS.

Note

The bit by bit access to a tag is only read access.

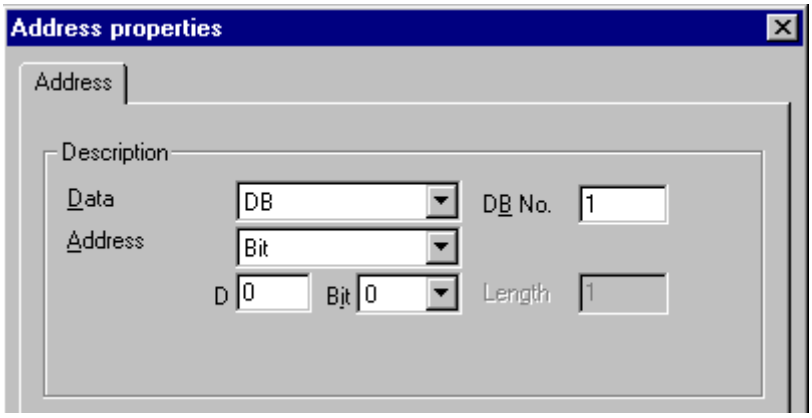
Requirements

1. The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.
2. A connection must be defined in the channel unit "FDL (CP5412/A2-1)".

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. Set the "Binary tag" data type in the "Data Type" field.

5. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



6. Enter the byte address in field "D" and the bit address in field "Bit". The label on the left field depends on the entry in the Data Area field, for e.g. "D" for data area "DB" and binary tag as the data type of the tag.
7. Click "OK" to close all open dialogs.

Note

You cannot change the "Bit" entry in the Addressing field because it is defined by the Binary tag data type of the WinCC tag.

10.4.4.3 How to Configure a Tag with Byte by Byte Access

Introduction

This section shows you how to configure a tag for byte by byte access for the address area in the AS.

Note

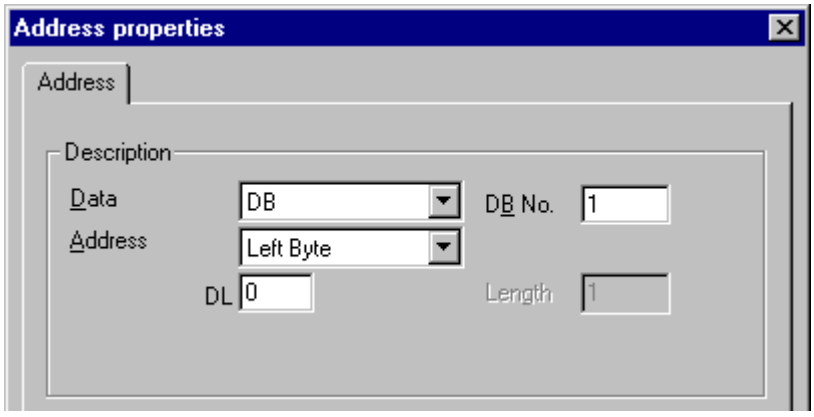
The byte-wise access to a tag is only read access.

Requirements

1. The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.
2. A connection must be defined in the channel unit "FDL (CP5412/A2-1)".

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. In the "Data Type" field, set the data type to "Unsigned 8-bit value".
5. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



6. You can choose between "Left byte" and "Right byte" in the "Addressing" field.
7. Enter the byte address in the "DL" field. The label on the field depends on the entry in the "Addressing" field, e.g. "DL" for "Left byte" addressing.
8. Click "OK" to close all open dialogs.

10.4.4.4 How to configure a tag with word by word access

Introduction

This section shows you how to configure a tag for word-wise access for the address area in the AS.

Note

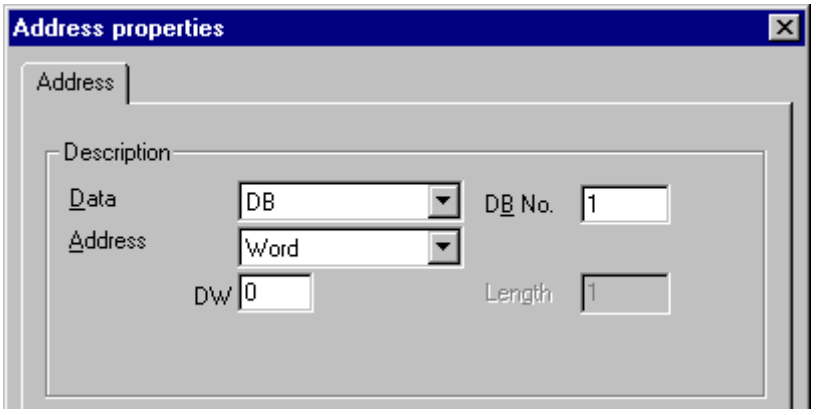
The word by word access to a tag is read and/or write access.

Requirements

1. The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.
2. A connection must be defined in the channel unit "FDL (CP5412/A2-1)".

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. In the "Data Type" field, set the data type to "Unsigned 16-bit value".
5. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



6. Enter the word address in the field below. The label on the field depends on the entry in the "Addressing" field, e.g. "DW" for "Word" addressing.
7. Click "OK" to close all open dialogs.

Note

You cannot change the "Word" entry in the "Addressing" field because it is defined by the "Unsigned 16-bit value" data type of the WinCC tag.

10.4.5 System parameters

10.4.5.1 System parameters of the channel unit

Introduction

If you require a configuration that deviates from the standard WinCC settings, you can make all the required changes using the "System Parameter" dialog of the channel unit.

The following individual points can be changed:

- the device name
- the Write/Read monitoring time

Device Name

Communication between WinCC and the automation system takes place via logical device names. These names are assigned during the installation of the communication module and are unit-specific. The device name represents the logical device name. This field is defined with the entry "/CP_L2_1:/SCP" as default.

Write/Read monitoring time

The write/read monitoring time is the maximum waiting time in seconds for write/read responses of the AS. If no response is made by the AS within the defined time, the connection is broken. This field is assigned a waiting time value of 30 seconds as default

Note

The system parameters apply for all CPs in the AS.

See also

How to change the write/read monitoring time of process values (Page 201)

How to Change the Device Name (Page 200)

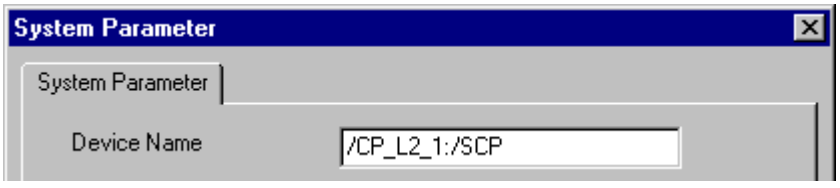
10.4.5.2 How to Change the Device Name

Requirements

- The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.

Procedure

1. In the channel unit shortcut menu, select "System parameters". The "System Parameters" dialog opens.



2. Enter the name of the access point in the "Device name" field. This name must match the setting that you have made under Windows via "Start" → "Settings" → "Control panel" → "Set PG/PC interface".
3. Close the dialog by clicking the "OK" button.

Note

The changes only take effect after WinCC is restarted.

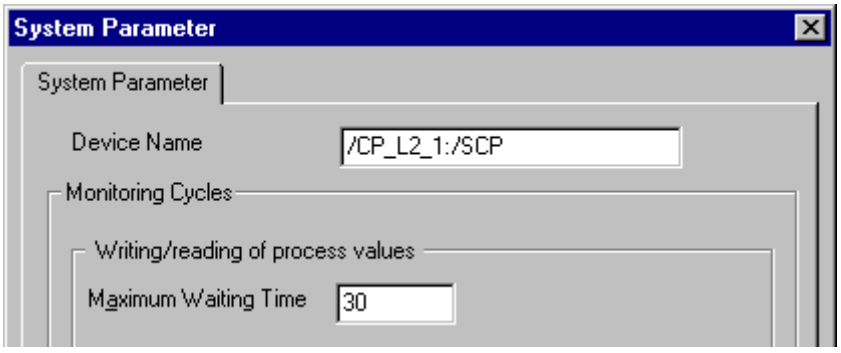
10.4.5.3 How to change the write/read monitoring time of process values

Requirements

- The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.

Procedure

1. In the channel unit shortcut menu, select "System parameters". The "System Parameters" dialog opens.



2. Enter the required value in seconds in the "Maximum waiting time" field. You can define a value between 1 and 3600 seconds. This field is assigned with a default value of 30 seconds.
3. Close the dialog by clicking the "OK" button.

Note

The changes only take effect after WinCC is restarted.

10.5 Special Functions

10.5.1 Special functions of the "SIMATIC S5 Profibus FDL" Channel

Introduction

The "SIMATIC S5 Profibus FDL" channel has some special functions, the functionality of which is described in this chapter.

See also

Configuring the communication types (Page 204)

Raw data tags of the "SIMATIC S5 Profibus FDL" channel (Page 202)

10.5.2 Raw data tags of the "SIMATIC S5 Profibus FDL" channel

10.5.2.1 Raw data tags of the "SIMATIC S5 Profibus FDL" channel

Introduction

A tag of the type "raw data type" is a data telegram.

Raw data tags are required for transferring user data blocks from/to the AS

A raw data tag used by SIMATIC S5 Profibus FDL can be a maximum of 220 bytes in length.

Raw data tag as byte array

A raw data tag as byte array is handled like a normal process tag that is addressed via the address and length of the data block (for e.g. DB 100, DW 20, length 40 Byte).

Writing raw data tags using scripts

If a raw data tag, which is longer than the tag length configured in WinCC, is written to S5 using a VB script, the write process is aborted.

Instead write the raw data tag via a C script using the "SetTagRaw" function. For this function, specify the length of the tags that are to be written.

See also

How to configure raw data tags (Page 203)

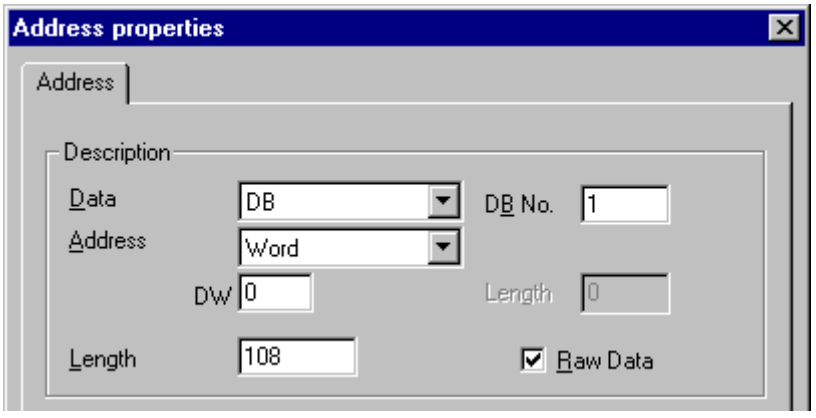
10.5.2.2 How to configure raw data tags

Requirements

1. The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.
2. A connection must be defined in the channel unit "FDL (CP5412/A2-1)".

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. Select "Raw Data Type" in the "Data type" field.
5. Click the "Select" button to open the "Address properties" dialog.
   Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



6. Mark the "Raw Data" check box.
7. Enter the length of the raw data block (in bytes) in the "Length" field.
8. In the "Data area" set the data area of the PLC where the data is located. If you select "DB" as data area, enter the number of the data block in the enabled "DB No." field.
9. Set up the addressing type in the "Addressing" field. The entries "Left byte", "Right byte", "Word" and "Double word" are possible for data type "Raw data type" of the WinCC tag.
10. Enter the value of the start address in the underlying field. The label on the left field depends on the entry in the Data Area and Addressing field, for e.g. "DW" for data area "DB" "Word" for addressing type.
11. Click "OK" to close all open dialogs.

10.5.3 Configuring the communication types

10.5.3.1 Configuring the communication types

Introduction

An FDL connection can be configured so that WinCC runs as an active or passive partner.

If WinCC is configured as an active partner, the values are requested by the WinCC station.

If WinCC is configured as a passive partner, the transfer of values to the WinCC station is initiated by the AS.

See also

How to configure a passive data transfer (Page 206)

How to configure an active data transfer (Page 204)

10.5.3.2 How to configure an active data transfer

Introduction

This section shows you how to configure an active data transfer to the address area in the AS.

Note

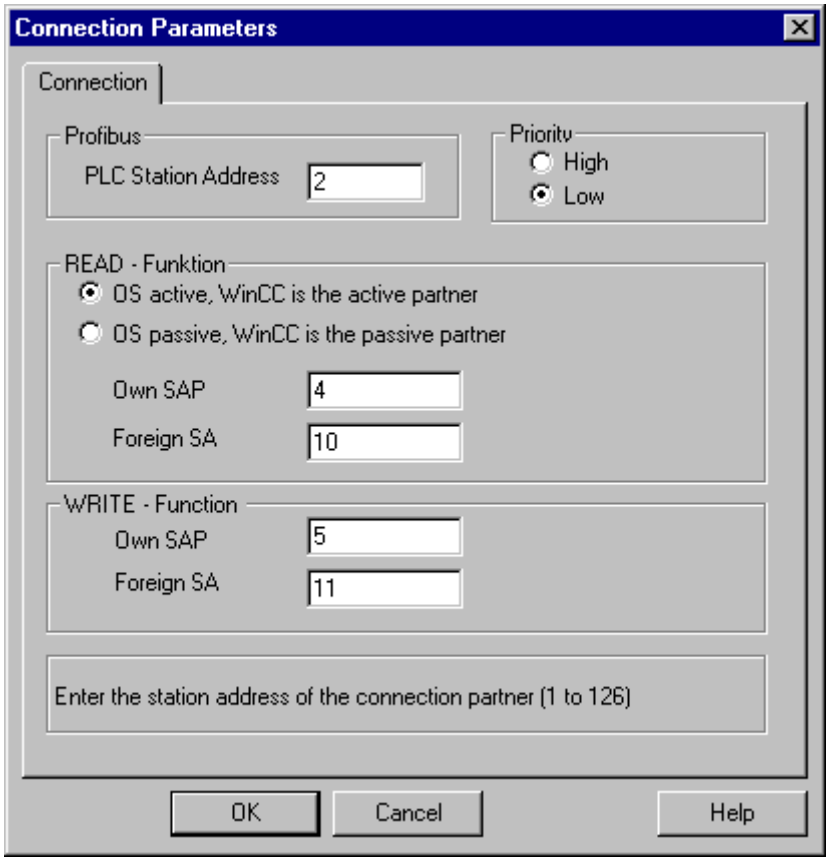
If more than one connection is configured, note that an SAP can only be assigned one time.

Requirements

1. The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.
2. A connection must be defined in the channel unit "FDL (CP5412/A2-1)".
3. You must know the SAP address defined by the AS.

Procedure

1. Select the entry "Connection parameters" from the shortcut menu of the connection.



2. Enter the station address of the AS in the "PLC Station Address" field on the "Connection" tab.
3. The following settings must be made in the "READ - Function" area:
4. Activate the option "OS active, WinCC is the active partner".
5. Enter the SAP-ID of the WinCC station in the "Own SAP" field.
6. Enter the SAP-ID of the AS in the "Foreign SAP" field.
7. The following settings must be made in the "WRITE - Function" area:
8. Enter the SAP-ID of the WinCC station in the "Own SAP" field.
9. Enter the SAP-ID of the AS in the "Foreign SAP" field.
10. Click "OK" to close all open dialogs.

10.5.3.3 How to configure a passive data transfer

Introduction

This section shows you how to configure a passive data transfer to the address area in the AS.

Note

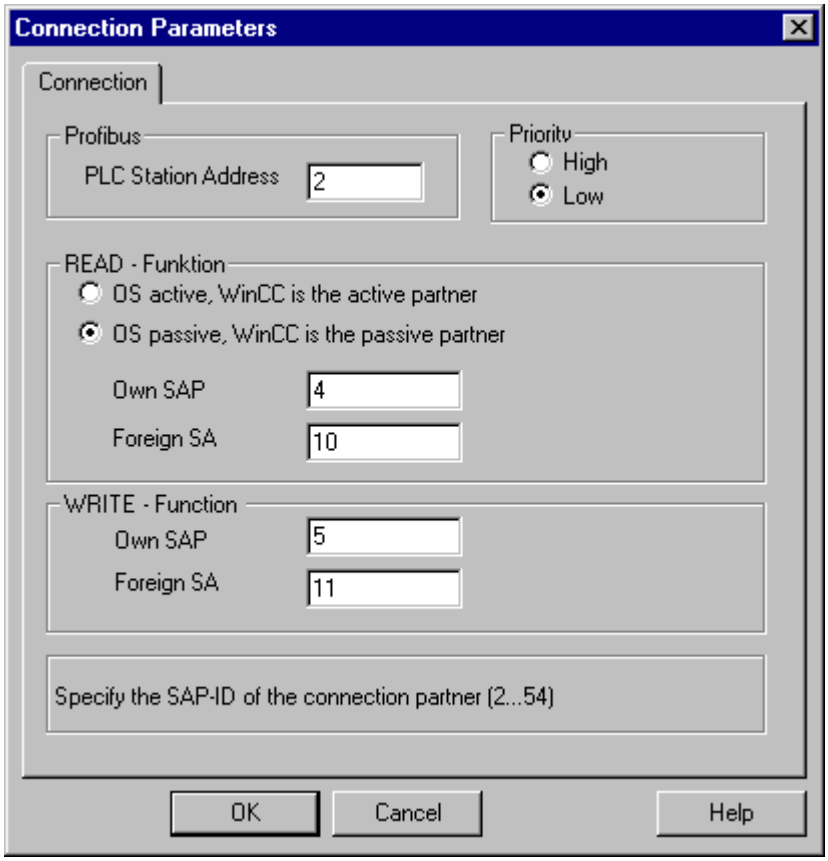
If more than one connection is configured, note that an SAP can only be assigned one time.

Requirements

1. The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.
2. A connection must be defined in the channel unit "FDL (CP5412/A2-1)".
3. You must know the SAP address defined by the AS.

Procedure

1. Select the entry "Connection parameters" from the shortcut menu of the connection.



2. Enter the station address of the AS in the "PLC Station Address" field on the "Connection" tab.

3. The following settings must be made in the "READ - Function" area:
4. Activate the option "OS passive, WinCC is the passive partner".
5. Enter the SAP-ID of the WinCC station in the "Own SAP" field.
6. Enter the SAP-ID of the AS in the "Foreign SAP" field.
7. The following settings must be made in the "WRITE - Function" area:
8. Enter the SAP-ID of the WinCC station in the "Own SAP" field.
9. Enter the SAP-ID of the AS in the "Foreign SAP" field.
10. Click "OK" to close all open dialogs.

10.6 Example of configuring the "SIMATIC S5 Profibus FDL" channel

10.6.1 Example of configuring the "SIMATIC S5 Profibus FDL" channel

Introduction

In this example, you will configure an I/O Field in the Graphics Designer and assign the necessary values to the data handling blocks in the AS.

See also

How to configure an I/O Field (Page 211)

How to configure the data handling blocks in the AS (Page 208)

10.6.2 How to configure the data handling blocks in the AS

Introduction

In this section, you will configure the standard function blocks OB 21 (L2ANLAUF) and OB 1 (L2SNDRCV) in the AS.

By default, the data traffic for the SIMATIC S5 connection via PROFIBUS FDL is handled using the following blocks.

Within the example, the following SAP numbers are used:

	WinCC	Programmable controller
SAP number for the READ function	12	6
SAP number for the WRITE function	11	4

Blocks

Function	Block
For the startup OB 20, 21, 22	FB-L2ANLAUF (FB 9)
For cyclic operation OB 1	FB-L2SNDRCV (FB 10)
As internal work data blocks for both FBs	DB-L2DBVC3 (DB 10) DB-L2DBVC4 (DB 11) DB-L2DBVC5 (DB 12)

Startup blocks

In the startup blocks, the communication parameters are specified, the work DBs are registered and the communications processor is synchronized.

These work steps are executed by calling function block FB9 L2ANLAUF, for example.

Cyclic block

The frame traffic is handled in the cyclic FB.

Received frames are entered in the destination data blocks. Should an error occur while this is being done, the frame is rejected and an error message is issued.

The user specifies the frames to be sent similar to the standard data handling blocks.

Feedback occurs after a completed transmission.

Requirements

- The data handling blocks SYNCHRON, CONTROL, SEND and RECEIVE must be available in the AS.

Procedure

1. A startup block (OB 20, 21, 22) is created in the STEP 5 software using menu item "Editor" → "STEP5 Block" → "in the program file".
   The name of the program block is "L2ANLAUF" in the example.
2. The following parameters must be preassigned:
   - Interface number (SSNR) of the CP (e.g. of CP5431)
   - The PROFIBUS address (RADR) of communication processor CP 5613 A3 on the WinCC computer.
     This number must be unique in the network.
   - Connection parameters of the utilized request types, e.g. parameters RVC4 for writing and RVC5 for reading, which specify the SAPs of the WinCC station. These SAPs are specified when the connection is created in WinCC.
   - Request numbers (ANR4 and ANR5), which are set during configuration of the FDL connections for the communications processor
   - Numbers of the work data blocks, DBX4 (for writing) and DBX5 (for reading)

```
OB  21
NETZWERK 1 von 1          Synchronisieren CP 5431
      :
      :SPA FB 9
NAME :L2ANLAUF
SSNR :     KF +0                          SSNR of CP 5431
TIM3 :     KT 000.0                       nr
TIM7 :     KT 000.0                       nr
RADR :     KF +8                          PROFIBUS-Address WinCC Station
RVC3 :     KF +0                          nr
RVC4 :     KF +4                          SAP WRITE
RVC5 :     KF +6                          SAP READ
RVC6 :     KF +0                          nr
RVC7 :     KF +0                          nr
ANR3 :     KF +0                          nr
ANR4 :     KF +134                        ANR WRITE
ANR5 :     KF +135                        ANR READ
ANR6 :     KF +0                          nr
ANR7 :     KF +0                          nr
DBX3 :     KY 000,000                     nr
DBX4 :     KY 000,011                     Work-DB WRITE
DBX5 :     KY 000,012                     Work-DB READ
DBX6 :     KY 000,000                     nr
DBX7 :     KY 000,000                     nr
S/R3 :     KF +0                          nr
      :
      :BE
```

3. An OB 1 (cyclic operation) is created in the STEP 5 software using menu item "Editor" → "STEP5 Block" → "in the program file".
   The name of the program block is "L2SNDRCV" in the example.

4. The communication with WinCC is performed, for example, using communications processor CP5431 and function block FB10 L2SNDRCV.
   If WinCC is to send and request data, only two relevant in/out parameters have to be specified for this purpose. These are the parameters DBX4 (for writing) and DBX5 (for reading), which specify the numbers of the two work data blocks of the utilized request types. These SAPs are specified when the connection is created in WinCC.

```
OB   1
NETZWERK 1 von 1          Communication Manual
      :
      :SPA FB 10                                  Communication
NAME  :L2SNDRCV
STR3  :     M 0.0                                 nr
STR7  :     M 0.0                                 nr
RDY   :     MB 0                                  nr
FAIL  :     MB 0                                  nr
TVC3  :     T 0                                   nr
TVC7  :     T 0                                   nr
DBX3  :     KY 000,000                            nr
DBX4  :     KY 000,011                            Work-DB WRITE
DBX5  :     KY 000,012                            Work-DB READ
DBX6  :     KY 000,000                            nr
DBX7  :     KY 000,000                            nr
      :
      :SPA FB 6                                   Add and Increment
NAME  :ADD_INC
      :
      :BE
```

5. Download the STEP 5 program to the programmable logic controller.
   This is done in the STEP 5 software using menu item "Object" → "Blocks" → "Transfer" → "PLC file".
   Select the "All blocks" option in the "Selection" field to download all previously created blocks to the automation system.

10.6.3 How to configure an I/O Field

Introduction

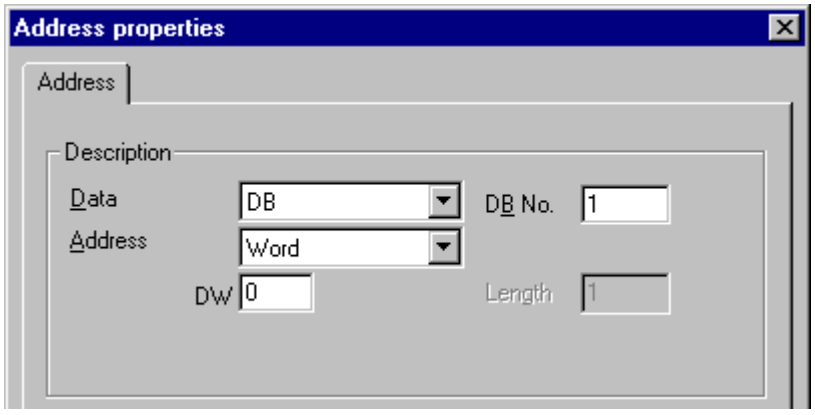
You will configure an I/O Field in this section.

Requirements

- The channel "SIMATIC S5 Profibus FDL" must be integrated in the project.

Procedure

1. Choose the "New Connection" option from the shortcut menu of the channel unit "FDL (CP5412/A2-1)" and set up a connection called "TestFDL".
2. Select the entry "Connection parameters" from the shortcut menu of the connection.
   The "Connection properties" dialog opens.
   Enter the station address of the AS in the "PLC Station Address" field.
   You can configure an FDL connection in such a way that WinCC is either an active or passive partner. If WinCC is configured as an active partner, the values are requested by the WinCC station. If WinCC is configured as a passive peer, the transfer of values to the WinCC station is initiated by the AS.
   Close all opened dialogs by clicking "OK"
3. Click the "Tags" tab below the table area.
4. Click in the top free cell of the "Name" column.
   Enter "FDLWord1_Test" as name.
5. In the "Data Type" field, set the data type to "Unsigned 16-bit value".
6. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



7. In the "Data area" set the data area of the PLC where the data is located. If you select "DB" as data area, enter the number of the data block in the enabled "DB No." field. Enter the byte address in the "DW" field.
8. Click "OK" to close all open dialogs.
9. You must integrate the smart object "I/O field" into a picture within Graphics Designer.
10. Connect the "I/O field" to a configured tag.
11. Activate the project by clicking the "Activate Runtime" button in the toolbar or by selecting "Activate Runtime" in the "File" menu. All changes to the configured tag are displayed in the "I/O field" in Runtime.

S5 Programmers Port AS511

11

11.1 WinCC channel "SIMATIC S5 Programmers Port AS511"

Introduction

The communication driver "SIMATIC S5 Programmers Port AS511" is utilized for the serial connection through a TTY interface to the SIMATIC S5 automation system.

This chapter describes

- how to configure the data transfer with the "SIMATIC S5 Programmers Port AS511" channel.
- how to configure a connection and a tag.

Channel Unit

The communication driver has one channel unit for controlling a COM port for the serial connection.

The following capability is available:

- Channel unit S5-AS511 for serial communication via a "Siemens-specific" protocol.

See also

Data type of the tags (Page 214)

11.2 Data type of the tags

Introduction

Define the required tags for a logical connection. From the WinCC viewpoint, you can access the following data types:

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

11.3 Configuring the Channel

11.3.1 Configuring the "SIMATIC S5 Programmers Port AS511" channel

Introduction

The following steps are required for configuring the channel "SIMATIC S5 Programmers Port AS511".

- Configuring the connection
- Configuring the tags

See also

Configuring the tags (Page 217)

How to configure the connection (Page 215)

11.3.2 How to configure the connection

Introduction

The process connection using a serial connection is possible with the SIMATIC S5 automation system. The AS 511 communication processor is used in the automation system.

No additional communication module is required in WinCC. Communication is set up via either the TTY port on a PG 760 or a COM Port that is part of the system's standard equipment. In the later case, an additional port converter is required V.24/V.28 <---> TTY.

This serial link supports transmission rates of up to 19200 baud.

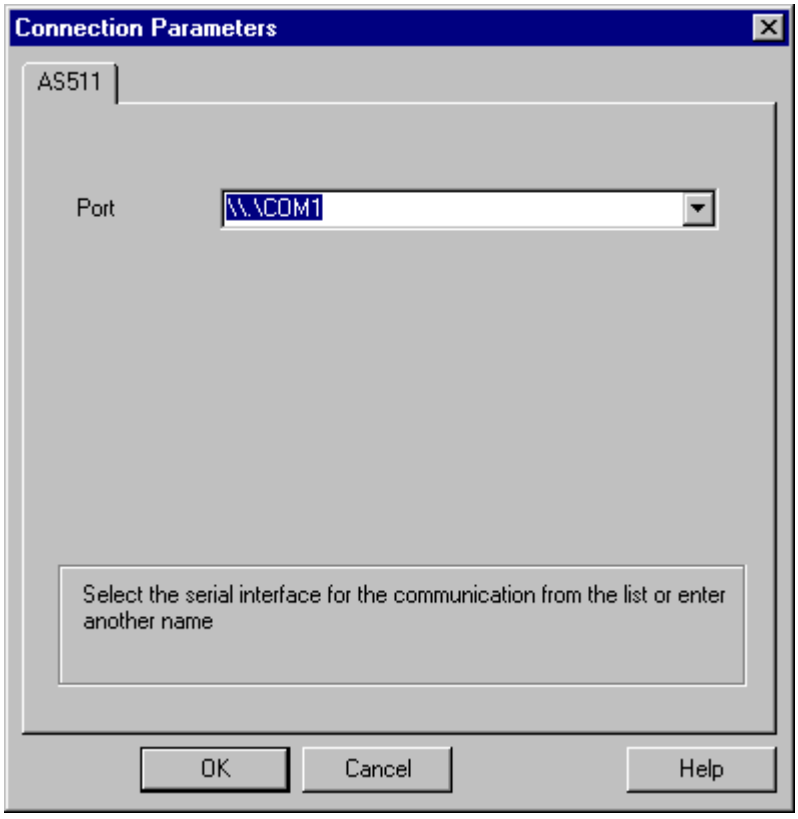
The following procedure can be used to assign one of the PC's serial ports to the AS511-NT drivers.

Note

During communication between WinCC and a S5 automation system via the "AS511" channel, data blocks may not be transferred, created or deleted in the AS. The memory in the S5 may also not be compressed. This last restriction is the result of the absolute addressing of memory in the S5. If changes are necessary, the link to WinCC must be disconnected.

Procedure

1. Select the connection under the channel unit "S5-AS511".
2. Select the entry "Connection parameters" from the shortcut menu of the connection.



3. In the "Port" field, select the port to be used for the serial link.

11.3.3 Configuring the tags

11.3.3.1 Configuring the tags

Introduction

For a connection between WinCC and the AS via channel "SIMATIC S5 Programmers Port AS511", tags of different data types can be created within WinCC. This is described in the following section.

Note

Addresses of the tags are not checked for plausibility in WinCC. If an address is used, which is not available in the AS, the status "Addressing error" will be set.

In DB and DX data blocks, reads and writes can only be made up to address 255.

Times cannot be written.

See also

How to Configure a Tag with Byte by Byte Access (Page 221)

How to configure a tag with bit-wise access (Page 220)

How to configure the address of a tag (Page 217)

11.3.3.2 How to configure the address of a tag

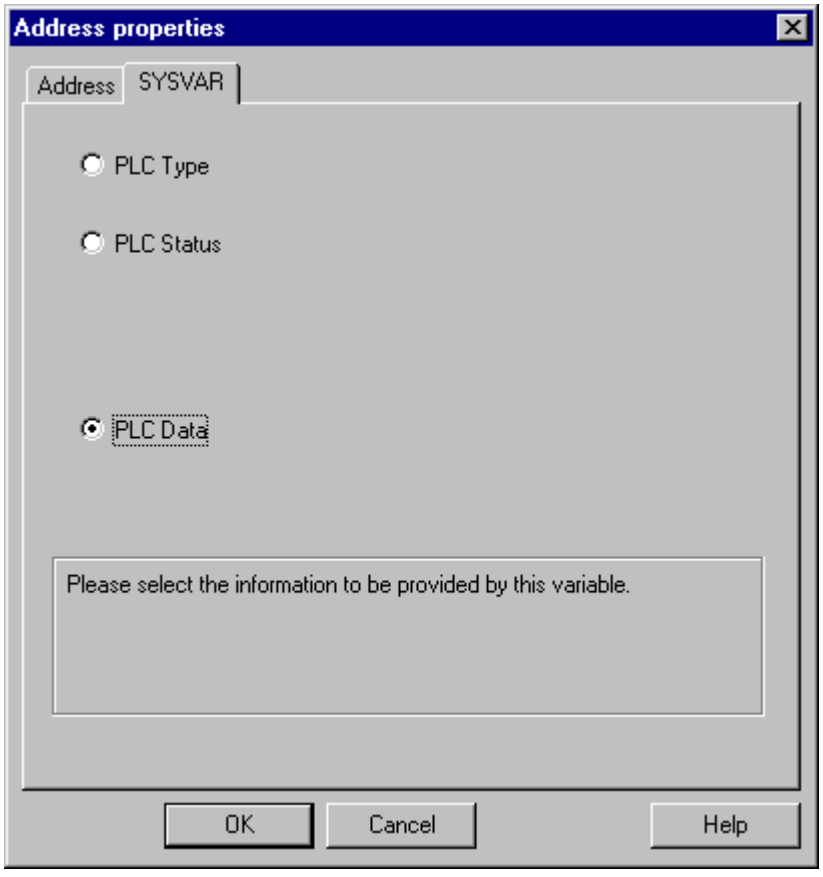
Introduction

The tag address is entered according to the address structure of the SIMATIC S5.

Procedure

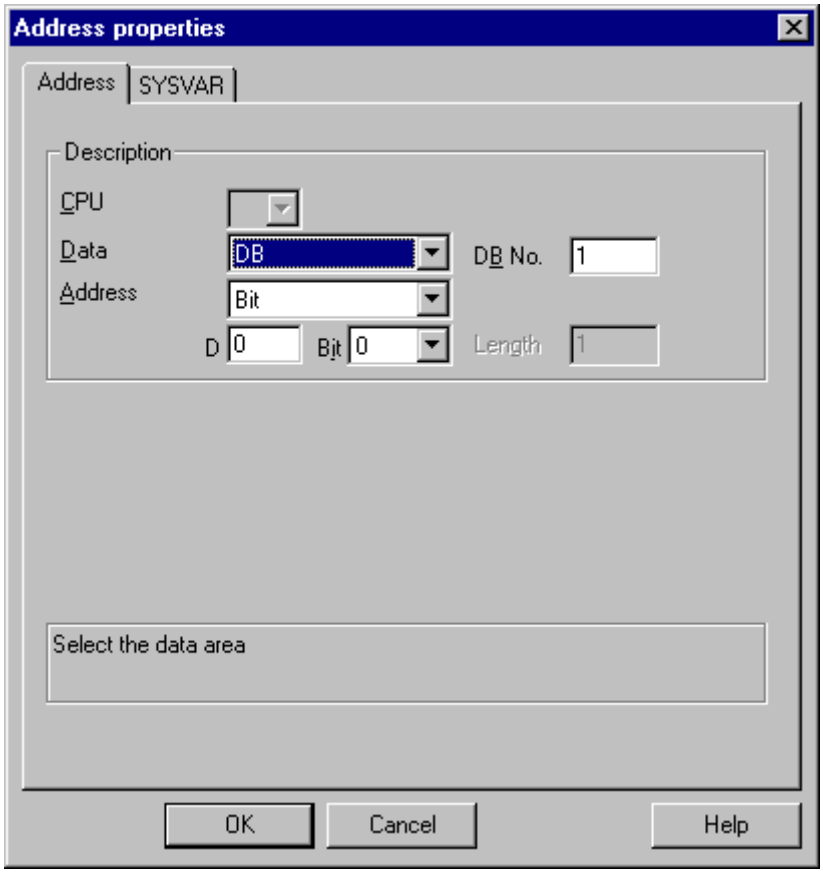
1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area and select the tag.
3. Select the required data type (e.g. signed 8-bit value) from the "General" tab.

4. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.
   Select the "SYSVAR" tab.



5. Click on the selection field to choose whether the tag should transfer the "PLC Type", the current status ("PLC Status") or other data ("PLC Data").

6. Only if you have selected "PLC data" will you have to click on the "Address" tab to define the S5 address of the tag.



7. Choose whether the tag is located in a data block, in an extended data block, in a flag area, an input range or an output range in the "Data Area" field.
8. If the tag is in a data block, the "DB No." field is also shown. Here, you enter the number of the data block.
9. The type of addressing is entered in the "Addressing" field. Normally, the default definition can be used.
10. Enter the address in the respective field (e.g. "DW ").

Frequently, the memory in the PLC can only be accessed by byte or word. When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to change individual bits in the memory of the PLC as well. For this purpose, the addressed memory area is read from the PLC for every single write request and

the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the PLC's memory.

Note

Changes that have been made by the PLC in a read data area are overwritten when writing back into the data area.

Depending on the type of tag, you can access the automation system's memory bit-wise or byte-wise.

Addresses of the tags are not checked for plausibility in WinCC. If an address is used, which is not available in the AS, the status "Addressing error" will be set.

In DB and DX data blocks, reads and writes can only be made up to address 255.

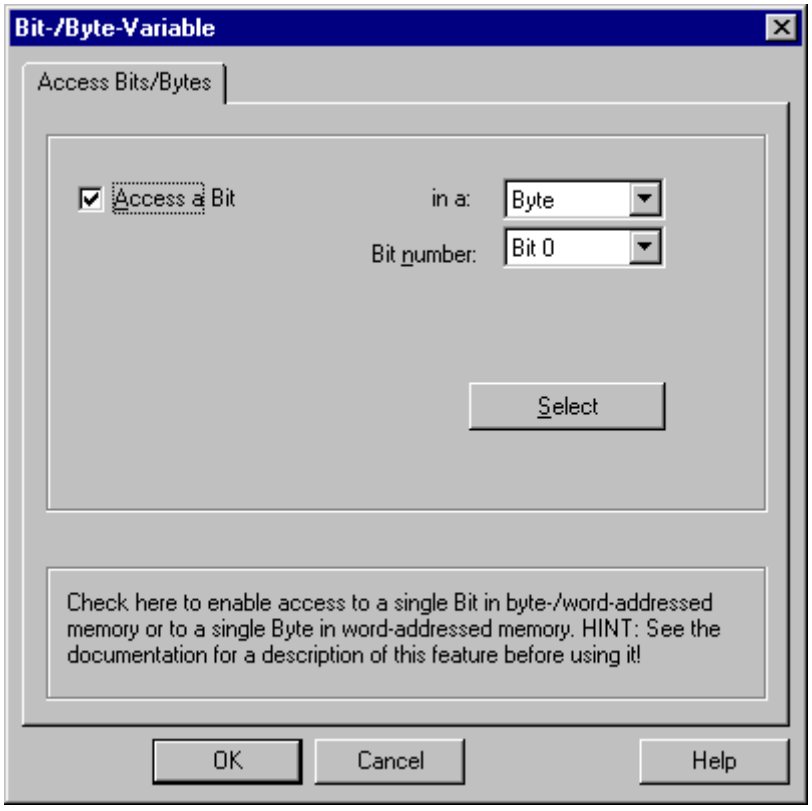
Times cannot be written.

11.3.3.3 How to configure a tag with bit-wise access

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. Set the "Binary tag" data type in the "Data Type" field.

5. Open the "Bit/Byte tag" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



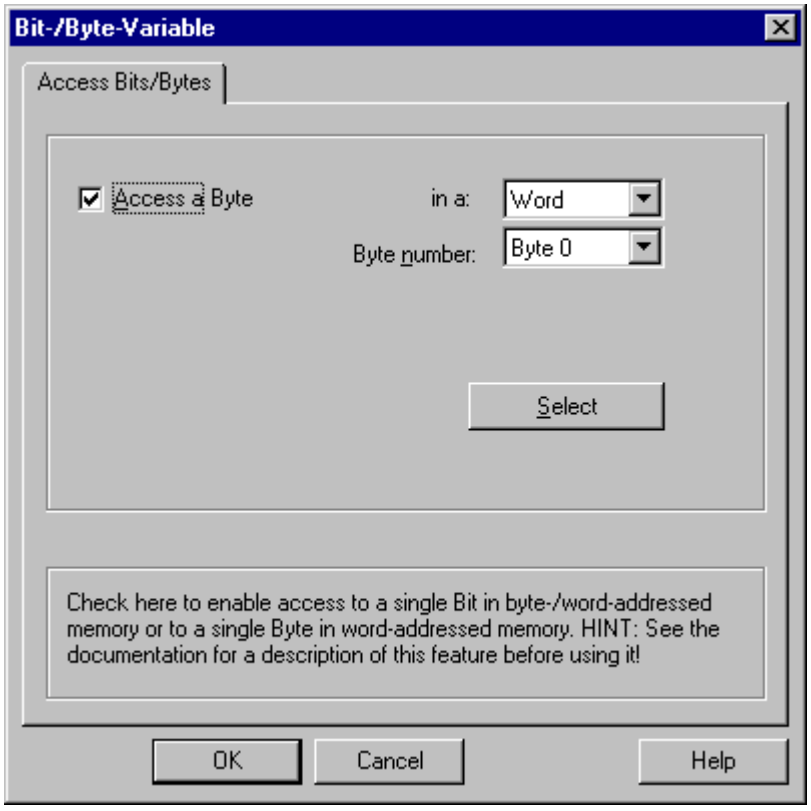
6. Click the "Select" button. The "Address properties" dialog is opened.
7. Select the addressing type of the PLC memory in the selection field.
8. Select the number of bit to be changed in the selection field.

11.3.3.4 How to Configure a Tag with Byte by Byte Access

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. In the field "Data Type", set the data type to "Unsigned 8-bit value" or "Signed 8-bit value".

5. Open the "Bit/Byte tag" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



6. Click the "Select" button. The "Address properties" dialog is opened.
7. Select the addressing type of the PLC memory in the selection field.
8. Select the number of byte to be changed in the selection field.

S5 Serial 3964R

12

12.1 WinCC channel "SIMATIC S5 Serial 3964R"

Introduction

The communication driver "SIMATIC S5 Serial 3964R" is implemented for the serial link between a WinCC station and a SIMATIC S5 automation system.

This chapter describes

- how to configure the data transfer with the "SIMATIC S5 Serial 3964R" channel.
- how to configure a connection and a tag.

Channel Unit

The communication driver has one channel unit for controlling a COM port for the serial link.

The following capability is available:

- Channel unit S5-RK512 (3964R) for serial communication via the 3964R or 3964 protocol.

See also

Configuring the tags (Page 227)

Configuring the "SIMATIC S5 Serial 3964R" channel (Page 225)

Data type of the tags (Page 224)

12.2 Data type of the tags

Introduction

Define the required tags for a logical connection. From the WinCC viewpoint, you can access the following data types:

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

12.3 Configuring the Channel

12.3.1 Configuring the "SIMATIC S5 Serial 3964R" channel

Introduction

The following steps are required for configuring the channel "SIMATIC S5 Serial 3964R".

See also

Configuring the tags (Page 227)

How to configure the connection (Page 225)

12.3.2 How to configure the connection

Introduction

The process connection using a serial connection is possible with the SIMATIC S5 automation system. On the automation system, the communication processor CP 544 or a second, plug-in serial port is used on the CPU module (module receptacle SI2).

No additional communication module is required in WinCC. Communication takes place by means of the default COM ports available on the system.

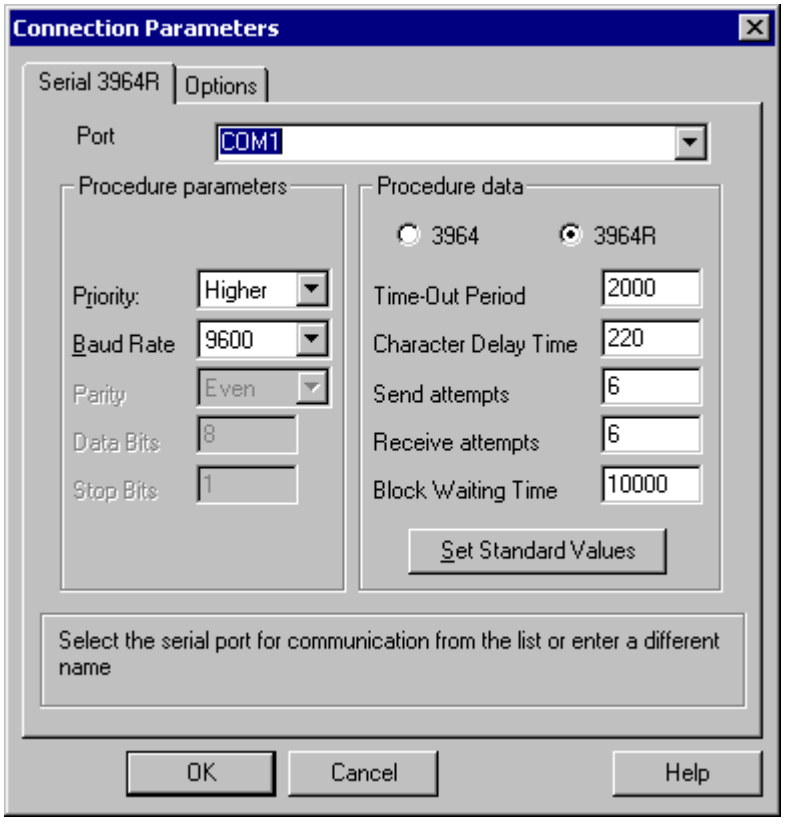
This serial link supports transmission rates of up to 19200 baud.

Note

When the SIMATIC S5 is actively sending with job type "Pseudowrite", the message length must not exceed 64 words.

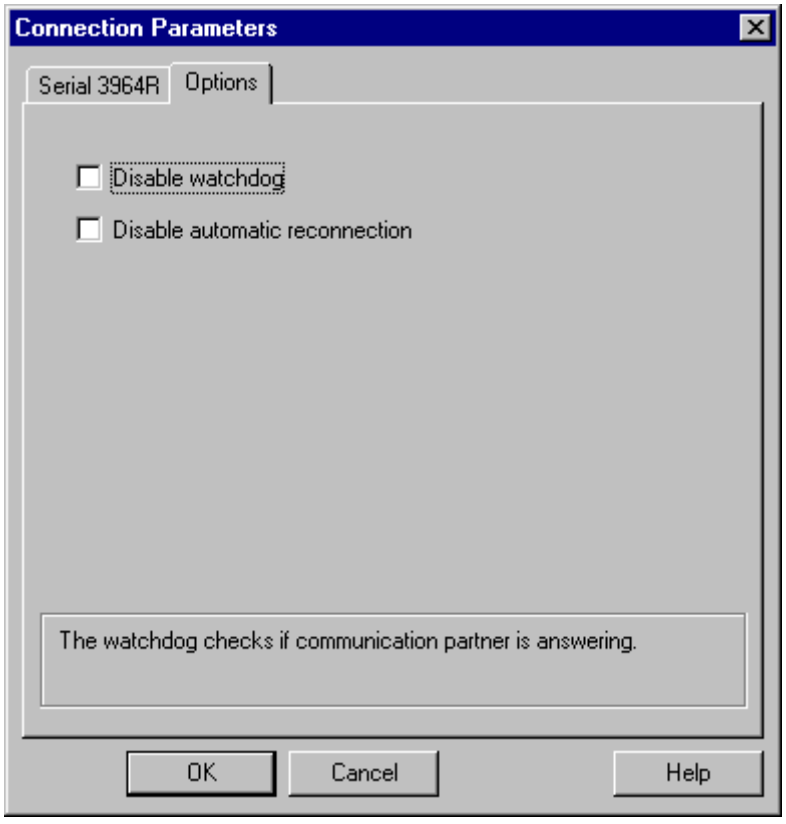
Procedure

1. Select a connection and select "Connection parameters" from the shortcut menu.
2. Select the Serial 3964R tab.



3. Select the communications port (COM1 or COM2) for the connection in the "Port" field.
4. Set the data transfer speed to the value used in the "Baud rate" field of the "Procedure parameters" area. The priority in the case of an initiation conflict (simultaneous line bid by WinCC and the automation system) is set in the "Priority" field.
   The set priority must be different from that set in SIMATIC S5.
5. In the "Procedure data" area, select either the "3964" or "3964R" line protocol. You should only change the default values for the procedure data (such as acknowledgment time, character delay time, etc.) in exceptional cases. Make sure that they match the parameters on the automation system.

6. Now select the "Options" tab.



7. You can disable cyclic life beat monitoring and disable the automatic reconnection on the "Options" tab.

12.3.3 Configuring the tags

12.3.3.1 Configuring the tags

Introduction

For a connection between WinCC and the AS via channel "SIMATIC S5 3964R", data types binary and byte can be defined within WinCC. The following describes how to configure a tag of these data types.

See also

How to Configure a Tag with Byte by Byte Access (Page 231)

How to configure a tag with bit-wise access (Page 229)

How to configure the address of the tag (Page 228)

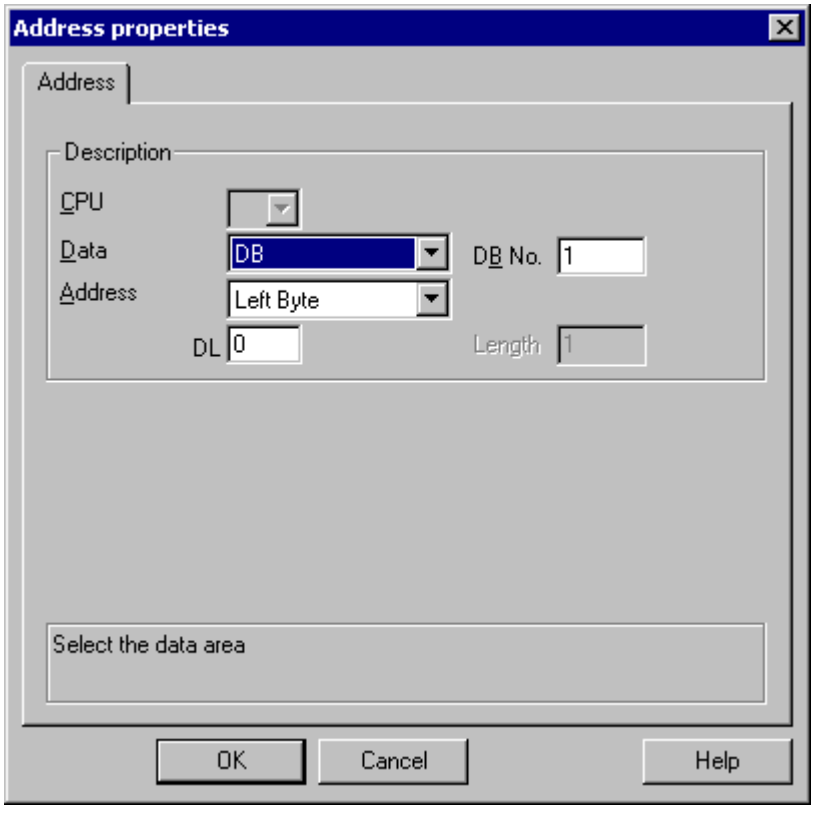
12.3.3.2 How to configure the address of the tag

Introduction

The tag address is entered according to the address structure of the SIMATIC S5.

Procedure

1. Select the tag and set the required data type for the tag (e.g. signed 8-bit value) in the field "Data Type".
2. Click the "Select" button. The "Address properties" dialog is opened.
   Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



3. Choose whether the tag is located in a data block, in an extended data block, in a flag area, an input range or an output range in the "Data Area" field.
4. If the tag is in a data block, the "DB No." field is also shown. Here, you enter the number of the data block.
5. The type of addressing is entered in the "Addressing" field. Normally, the default definition can be used.
6. Enter the address in the respective field (e.g. "DL ").

Note

Only read access is possible to the Inputs, Outputs, Timers and Counters address areas. Read and write access is possible to data blocks (DB, DX).

Do not use data word addresses which are greater than 255. Due to a system characteristic of the RK512, only data word addresses 0 to 255 are permissible.
It is possible to configure larger addresses, but this leads to data corruption on all configured tags of this connection.

Frequently, the memory in the PLC can only be accessed by byte or word. When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to change individual bits in the memory of the PLC as well. For this purpose, the addressed memory area is read from the PLC for every single write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the PLC's memory.

Note

Changes that have been made by the PLC in a read data area are overwritten when writing back into the data area.

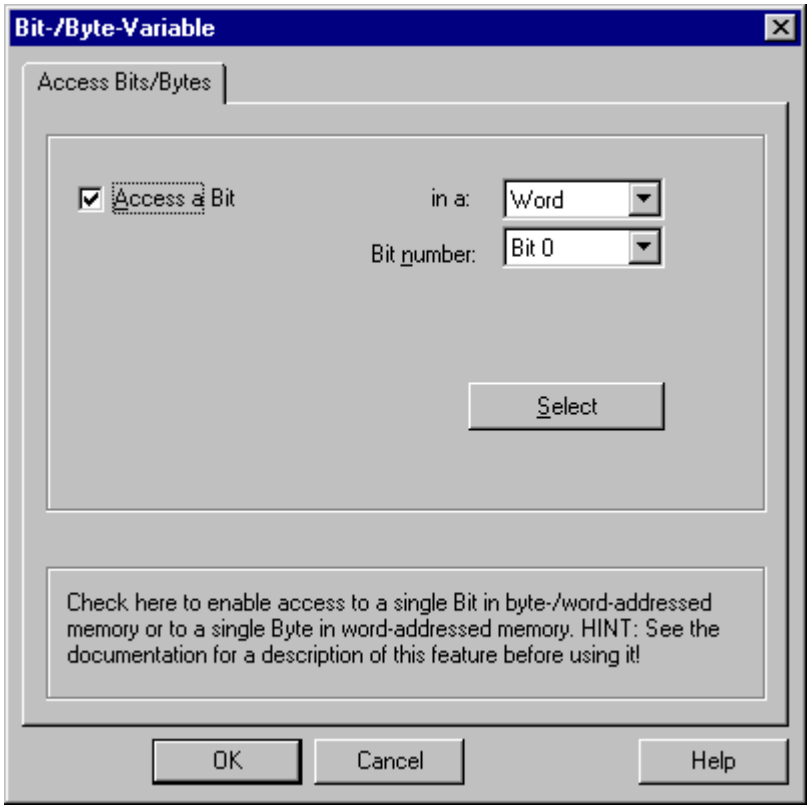
Depending on the type of tag, you can access the automation system's memory bit-wise or byte-wise.

12.3.3.3 How to configure a tag with bit-wise access

Procedure

1. Select the tag and set the "Binary tag" data type in the "Data Type" field.
2. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.
3. Click the "Select" button. The "Bit/Byte tag" dialog is opened.

4. Select the "Access to a bit" check box and define the addressing for the bit.



5. Click the "Select" button. The "Address properties" dialog is opened.
6. Select the addressing type of the PLC memory in the selection field.
7. Select the number of bit to be changed in the selection field.

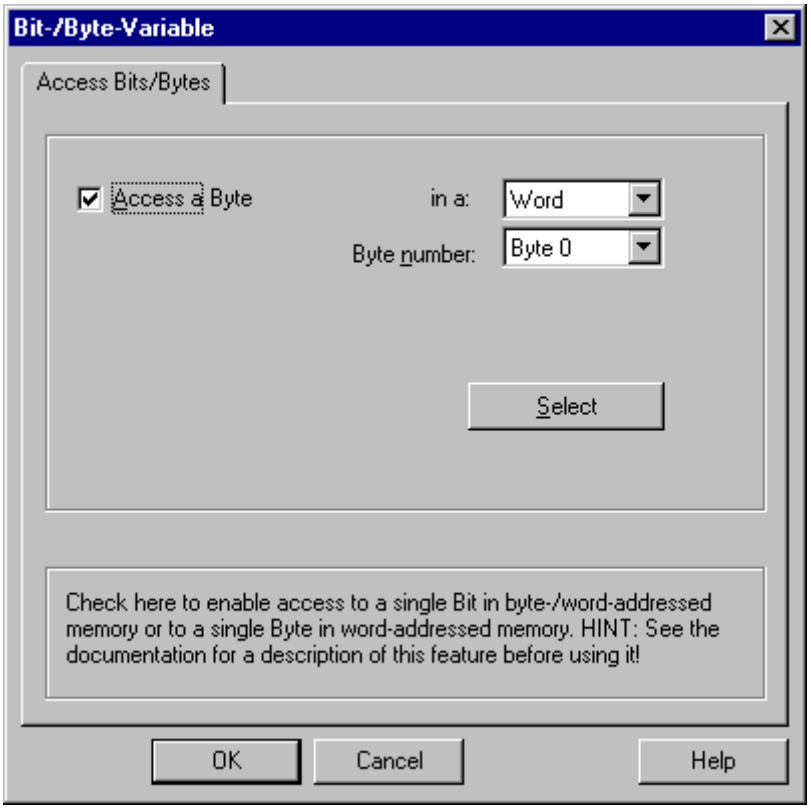
Note

With the S5, flags, inputs and outputs can be addressed byte by byte; data blocks (DB, DX) are addressed word by word.
Only read access is possible to the Inputs, Outputs, Timers and Counters address areas. Read and write access is possible to data blocks (DB, DX).

12.3.3.4 How to Configure a Tag with Byte by Byte Access

Procedure

1. Select the tag and set the data type in the "Data Type" field to "Unsigned 8-bit value" or "Signed 8-bit value".
2. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.
3. Click the "Select" button. The "Bit/Byte tag" dialog is opened.
4. Select the "Access to a byte" check box and define the addressing for the byte.



5. Click the "Select" button. The "Address properties" dialog is opened.
6. Select the addressing type of the PLC memory in the selection field.
7. Select the number of byte to be changed in the selection field.

Note

With the S5, flags, inputs and outputs can be addressed byte by byte; data blocks (DB, DX) are addressed word by word.
Only read access is possible to the Inputs, Outputs, Timers and Counters address areas. Read and write access is possible to data blocks (DB, DX).

SIMATIC S7 Protocol Suite 13

13.1 WinCC Channel "SIMATIC S7 Protocol Suite"

Introduction

The "SIMATIC S7 Protocol Suite" channel supports communication between a WinCC station and the SIMATIC S7 automation systems. The suite supports a variety of protocols and types of networks.

This section shows you how to:

- Configure various connections and tags for the channel
- Create a sample project
- Use the channel's special functions such as the AR_SEND function, raw data tags and software redundancy

Principle of operation

The "SIMATIC S7 Protocol Suite" channel is used to link SIMATIC S7-300 and SIMATIC S7-400 automation systems.

Depending on the communication hardware used, the system supports connections via the following channel units:

- Industrial Ethernet and Industrial Ethernet (II): For communication via a communication processor (such as CP 1612 A2; CP 1613 A2) with SIMATIC NET Industrial Ethernet.
- MPI: For communication via the internal MPI interface of a programming device (e.g. PG 760/PC RI45), or via an MPI communications processor or communications module (e.g. CP 5613 A3).
- Named Connections: For communication via a symbolic connection with STEP 7. These symbolic connections are configured using STEP 7 and are needed, for example, for high-availability communication with the AS S7-400 in combination with redundancy in H/F systems.
- PROFIBUS and PROFIBUS (II): For communication via a communications processor (such as CP 5613 A3) with SIMATIC NET PROFIBUS.
- Slot-PLC: To communicate with a Slot PLC (e.g. WinAC Pro) that is installed as a PC card in the WinCC computer.
- Soft-PLC: To communicate with a Software PLC (e.g. WinAC Basis), that is installed as an application on the WinCC computer.
- TCP/IP: to communicate with networks using the TCP/IP protocol.

You can find more information on channel and tag diagnostics under "Communication Diagnostics".

Note

SINUMERIK S7-300

You can configure communication with a SINUMERIK S7-300 as of WinCC V8.1 with the SINUMERIK (Page 411) channel.

After migrating to WinCC V8.1, you will find previously configured connections to a SINUMERIK S7-300 under the SinumerikNC channel.

Detailed procedures

You can find more information and detailed examples of channel configuration in the "WinCC V6 Communication Manual":

- http://support.automation.siemens.com/WW/view/en/21320307 (http://support.automation.siemens.com/WW/view/en/21320307)

You can find more information on channel and tag diagnostics under "Communication diagnostics".

See also

Software Redundancy - Connection-specific internal tags (Page 320)

"SIMATIC S7 Protocol Suite" Channel - Configuration (Page 240)

Overview of the supported data types (Page 239)

Channel unit selection (Page 235)

http://support.automation.siemens.com/WW/view/en/21320307 (http://support.automation.siemens.com/WW/view/en/21320307)

13.2 Channel unit selection

Introduction

To create a communication connection for an existing or planned network, a selection must be made for:

- A channel unit of the channel
- A suitable communications processor for the WinCC station
- A suitable communications module for a specific automation system

This section provides an overview of the different possible variations.

There are two different types of communications processors for WinCC:

- Communications processors for the so-called Hardnet.
  They have their own microprocessors and reduce the load on the computer CPU.
  It is possible to use two different protocols at the same time (multi-protocol operation).
- Communications processors for the so-called Softnet.
  They do not have their own microprocessors.
  Only one protocol can be used at a time (mono-protocol operation).

Assignment of the channel unit

The table below shows an assignment of a channel unit of the "SIMATIC S7 Protocol Suite" channel to a network and automation system.

Channel unit of the channel	Communication network	Automation system
MPI	MPI	S7-300 and S7-400
PROFIBUS + PROFIBUS (II)	PROFIBUS	S7-300 and S7-400
Industrial Ethernet + Industrial Ethernet (II)	Industrial Ethernet	S7-300 and S7-400
TCP/IP	Industrial Ethernet via TCP/IP	S7-300 and S7-400
Named connections	Industrial Ethernet or PROFIBUS	S7-400 H/F Systems
Slot PLC	"Soft K-Bus" (internal)	PC internal
Soft PLC	"Soft K-Bus" (internal)	PC internal

MPI

For communication with the S7-300 and S7-400 automation systems via MPI, the "MPI" channel unit is available in the "SIMATIC S7 Protocol Suite" channel.

The MPI network largely corresponds to the PROFIBUS network with specified parameters and limitations on station counts and transfer rates. The same communications processors and modules are used for communication via MPI as for the PROFIBUS network. The same communication protocols are also used.

Communication connections of the automation systems

The communication of the S7-300 and S7-400 automation systems over an MPI network can take place via the AS-internal MPI interface or using a suitable communications module. The table shows the recommended components.

System	CPU or communications module (recommended)
S7-300	CPU 31x CP 342-5 CP 343-5
S7-400	CPU 41x CP 443-5 Ext. CP 443-5 Basic

Communications processors for WinCC

The following table shows communication processors recommended for connecting a WinCC station to the MPI network. Only one communications processor per WinCC computer can be used for MPI communication. Each card also has suitable driver software for the respective communication protocol.

Communications processor (WinCC)	Design/Type
CP 5613 A3	PCI card/Hardnet
CP 5612	PCI card/Softnet

PROFIBUS

For communication with the S7-300 and S7-400 automation systems via PROFIBUS, the "PROFIBUS" and "PROFIBUS II" channel units are available in the "SIMATIC S7 Protocol Suite" channel.

The channel units support communication using Hardnet and Softnet modules.

Communication connections of the automation systems

The communication of the S7-300 and S7-400 automation systems over a PROFIBUS network can take place via the AS-internal interface or using a communications module. The table shows the recommended components.

System	CPU or communications module
S7-300	CPU 31x CP 342-5 CP 343-5
S7-400	CPU 41x CP 443-5 Ext. CP 443-5 Basic

Communications processors for WinCC

The following table shows the communications processors recommended for connecting a WinCC station to PROFIBUS.

The "PROFIBUS" channel units support communication using Hardnet and Softnet cards.

Use of up to two of these modules is possible in a WinCC station.

Each communications processor also has suitable driver software for the respective communication protocol.

Communications processor (WinCC)	Design/Type
CP 5613 A3	PCI card/Hardnet
CP5623	PCI card/Hardnet
CP 5612	PCI card/Softnet
CP5622	PCI card/Softnet

Number of connections

With WinCC, up to 8 MPI connections or PROFIBUS SOFTNET connections are licensed, for example, CP5622. Additional PROFIBUS SOFTNET licenses are not required.

With a corresponding SIMATIC NET license, you can also create more than 8 PROFIBUS connections. You need PROFIBUS Hardnet for this, for example, CP5623.

Industrial Ethernet and TCP/IP

In WinCC, multiple channel units for communication via Industrial Ethernet are available in the "SIMATIC S7 Protocol Suite" channel.

- "Industrial Ethernet" and "Industrial Ethernet (II)" channel units for "ISO" protocol with S7 functions
- "TCP/IP" channel unit for "ISO-on-TCP" protocol with S7 functions

The channel units support communication using Hardnet and Softnet modules.

Communication modules for automation systems

For communication of the S7-300 or S7-400 automation systems via Industrial Ethernet with "ISO" or "ISO-on-TCP" protocol, these are equipped with a suitable communications module. The table shows the recommended components.

System	Communications module for Industrial Ethernet	Communications module for TCP/IP protocol
S7-300	CP 343-1	CP 343-1 TCP
S7-400	CP 443-1	CP 443-1 TCP CP 443-1 IT

Communications processors for WinCC

The communication of a WinCC station via Industrial Ethernet with the "ISO" or "ISO-on-TCP" protocol takes place using the recommended communications processors listed in the table.

Each communications processor also has suitable driver software for the respective communication protocol.

Communications processor (WinCC)	Design/Type
CP 1612 A2	PCI card/Softnet
CP 1613 A2	PCI card/Hardnet

13.3 Overview of the supported data types

Introduction

For configuring a tag, you need to define data type and type conversion according to the data format in AS.

The table shows the data types supported by the channel and the use of type conversions.

Supported data types

Data Types	Type conversion
Binary tag	No
Signed 8-bit value	Yes
Unsigned 8-bit value	Yes
Signed 16-bit value	Yes
Unsigned 16-bit value	Yes
Signed 32-bit value	Yes
Unsigned 32-bit value	Yes
Floating-point number 32-bit IEEE 754	Yes
Text tag, 8-bit font	No
Raw data type	No

You will find additional information about type conversion in the "Communication" section.

13.4 Configuring the Channel

13.4.1 "SIMATIC S7 Protocol Suite" Channel - Configuration

Introduction

This section will show you how to configure the "SIMATIC S7 Protocol Suite" channel.

1. Installing the Channel
2. Channel unit selection
3. Configuring a connection
4. Tag configuration

System parameter configuration

Further information regarding the diagnosis of the channel, connection and tags can be found under "Communication Diagnosis".

See also

System Parameters of the Channel Unit (Page 272)

Configuring the tags (Page 267)

Channel units of the "SIMATIC S7 Protocol Suite" channel (Page 250)

13.4.2 How to configure the "SIMATIC S7 Protocol Suite" channel

Introduction

This section will show you how to install the "SIMATIC S7 Protocol Suite" channel.

1. Installing the Channel
2. Channel unit selection
3. Creating a connection
4. Inserting a tag
5. Configuring the system parameters in a customized WinCC installation

Prerequisites:

- The communication module is built in.
- The hardware driver has been installed.
- Cable connection to AS exists.

Procedure

1. In the navigation area of the tag management, select the entry "Add new driver" in the shortcut menu of node "Tag Management".
2. Select the driver "SIMATIC S7 Protocol Suite".
   The channel is created.
   The communication driver and the associated channel units are displayed in Tag Management.
3. Select the desired channel unit and select the "New Connection" entry in the shortcut menu.
4. Enter the name of the connection.
5. To create the system tags for connection establishment and connection status, select the "Create tags for activation/deactivation" entry in the shortcut menu of the connection.
   The following tags are created in the internal tag group "ConnectionStates":
   - @<Connectionname>@ForceConnectionStateEx
   - @<Connectionname>@ConnectionStateEx
6. Click the "Tags" tab below the table area.
7. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
8. Select the desired data type in the "Data Type" field.
   You have the option of defining a start value and a substitute value for the tag in the "Properties" area.
   If you want a detailed description for configuring tags of the connection of a particular channel unit, close the dialog and continue with the topic "Configuring tags" within the channel unit involved.
9. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.
10. Close both of the dialogs by clicking the "OK" button.
11. If your WinCC system and the communication hardware is non-standard, you also need to set the system parameters to the non-standard values.
    Additional information on this topic may be found under "System parameters".

See also

How to download AS symbols offline (Page 241)

13.4.3 How to download AS symbols offline

Introduction

You can configure the following S7 channels offline:

- SIMATIC S7 Protocol Suite
- SIMATIC S7-1200, S7-1500 Channel

To this purpose export, for example, the data records from the existing TIA Portal project and load the export file in the WinCC project.

Supported export formats

The following file formats are supported for the import:

Format	Contents	Description
*.bin	Binary data	Export from the WinCC Tag Management: • "Tag Management" view > Shortcut menu of the connection: AS Symbols > Save to file Not supported by the "SIMATIC S7 Protocol Suite" channel.
*.sdz	Structured export	Export from the WinCC Tag Management: • "Symbols" view > Menu: Edit > Export Also exports the structure information from the navigation area.
*.zip	TIA Portal export file	Export from the TIA Portal with the "SIEMENS SIMATIC SCADA Export" tool

"SIEMENS SIMATIC SCADA Export" for TIA Portal

To export data records from a TIA Portal project, use the "SIEMENS SIMATIC SCADA Export" tool.

In the TIA Portal project, select the "Export to SIMATIC SCADA" entry in the shortcut menu of the PLC.

The tool for the various TIA Portal versions is available for download in Industry Online Support:

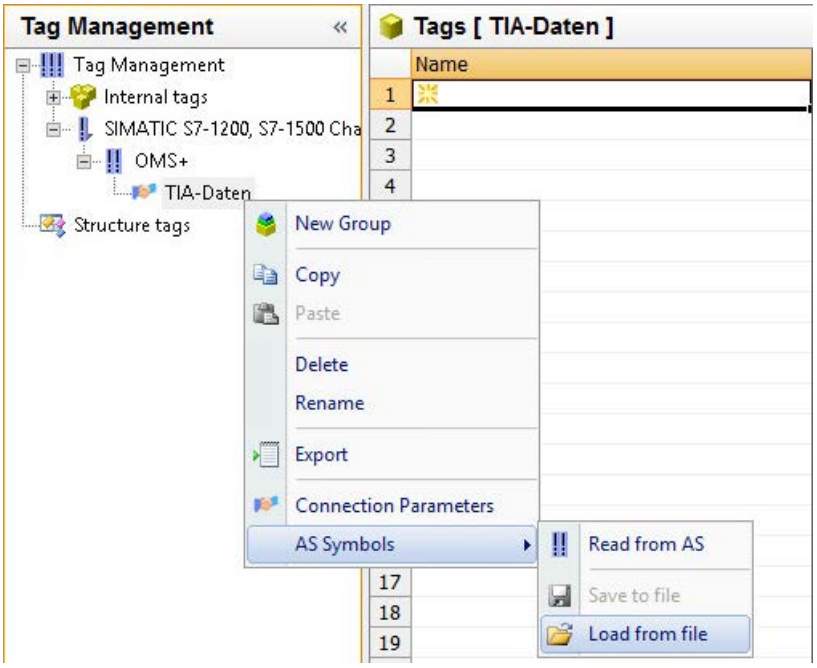
- Industry Online Support: Download "SIMATIC SCADA Export for TIA Portal" (ID 109748955) (https://support.industry.siemens.com/cs/ww/en/view/109748955)
- Industry Online Support: "SIMATIC SCADA Export" documentation (ID 101908495) (https://support.industry.siemens.com/cs/ww/en/view/101908495)

Requirement

- The AS was compiled in the TIA Portal.
- The corresponding configuration data of the PLC is exported and is available, for example, as a .zip file.
- The communications processor and associated hardware driver are installed in the WinCC project.
- A connection is created in the "SIMATIC S7-1200, S7-1500 Channel" or "SIMATIC S7 Protocol Suite".
- The "Tag Management" editor is open.

Procedure

1. Select "AS Symbols > Load from file" from the shortcut menu of the connection.



2. Select the desired data records to be loaded.
   The available controller data is loaded.

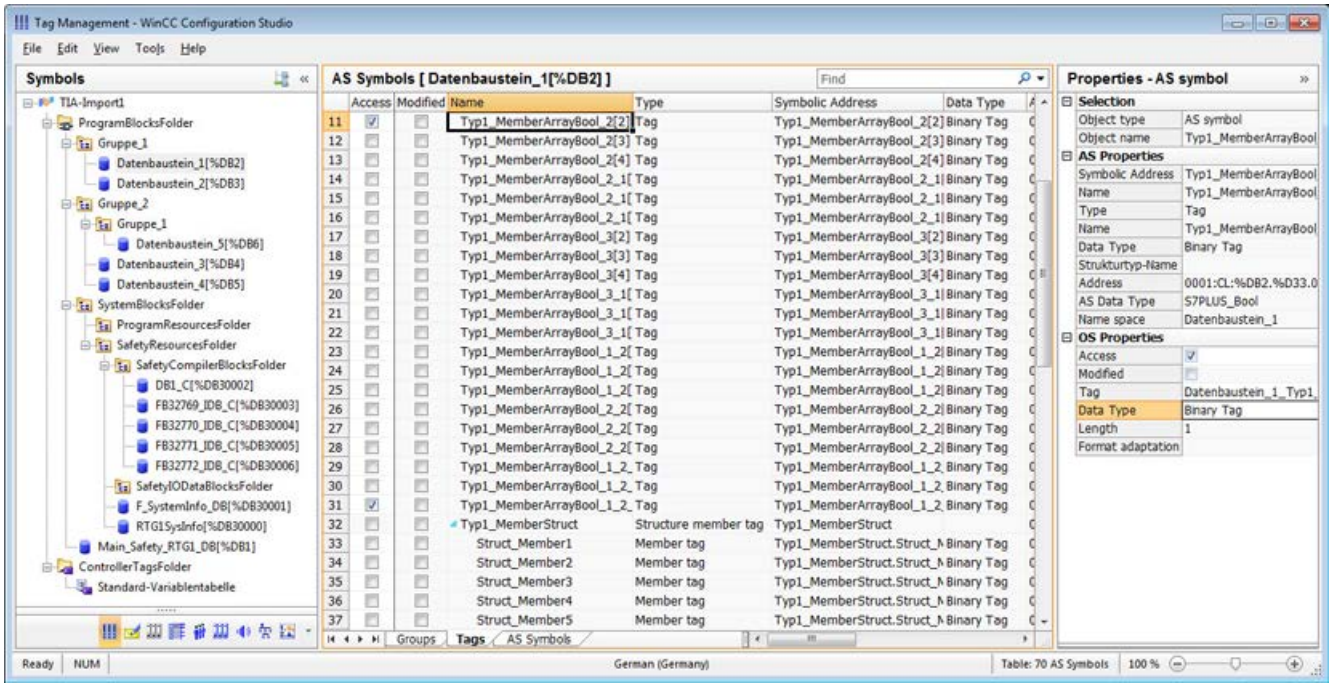
Result

The configuration has been imported and the "Symbols" view opens.

The loaded data is displayed in the "AS Symbols" tab in the table area and is available for further processing.

If the loaded data also contains structures, the "AS structures" tab is displayed additionally.

After the editor is closed, the "AS Symbols" and "AS structures" tabs are hidden once again.



Display of the symbols

You use the following button to switch in Tag Management between the default view and the "Symbols" view: 

The button is available only after the data records have been loaded.

Navigation area

The representation of the data in the structure tree corresponds to the hierarchy from the TIA Portal.

Table area

The check boxes in the "Modified" column are selected automatically when a found WinCC tag does not match the AS tags. This also allows you to filter by these.

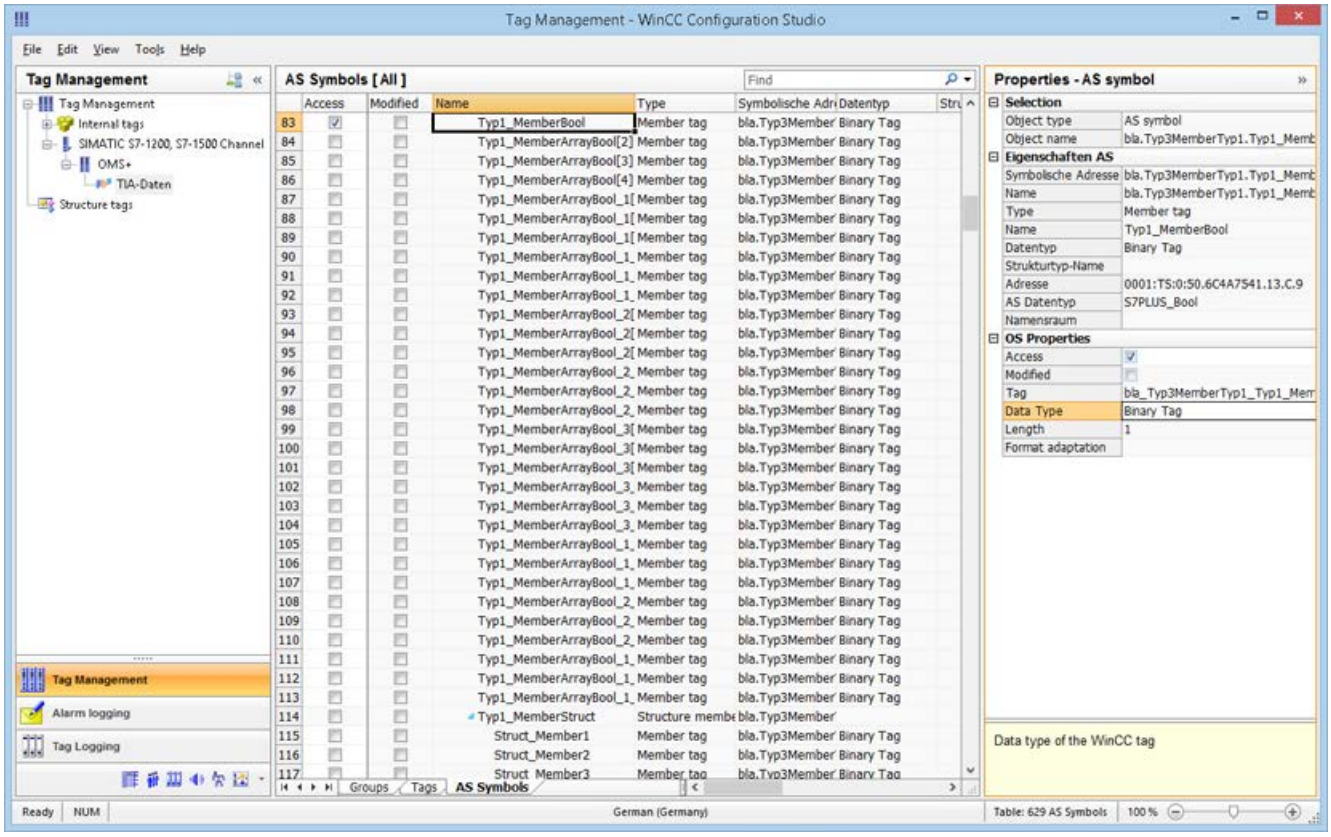
By selecting the check boxes in the "Access" column, you create a WinCC tag from the found AS tags.

AS symbols in Tag Management

You also have access to the AS symbols in Tag Management via the "AS Symbols" tab.

In contrast to the data block-specific "Symbols" view, all the available tags of the controller are shown here.

This view also shows previously configured tags that are no longer present on the AS.



See also

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 240)

How to export AS project data (Page 248)

How to configure AS structures (Page 245)

Industry Online Support: "SIMATIC SCADA Export" documentation (ID 101908495) (https://support.industry.siemens.com/cs/ww/en/view/101908495)

Industry Online Support: Download "SIMATIC SCADA Export for TIA Portal" (ID 109748955) (https://support.industry.siemens.com/cs/ww/en/view/109748955)

13.4.4 How to configure AS structures

Introduction

If you load AS symbols, structured data types of the control system (UDT) are also imported. Structures of the "STRUCT" type are not taken into consideration.

The procedure depends on the communication channel:

- SIMATIC S7 Protocol Suite:
  - Load from file
- SIMATIC S7-1200, S7-1500 Channel
  - Load from file
  - Load from AS

AS structures in Tag Management

The AS structures are displayed in the default view and in the "Symbols" view on the "AS structures" tab.

You have the following possibilities to use the AS structures in WinCC:

- Create a WinCC structure type for the AS structure tag.
  The structure is created as a structure type under "Structure tags" in the WinCC Tag Management.
  A structure type element is also created for each contained "Tag type member".
- Assign a WinCC structure type to the AS structure tag.
  Then select a structure type element of the selected structure type for each "Tag type member".

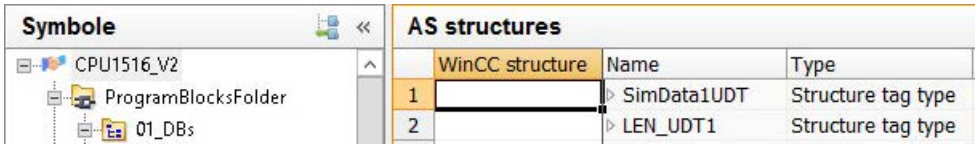
You change the name of the WinCC structure type and the structure type elements in the Tag Management. The assignment of the AS structure is automatically adjusted.

Requirement

- You have access to the configuration data of the PLC by one of the following methods:
  - A connection to the PLC is established in Runtime.
  - The exported configuration data is available, for example, as a zip file.
- A connection is created in the "SIMATIC S7-1200, S7-1500 Channel" or "SIMATIC S7 Protocol Suite".

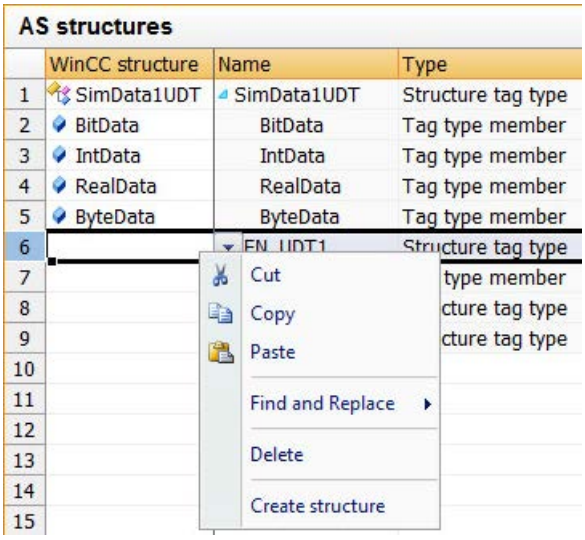
Procedure

1. Load the AS symbols via "Read from AS" or "Load from file".
   The loaded messages are displayed in the "Symbols" Tag Management view.
   The loaded structures are displayed on the "AS structures" tab.
   The structure names are transferred when loading from the AS.

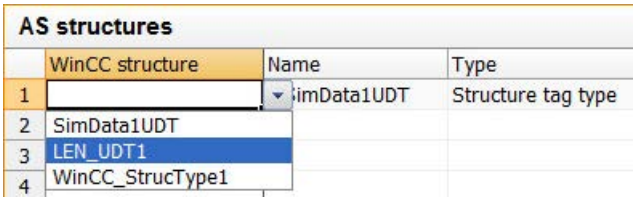


2. Click "AS structures".
   To display the elements of a structure, click the arrow in front of the structure name.

3. Select the entire row of a structure and select the "Create structure" entry in the shortcut menu.



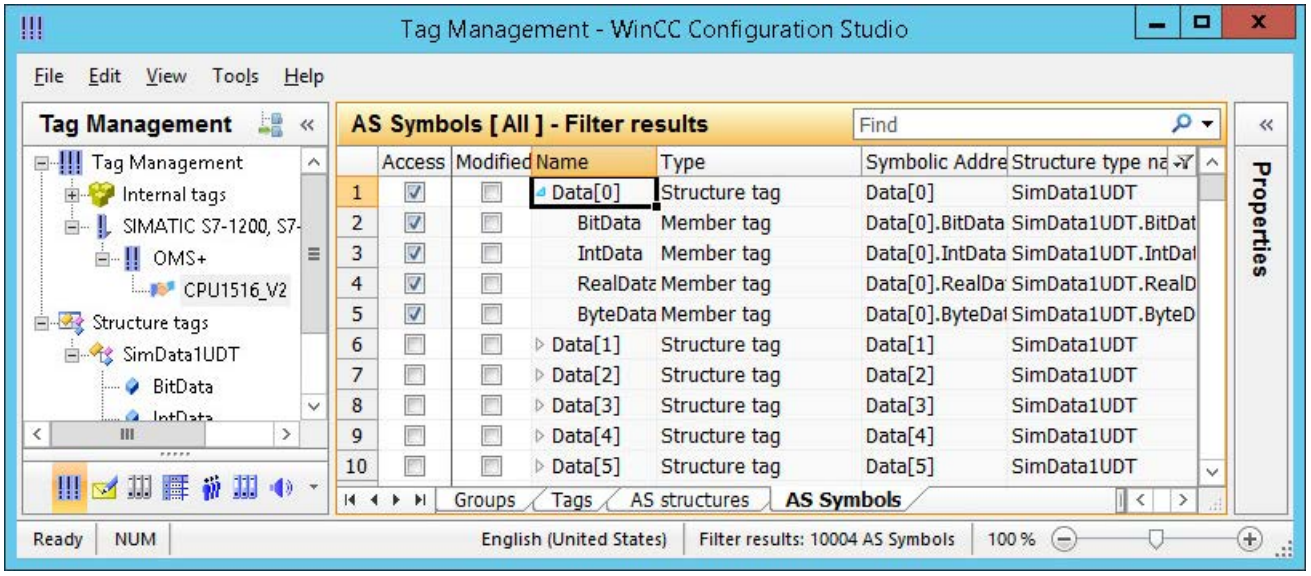
Alternatively, select a structure type that has already been created in the WinCC Tag Management.
Then assign a structure type element to the "Tag type member".



A structure type is created in the WinCC Tag Management for each "Structure tag type" of the AS structures.
A structure type element is created for each "Tag type member".

4. Select the "AS Symbols" tab in the "Tag Management" view.

5. To only have structure tags and member tags displayed, filter for the desired AS structure in the "Structure type name" column.



6. To access an AS structure tag in the WinCC Tag Management, activate the "Access" field.
   The contained member tags are automatically activated.
   The AS structure tag is created as a structure tag in the WinCC Tag Management.

Result

Through the structure types and structure tags in WinCC Tag Management you have access to the AS structure tags.

In this way you can, for example, access AS structures in WinCC faceplate types and represent them in faceplate instances.

See also

How to download AS symbols offline (Page 241)

How to export AS project data (Page 248)

13.4.5 How to export AS project data

Exporting AS symbols

You use the export files for the offline configuration.

You can export AS project data to the following formats:

Communication channel	Exported data	Format of the export file
SIMATIC S7-1200, S7-1500 Channel	AS symbols and AS structures	Binary data: *.bin Structured export: *.sdz
SIMATIC S7 Protocol Suite	AS symbols and AS structures	Structured export: *.sdz

Requirement

- A connection is created in the "SIMATIC S7-1200, S7-1500 Channel" or "SIMATIC S7 Protocol Suite".
- You have loaded AS project data and configured it in WinCC.

Procedure: Exporting binary data

1. Select the connection in the Tag Management.
2. Select the "AS Symbols > Save to file" entry from the shortcut menu.
   The "Export" dialog opens.
3. Select the storage path and enter a file name.
   Close the dialog with the "Export" button.
   The configuration data is exported as a binary data set to a .bin file.

Procedure: Exporting structured data

1. Select the "Symbols" view in the Tag Management.
2. Select the "Edit > Export" menu command.
3. Select the storage path and enter a file name.
   Close the dialog with the "Export" button.
   The configuration data is exported to an *.sdz file.
   The structured export also contains the structure information from the navigation area.

See also

How to download AS symbols offline (Page 241)

How to configure AS structures (Page 245)

13.4.6 Channel units

13.4.6.1 Channel units of the "SIMATIC S7 Protocol Suite" channel

Introduction

The following chapters describe how to configure the channel units and a corresponding connection. There can be multiple connections in the same channel unit.

See also

"TCP/IP" channel unit (Page 264)

"Soft PLC" channel unit (Page 263)

"Slot PLC" channel unit (Page 261)

Channel Units "PROFIBUS (I + II)" (Page 258)

"Named Connections" channel unit (Page 255)

"MPI" channel unit (Page 253)

Channel Units "Industrial Ethernet" + "Industrial Ethernet (II)" (Page 250)

13.4.6.2 "Industrial Ethernet (I+II)" channel units"

Channel Units "Industrial Ethernet" + "Industrial Ethernet (II)"

Principle of operation

The channel unit "Industrial Ethernet" is used to connect WinCC to the S7 automation systems via the Industrial Ethernet. Communication is possible via the communications modules (CP), e.g. in the case of automation system S7-300 via CP 343-1 and in the case of S7-400 via CP 443-1.

In WinCC different communications processors can be used, e.g. CP 1613 A2. A second communications processor can be addressed via the "Industrial Ethernet (II)" channel unit. Because communication takes place via the "ISO" transport protocol, it is not necessary to configure the logical connection in the local database.

The function and configurations regarding these channel units are identical.

Unit-typical terminology

Communications processor

A communications processor (CP) is a module via which the communication of the WinCC computer with a specific network takes place.

"ISO" transport protocol

ISO transport is a layer of the ISO-OSI reference model and offers services related to the transfer of data via connections. The transport layer handles data flow control, blocking and acknowledgment tasks.

The protocol defines the structure of the data traffic with regards to content on the physical line. It defines, among other things, the mode of operation, the procedure when establishing a connection, data backup or the transmission speed.

Industrial Ethernet

The Industrial Ethernet is the most efficient subnet in the industrial environment. It is suitable for the factory and cell levels and facilitates the exchange of large data volumes over large distances between a large number of stations.

The Industrial Ethernet is standardized as open communication network in accordance with the IEEE 802.3 standard. Its prime advantages are its speed, simple extendibility and openness as well as high availability and worldwide utilization. The configuration process requires a minimum of effort.

See also

Configuring the tags (Page 267)

How to configure a "Industrial Ethernet" channel unit connection (Page 251)

How to configure a "Industrial Ethernet" channel unit connection

Introduction

In addition to the channel unit, WinCC also requires a logical connection to communicate with the PLC. All the specific parameters are defined while establishing a logical connection.

For S7 automation systems, a communications module, e.g. a CP 343-1 in the S7-300 or a CP 443-1 in the S7-400, is used for the communication.

A communications processor, e.g. CP 1613 A2, is used in WinCC. A second communications processor can be addressed/increased via the "Industrial Ethernet II" channel unit.

You can find more information on channel, connection, and tag diagnostics under "Communication Diagnostics".

Note

S7-300/S7-400: Rack/slot number of the CPU

When using an S7-300 or S7-400 with an external communications module, you must enter the rack/slot number of the CPU.

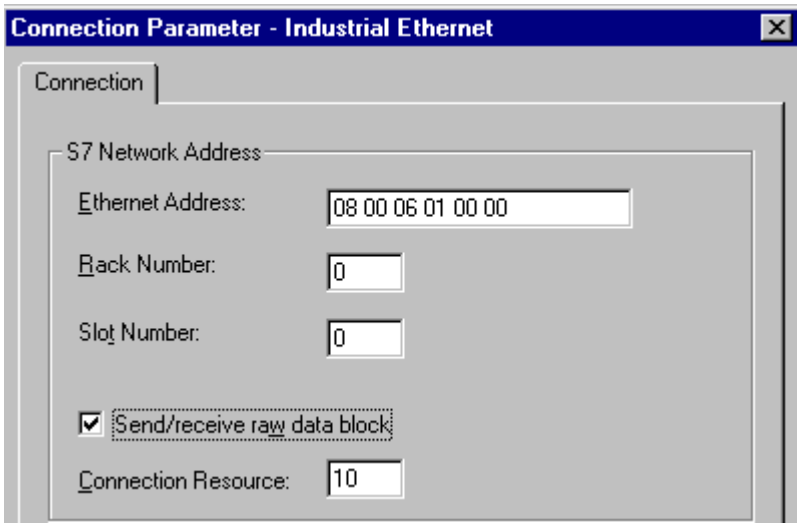
If the wrong rack or slot number is entered, the communication connection will not be established.

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

Procedure

1. Select the entry "New Connection" in the shortcut menu of the channel unit "Industrial Ethernet".
   A new connection is created.
2. Enter a connection name, e.g. "Test_Ind_Eth".
3. Select the "Connection parameters" connection from the shortcut menu.
   The "Connection parameters - Industrial Ethernet" dialog opens.



4. Enter the station address of the automation system on the bus in the field "Ethernet Address".
5. Enter the number of the rack in which the CPU that is to be addressed is located in the "Rack Number" field.
6. The CPU's slot number in the specified rack must be entered in the "Slot Number" field.
7. Activate the check box "Send/Receive Raw Data Block" if you wish to transfer BSEND/BRCV data blocks via the connection.
   If the check box is active, the field "Connection Resource" can be edited.
   Enter the hexadecimal value for the connection resource.
   This connection resource will be assigned by STEP7 when the connection is configured in the PLC.
8. Close both of the dialogs by clicking the "OK" button.

See also

Configuring the tags (Page 267)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 240)

13.4.6.3 "MPI" channel unit

"MPI" channel unit

Principle of operation

The "MPI" channel unit is used to connect WinCC to the SIMATIC S7-300 and S7-400 automation systems via MPI.

This can be done in WinCC via

- the internal MPI interface of programming devices, such as PG 760/PC RI45
- a communications processor, e.g. CP 5613 A3 (PCI card)

The so-called MPI module (ISA card) is also suitable - it exists but is no longer available. It has been replaced by the communications processors.

In the AS, the connection is made via the MPI interface of the CPU or a corresponding communications module.

Unit-typical terminology

MPI

MPI means Multi Point Interface and is a communication connection in which multiple stations are possible. The connection to the communication network is made as follows:

- In the AS via the MPI interface of the CPU or using a communications module
- In WinCC via the built-in MPI interface, e.g. of a programming device, or using a communications processor (network card).

Communications processor

A communications processor (CP) is a module via which the communication of the WinCC computer with a specific network takes place.

See also

Configuring the tags (Page 267)

How to configure a "MPI" channel unit connection (Page 253)

How to configure a "MPI" channel unit connection

Introduction

In addition to the channel unit, WinCC also requires a logical connection to communicate with the PLC. All the specific parameters are defined while establishing a logical connection.

S7-300 and S7-400 automation systems either use the internal MPI interface or a communications module such as CP 342-5 (SIMATIC S7-300) or CP 443-5 (SIMATIC S7-400).

If WinCC is installed on a PG 760/PC RI45, the internal MPI interface can be used; otherwise, you need to have a built-in MPI module. Alternatively, you can also use a communications module.

You can find more information on channel, connection, and tag diagnostics under "Communication Diagnostics".

Note

S7-300/S7-400: Rack/slot number of the CPU

When using an S7-300 or S7-400 with an external communications processor, you must enter the rack/slot number of the CPU.

If the wrong rack or slot number is entered, the communication connection will not be established.

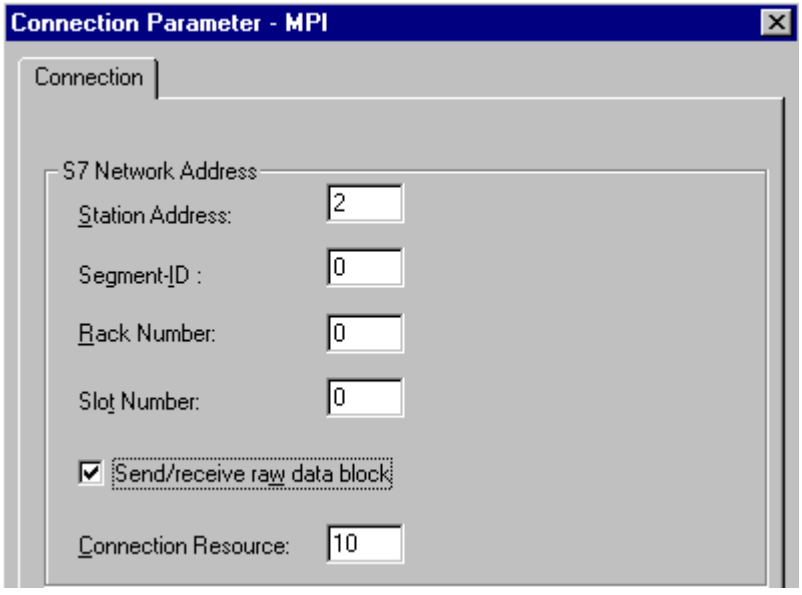
When using an S7-300, the rack/slot number = 0 must be entered for the link via the internal MPI interface of the CPU.

Requirements

- The "SIMATIC S7 Protocol Suite" (communication) driver must be integrated into the project.

Procedure

1. Select the entry "New Connection" in the shortcut menu of the channel unit "MPI". A new connection is created.
2. Enter "Test_MPI" as connection name.
3. Select the "Connection parameters" connection from the shortcut menu. The "Connection parameters - MPI" dialog opens.



4. Enter the station address in the Station Address field of the automation system on the bus in the appropriate field.
5. The field "Segment ID" is currently not supported. The value must remain at "0".
6. Enter the number of the rack in which the CPU that is to be addressed is located in the "Rack Number" field.
7. Enter the "Slot Number" of the CPU in the specified rack.
8. Activate the check box "Send/Receive Raw Data Block" if you wish to transfer BSEND/BRCV data blocks via the connection.
   If the check box is flagged, the field "Connection Resource" will also be active.
   Enter the hexadecimal value for the connection resource.
   This connection resource will be assigned by STEP7 when the connection is configured in the PLC.
9. Close both of the dialogs by clicking the "OK" button.

See also

Configuring the tags (Page 267)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 240)

13.4.6.4 "Named Connections" channel unit

"Named Connections" channel unit

How it works

This channel unit is used for setting up a symbolic connection configured using STEP 7. WinCC can thus access redundant and non-redundant connection via a symbolic connection name. The symbolic connections are required, for example, for high availability communication using S7-400 PLC in connection with redundancy in H/F systems.

The symbolic connection names are configured in STEP 7 using the NETPRO application. The connection names, connection parameters and the application names are stored in a database (*.XDB). This databased is automatically stored by the PLC/OS Engineering Tool "Mapper" in the corresponding WinCC project directory; however, it can also be copied outside this directory, for e.g. if you are not using the "Mapper".

Note

There should only be one XDB file per communication participant in the WinCC system.

Hence, a XDB file should not be copied and used on multiple WinCC computers.

You have the following options to activate this database in WinCC:

- If the XDB file is located outside the project directory (for e.g. because the Mapper tool is not used), you need to enter the path and name of the XDB file in the "Set PG/PC interface" (Control Panel) in the STEP 7 Configuration tab before starting WinCC.
  On starting WinCC, this XDB file is read from this external directory provided no file exists within the project directory. This procedure is helpful when multiple projects have to use the same centrally stored database.
- If the Mapper tool is used, it automatically copies the XDB file to the WinCC project directory. On starting WinCC and opening the project, the data is read from the S7 channel and entered in the registration database of Windows.

Thereafter, a connection can be configured in WinCC by assigning one of the symbolic connection names to the selected application name.

Note

The application name and connection name can also be entered manually.

It is necessary to check the correct writing of the name configured in STEP 7 because there is no name validation in the CS mode.

This may be necessary in the following cases, for example:

- No XDB file is available for the symbolic connection name. In this case transfer the configuration directly to the "Components configurator".
- The project is to be transferred to another computer.

Typical unit terminology

Communication processor

A communications processor (CP) is a module that supports communication between the PLC and a specific network.

See also

Configuring the tags (Page 267)

How to configure a "Named Connections" channel unit connection (Page 256)

SiePortal: PG/PC interface for the WinCC communication with the control system (ID 38717202) (https://support.industry.siemens.com/cs/ww/en/view/38717202)

How to configure a "Named Connections" channel unit connection

Introduction

In addition to the channel unit, WinCC also requires a logical connection to communicate with the S7-400 PLC via a symbolic connection.

For setting up a logical connection, one of the symbolic connection names listed in the "Connection name" field is assigned to a selected application name.

The symbolic connection names and application names are configured in STEP 7.

Further information regarding the diagnostics of the channel, connection and tags can be found under "Communication Diagnostics".

Note

The application name and connection name can also be entered manually.

It is necessary to check the correct writing of the name configured in STEP 7 because there is no name validation in the CS mode.

This may be necessary in the following cases, for example:

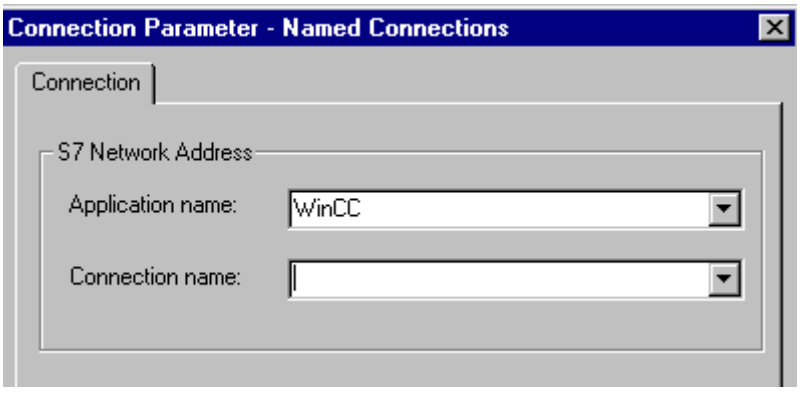
- No XDB file is available for the symbolic connection name. In this case transfer the configuration directly to the "Components configurator".
- The project is to be transferred to another computer.

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

Procedure

1. Select the entry "New Connection" in the shortcut menu of the channel unit "Named Connections".
   A new connection is created.
2. Enter a connection name, for example, "Test_NC".
3. Select the "Connection parameters" in the shortcut menu of the connection.
   The "Connection parameter - Named Connections" dialog opens.



4. In the Application name field, enter the application name that has been configured in STEP 7. Default value is WinCC.
5. In the Connection name field, enter the symbolic connection name that has been configured in STEP 7.
6. Close both of the dialogs by clicking the "OK" button.

See also

Configuring the tags (Page 267)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 240)

13.4.6.5 "PROFIBUS (I+II)" channel units

Channel Units "PROFIBUS (I + II)"

Principle of operation

The channel unit is used to connect WinCC to the SIMATIC S7-300 and S7-400 automation systems via a PROFIBUS network.

For S7 automation systems, a communications module is used, e.g. CP 342-5 in the S7-300 or CP 443-5 in the S7-400.

A communications processor, e.g. CP 5613 A3, is used in WinCC.

A second communications processor can be addressed via the "PROFIBUS II" channel unit. As a result, the maximum number of connections is increased.

Unit-typical terminology

PROFIBUS

PROFIBUS is an open, vendor-neutral communication system for the cell and field levels and is designed for a maximum of 127 stations. PROFIBUS is based on the European Standard EN 50170, Volume 2, PROFIBUS. PROFIBUS uses token passing with lower-level master-slave as the access method.

Communications processor

A communications processor (CP) is a module via which the communication of the WinCC computer with a specific network takes place.

See also

Configuring the tags (Page 267)

How to configure a "PROFIBUS" channel unit connection (Page 258)

How to configure a "PROFIBUS" channel unit connection

Introduction

In addition to the channel unit, WinCC must also have a logical connection to communicate with the PLC. All the specific parameters are defined while establishing a logical connection.

For S7 automation systems, a communications module is used, e.g. a CP 342-5 in an S7-300 or a CP 443-5 in an S7-400.

A communications processor, e.g. CP 5613 A3, is used in WinCC. A second communications processor can be addressed via the "PROFIBUS II" channel unit.

You can find more information on channel, connection, and tag diagnostics under "Communication Diagnostics".

Note

Connection in the off state

When starting up the PROFIBUS communication, PROFIBUS errors can occur if the communication processor was connected to the PROFIBUS while the WinCC computer was ON.

Therefore, it is recommended that the computer be switched OFF before connecting it to the PROFIBUS.

Otherwise, (in accordance with the PROFIBUS standard) multiple tokens may be generated on the bus, which will cause a bus error.

S7-300/S7-400: Rack/slot number of the CPU

When using an S7-300 or S7-400 with an external communications module, you must enter the rack/slot number of the CPU.

If the wrong rack or slot number is entered, the communication connection will not be established.

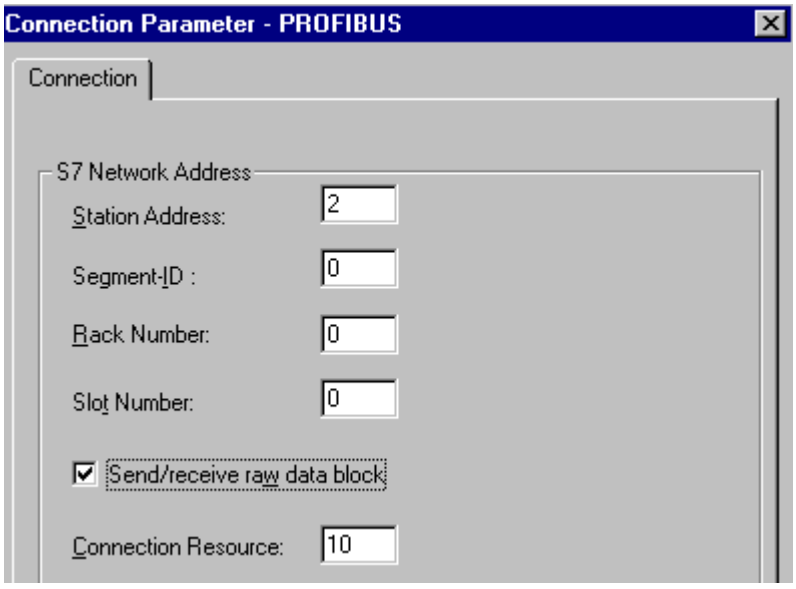
Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

Procedure

1. Select the entry "New Connection" in the shortcut menu of the channel unit "PROFIBUS". A new connection is created.
2. Enter "Test_PROFIBUS" as connection name.

3. Select the "Connection parameters" connection from the shortcut menu.
   The "Connection parameters - PROFIBUS" dialog opens.



4. Enter the "Station Address" of the automation system on the bus in the appropriate field.
5. The field "Segment ID" is currently not supported. The value must remain at "0".
6. Enter the "Rack Number" in which the CPU that is to be addressed is located.
7. Enter the "Slot Number" of the CPU in the specified rack.
8. Activate the check box "Send/Receive Raw Data Block" if you wish to transfer BSEND/BRCV data blocks via the connection.
   If the check box is flagged, the field "Connection Resource" will also be active.
   Enter the hexadecimal value for the connection resource.
   This connection resource will be assigned by STEP7 when the connection is configured in the PLC.
9. Close both of the dialogs by clicking the "OK" button.

See also

Configuring the tags (Page 267)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 240)

13.4.6.6 "Slot PLC" channel unit

"Slot PLC" channel unit

Principle of Operation

Channel unit "Slot PLC" serves the communication between WinCC and up to four Slot PLC (WinAC Pro) installed in the WinCC computer. Since the Slot PLC has an integrated interface, no additional communication hardware is required for the connection between WinCC and Slot PLC.

See also

Configuring the tags (Page 267)

How to Configure a "Slot PLC" Channel Unit Connection (Page 261)

How to Configure a "Slot PLC" Channel Unit Connection

Introduction

In order to communicate with the installed SPS cards, WinCC requires a logical connection in addition to the channel unit. All the specific parameters are defined while establishing a logical connection.

Further information regarding the diagnostics of the channel, connection and tags can be found under "Communication Diagnostics".

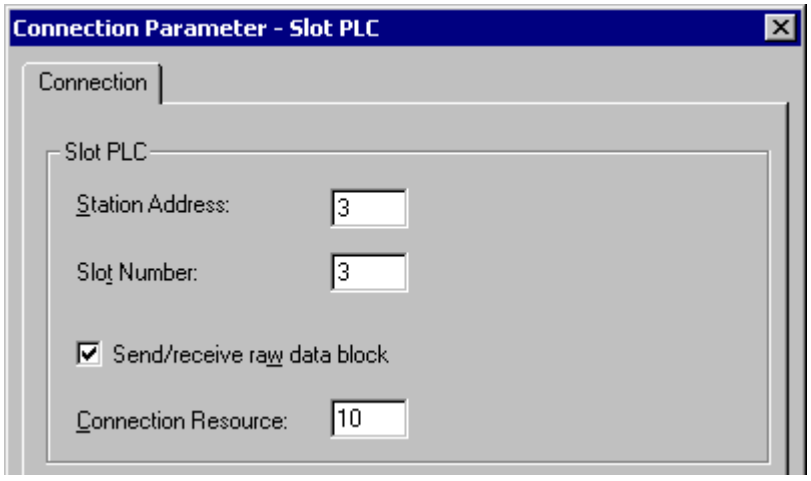
Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.
- If several Slot PLC are to be configured, Slot PLC Version 3.4 is required.

Procedure

1. Select the entry "New connection" in the shortcut menu of the channel unit "Slot PLC". A new connection is created.
2. Enter a connection name, for example, "Test_SPLC".

3. Select the "Connection parameters" connection from the shortcut menu.
   The "Connection parameters - Slot PLC" dialog opens.



4. In the field "Station address", enter the station address of the Slot PLC on the Soft K-Bus.
5. In the field "Slot No.", enter the number of the slot in which the Slot PLC is installed.
6. Activate the check box "Send/Receive Raw Data Block" if you wish to transfer BSEND/BRCV data blocks via the connection.
7. If the check box is flagged, the field "Connection Resource" will also be active. Enter the hexadecimal value for the connection resource. This connection resource will be assigned by STEP 7 when the connection is configured within the PLC.
8. Close both of the dialogs by clicking the "OK" button.

Note

Connection parameters "Station Address" and "Slot No." must be identical for several installed Slot PLCs and must start with "Slot No." "3".

See also

Configuring the tags (Page 267)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 240)

13.4.6.7 "Soft PLC" channel unit

"Soft PLC" channel unit

Principle of Operation

Channel unit "Slot PLC" serves the communication between WinCC and a Soft PLC (WinAC Basic) installed in the WinCC computer. No other communication hardware is required for connecting WinCC to the Soft PLC.

See also

Configuring the tags (Page 267)

How to configure a connection on the "Soft PLC" channel unit (Page 263)

How to configure a connection on the "Soft PLC" channel unit

Introduction

In addition to the channel unit, WinCC must also have a logical connection to communicate with the Soft PLC. All the specific parameters are defined while establishing a logical connection.

Further information regarding the diagnostics of the channel, connection and tags can be found under "Communication Diagnostics".

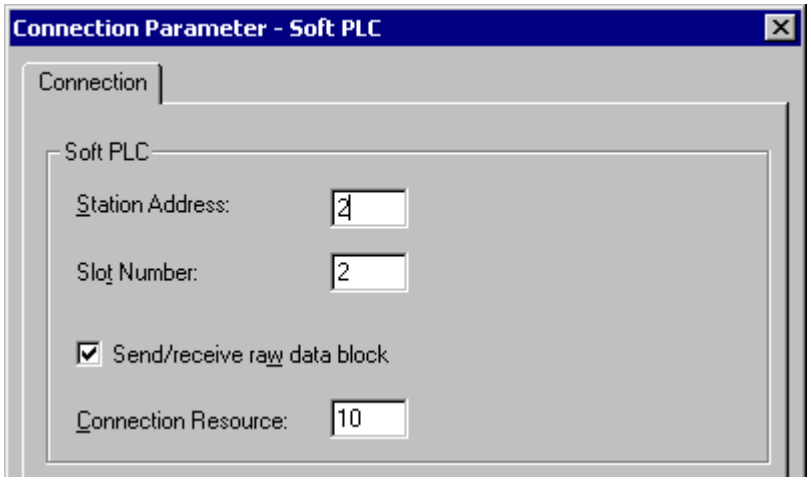
Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

Procedure

1. Select the entry "New connection" in the shortcut menu of the channel unit "Soft PLC". A new connection is created.
2. Enter a connection name, for example, "Test_SOFTPLC".

3. Select the "Connection parameters" connection from the shortcut menu.
   The "Connection parameters - Slot PLC" dialog opens.



4. In the field "Station address", enter the station address of the Soft PLC on the Soft K-Bus.
5. In the field "Slot No.", enter the number of the slot. The slot number is configured in the hardware configuration of Soft PLC and is required when you want to use multiple Soft PLC in the same WinCC computer.
6. Activate the check box "Send/Receive Raw Data Block" if you wish to transfer BSEND/BRCV data blocks via the connection.
7. If the check box is flagged, the field "Connection Resource" will also be active. Enter the hexadecimal value for the connection resource. This connection resource will be assigned by STEP 7 when the connection is configured within the PLC.
8. Close both of the dialogs by clicking the "OK" button.

See also

Configuring the tags (Page 267)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 240)

13.4.6.8 "TCP/IP" channel unit

"TCP/IP" channel unit

Principle of operation

The "TCP/IP" channel unit is used to connect WinCC to the SIMATIC S7-300 and S7-400 automation systems via an Industrial Ethernet with "ISO-on-TCP Transport" protocol.

The protocol corresponds to the standard TCP/IP with the extension RFC 1006. This extension is necessary, since TCP/IP uses communication where there is no blocking of data.

In the case of the S7-300 automation system, communication takes place via a communications module, e.g. CP 343-1 TCP, and via CP 443-1 TCP or CP 443-1 IT in the case of S7-400.

A communications processor, e.g. CP 1613 A2, is used in WinCC.

Because communication takes place via the ISO-on-TCP transport protocol, it is not necessary to configure the logical connection in the local database.

Unit-typical terminology

Communications processor

A communications processor (CP) is a module via which the communication of the WinCC computer with a specific network takes place.

ISO transport protocol

ISO transport is a layer of the ISO-OSI reference model and offers services related to the transfer of data via connections. The transport layer handles data flow control, blocking and acknowledgment tasks.

The protocol defines the structure of the data traffic with regards to content on the physical line. It defines, among other things, the mode of operation, the procedure when establishing a connection, data backup or the transmission speed.

Industrial Ethernet

The Industrial Ethernet is the most efficient subnet in the industrial environment. It is suitable for the factory and cell levels and facilitates the exchange of large data volumes over large distances between a large number of stations.

The Industrial Ethernet is standardized as open communication network in accordance with the IEEE 802.3 standard. Its prime advantages are its speed, simple extendibility and openness as well as high availability and worldwide utilization. The configuration process requires a minimum of effort.

See also

Configuring the tags (Page 267)

How to configure a "TCP/IP" channel unit connection (Page 265)

How to configure a "TCP/IP" channel unit connection

Introduction

In addition to the channel unit, WinCC also requires a logical connection to communicate with the PLC. All the specific parameters are defined while establishing a logical connection.

Use the following communications modules, for example, for the communication:

- S7-300 automation system: CP 343-1 TCP
- S7-400 automation system: CP 443-1 TCP or CP 443-1 IT

A communications processor, e.g. CP 1613 A2, is used in WinCC.

You can find more information on channel, connection, and tag diagnostics under "Communication Diagnostics".

Note

S7-300/S7-400: Rack/slot number of the CPU

When using an S7-300 or S7-400 with an external communications module, you must enter the rack/slot number of the CPU.

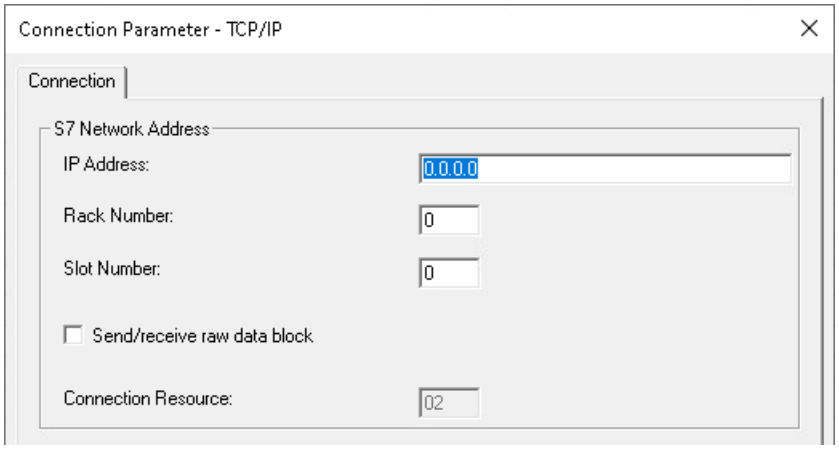
If the wrong rack or slot number is entered, the communication connection will not be established.

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

Procedure

1. Select the entry "New Connection" in the shortcut menu of the "TCP/IP" channel unit.
   A new connection is created.
2. Enter "Test_TCP" as connection name.
3. Select the "Connection parameters" connection from the shortcut menu.
   The "Connection parameters - TCP/IP" dialog opens.



4. Enter the Internet protocol address of the automation system on the bus in the field "IP Address".
5. Enter the number of the rack in which the CPU that is to be addressed is located in the "Rack Number" field.
6. The CPU's slot number in the specified rack must be entered in the "Slot Number" field.

7. Activate the check box "Send/Receive Raw Data Block" if you wish to transfer BSEND/BRCV data blocks via the connection.
   If the check box is flagged, the field "Connection Resource" will also be active.
   Enter the hexadecimal value for the connection resource.
   This connection resource will be assigned by STEP7 when the connection is configured in the PLC.
8. Close both of the dialogs by clicking the "OK" button.

See also

Configuring the tags (Page 267)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 240)

13.4.7 Configuring the tags

13.4.7.1 Configuring the tags

Introduction

The following sections describe how to configure the tags. It is different in the way the data area in the PLC is accessed and the data type of the WinCC tags.

Further information regarding the diagnosis of the channel, connection and tags can be found under "Communication Diagnosis".

See also

How to Configure a Text Tag (Page 271)

How to Configure a Tag with Word by Word Access (Page 270)

How to Configure a Tag with Byte by Byte Access (Page 268)

How to Configure a Tag with Bit by Bit Access (Page 267)

13.4.7.2 How to Configure a Tag with Bit by Bit Access

Introduction

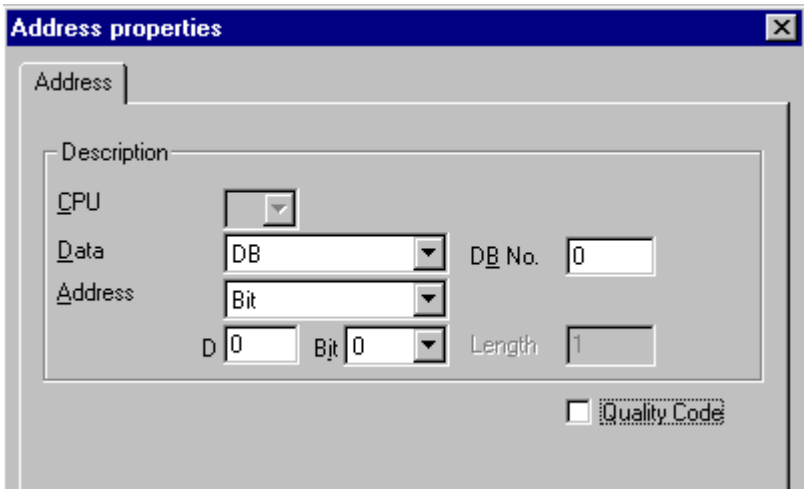
This section shows you how to configure a tag for bit by bit access for the address area in PLC.

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.
- A connection e.g. "Test_Ind_Eth" must be created in a channel unit, e.g. "Industrial Ethernet".

Procedure

1. Select the connection "Test_Ind_Eth".
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   In the "Name" field, enter "ETH_Var1_bit" as the name for the tag.
4. Set the "Binary tag" data type in the "Data Type" field.
5. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.
   In the "Data area" field, set the data area of the automation system where the data are located. If you select "DB" as data area, enter the number of the data block in the enabled "DB No." field.



6. You cannot change the "Bit" entry in the Addressing field because it is defined by the Binary Variable data type of the WinCC tag.
7. Enter the byte and bit address in the two fields below it. The label on the left field depends on the entry in the Data Area field, for e.g. "D" for data area "DB" and Binary Variable as type.
8. Check the quality code check-box if the tag is with quality code that is to be used in WinCC. For this, the code must also exist in the PLC. The check-box is enables only if the data area is selected as "DB".
9. Close both of the dialogs by clicking the "OK" button.

See also

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 240)

13.4.7.3 How to Configure a Tag with Byte by Byte Access

Introduction

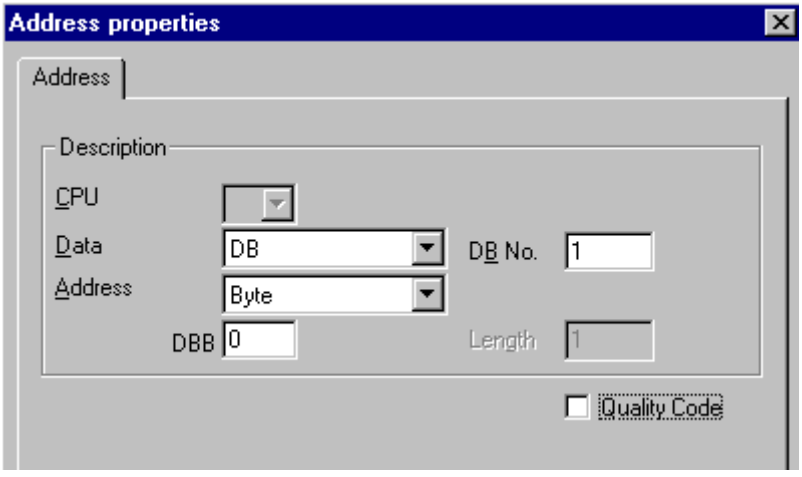
This section shows you how to configure a tag for byte by byte access for the address area in PLC.

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.
- A connection e.g. "Test_Ind_Eth" must be created in a channel unit, e.g. "Industrial Ethernet".

Procedure

1. Select the connection "Test_Ind_Eth".
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   In the "Name" field, enter "ETH_Var1_byte" as the name for the tag.
4. In the "Data Type" field, set the data type to "Unsigned 8-bit value".
5. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.
   In the "Data area" field, set the data area of the automation system where the data are located. If you select "DB" as data area, enter the number of the data block in the enabled "DB No." field.



6. You cannot change the "Bit" entry in the Addressing field because it is defined by the "Unsigned 8-bit value" data type of the WinCC tag.
7. Enter the byte address in the field below. The label on the left field depends on the entry in the Data Area field, for e.g. "D" for data area "DB" and "Unsigned 8-bit value" as type.
8. Check the quality code check-box if the tag is with quality code that is to be used in WinCC. For this, the code must also exist in the PLC. The check-box is enables only if the data area is selected as "DB".
9. Close both of the dialogs by clicking the "OK" button.

See also

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 240)

13.4.7.4 How to Configure a Tag with Word by Word Access

Introduction

This section shows you how to configure a tag for word by word access for the address area in PLC.

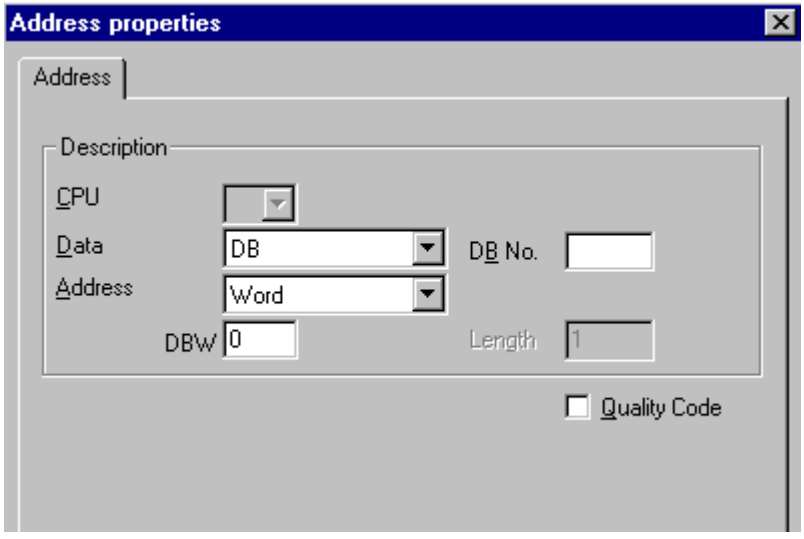
This procedure is also applicable for tags with length of 4 byte ("double word") and more.

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.
- A connection e.g. "Test_Ind_Eth" must be created in a channel unit, e.g. "Industrial Ethernet".

Procedure

1. Select the connection "Test_Ind_Eth".
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   In the "Name" field, enter "ETH_Var3_word" as the name for the tag.
4. In the "Data Type" field, set the data type to "Unsigned 16-bit value".
5. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.
   In the "Data area" field, set the data area of the automation system where the data are located. If you select "DB" as data area, enter the number of the data block in the enabled "DB No." field.



6. You cannot change the "Word" entry in the Addressing field because it is defined by the "Unsigned 16-bit value" data type of the WinCC tag.
7. Enter the numeric value of the address in the Addressing field. The label on the left field depends on the entry in the Data Area field, for e.g. "DBW" for "Unsigned 16-bit value" as type.

8. Check the quality code check-box if the tag is with quality code that is to be used in WinCC. For this, the code must also exist in the PLC. The check-box is enables only if the data area is selected as "DB".
9. Click "OK" to close all open dialogs.

See also

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 240)

13.4.7.5 How to Configure a Text Tag

Introduction

This section show you how to configure a text tag.

For a text tag in the SIMATIC S7 Protocol Suite channel, WinCC only supports S7 string type consisting of a control word and the actual user data of the string:

- To configure a text tag in WinCC, enter the address of the control word that exists in the PLC memory before the user data. The first byte of the control word contains the customized maximum length of the string, the second byte the actual length.
- To insert the data structure in the PLC memory, you must note that the length of the text tag configured in WinCC is extended by 2 bytes of the control word. If the data structures of the text tag are inserted in the memory directly one after the other, then the subsequent data will get overwritten.
- New mapping is required for switching the PCS-7 version from V4.01 to V5.0 SP1 because in the versions before V5.0 the address of the user data was also mentioned while configuring the text tags; from version V5.0 onwards the address of the control word is to be given.
- While reading, the control word is read along with the user data and the current length is evaluated in the second byte. Only the user data according to the current length included in the second control byte is transferred at the text tags of WinCC.
- While writing, the actual length of the string is ascertained ("0" character) and the control byte with the current length is sent to the PLC along with the user data.

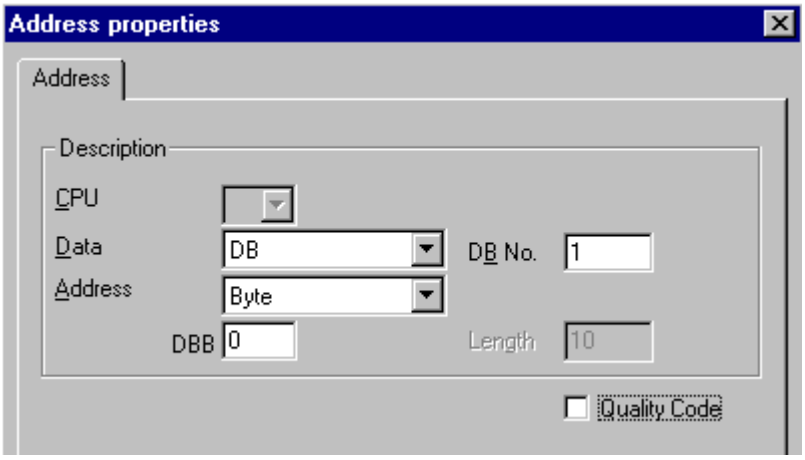
Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.
- A connection e.g. "Test_Ind_Eth" must be created in a channel unit, e.g. "Industrial Ethernet".

Procedure

1. Select the connection "Test_Ind_Eth".
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   In the "Name" field, enter "ETH_Var3_Text" as the name for the tag.
4. In the Data Type field, set "Text tag, 8-bit font" as the data type.

5. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.
   In the "Data area" set the data area of the PCL where the data is located. If you select "DB" as data area, enter the number of the data block in the enabled "DB No." field.



6. The entry in the Addressing field can only be changed to Byte or Word because it is defined by the "Text tag, 8-bit font" data type of the WinCC tag.
7. Enter the numeric value of the address in the Addressing field. Mention the address of the control word. The label on the left field depends on the entry in the Data Area field, for e.g. "DBW" for Word as type.
8. Check the quality code check-box if the tag is with quality code that is to be used in WinCC. For this, the code must also exist in the PLC. The check-box is enables only if the data area is selected as "DB".
9. Click "OK" to close all open dialogs.

See also

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 240)

13.4.8 System parameters

13.4.8.1 System Parameters of the Channel Unit

Introduction

If you require a configuration that deviates from the WinCC standard settings, you will therefore be able to make all the required changes using the "System Parameter" dialog of the channel unit.

The following can be modified:

- the logical device name
- the use of the cyclic read service in the AS

Logical Device Name

Communication between WinCC and the automation system takes place via logical device names. These names are assigned during the installation of the communications card and are unit-specific. This field will be filled with a default unit-specific entry, e.g. "MPI" in the case of channel unit "MPI".

Using cyclic reading services in the PLC

It is possible to specify whether or not the cyclic read services of S7-PLC (also referred to as cyclic tag services) should be used. These cyclic read services group the tags that are to be read cyclically into individual request and transfer these to the PLC. The PLC will transfer the required data immediately on receipt of the request and will also transfer the data each time the cycle time elapses.

When the cyclic read services are activated, modification transfers can also be used. The data will then only be transferred when the values have changed. The function must be supported by the PLC.

Note

The system parameters on the SIMATIC S7 and Unit tabs are unit-specific and can thus be set separately for each channel unit of the channel.

See also

How to Change the Logical Device Name (Page 276)

How to Configure the System Parameters (Page 274)

Cyclic read services in PLC (Page 273)

13.4.8.2 Cyclic read services in PLC

Introduction

In the system parameters of the "SIMATIC S7 Protocol Suite" channel, it is possible to specify whether or not the cyclic read services of the S7-AS (also referred to as cyclic tag services) should be used. These cyclic read services group the tags that are to be read cyclically into individual requests and transfer them to the AS. The AS will transfer the required data immediately on receipt of the request and will also transfer the data each time the cycle time elapses. When the requested data is no longer required, e.g. in the case of a screen change, WinCC will delete the cyclic read service in the AS.

In normal cases, use should be made of the cyclic read services in the AS. For this reason, the corresponding check box is already activated (default setting) in the system parameters of the channel unit. This setting should only be changed if you do not wish to use the cyclic services.

Change-driven transfer can only be used when the cyclic read services are activated. The data will only then be transferred from the AS when a value has changed and only once per AS cycle. This feature must be supported by the AS.

The use of cyclic read services and change-driven transfers relieves both the AS and AS-OS communication, since read requests need not be continually sent to the AS and processed there.

In the case of acyclic read services, the tags that are to be read are combined in an individual request and transferred to the AS. The AS only sends the required data once. The formation of the cycle for the request is carried out by WinCC.

Number of cyclic read services in a CPU

The number of cyclic read services will depend on the resources that are available in the S7 AS. A maximum of four cyclic services are available for an S7-300 and a maximum of 32 for an S7-416 or 417. This number applies for all stations communicating with the AS, i.e. if several WinCC systems are communicating with an S7 AS, they will have to share the resources that are available. If the maximum number of services is exceeded, access to a further cyclic read service will be refused. WinCC will then have to request this data using acyclic read requests and will also have to perform cycle management itself.

Requesting external tags in scripts

Utilization of the cyclic read services has no influence on the initial update once a picture has been opened, provided the selected picture does not contain any scripts that request external tags using the "GetTagWord()" function. If scripts are executed with "GetTagWord()" when a screen is selected, the incorrect configuration of this script could result in new tag requests being sent to this channel repeatedly following a screen change. If external tags are required in a script, "Tag" should be entered as a trigger event.

13.4.8.3 How to Configure the System Parameters

Introduction

In this section, we will show you how to configure the system parameters of the "SIMATIC S7 Protocol Suite" channel.

The "System Parameters" dialog comprises two tabs:

- "SIMATIC S7" tab
- "Unit" tab

The system parameters on the "SIMATIC S7" and "Unit" tabs are unit-specific and can thus be set separately for each channel unit of the channel.

These tabs are identical for all channel units of the S7 channel. Consequently, the dialog for the "MPI" channel unit is used in all examples.

Any changes that are made to the parameter values will only take effect after WinCC has been restarted.

Note

Copying the project: Settings

When copying the project to another PC, the settings on the "Unit" tab are retained.

The settings on the "SIMATIC S7" tab are not retained.

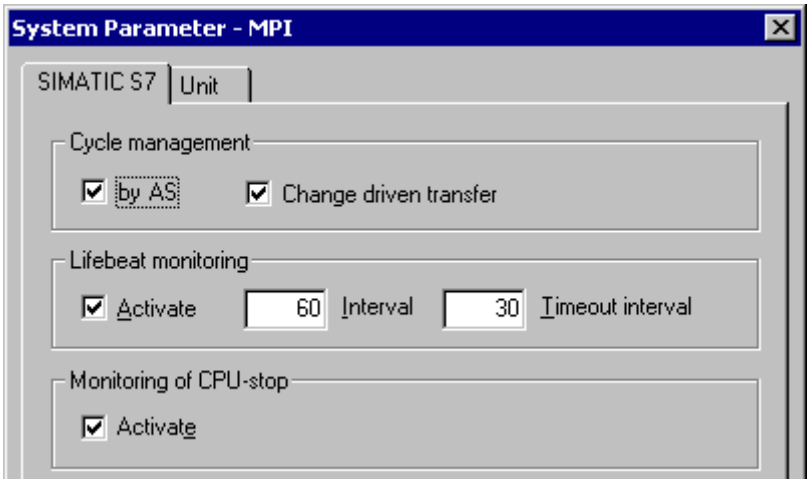
Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

Procedure

1. Select the "SIMATIC S7 Protocol Suite" channel in the Tag Management.
2. Open the "System Parameter" dialog using the shortcut menu of the desired channel unit.
3. Select the "SIMATIC S7" tab.
4. Activate the options in the "Cycle management" area if necessary:
   - "by AS":
     Cycle management when reading tags is switched on by the channel.
   - "Change driven transfer":
     The change-driven transfer is activated.

   If available, the cyclic services in the PLC will then be used here.
   You can find more information under "Cyclic read services in PLC (Page 273)".



5. To use the "Lifebeat monitoring" function, activate the option and enter the seconds values:
   - "Interval":
     Seconds value for the time interval for sending lifebeat messages.
   - "Timeout interval":
     Seconds value for monitoring the response to a lifebeat message.

   Note

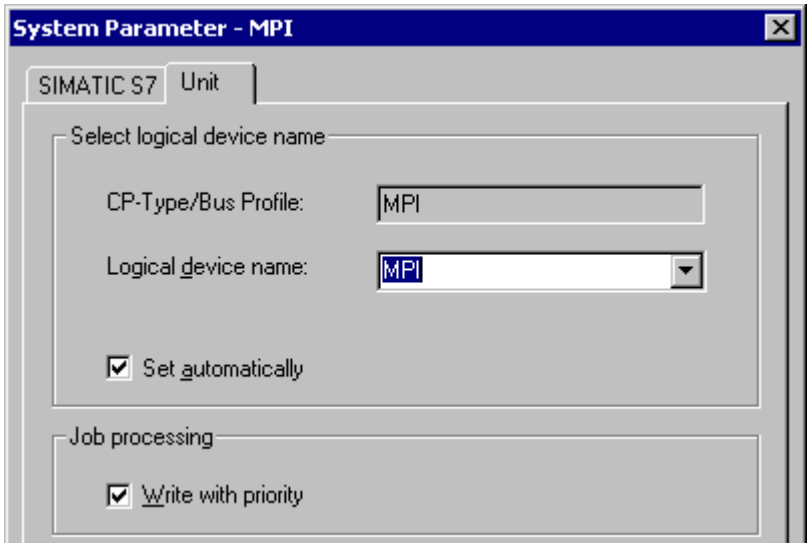
   Signoflife monitoring

   The lifebeat monitoring does not affect the monitoring mechanisms of other protocols and channels.

   Values for the timeout interval > 30 seconds are not useful, as TCP keep-alive monitoring terminates the connection after 30 seconds.

6. To display communication as faulty when the S7 CPU is stopped, activate the "CPU stop monitoring" option.

7. Select the "Unit" tab.
   A name, which will depend on the communications processor installed, will be displayed in the "Logical device name" field.
   Change this name only if you selected a different name when installing the communications processor being used.
   You can find more information under "How to Change the Logical Device Name (Page 276)".



8. If the device name is to be set automatically when Runtime is activated, activate the "Set automatically" option.
   The option is only active if only one communications processor is installed for this type of communication.
9. To process write jobs with a higher priority than read jobs, activate the "Write with priority" option.
10. Click "OK" to close the dialog.

See also

How to Change the Logical Device Name (Page 276)

Cyclic read services in PLC (Page 273)

13.4.8.4 How to Change the Logical Device Name

Introduction

Communication with the S7 takes place via logical device names. These names are assigned during the installation of the communications processor and are unit-specific.

Certain presets have now been established for the device names depending on the communications processor that has been installed. These are listed in the table "Default Device Names" below.

The tabs for all units of the S7 channel are identical and, for this reason, the dialog for the channel unit "MPI" is shown in the description.

Note

Device names must be identical

Logical device names must be exactly the same - to the letter - as in the device settings.

This being the case, the default logical device names for the "Industrial Ethernet" and "PROFIBUS" have, for example, a colon at the end of the name.

Any changes that are made to the parameter values will only take effect after WinCC has been restarted.

Default Device Names

Channel unit	Default Device Name
Industrial Ethernet	CP_H1_1:
Industrial Ethernet (II)	CP_H1_2:
MPI	MPI
Named connections	VM/
PROFIBUS	CP_L2_1:
PROFIBUS (II)	CP_L2_2:
Slot PLC	SLOT_PLC
Soft PLC	SOFT_PLC
TCP/IP	CP-TCPIP

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.
- A connection must be created to a channel unit e.g. "MPI".

Procedure

1. Select the desired channel unit in Tag Management.
2. Open the dialog window "System parameters" using the pop-up menu.
3. Select the Unit tab.
4. Specify a device name in the field "Logical device name".
   You can either select an entry from the selection list or enter a new name manually.
   The possible device names are determined by the "Configure PG/PC Interface" (Control Panel) tool. If this has not been installed, only the device name that is currently set will be displayed.
   If you specify a different logical device name, a message will be displayed.
   Add the entry only manually if you use a communications card on the target station which is not installed on the configuring station.
5. Close the dialog by clicking the "OK" button.

More information about the tool "Set PG/PC interface":

- SiePortal: PG/PC interface for the WinCC communication with the control system (ID 38717202) (https://support.industry.siemens.com/cs/ww/en/view/38717202)

See also

SiePortal: PG/PC interface for the WinCC communication with the control system (ID 38717202) (https://support.industry.siemens.com/cs/ww/en/view/38717202)

13.5 Special functions

13.5.1 Special functions of the "SIMATIC S7 Protocol Suite" Channel

Introduction

The SIMATIC S7 Protocol Suite contains some special functions; their functionality is described in this chapter.

See also

Software Redundancy (Page 319)

Raw data tags of the Channel "SIMATIC S7 Protocol Suite" (Page 310)

Data exchange with the S7 function block AR_SEND (Page 279)

13.5.2 Data exchange with the S7 function block AR_SEND

13.5.2.1 Data exchange with the S7 function block AR_SEND

Introduction

The S7 function block AR_SEND in the S7-400 AS is used to transfer process values to the process value archives.

Principle of Operation

To transfer PLC process values to a process value archive in WinCC, the S7-400 PLC has an integrated function component called SFB 37 "AR_SEND".

The basic function of AR_SEND component can supply data to archive tag. Data can be supplied to multiple tags if the AR_ID-Subnumber is used. If AR_SEND component is used, the process values are not sent individually to the archive; they are first collect in PLC and transferred as a package. This reduces the load on the used network.

In a PLC, you can use a CPI-dependent number of AR_SEND components (for e.g. CPU 416 max. 32 AR_SEND). A AR_ID can in turn be assigned to aach AR_SEND component. The sub-number is used to increase the amount of transferable process data because up to 4095 sub-numbers are possible for each AR_ID.
In reality, the number of archive tags per AR_SEND component is limited by the maximum length of the data area to be transferred. For more information about "The Structure and Parameters of Data Block Structures", please see the description of the "Number of Process Values" parameter.

AR_ID and AR_ID-Subnumber establish the assignment between data in the PLC and the archive tags and are defined using other parameters while configuring the data structure in

the databases in the PLC.
This assignment is configured in WinCC where as the other parameters are evaluated automatically.

SFB 37 "AR_SEND" must have first been configured in the PLC and the data block structure must have been done because configuration in WinCC is based on these values in the PLC. For information about configuring the AR_SEND function component can be found in the S7-400 PLC documentation.

Overview of AR_SEND Variants

Variants: AR_SEND for ...	Number of process-controlled archive tags for each AR_SEND	Intended use
an archive tag	a	For transferring process values for an archive tag where the process values can also be read in very small time intervals.
multiple archive tags	corresponds to the number AR_ID-Subnumbers	For transferring process values for multiple archive tags where the process values can also be read in very small time intervals.
multiple archive tags (optimized)	corresponds to the number AR_ID-Subnumbers	For cyclic data supply to maximum number of archive tags each one value each at one time

See also

AR_SEND variant for multiple archive tags (Page 292)

How to configure the AR _SEND variant for multiple archive tags (Page 308)

How to configure the AR _SEND variant for an archive tag (Page 306)

AR_SEND variant for multiple archive tags (optimized) (Page 304)

AR_SEND variant for an archive tag (Page 288)

13.5.2.2 Data Block - Structure and Parameters

Introduction

Before data is transferred with the "AR_SEND" function block, the data to be transferred is provided as one or more data blocks in the AS.

The structure of a data block depends on various parameters, for example on the AR_SEND variant used, the use of a time stamp or the data type of the process value.

The parameters used in data blocks are described below.

The individual parameter values are set in the data block in the AS and in the parameterization of the "AR_SEND" function block.

The parameterization is checked when the data block is evaluated in WinCC. If WinCC detects an error in the structure of the data block or if the archive tag configuration does not match

the received data, an entry with the following structure is recorded in the WinCC Diagnostics logbook:

- "Date, Time, 1003080 ,4 ,user name, computer name, NRMS7PMC, PdeReceive: Unknown parameter AR_SEND from connection connectionname ...+ more information for error description"

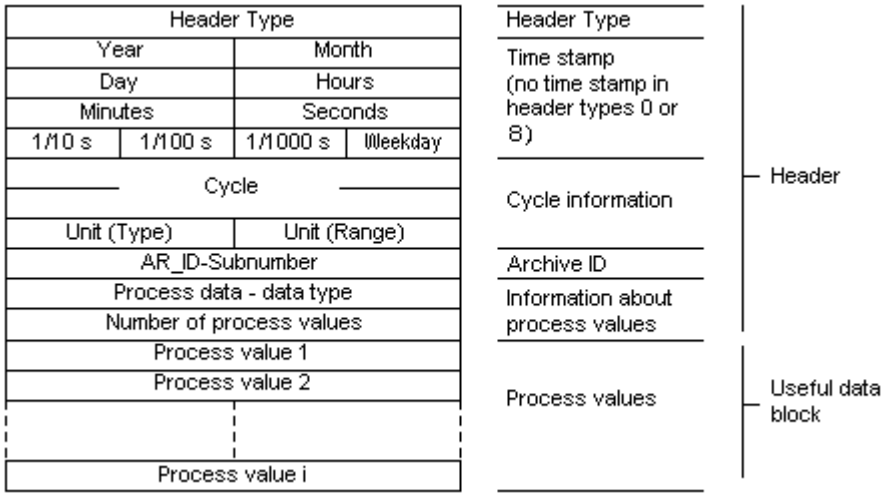
If the message system has been configured with the WinCC System Messages, this diagnostics entry will also trigger the OS process control message numbered 1003080. The text of the logbook entry is contained in this message's comment.

Structure of a data block

Each data block consists of a header and a user data area:

- The header contains information about the process values and their cycle, and possibly a time stamp.
- The user data area contains the actual process values.

One or more data blocks form the data area to be transferred.



Note

In the data blocks, each line represents two bytes.

Process values can be one or more bytes long, depending on their data type.

Further information can be found in the description of the "Number of Process Values" parameter.

Description of the parameters

Header type

The header type defines the type of information that is contained in the header.

Header type	Time stamp	AR_ID Subnumber
0	Header without time stamp	Header without AR_ID Subnumber
1	Header with time stamp	Header without AR_ID Subnumber
8	Header without time stamp	Header with AR_ID Subnumber
9	Header with time stamp	Header with AR_ID Subnumber

Note

In the case of the header types 0 and 8, the bytes for the time stamp is not included in the header.

Since these bytes are also not reserved in the data block, the header will be shortened accordingly by 8 bytes.

AR_ID Subnumber

Specifies the assignment between the AS user data and the WinCC archive tags and is configured in two places:

- In WinCC, when configuring the process controlled archive tags
- In the PLC, when setting up the user data area to be transferred

The subnumber is only relevant for the header types 8 or 9.

Valid values for the subnumber are in the range from 1 to 4095.

The parameter is output in WinCC as a hexadecimal value (1 - 0FFF).

Time stamp

The time stamp contains the date and the time in SIMATIC S7 BCD format.

The weekday entry is not evaluated by WinCC.

Note

Daylight saving time/standard time: Time setting of the automation system

The automation system S7 does not recognize the daylight saving time / standard time switchover.

The correction of the time stamp to daylight saving time or standard time is carried out in WinCC by the standardization DLL. The corrected time and a daylight savings time / standard time ID are then available in the WinCC applications. The corrected time and ID are then added to the archive e.g. in Tag Logging.

To ensure the correct time setting, select the same time setting in the AS as in WinCC.

In WinCC, you find the time setting of the PLC "Parameters > Time setting" in the "Properties - Project" area of the "Computer" editor.

Cycle

The cycle in which the process values are read.

This parameter is a factor for the units of time specified under Unit (Range).

Data length: Double word.

Example:

- "Cycle" = 10 ; "Unit(range)" = 4 means: Reading cycle for process values = 10 seconds

Unit (Type)

Specifies the type of time information and has an effect on the parameter "Number of Process Values".

No.	Meaning
1	The process values are read at equal intervals. Start time is specified in the time stamp of the header and is mandatory. The time interval between the process values is defined by the time units in the "Units (Range)" and the factor "Cycle".
2	Each process value has a time stamp. Any time stamp specified in the header is not evaluated. The format corresponds to the time stamp in the header with a length of 8 bytes.
3	Each process value has a relative time difference in units of time with a data length of 2 words. The absolute time is the sum of the time stamp in the header (= start time) and the relative time difference in the time unit set in "Unit(Range)". A time stamp entry in the header is mandatory.
4	Each process value contains the AR_ID-Subnumber. The time stamp given in the header applies to the process value. A time stamp entry in the header is mandatory.

Unit (range)

Specifies the units of time used for Unit (Type) = 1 or 3.

No.	Meaning
1	Reserved
2	Reserved
3	Milliseconds
4	Seconds
5	Minutes
6	Hours
7	Days

Data type of the process data

The process values are stored directly in the S7 format.

No.	S7 data type	WinCC data type
0	BYTE	BYTE
1	WORD	WORD

No.	S7 data type	WinCC data type
2	INT	SWORD
3	DWORD	DWORD
4	DINT	SDWORD
5	REAL	FLOAT

Number of process values

Depending on the entry in "Unit (Type)", the transferred data area can contain a specific number of process values.

The number is limited by the maximum length of the transferred data area - 16 Kbytes.

The resource limitations when using the S7 functions "AR_SEND" and "BSEND/BRCV" for communication with S7-400 are to be taken into account. This means that the maximum data volume that can be sent simultaneously using AR_SEND and/or BSEND/BRCV functions from AS to WinCC is limited to a maximum of 16 kByte.

Note

In the case of the AR_SEND variant "Multiple Archive Tags", the following limitation applies for this parameter:

The data blocks for the various archive tags must always begin on a word boundary.

Therefore, in the case of the combination "Data Type Process Value" = 0 (BYTE) and "Unit (Type)" = 1 (Process value with equally spaced time intervals) an even number of process values (=Bytes) must be entered for the parameter "Number of Process Values".

This restriction only applies for this AR_SEND variant and this combination of data type and "Unit (Type)".

Examples:

- 1x BSEND with a max. of 16 Kbytes
- or 1x AR_SEND with 8 Kbytes + 1x BSEND with 8 Kbytes
- or 1x AR_SEND with 10 Kbytes + 1x AR_SEND with 2 Kbytes + 1x BSEND with 4 Kbytes

Unit (Type)	Meaning for the number of process values
1	Process values read out at equal intervals: Therefore, 8000 process values of the data type WORD or INT or 4000 values of data type DWORD, DINT or REAL can be transferred.
2	Process values with time stamp: Each element of the user data area consists of a time stamp (8 bytes) and a value. Therefore, 1600 process values of the data type WORD or INT or 1333 values of data type DWORD, DINT or REAL can be transferred.

Unit (Type)	Meaning for the number of process values
3	Process values with time difference: Each element of the user data area consists of a time difference (4 bytes) and a value. Therefore, 2666 process values of the data type WORD or INT or 2000 values of data type DWORD, DINT or REAL can be transferred.
4	Process value contains AR_ID-Subnumber (AR-SEND with multiple tags - optimized) In Type 4, the process value consists of one word with the AR_ID-Subnumber (Value range: 1 - 0x0FFF) and one value. Thus, the user data area consists of an array of process values preceded by AR_ID Subnumbers. Therefore, 3992 process values of the data type WORD or INT or 2660 values of data type DWORD, DINT or REAL can be transferred.

Note

The AR_ID Subnumbers given in the data blocks must all be configured in WinCC. WinCC will stop interpreting the user data, if a subnumber that is not configured is found.

The data blocks for the various archive tags must always begin on a word boundary.

Therefore, with the data type BYTE and "Unit (Type)" = 1 (Process value with equally spaced time intervals), an even number of process values (=Bytes) must be entered for the parameter "Number of Process Values".

This restriction only applies for this AR_SEND variant and this combination of data type and "Unit (Type)".

See also

How to configure the AR _SEND variant for multiple archive tags (Page 308)

How to configure the AR _SEND variant for an archive tag (Page 306)

AR_SEND variant for multiple archive tags (optimized) (Page 304)

AR_SEND variant for multiple archive tags (Page 292)

AR_SEND variant for an archive tag (Page 288)

13.5.2.3 Overview of the properties of the AR_SEND variants

Introduction

On the basis of examples, the tables show the properties and possible parameter values for different AR_SEND variants.

The tables do not display all of the possible combinations.

The columns "Header Type" to "Process Value Data Type" are presented in the order that they appear in the header.

Note

The values for AR_ID and AR_ID Subnumber are set together with those of the other parameters while configuring the function block "AR_SEND" and the data structure in the data block in the AS.

Variants for an Archive Tag

Example / Property	E.g.-No.	Header type	Date / Time (Time-stampt in header)	Cycle factor	Unit (Type)	Units (range)	AR_ID-Subno.	Data type of process val.	max. number of proc.val.	Process value structure in the e.g.
Each process value (byte) with its own time stamp	1	0	does not exist	0	2	0	0	0 1; 2 3; 4; 5	3200 1600 1333	8 byte time stamp + 1 Byte process value
Process value with equally spaced time stamp	2	1	Relevant	>=1	1	3 to 7	0	0 1; 2 3; 4; 5	16000 8000 4000	1 word process value
Each process value (word) with its own time stamp	3	1	not relevant	0	2	0	0	0 1; 2 3; 4; 5	3200 1600 1333	8 byte time stamp + 1 word process value
Each process value with time difference	4	1	Relevant	>=1	3	3 to 7	0	0 1; 2 3; 4; 5	5332 2666 2000	8 byte time stamp + 1 Byte process value

Variants for Multiple Archive Tags

Example / Property	E.g.-No.	Header type	Date / Time (Time-stampt in header)	Cycle factor	Unit (Type)	Units (range)	AR_ID-Subno.	Data type of process val.	max. number of proc.val.	Process Value structure in the e.g.
Each process value (byte) with its own time stamp	5	8	does not exist	0	2	0	1 to 4095	0 1; 2 3; 4; 5	3200 1600 1333	8 byte time stamp + 1 Byte process value
Process value with equally spaced time stamp	6	9	Relevant	>=1	1	3 to 7	1 to 4095	0 1; 2 3; 4; 5	16000 8000 4000	1 word process value
Each process value (word) with its own time stamp	7	9	not relevant	0	2	0	1 to 4095	0 1; 2 3; 4; 5	3200 1600 1333	8 byte time stamp + 1 word process value
Each process value with time difference	8	9	Relevant	0	3	3 to 7	1 to 4095	0 1; 2 3; 4; 5	5332 2666 2000	8 byte time stamp + 1 Byte process value

Variants for Multiple Archive Tags - Optimized

Example / Property	E.g.-No.	Header type	Date / Time (Time-stampt in header)	Cycle factor	Unit (Type)	Units (range)	AR_ID-Subno.	Data type of process val.	max. number of proc.val.	Process value structure in the e.g.
Each process value with an AR_ID Sub-number	9	1	Relevant	0	4	0	0	1; 2 3; 4; 5	3992 2660	1 Word Sub-number + 1 Word process value

See also

AR_SEND variant for multiple archive tags (optimized) (Page 304)

AR_SEND variant for multiple archive tags (Page 292)

AR_SEND variant for an archive tag (Page 288)

13.5.2.4 AR_SEND variant for an archive tag

Introduction

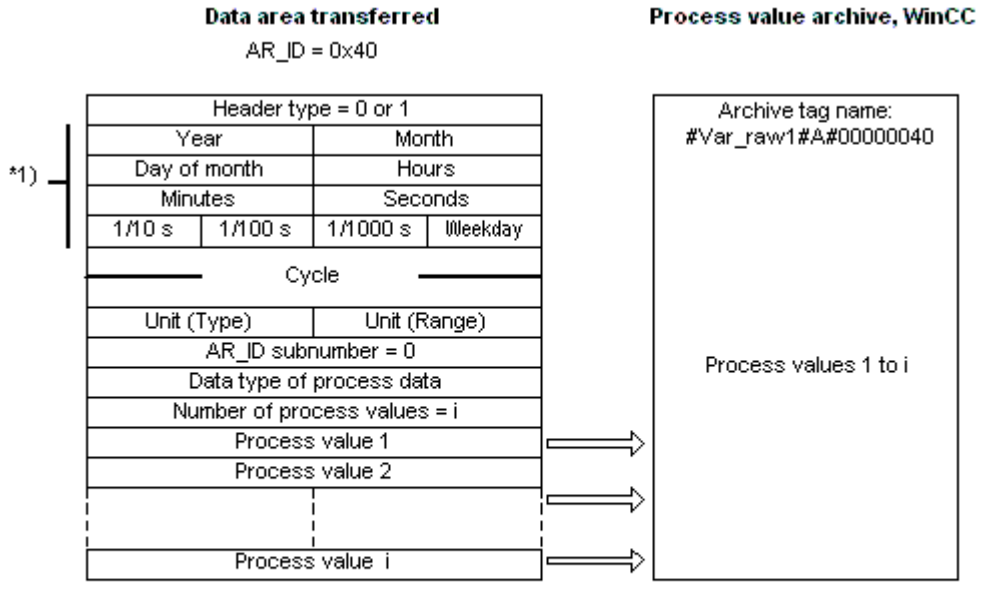
This variant can be used to supply an archive tag with process values. It can also be used with older versions of WinCC (prior to V5.0).

The properties of this variant:

- The Header Type must be 0 or 1, i.e. without AR_ID Subnumber and with/without time stamp.
- The AR_ID Subnumber in the header will not be evaluated.
- In WinCC, the archive tag name does not contain an AR_ID Subnumber, since only the process values for an archive tag will be transferred.

An example of the data area's structure

The data area to be transferred consists of one data block.



*1) = time stamp omitted with header type 0

See also

Example 4 for data block structure: An archive tag; each process value with relative time stamp (time difference) (Page 291)

Example 3 for data block structure: An archive tag; each process value has its own time stamp (Page 290)

Example 2 for data block structure: One archive tag; equally spaced time stamp (Page 290)

Example 1 for data block structure: An archive tag; each process value has a time stamp (Page 289)

Overview of the properties of the AR_SEND variants (Page 285)

13.5.2.5 Example 1 for data block structure: An archive tag; each process value has a time stamp

Introduction

In this example, the process values are transferred for one archive tag only. There is no time stamp in the header and the corresponding number of bytes are also not reserved. Hence, each process value (1byte) is preceded by a time stamp (8byte).

Data type of the process values is BYTE.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 0			
2.0	Cycle =0			
4.0				
6.0	Unit (Type) = 2		Unit (Range) = 0	
8.0	AR_ID-Subnumber = 0			
10.0	Process data - data type = 0			
12.0	Number of process values = 3 (max. 3200)			
14.0	Year=2001		Month=10	
16.0	Day=05		Hours=13	
18.0	Minutes=40		Seconds=00	
20.0	1/10 s	1/100 s	1/1000 s	Weekday
22.0	Process value 1		-	
24.0	Year=2001		Month=10	
26.0	Day=05		Hours=14	
28.0	Minutes=40		Seconds=00	
30.0	1/10 s	1/100 s	1/1000 s	Weekday
32.0	Process value 2		-	
34.0	Year=2001		Month=10	
36.0	Day=05		Hours=15	
38.0	Minutes=40		Seconds=00	
40.0	1/10 s	1/100 s	1/1000 s	Weekday
42.0	Process value 3		-	

See also

Data Block - Structure and Parameters (Page 280)

13.5.2.6 Example 2 for data block structure: One archive tag; equally spaced time stamp

Introduction

In this example, the process values are transferred for one archive tag.

The equally spaced time stamp of one second is formed using the parameters "Cycle" = 1 and "Unit (Range)" = 4 (= seconds).

Data type of the process values is WORD.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 1			
2.0	Year=2001		Month=10	
4.0	Day=05		Hours=13	
6.0	Minutes=40		Seconds=00	
8.0	1/10 s	1/100 s	1/1000 s	Weekday
10.0	Cycle =1			
12.0				
14.0	Unit (Type) = 1		Unit (Range) = 4	
16.0	AR_ID-Subnumber = 0			
18.0	Process data - data type = 1			
20.0	Number of process values = 8 (max. 8000)			
22.0	Process value 1			
24.0	Process value 2			
26.0	Process value 3			
28.0	Process value 4			
30.0	Process value 5			
32.0	Process value 6			
34.0	Process value 7			
36.0	Process value 8			

See also

Data Block - Structure and Parameters (Page 280)

13.5.2.7 Example 3 for data block structure: An archive tag; each process value has its own time stamp

Introduction

In this example, the process values are transferred for one archive tag only. The time stamp in the header is not important. Hence, each process value (1Word) is preceded by a time stamp (8byte).

Data type of the process values is SWORD.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 1			
2.0	Year=0		Month=0	
4.0	Day=0		Hours=0	
6.0	Minutes=0		Seconds=0	
8.0	1/10 s	1/100 s	1/1000 s	Weekday
10.0	Cycle =0			
12.0				
14.0	Unit (Type) = 2		Unit (Range) = 0	
16.0	AR_ID-Subnumber = 0			
18.0	Process data - data type = 2			
20.0	Number of process values = 2 (max. 1600)			
22.0	Year=2001		Month=10	
24.0	Day=05		Hours=13	
26.0	Minutes=40		Seconds=00	
28.0	1/10 s	1/100 s	1/1000 s	Weekday
30.0	Process value 1			
32.0	Year=2001		Month=10	
34.0	Day=05		Hours=14	
36.0	Minutes=40		Seconds=00	
38.0	1/10 s	1/100 s	1/1000 s	Weekday
40.0	Process value 2			

See also

Data Block - Structure and Parameters (Page 280)

13.5.2.8 Example 4 for data block structure: An archive tag; each process value with relative time stamp (time difference)

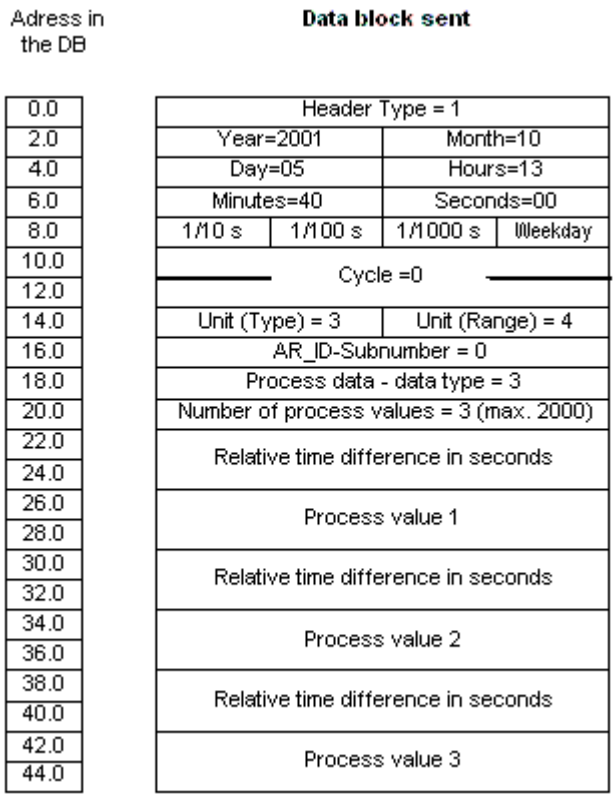
Introduction

In this example, the process values are transferred for one archive tag with time stamp.

The parameter "Unit(Type)" = 3 gives each process value a time difference (4Byte) for the time stamp in the header. The unit of time difference is set by the parameter "Unit(range)" = 4 in seconds.

Data type of the process values is DWORD.

Data block structure in the data component



Adress in the DB	Data block sent			
0.0	Header Type = 1			
2.0	Year=2001		Month=10	
4.0	Day=05		Hours=13	
6.0	Minutes=40		Seconds=00	
8.0	1/10 s	1/100 s	1/1000 s	Weekday
10.0	Cycle =0			
12.0				
14.0	Unit (Type) = 3		Unit (Range) = 4	
16.0	AR_ID-Subnumber = 0			
18.0	Process data - data type = 3			
20.0	Number of process values = 3 (max. 2000)			
22.0	Relative time difference in seconds			
24.0				
26.0	Process value 1			
28.0				
30.0	Relative time difference in seconds			
32.0				
34.0	Process value 2			
36.0				
38.0	Relative time difference in seconds			
40.0				
42.0	Process value 3			
44.0				

See also

Data Block - Structure and Parameters (Page 280)

13.5.2.9 AR_SEND variant for multiple archive tags

Introduction

With this variant, you can supply process values to one or more archive tags. For each archive tag, an AR_ID Subnumber will be assigned and a data block will be created in the data area to be transferred.

"x" process values can be transferred for each AR_ID Subnumber. For more information about "The Structure and Parameters of Data Block Structures", please see the description of the "Number of Process Values" parameter.

The time stamp for the value of an archive tag is taken or derived from the data area to be transferred in accordance with the given "Unit (Type)" and "Unit (Range)". It is then sent on to the WinCC process value archive.

The properties of this variant:

- The Header Type must be 8 or 9 (with/without time stamp and with AR_ID Subnumber).
- For every AR_ID Subnumber, a data block must be created in the data area to be transferred.
- The AR_ID Subnumber in each data block must be greater than zero.
- In WinCC, the archive tag name has an AR_ID Subnumber.

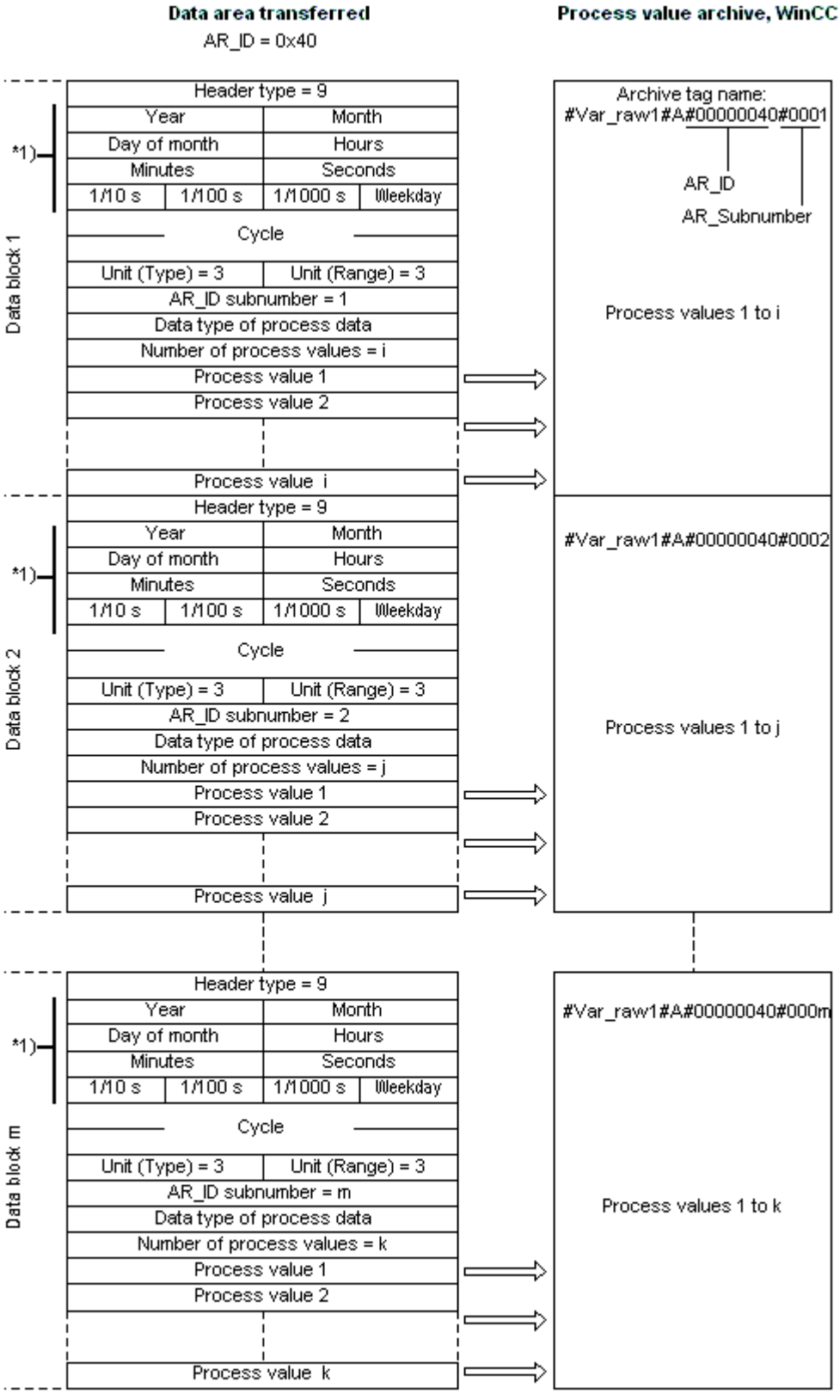
Note

The AR_ID Subnumbers given in the data blocks must all be configured in WinCC. WinCC will stop interpreting the user data, if a not-configured subnumber is found.

The data blocks for the various archive tags must always begin on a word boundary. Therefore, in the case of the combination "Data Type Process Value" = 0 (BYTE) and "Unit (Type)" = 1 (Process values with equally spaced time intervals) an even number of process values (=Bytes) must be entered for the "Number of Process Values" parameter. This restriction only applies for this AR_SEND variant and this combination of data type and "Unit (Type)".

An example of the data area's structure

The data area to be transferred consists of one or more data blocks corresponding to the number of archive tags to be supplied.



*1) = omitted with header type 0 or 8

See also

Example 8 for data block structure: Multiple archive tags; process values with relative time stamp (time difference) (Page 302)

Example 7 for data block structure: Multiple archive tags; each process value has its own time stamp (Page 300)

Example 6 for data block structure: Multiple archive tags; equally spaced time stamp (Page 298)

Example 5 for data block structure: Multiple archive tags; each process value has its own time stamp (Page 296)

Overview of the properties of the AR_SEND variants (Page 285)

13.5.2.10 Example 5 for data block structure: Multiple archive tags; each process value has its own time stamp

Introduction

In this example, process values are transferred for multiple archive tags.

The data blocks for the different archive tags are located one after the other in the data component. A different AR_ID-Subnumber is entered in each data block.

There is no time stamp in the header and the corresponding number of bytes for it are also not reserved. Hence, each process value (1byte) is preceded by a time stamp (8byte).

Data type of the process values is BYTE.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 8			
2.0	Cycle =0			
4.0				
6.0	Unit (Type) = 2		Unit (Range) = 0	
8.0	AR_ID-Subnumber = 1			
10.0	Process data - data type = 0			
12.0	Number of process values = 3			
14.0	Year=2001		Month=10	
16.0	Day=05		Hours=13	
18.0	Minutes=40		Seconds=00	
20.0	1/10 s	1/100 s	1/1000 s	Weekday
22.0	Process value 1		-	
24.0	Year=2001		Monat=10	
26.0	Day=05		Stunden=14	
28.0	Minutes=40		Sekunden=00	
30.0	1/10 s	1/100 s	1/1000 s	Weekday
32.0	Process value 2		-	
34.0	Year=2001		Month=10	
36.0	Day=05		Hours=15	
38.0	Minutes=40		Seconds=00	
40.0	1/10 s	1/100 s	1/1000 s	Weekday
42.0	Process value 3		-	
44.0	Header Type = 8			
46.0	Cycle =0			
48.0				
50.0	Unit (Type) = 2		Unit (Range) = 0	
52.0	AR_ID-Subnumber = 2			
54.0	Process data - data type = 0			
56.0	Number of process values = 2			
58.0	Year=2001		Month=10	
60.0	Day=05		Hours=12	
62.0	Minutes=40		Seconds=00	
64.0	1/10 s	1/100 s	1/1000 s	Weekday
66.0	Process value 1		-	
68.0	Year=2001		Month=10	
70.0	Day=05		Hours=13	
72.0	Minutes=40		Seconds=00	
74.0	1/10 s	1/100 s	1/1000 s	Weekday
76.0	Process value 2		-	

See also

Data Block - Structure and Parameters (Page 280)

13.5.2.11 Example 6 for data block structure: Multiple archive tags; equally spaced time stamp

Introduction

In this example, process values are transferred for multiple archive tags. The data blocks for the different archive tags are located one after the other in the data component. A different AR_ID-Subnumber is entered in the header in each data block.

The equally spaced time stamp of one second is formed using the parameters "Cycle" = 1 and "Unit (Range)" = 4 (= seconds).

Data type of the process values is WORD.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 9			
2.0	Year=2001		Month=10	
4.0	Day=05		Hours=13	
6.0	Minutes=40		Seconds=00	
8.0	1/10 s	1/100 s	1/1000 s	Weekday
10.0	Cycle =1			
12.0				
14.0	Unit (Type) = 1		Unit (Range) = 4	
16.0	AR_ID-Subnumber = 1			
18.0	Process data - data type = 1			
20.0	Number of process values = 8			
22.0	Process value 1			
24.0	Process value 2			
26.0	Process value 3			
28.0	Process value 4			
30.0	Process value 5			
32.0	Process value 6			
34.0	Process value 7			
36.0	Process value 8			
38.0	Header Type = 9			
40.0	Year=2001		Month=10	
42.0	Day=05		Hours=12	
44.0	Minutes=40		Seconds=00	
46.0	1/10 s	1/100 s	1/1000 s	Weekday
48.0	Cycle =1			
50.0				
52.0	Unit (Type) = 1		Unit (Range) = 4	
54.0	AR_ID-Subnumber = 2			
56.0	Process data - data type = 1			
58.0	Number of process values = 5			
60.0	Process value 1			
62.0	Process value 2			
64.0	Process value 3			
66.0	Process value 4			
68.0	Process value 5			

See also

Data Block - Structure and Parameters (Page 280)

13.5.2.12 Example 7 for data block structure: Multiple archive tags; each process value has its own time stamp

Introduction

In this example, process values are transferred for multiple archive tags. The data blocks for the different archive tags are located one after the other in the data component. A different AR_ID-Subnumber is entered in each data block.

The time stamp in the header is not important. Hence, each process value (1Word) is preceded by a time stamp (8byte).

Data type of the process values is SWORD.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 9			
2.0	Year=0		Month=0	
4.0	Day=0		Hours=0	
6.0	Minutes=0		Seconds=0	
8.0	1/10 s	1/100 s	1/1000 s	Weekday
10.0	Cycle =0			
12.0				
14.0	Unit (Type) = 2		Unit (Range) = 0	
16.0	AR_ID-Subnumber = 1			
18.0	Process data - data type = 2			
20.0	Number of process values = 3			
22.0	Year=2001		Month=10	
24.0	Day=05		Hours=13	
26.0	Minutes=40		Seconds=00	
28.0	1/10 s	1/100 s	1/1000 s	Weekday
30.0	Process value 1			
32.0	Year=2001		Month=10	
34.0	Day=05		Hours=14	
36.0	Minutes=40		Seconds=00	
38.0	1/10 s	1/100 s	1/1000 s	Weekday
40.0	Process value 2			
42.0	Year=2001		Month=10	
44.0	Day=05		Hours=15	
46.0	Minutes=40		Seconds=00	
48.0	1/10 s	1/100 s	1/1000 s	Weekday
50.0	Process value 3			
52.0	Header Type = 9			
54.0	Year=0		Month=0	
56.0	Day=0		Hours=0	
58.0	Minutes=0		Seconds=0	
60.0	1/10 s	1/100 s	1/1000 s	Weekday
62.0	Cycle =0			
64.0				
66.0	Unit (Type) = 2		Unit (Range) = 0	
68.0	AR_ID-Subnumber = 2			
70.0	Process data - data type = 2			
72.0	Number of process values = 2			
74.0	Year=2001		Month=10	
76.0	Day=05		Hours=12	
78.0	Minutes=40		Seconds=00	
80.0	1/10 s	1/100 s	1/1000 s	Weekday
82.0	Process value 1			
84.0	Year=2001		Month=10	
86.0	Day=05		Hours=13	
88.0	Minutes=40		Seconds=00	
90.0	1/10 s	1/100 s	1/1000 s	Weekday
92.0	Process value 2			

See also

Data Block - Structure and Parameters (Page 280)

13.5.2.13 Example 8 for data block structure: Multiple archive tags; process values with relative time stamp (time difference)

Introduction

In this example, the process values are transferred for one archive tag with time stamp.

The data blocks for the different archive tags are located one after the other in the data component. A different AR_ID-Subnumber is entered in each data block.

The parameter "Unit(Type)" = 3 gives each process value a time difference (4Byte) for the time stamp in the header. The unit of time difference is individually defined by the "Unit(Range)" parameter for each archive tag and hence for each data block.

Data type of the process values is DWORD.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 9			
2.0	Year=2001		Month=10	
4.0	Day=05		Hours=13	
6.0	Minutes=40		Seconds=00	
8.0	1/10 s	1/100 s	1/1000 s	Weekday
10.0	Cycle =0			
12.0				
14.0	Unit (Type) = 3		Unit (Range) = 5	
16.0	AR_ID-Subnumber = 1			
18.0	Process data - data type = 3			
20.0	Number of process values = 3			
22.0	Relative time difference in minutes			
24.0				
26.0	Process value 1			
28.0				
30.0	Relative time difference in minutes			
32.0				
34.0	Process value 2			
36.0				
38.0	Relative time difference in minutes			
40.0				
42.0	Process value 3			
44.0				
46.0	Header Type = 9			
48.0	Year=2001		Month=10	
50.0	Day=05		Hours=12	
52.0	Minutes=40		Seconds=00	
54.0	1/10 s	1/100 s	1/1000 s	Weekday
56.0	Cycle =0			
58.0				
60.0	Unit (Type) = 3		Unit (Range) = 6	
62.0	AR_ID-Subnumber = 2			
64.0	Process data - data type = 3			
66.0	Number of process values = 2			
68.0	Relative time difference in hours			
70.0				
72.0	Process value 1			
74.0				
76.0	Relative time difference in hours			
78.0				
80.0	Process value 2			
82.0				

See also

Data Block - Structure and Parameters (Page 280)

13.5.2.14 AR_SEND variant for multiple archive tags (optimized)

Introduction

This variant is to be used when the maximum number of archive tags should each be supplied a process value at one time. In this case, the data area to be transferred consists of just one data block and each process value has just its AR_ID Subnumber and its associated value.

The data type is the same for the process values of all of the archive tags in this data block.

The properties of this variant:

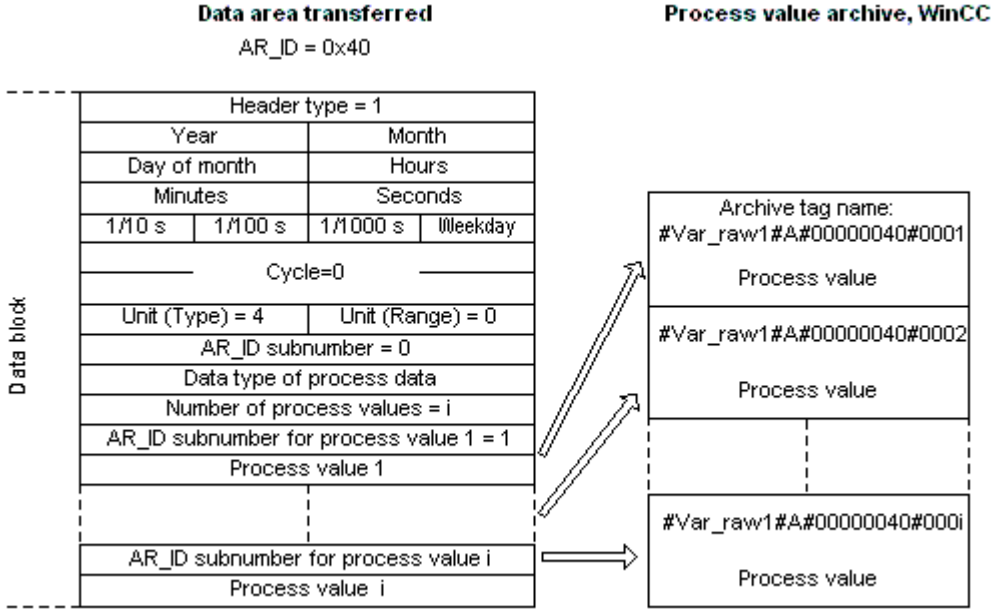
- The Header Type must be 1 (with time stamp and without AR_ID Subnumber).
- The AR_ID Subnumbers for the associated process values in the data block must be greater than zero. The AR_ID Subnumber in the header will not be evaluated.
- The "Unit (Type)" parameter must be 4, i.e. the process value has an AR_ID Subnumber.
- The "Units (Range)" parameter must be 0, i.e. the time stamp in the Header is valid for all process values and there are no relative times.
- In WinCC, the archive tag name has an AR_ID Subnumber.

Note

If a process value has an AR_ID Subnumber for which no WinCC archive tag is found, this will result in an entry in the WinCC Diagnosis Log. The remaining process values will then continue to be processed.

An example of the data area's structure

The data area to be transferred consists of just one data block.



See also

Overview of the properties of the AR_SEND variants (Page 285)

Example 9 for data block structure: multiple archive tags;optimized (Page 305)

13.5.2.15 Example 9 for data block structure: multiple archive tags;optimized

Introduction

In this example, the process values are transferred for one archive tag with time stamp. The time stamp is applicable to all archive tags.

The corresponding AR_ID-Subnumber is placed before each process value.

Data type of the process values is WORD.

Data block structure in the data component

Adress in the DB	Data block sent			
0.0	Header Type = 1			
2.0	Year=2001		Month=10	
4.0	Day=05		Hours=13	
6.0	Minutes=40		Seconds=00	
8.0	1/10 s	1/100 s	1/1000 s	Weekday
10.0	Cycle =0			
12.0				
14.0	Unit (Type) = 4		Unit (Range) = 0	
16.0	AR_ID-Subnumber = 0			
18.0	Process data - data type = 1			
20.0	Number of process values = 5 (max. 3992)			
22.0	AR_ID-Subnumber			
24.0	Process value 1			
26.0	AR_ID-Subnumber			
28.0	Process value 2			
30.0	AR_ID-Subnumber			
32.0	Process value 3			
34.0	AR_ID-Subnumber			
36.0	Process value 4			
38.0	AR_ID-Subnumber			
40.0	Process value 5			

See also

Data Block - Structure and Parameters (Page 280)

13.5.2.16 How to configure the AR _SEND variant for an archive tag

Introduction

There are a number of variants for using the AR_SEND function to exchange data.

Only the AR_ID is used for the "One archive tag" variant. The AR_ID sub-number is not used.

AR_ID

The AR_ID establishes the assignment between the data in the AS and the archive tags.

The AR_ID is configured together with other parameters when the data blocks and the SFB 37 "AR_SEND" function block are configured in the AS.

In WinCC, this assignment is performed in the properties for the process-controlled tag. This assignment is the only configuration that is necessary in WinCC and is described in this section.

The other parameters in WinCC do not need to be configured, since they are evaluated automatically.

Note

AR_ID sub-number: Header type

To use this variant with just the AR_ID, the header type must be configured as 0 or 1. The AR_ID sub-number must be set to zero.

Migrated projects: WinCC prior to V5.0

Since they do not have an AR_ID sub-number, all archive tags configured before WinCC Version V5.0 can be used with this variant.

As of WinCC Version V5.1 Hotfix 4, it has been possible to either specify an alias for the archive tag name with process-controlled tags or to use the internal name generated by the system.

As of Version V5.1 Hotfix 4, the name generated by the system contains the name of the assigned raw data tag instead of the raw data ID from

In projects migrated to a version from V5.1 Hotfix 4, the archive tag names can be retained for use in their original format or can be converted. Conversion is performed by opening the properties dialog of the process-controlled archive tags and closing it after saving. An alias does not have to be assigned.

If all external tags in a project are remapped using the "AS-OS transfer" function, the archive tag name must be converted once to the new structure.

The new structure is then maintained.

Requirements

- The "AR_SEND" function block and the data block structure are configured in the AS. The structure is known.
- The "SIMATIC S7 Protocol Suite" channel is integrated into the project.
- A connection has been created in a channel unit, e.g. "Test_Ind_Eth" under "Industrial Ethernet".
- A process value archive is configured in the "Tag Logging" editor.

Procedure

1. In the "SIMATIC S7 Protocol Suite" channel, select the connection to be used for the data transfer.
2. In the "Tags" tab, click in the topmost free cell in the "Name" column.
   Enter the name "Var_raw_arsend" for the tag.
   The name can be no longer than 24 characters.
   The raw data tag name becomes part of the archive tag name during the configuration of the process-controlled tag and is therefore limited to this length.
3. Select "Raw data type" in the "Data type" field.
4. To open the "Address properties" dialog, click the button in the "Address" field.

5. Select the "Raw data" option.
   In the "Raw data type" area, select the "Archive data link" type, and close the dialog with "OK".
6. Open the "Tag Logging" editor.
   Select the archive in the navigation area under "Process value archives" and the "Process-controlled tags" tab in the table area.
7. To add the "Var_raw_arsend" raw data tag, click the ... button in the "Raw data tag" field.
8. You can edit the properties in the "Properties" area:
   - Select the entry "nrms7pmc.nll" in the "Format DLL" field.
   - Enter the "AR_ID" as a hexadecimal value in the "Block ID" field.
     The value is specified by the AS configuration.
   - The "Sub-number" field remains empty as no sub-number is used in this AR_SEND variant.

   The internal archive tag name generated by the system is displayed in the "Tag name" field. The name contains the name of the assigned raw data tag and the AR_ID.
9. If necessary, add an alias for this archive tag in the "Archive tag name" field.
   If you do not enter an alias, the internal archive tag name is used for management in the process value archive and addressing of the archive tag in WinCC.
10. Close Tag Logging.

13.5.2.17 How to configure the AR _SEND variant for multiple archive tags

Introduction

There are different variants for the data exchange of multiple archive tags via the AR_SEND function:

"Multiple archive tags"	Value assignment of multiple archive tags with multiple values at different times
"Multiple archive tags – optimized"	Value assignment of as many archive tags as possible, each with a value at a specific point in time

AR_IDs and AR_ID sub-numbers

AR_IDs and AR_ID sub-numbers are used in the variants described.

AR_IDs and AR_ID sub-numbers establish the assignment between the data in the AS and the archive tags. They are defined in the AS with other parameters when configuring the data blocks and the SFB 37 "AR_SEND" function block.

In WinCC, this assignment is performed in the properties for the process-controlled tag. This assignment is the only configuration that is necessary in WinCC and is described in this section.

The other parameters in WinCC do not need to be configured, since they are evaluated automatically.

Note

AR_ID sub-number: Header type

For the AR_ID sub-number, header type 8 or 9 must be configured.

Migrated projects: WinCC prior to V5.0

Archive tags configured before WinCC Version V5.0 have no AR_ID sub-number and can therefore only be used in the "One tag" variant.

As of WinCC Version V5.1 Hotfix 4, it has been possible to either specify an alias for the archive tag name with process-controlled tags or to use the internal name generated by the system.

As of version V5.1 Hotfix 4, the name generated by the system contains the name of the assigned raw data tag instead of the raw data ID.

In projects migrated to a version from V5.1 Hotfix 4, the archive tag names can be retained for use in their original format or can be converted. Conversion is performed by opening the properties dialog of the process-controlled archive tags and closing it after saving. An alias does not have to be assigned.

If all external tags in a project are remapped using the "AS-OS transfer" function, the archive tag name must be converted once to the new structure.

The new structure is then maintained.

Requirements

- The "AR_SEND" function block and the data block structure are configured in the AS. The structure is known.
- The "SIMATIC S7 Protocol Suite" channel is integrated into the project.
- A connection has been created in a channel unit, e.g. "Test_Ind_Eth" under "Industrial Ethernet".
- A process value archive is configured in the "Tag Logging" editor.

Procedure

1. In the "SIMATIC S7 Protocol Suite" channel, select the connection to be used for the data transfer.
2. In the "Tags" tab, click in the topmost free cell in the "Name" column.
   Enter the name "Var_raw_arsend" for the tag.
   The name can be no longer than 24 characters.
   The raw data tag name becomes part of the archive tag name during the configuration of the process-controlled tag and is therefore limited to this length.
3. Select "Raw data type" in the "Data type" field.
4. To open the "Address properties" dialog, click the ... button in the "Address" field.

5. Select the "Raw data" option.
   In the "Raw data type" area, select the "Archive data link" type, and close the dialog with "OK".
6. Open the "Tag Logging" editor.
   Select the archive in the navigation area under "Process value archives" and the "Process-controlled tags" tab in the table area.
7. To add the "Var_raw_arsend" raw data tag, click the ... button in the "Raw data tag" field.
8. You can edit the properties in the "Properties" area:
   - Select the entry "nrms7pmc.nll" in the "Format DLL" field.
   - Enter the "AR_ID" as a hexadecimal value in the "Block ID" field.
     The value is specified by the AS configuration.
   - Enter the AR_ID sub-number as a hexadecimal value under "Sub-number".
     The value is also specified by the configuration in AS.

   The internal archive tag name generated by the system is displayed in the "Tag name" field. The name contains the name of the assigned raw data tag and the AR_ID.
9. If necessary, add an alias for this archive tag in the "Archive tag name" field.
   If you do not enter an alias, the internal archive tag name is used for management in the process value archive and addressing of the archive tag in WinCC.
10. Close Tag Logging.

13.5.3 Raw data tags of the Channel "SIMATIC S7 Protocol Suite"

13.5.3.1 Raw data tags of the Channel "SIMATIC S7 Protocol Suite"

Introduction

- A tag of the type raw data type is a data telegram on a transport level. The contents of the raw data tag are not fixed and therefore only the sender and the receiver can interpret the transmitted data. There are no format changes in WinCC for this data type. Maximum length is 65535 Byte.
- WinCC distinguishes between two types of raw data tags: Raw data tag for free application use and raw data tag for handling S7 functions.

Raw data tag for free application use

Raw data tags for free application use are used for transferring user data blocks between WinCC and PLC and handle only user data. It distinguishes between:

Raw data tag as byte array

Raw data tag for BSEND/BRCV functions

Raw data tag for handling S7 functions

These raw data tags do not have any channel-specific header and are normally used by the message system and for process data entry in WinCC.

No further description is needed here as these are tags and functions internal to the channel.

See also

Raw data tag for BSEND/BRCV functions of S7 communication (Page 314)

Raw data tag as byte array (Page 311)

13.5.3.2 Raw data tag as byte array

Introduction

Raw data tags as byte array are used for transferring user data blocks between WinCC and PLC and handle only user data.

A raw data tag as byte array is handled in the channel like a normal process tag that is addressed via the address and length of the data area (for e.g. DB 100, DW 20, length 40 Byte).

Raw data length

The raw data length is limited to one transferable data block and must be fully transferable using a PDU (Protocol Data Unit).

The maximum length of the data blocks that can be transferred using the communication driver depends on the PDU length negotiated while establishing the connection minus the header and additional information.

The PDU lengths normally used in SIMATIC S7 thus result in the following maximum lengths:

- S7-300: PDU length 240 bytes, max. data block length 208 bytes
- S7-400: PDU length 480 Byte, max. data block length 448 Byte

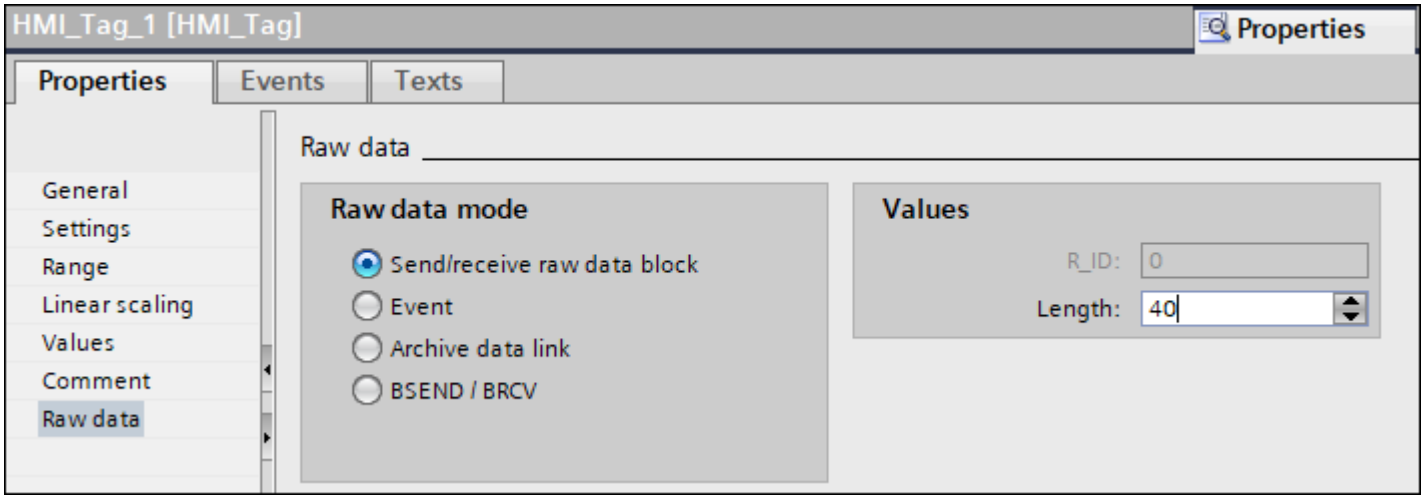
Data must be blocked if larger data blocks are to be transferred.

In PLC, the S7 software forms the blocks; in WinCC through scripts.

How to Configure a Raw Data Tag as Byte Array

The raw data tags for transferring data blocks are configured as raw data of "Send/receive block" type with one address and one length detail.

The following illustration shows a configuration example for a data area with length of 40 bytes in the data component 100 from data word 20:



Read a Raw data tag as Byte Array

Raw data tag is read in the same way as a "normal" process tag. The corresponding data block is requested in AS and transferred to the user when the data is received.

Data is transferred always at the initiative of WinCC. Sporadic or event-controlled data reception at the initiative of AS cannot be done using this raw data tag.

Write a Raw data tag as Byte Array

Raw data tag is written in the same way as a "normal" process tag.

After sending the data block and receiving a positive acknowledgment from AS, the data block is transferred to the image of the Data Manager.

See also

How to Configure a Raw Data Tag as Byte Array (Page 312)

13.5.3.3 How to Configure a Raw Data Tag as Byte Array

Introduction

This section will show you how to configure as byte array a raw data tag of the "SIMATIC S7 Protocol Suite" channel.

The configuration is identical for all channel units of the channel.

The "MPI" channel unit and its connection is used in the example.

Raw data length

The raw data length is limited to one transferable data block and must be fully transferable using a PDU (Protocol Data Unit).

The maximum length of the data block that can be transferred using the communication driver depends on the PDU length negotiated while establishing the connection, minus the header and additional information.

The PDU lengths normally used in SIMATIC S7 thus result in the following maximum lengths:

- S7-300: PDU length 240 bytes, max. data block length 208 bytes
- S7-400: PDU length 480 Byte, max. data block length 448 Byte

Wrong length will cause the read/write job to be rejected with a display.

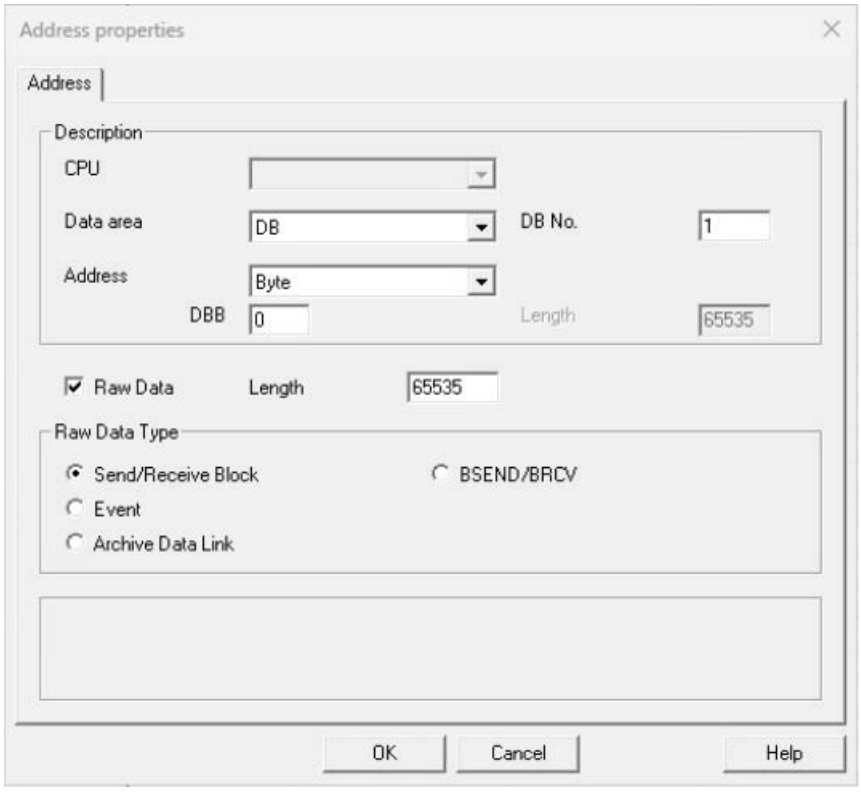
Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.
- A connection must be created to a channel unit e.g. "MPI".

Procedure

1. In the channel "SIMATIC S7 Protocol Suite", select the connection which should be used for the data transfer.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name "Var1_raw_byte" for the tag.
4. Select "Raw data type" in the "Data type" field.
5. Open the "Address properties" dialog.
   To do this, click in the "Address" field and then on the ... button.
6. Select the "Raw data" option.
   In the "Raw data type" area select the type "Send/receive block".
   The display of the fields in the "Address description" area and the field next to the check box "Raw data" depend on this setting.
7. Enter the length of the raw data block (in bytes) in the Length field.

8. In the "Data area" set the data area of the PCL where the data is located.
   If you select "DB" as data area, enter the number of the data block in the enabled "DB No." field as well.
   The "CPU" field is not enabled when the "MPI" channel unit is connected.



9. In the "Addressing" field, set the addressing type and below it the start address, e.g.:

Type of addressing	Start address
Byte	DBB
Word	DBW
Double word	DD

10. Click "OK" to close the dialog.

13.5.3.4 Raw data tag for BSEND/BRCV functions of S7 communication

Introduction

Raw data tags for "BSEND/BRCV" functions are used for transferring user data blocks between WinCC and AS and handle only user data.

This raw data type can be used to access the "BSEND/BRCV" functions of S7 communication.

The "BSEND/BRCV" raw data communication via Named Connections is supported for the following automation systems:

- S7-400
- S7-300
  - CPU319-3 PN/DP V2.5 or higher
  - CPU317-2 PN/DP V2.6 or higher
  - CPU315-2 PN/DP V3.1 or higher

  For S7-300 controllers, a firmware version V3.x or higher is recommended.
  The raw data communication cannot be established via a communications processor.
- WinAC RTX 2010

The initiative of data transfer always lies with the sending partner; hence "BSEND/BRCV" functions can also be used to implement event-controlled or sporadic data block transfers.

For resource reasons, it is advisable to keep the number of BSEND/BRCV raw data tags low.

Resource limitation for the use of S7 functions "AR_SEND" and "BSEND/BRCV"

The maximum data volume that can be sent simultaneously using AR_SEND and/or BSEND/BRCV functions from AS to WinCC is limited:

- To 16 KB for the S7-400
- To 8 KB for the S7-300

Examples with the S7-400:

- 1x BSEND with a max. of 16 Kbytes
- 1x AR_SEND with 8 kBytes + 1x BSEND with 8 kBytes
- 1x AR_SEND with 10 Kbytes + 1x AR_SEND with 2 Kbytes + 1x BSEND with 4 Kbytes

Note

Coordination of the write jobs

If the data block of a write job is transferred to AS and has not yet been deleted or fully deleted from the receiving buffer, then the next write job will be rejected with an error message.

During such an error display, write jobs with R_ID > 0x8000 0000 are written to a connection-specific queue and the system tries to repeat the write job for 6 seconds.

The responsibility for time co-ordination for transfer rests with the user and needs to be noted as shorter time intervals for write jobs.

Configuring a PBK Connection for Using "BSEND/BRCV" functions

"BSEND/BRCV" functions can only be used via a "hard-configured connection", a so-called PBK connection (programmed component communication).

To configure a hard-configured connection, you must specify a connection resource (hex: 10 ... DF) in the connection parameters.

This connection resource will be assigned by STEP 7 when the connection is configured within the PLC.

The connection must be configured as passive connection end-point in the automation system.

Read/write jobs

A hard-configured connection can also be used to handle "normal" read and write jobs.

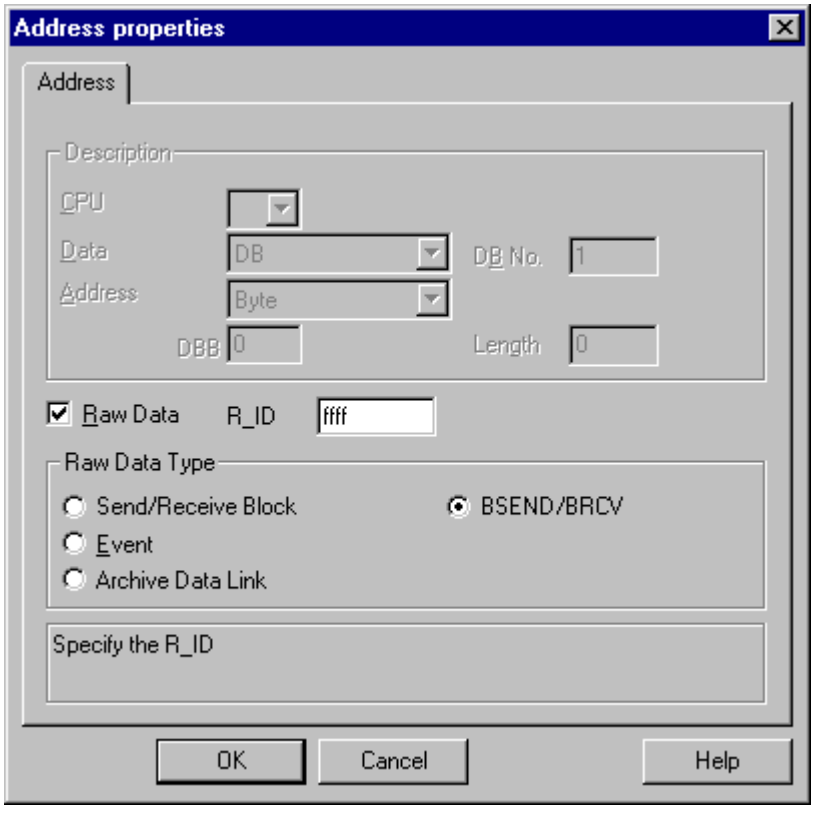
If very large data areas are to be transferred via the connection, then the data blocks are transferred in multiple PDUs.

For performance reasons, it would therefore be better to create a separate connection for "BSEND/BRCV" functions.

Configuring Raw data tag for BSEND/BRCV functions

Raw data tags for transferring "BSEND/BRCV" data blocks are configured as raw data of type "BSEND/BRCV" with a "R_ID".

The data length is derived implicitly from the sent or received data volume.



"R_ID" Parameter

For the "BSEND/BRCV" functionality, you must specify a 32-bit long R_ID as hexadecimal number.

The R_ID is assigned at the time of configuration in AS and is used for distinguishing multiple data block transfers over one connection.

The send and receive calls are always notified with reference to this R_ID in the underlying communication sub-system (SIMATIC Device Drivers).

A raw data tag is thus assigned to one unique R_ID.

Sending a "BSEND/BRCV" raw data tag

Sending a "BSEND/BRCV" raw data tag takes place in the same way as writing a "normal" process tag.

After sending the data block and receiving a positive acknowledgment from AS, the data block is transferred to the image of the Data Manager.

Receiving a "BSEND/BRCV" raw data tag

"BSEND/BRCV" raw data is sporadically sent to the channel on the initiative of the AS.

Hence it is not possible to explicitly read S7 raw data tags.

Synchronization

The BSEND/BRCV mechanisms do not include any synchronization functions.

If no user has logged in to receive the data during the start-up phase, the data blocks sent by AS will bounce on the receiver side.

Hence, the user has to take care of the synchronization and, for e.g. release the sending direction on the AS by setting a flag with a data word.

See also

How to Configure a Raw Data Tag for ""BSEND/BRCV" functions (Page 317)

13.5.3.5 How to Configure a Raw Data Tag for ""BSEND/BRCV" functions

Introduction

This section will show you how to configure a raw data tag of the "SIMATIC S7 Protocol Suite" channel for "BSEND/BRCV" functions.

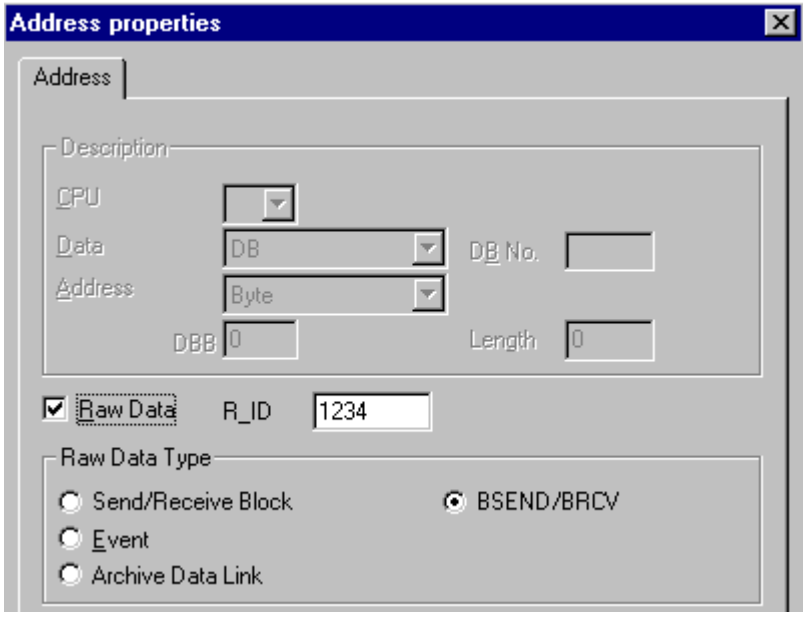
The configuration is identical for all channel units of the channel. The "MPI channel unit and its connection is used in the example.

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.
- A connection must be created to a channel unit e.g. "MPI".

Procedure

1. In the channel "SIMATIC S7 Protocol Suite", select the connection which should be used for the data transfer.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name "Var2_raw_bsend" for the tag.
4. Select "Raw data type" in the "Data Type" field.
5. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.
6. Mark the "Raw Data" check box. In the "Raw Data Type" area select the type "BSEND/BRCV".
   The display of the fields of the "Address description" area is now deactivated.
7. Enter the hexadecimal value of the ID in the "R_ID" field. The R_ID is assigned in AS at the time of configuration



8. Close both of the dialogs by clicking the "OK" buttons.

13.5.4 Software Redundancy

13.5.4.1 Software Redundancy

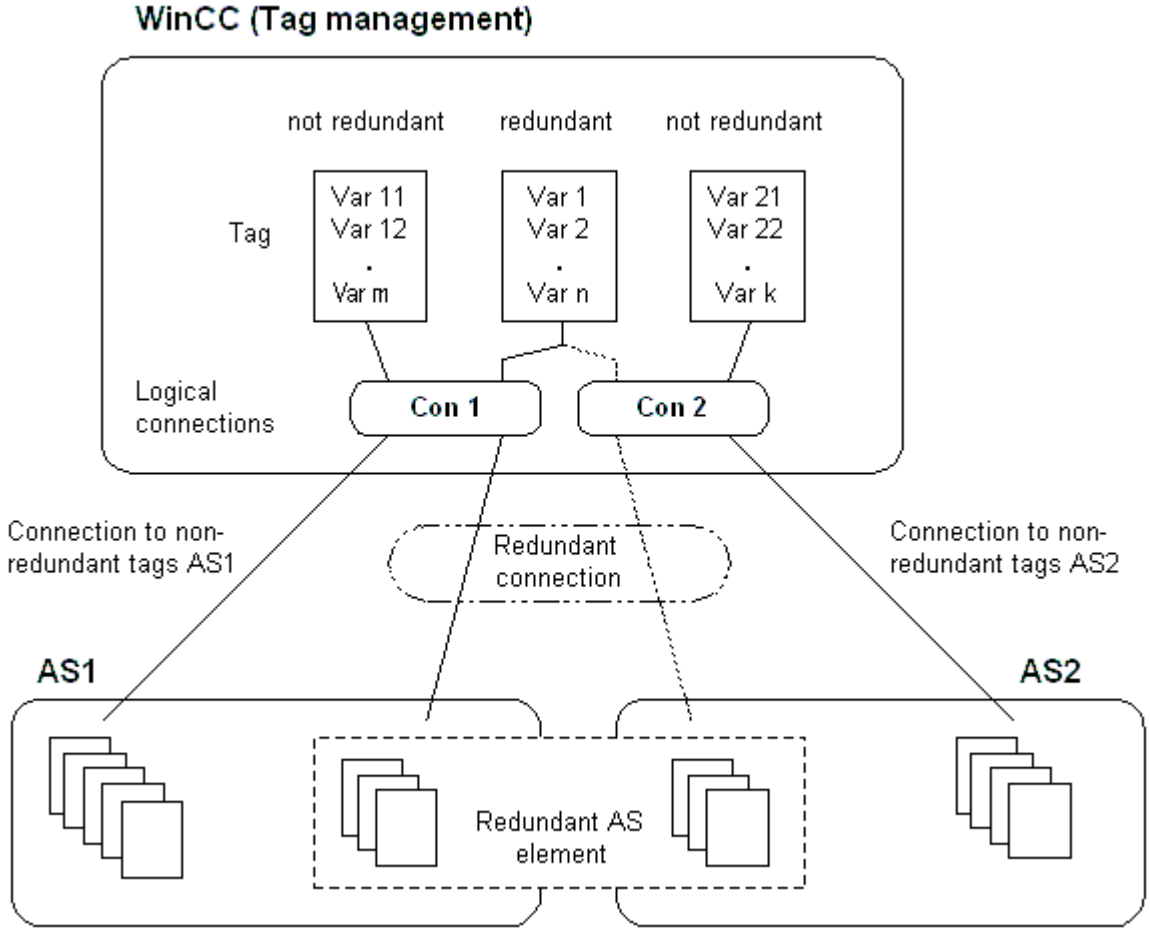
Introduction

The software redundancy offers a cost-effective option for monitoring the safety-related parts of the system that are not time-critical through a redundant connection of two automation systems S7-300 or S7-400.

Note

The software redundancy of this channel does not have the same functions as the H Layer Redundancy of SIMATIC S7-400 H.

Configurations in AS and in WinCC are required for functioning.



AS

If a redundant connection exists between two automation systems, then in the event of failure of one AS the other one can take over the monitoring. Monitoring can cover the entire process or just parts thereof.

In addition to the user program, the software redundancy program package is installed on both automation systems. This software package is not included in the scope of WinCC delivery. For data synchronization, you need a redundant link via MPI, PROFIBUS-DP, or Industrial Ethernet between the two automation systems, whereby the existing communication connections can be used.

WinCC

Redundancy coupling can only be configured between connections of the same channel unit. At the time of configuration, only one connection, the so-called main connection is configured. The reserve connection is created via the "Set up redundant connection" Dynamic Wizard.

The Wizard also inserts the connection-specific internal tags and a script. This controls the switching between connections during runtime and also triggers the corresponding messages.

During runtime, it is possible to use the script to automatically switch between the connections when there is an error. However, it is possible to also switch manually without the script by describing the connection-specific internal tags "@<Connection name>@ForceConnectionAddress".

For AS2, you need a second fixed configured connection if data is to be used from the non-redundant part of AS2 in WinCC.

The use of software redundancy does not mean that the hard configured connections can only be used for the redundancy. Each single connection can also be used without redundancy.

See also

How to Clear a Software Redundancy in WinCC (Page 327)

How To Configure a Software Redundancy (Page 324)

Software Redundancy - Connection-specific internal tags (Page 320)

13.5.4.2 Software Redundancy - Connection-specific internal tags

Introduction

Connection-specific internal tags are used to control the redundant connections. These tags are setup using the Dynamic Wizard "Redundant Connection Configuration" and are gathered in a Tag Group called "@<connectionname>" in the associated connection.

These tags can be used to determine the connection status and to control the establishment of a connection. Thus, they can also be used to implement other applications, e.g. the addressing of multiple PLCs via a single connection. However, if these tags are to be used without the Software Redundancy package, they must be created manually.

Name Format

The name of a connection-specific internal tag is composed of the name of the associated connection and an identifier.

"@<connectionname>@<identifier>"

The connection name is prefixed by a "@" to identify it as a system tag. The identifier is placed as a separator before the connection name.

Example: "@CPU_3@ConnectionState"

<connectionname> = CPU_3

<identifier> = ConnectionState

Note

Connection-specific internal tags are counted as external tags (eight external tags per connection).

WinCC's data manager only permits access to external tags when the associated connection is ready. The connection-specific internal tags can, however, be written and read regardless of the connection status.

In Runtime, the current values of some of the connection-specific internal tags may be called using the "WinCC Channel Diagnosis" tool. When the main connection is selected, the tags will be displayed in the "Counters" column. In addition, in WinCC Explorers' Tag Management, it is possible to display the tag's current value as a tooltip.

An Overview of the Tags

The following identifiers are available for the internal tags of Software Redundancy:

ConnectionState

Meaning	Connection Status This tag can be used to find out the current connection status.
Type	DWORD
Access	Read
Default value	0 = "faulty"
Values	0 = Connection faulty 1 = Connection ready-to-use 2 = Connection redundant (only in case of redundancy in H systems)

ConnectionError

Meaning	Cause of fault The tag contains a description of the cause for the fault. Default = 0, i.e. connection not yet established or without error. When establishing a connection, the tag is loaded with 0 (no error) again. The error code is interpreted in a channel-specific manner. The S7 channel passes the SIMATIC Device Driver's error code in this tag.
Type	DWORD
Access	Read
Default value	0 = "No error"
Values	0 = No error <> 0 = S7 Error codes

ConnectionErrorString

Meaning	Cause of fault as string The tag contains the reason for the connection error as string. The string is put out in the language currently selected. Default = ", i.e. connection not yet established or without error. In S7 channel, the following text is output in "English" regardless of the selected language.
Type	TEXT8 [128]
Access	Read
Default value	"" = "No error"
Values	"No Error" "Error hhhh" = Error hhhh has occurred (whereby hhhh = S7 error code hexadezimal)

ConnectionErrorCount

Meaning	Counter for connection error The value of this tag is incremented by 1 every time there is a connection error. When there is an overflow, the counting starts back at 0.
Type	DWORD
Access	Read
Default value	0

ConnectionEstablishMode

Meaning	Connection Establish Mode This tag can be used to set the automatic mode to establish a connection. If enabled, the S7 channel attempts to reestablish a failed connection at intervals of approx. 4 seconds. If the value in this tag = 0, there will be no attempt after every 4 seconds to reestablish the connection after a fault; rather it will remain disconnected.
Type	DWORD
Access	Write
Default value	1
Values	A Write to the tag brings about the following actions: 0 = Manual connection establish mode Action: Deactivate automatic connection establishment <>0 = Automatic connection establishment mode Action: Activate automatic connection establishment mode

ForceConnectionState

Meaning	Preferred connection status This tag can be used to inform the channel about the preferred connection status. Usually this tag has the value 1, i.e. the channel attempts to establish the connection (at regular intervals of approx. 4 seconds, if applicable). If the value 0 is written to this tag, the channel interrupts the connection.
Type	DWORD
Access	Write
Default value	1
Values	Any write to the tag has the following effects: 0 = preferred connection status: Connection broken Action: if connection established, initiate disconnection 1 = Preferred Connection Status: Connection broken Action: if connection disconnected, initiate establishment of connection

ForceConnectionAddress

Meaning	Select the connection address This tag defines which of the connection addresses is to be used to establish the connection.
Type	DWORD
Access	Write

Default value	0
Values	If ConnectionEstablishMode is set to "Automatic", the connection will be setup automatically to the corresponding address. A Write to the tag brings about the following actions: 0 = Connection via configured connection parameters Action: If @ForceConnectionAddress has been earlier set to 1, then initiate disconnection. 1 = Connection via alternative connection parameters Action: If @ForceConnectionAddress previously on 0, then initiate disconnect.

AlternateConnectionAddress

Meaning	Alternative Connection Address In this tag, you can enter the alternative connection address string. The string is the same as the one that will be displayed as the connection parameter in WinCC Explorer. The string is channel-specific. Upon system startup (Runtime), the configured address is entered here as default for the S7 channel. If an address has not been configured yet, the text "Illegal Address" is entered for the S7 channel. Example of address detail for a S7-AS with station address 3 via MPI: "MPI,3 0,,0,0,02"
Type	TEXT8 [255]
Access	Write
Default value	"..." = "configured address
Values	Writing to this tag gives rise to the following actions: - If the address changes due to the write process, then the connection is disconnected fro the setting "Connection via alternative connection parameter". - If connection mode is set to "automatic", then the connection is automatically established with the address that has just been written.

13.5.4.3 How To Configure a Software Redundancy

Introduction

This section describes how to configure the software redundancy for connections of the Channel "SIMATIC S7 Protocol Suite" in WinCC. The PLC must also be configured to use this function, but this will not be described in this document.

Requirements

1. The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.
2. A connection must be created in one of this channel's channel-units to which a redundant connection should be configured.

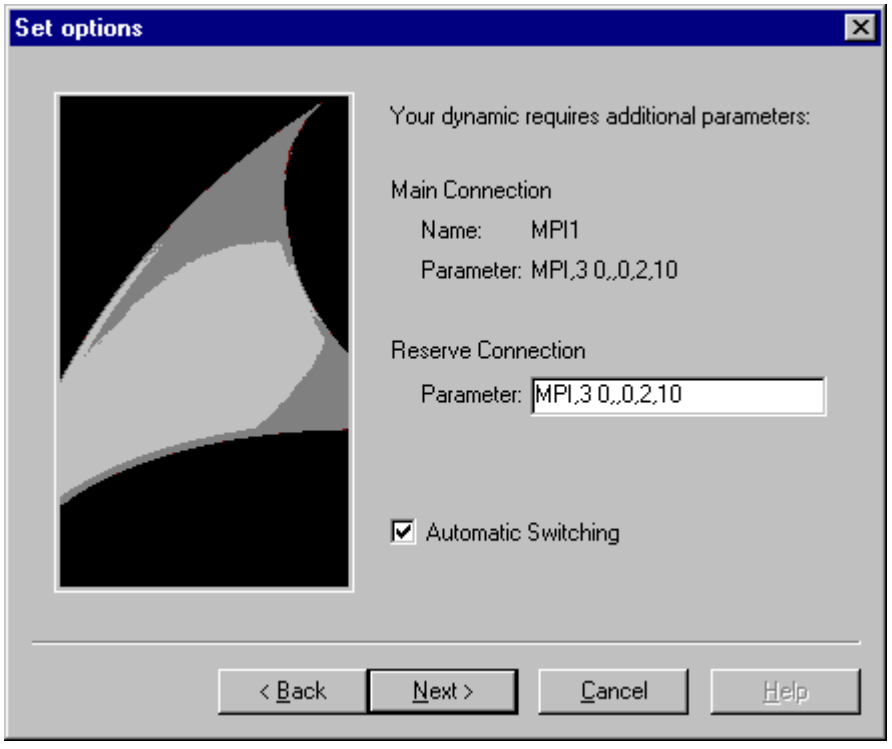
Procedure

1. In the computer's startup parameters, activate the "Global Script Runtime", "Alarm Logging Runtime" and "Graphics Runtime" modules.
   For further information, please see "Checking startup parameters".
2. Load WinCC's system messages into Alarm Logging. These system messages include messages about software redundancy.
   You can find additional information about the topic under "Read WinCC system messages in Alarm Logging".
3. Open a picture in Graphics Designer. In the "Dynamic Wizard" window select the "System Functions" tab. Double-click to start the Dynamic Wizard "Setup Redundant Connection".



4. The procedure for using the wizard is described briefly in the "Welcome". Click "Next" to open the "Set Options" dialog.
5. Select the connection that should be used as the main connection and then click on "Next". The wizard will now create the connection-specific internal tags and will save them in a tag group "@" under the main connection.

6. Enter the address of the PLC to which the reserve connection should be established in the "Parameter" field.
   Mark the "Automatic Switching" check box to have the wizard generate a script for automatically switching connections.
   Click "Next"
   A graphic of the redundancy - showing a MPI connection to two PLCs - will appear in the picture:



7. All of the settings made will be displayed once more in the "Finished!" dialog. If you want to make any corrections, simply click on "Back". Click "Finish".
   The Wizard will now generate a script and save it under "@<connectionname>.pas" in the directory "C-Editor \ Actions \ Actions : <computername> of the Global Script Editor.

Note

In the following procedure, the "Setup redundant connection" wizard will be used. This wizard generates - when Step 6 is completed - the connection-specific internal variables. If the wizard is canceled at this point or the procedure is not completed by clicking on "Finish", these tags will remain unchanged.

See also

How To Load WinCC's system messages into Alarm Logging (Page 328)

How to Check the WinCC Startup Parameters (Page 327)

13.5.4.4 How to Clear a Software Redundancy in WinCC

Introduction

This section describes how to delete the software redundancy for connections of the Channel "SIMATIC S7 Protocol Suite" in WinCC. The PLC must also be configured to use this function to return to non-redundant connections, but this is not described in this document.

Requirements

- The WinCC project must be deactivated.

Procedure

A software redundancy is deleted in two steps:

- Delete the tag group "@<computername" including its tags in "Tag Management".
- Delete the script "@<connectionname>.pas" in "Global Script".

Procedure

1. In the Tag Management, select the connection that should be configured as the main connection for the software redundancy. It contains a tag group "@<connectionname" with the software redundancy's connection-specific internal tags. Delete this tag group.
2. Delete the script for the Action "@<connectionname>.pas". To do this, open the C-Editor in "Global Script". Several subdirectories will be displayed.
3. Select the "Actions \ Action : " directory. <computername>". In the data window, delete the script "@<connectionsname>.pas" for the "Actions" type.
4. Close the "Global Script" editor.

13.5.4.5 How to Check the WinCC Startup Parameters

Procedure

1. Open the "Computer" editor in the WinCC Configuration Studio.
2. Select the computer name in the navigation area.
   The "Processes when starting WinCC Runtime" and "Additional applications" tabs are displayed in the data area.
3. Activate the required runtime applications on the "Processes when starting WinCC Runtime" tab.
4. To add more applications to the startup list, switch to the "Additional applications" tab
5. In the "Application" column, click the "..." button in the first empty box and select the desired application.

13.5.4.6 How To Load WinCC's system messages into Alarm Logging

Introduction

In this section you will see how you can load the WinCC system messages into the project.

Procedure

1. Open Alarm Logging.
2. Select the "System Messages" node in the navigation area.
3. You can activate the "Used" option for system messages that you use either in the table area or in the Properties area.
4. Select the command "Update used" from the shortcut menu of the "System Messages" node.

13.5.4.7 Error codes during connection disturbances

The "S7CHNdeu.chm" file contains a list of error codes.

You can find this file in the installation path under \SIEMENS\WinCC\bin.

Documentation of Error Codes

SIMATIC S7-1200, S7-1500 Channel 14

14.1 "SIMATIC S7-1200, S7-1500 Channel" channel

The "SIMATIC S7-1200, S7-1500 Channel" is used for communication between a WinCC station and the automation systems S7-1200 and S7-1500.

The TCP/IP protocol is used for the communication.

The channel also supports redundant S7-1500R/H systems. Additional information: "Configuring the channel > Redundant System S7-1500R/H (Page 360)"

Channel unit

The "SIMATIC S7-1200, S7-1500 Channel" comes with the "OMS+" channel unit.

Diagnostics of channels

To display faults and errors in the controllers in Runtime, use the WinCC SysDiagControl.

You can find more information under "Communication Diagnostics > Diagnostic channel SIMATIC S7-1200/S7-1500".

Recommended communications processors

The following communications processors are recommended for the communication of a WinCC station with the automation systems S7-1200 or S7-1500:

- CP 1612 A2
- CP 1613 A2
- CP 1623
- CP 1628

For the communication connection, it is recommended to use the "Secure Communication" of STEP 7 in the TIA Portal.

See also

Redundant System S7-1500R/H (Page 360)

"Secure Communication" (Page 336)

14.2 Overview of the supported data types

Introduction

The data type and the format adaptation to the data format in the automation system (AS) are specified when the tag is configured.

The table shows the data types supported by the channel and the use of type conversions.

Supported data types / format adaptations

Data Types	Type conversion
Binary tag	No
Signed 8-bit value	Yes
Unsigned 8-bit value	Yes
Signed 16-bit value	Yes
Unsigned 16-bit value	Yes
Signed 32-bit value	Yes
Unsigned 32-bit value	Yes
Floating-point number 32-bit IEEE 754	Yes
Floating-point number 64-bit IEEE 754	Yes
Text tag, 8-bit font	No
Text tag, 16-bit character set	No
Raw data tag	No
Date/time	Yes

14.3 Configuring the channel

14.3.1 Configuration of the "SIMATIC S7-1200, S7-1500 Channel" channel

Introduction

WinCC needs a logical connection for communication of WinCC with the automation system.

This section describes how to configure the "SIMATIC S7-1200, S7-1500 Channel".

To set up a communication channel, select "Add new driver > SIMATIC S7-1200, S7-1500 Channel" in the shortcut menu of the Tag Management.

Connection parameters

S7 network address

The address depends on the selected product family:

- For the S7-1200 or S7-1500 product family, enter the IP address of the TCP/IP connection.
- For the product family WinAC S7-1500, enter the station address for the S7-1507S.

Note that the access point must reference an interface that is suitable for the selected product family.

TCP/IP connection

When using the TCP/IP protocol, you must define the IP address of the automation system for the logical connection.

The IP address consists of four numerical values, separated by dots. The numerical values must be within the range "0-255".

Note

Timeout Behavior

Interrupted connections are not detected immediately when using the TCP/IP protocol.

The check-back message can take up to a minute.

Station address

The station address lies within the number range of a PROFIBUS address.

You can find the station address in the "Properties" dialog under "Index" during the configuration of the S7-1507S.

Choose the interface "PC Internal (local)" as the access point.

Protect connection access with password

For connections using channel "SIMATIC S7-1200, S7-1500 Channel", you can protect access to the automation system with a password.

The levels 1, 2 and 3 are defined on the automation system for this access protection.

Apply the configured password for the required level during the configuration in WinCC.

The level configured at the AS is used automatically if no password is set.

Specifying and determining the connection status

You can create the following system tags in the internal tag group "ConnectionStates" for each connection:

- Establishing / terminating a connection:
  @<Connectionname>@ForceConnectionStateEx
- Querying the connection status:
  @<Connectionname>@ConnectionStateEx

You can find additional information on the connection status in the WinCC Information System, under "WinCC process communication > AUTOHOTSPOT".

Software redundancy

If you work with the software redundancy of the S7-1500R/H, create the following tags:

- Establish / terminate redundant connection:
  @<Connectionname>@ForceConnectionState
- Querying redundant connection status:
  @<Connectionname>@ConnectionState

The system tags for the software redundancy are described under "Software redundancy for S7-1500R/H (Page 365)".

Configuring the tags

For a connection between WinCC and the automation system using channel "SIMATIC S7-1200, S7-1500 Channel", tags of various data types are created in WinCC. You configure process tags for the respective connection or load the AS symbols of the automation system into the WinCC Tag Management.

The configuration of the tags differs by the addressing of the data area in the automation system.

Note

AS configuration only in the TIA Portal

You can only change the configuration of the automation system in the TIA Portal.

Overwriting data of the HMI system or the Web server

During operation, data of the HMI system or the web server can be overwritten in the S7-1500. In operation, the two processes (PLC system and HMI system) run in parallel independently of each other.

If both systems attempt write access to the same tag, the data might be overwritten on the system side.

Additional information is provided on the Internet:

- FAQ with entry ID 109478253: "Why is data of the HMI system or the web server sometimes overwritten in the S7-1500? (https://support.industry.siemens.com/cs/ww/en/view/109478253)"

S7 connection: Software Controllers as of V30.0

You can create connections to the Software Controller as of version V30.0.

- You export the configuration and the certificate from TIA Portal.
- The IP address is configured in the connection.
- The data of Software Controller V30.0 is available in Runtime.

See also

Configuring raw data tags (Page 333)

Software redundancy for S7-1500R/H (Page 365)

"Secure Communication" (Page 336)

https://support.industry.siemens.com/cs/ww/en/view/109478253 (https://support.industry.siemens.com/cs/ww/en/view/109478253)

14.3.2 Configuring raw data tags

Introduction

The "SIMATIC S7-1200, S7-1500 Channel" channel supports the "Raw data tag" data type.

Raw data tags in the channel "SIMATIC S7-1200, S7-1500 Channel"

Raw data tags as byte arrays are used for transferring user data blocks between WinCC and PLC and handle only user data.

Only the acyclic read service of the controller is supported for raw data tags, e.g. the tag request via C scripts.

The "SIMATIC S7-1200, S7-1500 Channel" channel does not support cyclic read services for raw data tags.

Note

No raw data communication with S7-1200/S7-1500 controllers

The "SIMATIC S7-1200, S7-1500 Channel" channel does not support any raw data communication via the S7 communication "BSEND/BRCV" function.

Addressing the raw data tag

A raw data tag as byte array is handled in the channel like a normal process tag that is addressed via address and length of the data area, for example DB 1, DBB10, length 100 bytes.

Only "Byte" is possible for the "Raw data tag" data type of the WinCC tag. Except for the length of the raw data range, the parameters are preset and cannot be changed.

Length of the data blocks

Maximum length of data blocks that can be sent by the communication driver:

- S7-1200 / S7-1500: Data block length max. 8192 bytes (8 KB)

Exchanging large data volumes

Application example 37873547 describes how you use raw data to transfer large amounts of data from the controller to WinCC:

- SiePortal: Exchanging large amounts of data between S7-300/400/1500 and WinCC (Entry37873547) (https://support.industry.siemens.com/cs/ww/en/view/37873547)

See also

Configuration of the "SIMATIC S7-1200, S7-1500 Channel" channel (Page 331)

SiePortal: Exchanging large amounts of data between S7-300/400/1500 and WinCC (Entry37873547) (https://support.industry.siemens.com/cs/ww/en/view/37873547)

14.3.3 How to configure a connection

Introduction

The following steps are required for configuring the "SIMATIC S7-1200, S7-1500 Channel":

1. Configuring a connection
2. Configuring tags

Information on the secure communication of STEP 7 in the TIA Portal:

- "Secure Communication" (Page 336)

Requirements

- The communication driver for "SIMATIC S7-1200, S7-1500 Channel" is installed and integrated into the project.
- The SIMATIC project is configured and is available in the automation system.

Procedure

1. Open the menu structure for the "SIMATIC S7-1200, S7-1500 Channel" communication driver in the "Tag Management" editor of the WinCC Explorer.
2. Select the entry "New connection" from the shortcut menu of the channel unit "OMS+".
3. Enter the name of the connection.
4. Select the entry "Connection parameters" from the shortcut menu of the connection. The "Connection" dialog opens.
5. Select an automation system from the "Product family".
6. Enter the IP address of the automation system or the station address.
7. Select an "Access point".
   The access point must reference an interface that belongs to the selected product family.
8. Enter the password for access protection of the automation system:
   - Click "Change".
   - Enter the password for required level.
   - Repeat your entry in the "Repeat password" field.
9. The option "Only allow a secure PLC connection" is activated as the default value.
   This means that the PLC certificates are used for the communication.

   Note

   Secure PLC connection: Only deactivate for test

   Deactivate the option "Only allow a secure PLC connection" only in the test environment.

   For production operations you always use the "Secure Communication".

10. Click "OK" to close the dialog.
11. To create the system tags for connection establishment and connection status, select the "Create tags for activation/deactivation" entry in the shortcut menu of the connection. The following tags are created in the internal tag group "ConnectionStates":
    - @<Connectionname>@ForceConnectionStateEx
    - @<Connectionname>@ConnectionStateEx

See also

"Secure Communication" (Page 336)

14.3.4 "Secure Communication"

"Secure Communication" protects communication with a PLC by means of a certificate.

With TIA Portal from V17, "Secure Communication" is available for S7 PLCs of the following series:

- S7-1200 as of firmware 4.5
- S7-1500 as of firmware 2.9

You can also switch between the connections of the communication channel when using "Secure Communication" in Runtime.

You need to change connections, for example, when you replace hardware or install hardware updates.

Note

Secure PLC connection: Only deactivate for test

Deactivate the secure PLC connection only in the test environment.

For production operations you always use the "Secure Communication".

More information

- Changing connections ("Change Connection") (Page 336)
- "Configurations > Certificates in WinCC > Certificates for communication with S7 PLCs > AUTOHOTSPOT"

See also

Configuration of the "SIMATIC S7-1200, S7-1500 Channel" channel (Page 331)

"SIMATIC S7-1200, S7-1500 Channel" channel (Page 329)

Industry Online Support: "WinCC V7 - Secure Communication" (ID 109798498) (https://support.industry.siemens.com/cs/ww/en/view/109798498)

Industry Online Support: STEP 7 (TIA Portal) - Documentation "Secure Communication" (https://support.industry.siemens.com/cs/products?search=%22secure%20communication%22&dtp=Manual&mfn=ps&pnid=24471&lc=en-US)

SiePortal: STEP 7 (TIA Portal) - Documentation: Signatures and certificates (ID 109798671) (https://support.industry.siemens.com/cs/ww/en/view/109798671/143786688779)

14.3.5 Changing connections ("Change Connection")

You need to change connections, for example, when you replace hardware or install hardware updates.

You can switch between the connections of the communication channel also when using "Secure Communication" in Runtime.

More information

For "Secure Communication": AUTOHOTSPOT

For system tags:

- AUTOHOTSPOT
- AUTOHOTSPOT

Requirement

A connection change depends on the installed firmware.

Firmware of CPUs < V2.9

- Change between two CPUs is possible if a firmware earlier than V2.9 is used on both CPUs. The connection is always established without "Secure Communication".

CPUs with firmware ≥ V2.9

If "Secure Communication" is configured for the PLCs, the combination possibilities of the CPU are dependent on the type of the certificates installed on the PLC:

Source CPU	CPU after connection change *. *	Comments
Self-signed S7 certificate	Self-signed S7 certificate - not trustworthy	Manual trust required
	CA-based S7 certificate - root certificate not trustworthy	Import of certificate data from TIA Portal required
	CA-based S7 certificate - root certificate trustworthy	Neither manual trust nor import of the certificate data necessary Combination occurs, for example, when the root certificate has already been imported into WinCC.
CA-based S7 certificate and its root certificate	Self-signed S7 certificate - not trustworthy	Manual trust required
	CA-based S7 certificate - root certificate not trustworthy	Import of certificate data from TIA Portal required
	CA-based S7 certificate - root certificate trustworthy	Neither manual trust nor import of the certificate data necessary Combination occurs, for example, when the root certificate has already been imported into WinCC.

*) You can also switch to a CPU configured with the same connection parameters as the original CPU.

System tags for changing connections

To change connections, you create the required system tags in the Tag Management.

For each communication connection, you create system tags that contain the corresponding connection name:

- @<Connection name>@<System tag for changing connections>

Tag	Use	Value	Explanation
@<...>@ForceConnectionState	Establish / close connection in the communication channel	1 / 0	Behavior when Runtime is activated: • Start value = 1: The connection is established. • Start value = 0: The connection remains deactivated. Data type: Unsigned 32-bit value Access: reading / writing
@<...>@AlternativeAddress	Alternative CPU connection	String	Properties of the alternative connection The tag must have a start value, for example: • AccessPoint=S7ONLINE;IPAddress=111.111.111.111; The value can be changed subsequently. Data type: Text tag 8-bit character set; length = 255 Access: reading / writing
@<...>@UseAlternativeAddress	Use an alternative connection	1 / 0	Determines the currently used connection: • 1: Alternative connection • 0: Connection to the original connection Data type: Unsigned 32-bit value Access: reading / writing

Example scenario

Initial situation:

- The WinCC project is in Runtime.
- The connection to the "PLC1" CPU is active.
- The "@<PLC1>@AlternativeAddress" system tag contains the valid address of the second CPU, "PLC2".

Changing connections:

- The connection is deactivated:
  - @<PLC1>@ForceConnectionState = 0
- The connection parameters are changed:
  - @<PLC1>@UseAlternativeAddress = 1

  The connection parameters from "@<PLC1>@AlternativeAddress" are adopted.
- The connection is re-established:
  - @<PLC1>@ForceConnectionState = 1

  WinCC establishes the alternative connection to the "PLC2" CPU.

Import of the certificates from TIA Portal

More information: AUTOHOTSPOT

Manual trust

For manual trust, manually move certificate file in the WinCC certificate store. ("Manual Trust")

Note

Restriction

- Manual trust is only supported for the switch of the connection in Runtime.
- You cannot manage root certificates in the WinCC certificate store.
  If you use a CA-based S7 certificate and do not trust its root certificate, import the S7 certificate from TIA Portal to WinCC.
  During loading, the complete certificate chain of the S7 certificate is installed on the WinCC. The WinCC PC trusts the root certificate of the S7 certificate after loading. This means that the PC also trusts all S7 certificates which have the same root certificate.
  More information on root certificates:
  - Industry Online Support: STEP 7 (TIA Portal) - Documentation: Signatures and certificates (https://support.industry.siemens.com/cs/ww/en/view/109798671/143786688779)

Storage location of the certificate store

<Installation path>\Siemens\Automation\device-certificate-store

Example:

- "C:\ProgramData\Siemens\Automation\device-certificate-store"

In the certificate store, you can save certificate revocation lists, as well as manually confirm or reject certificates as trusted ("Manual Trust").

Structure of the certificate store

The certificate store has the following folders:

- "trusted\certs"
  The S7 certificates that the PC trusts
- "trusted\certs\crl"
  Certificate revocation lists contain certificates that have been revoked.
- "untrusted\certs"
  The certificates that the PC does not trust.

Procedure

If the S7 certificate of the target CPU after the connection change in WinCC is unknown, it is stored in the folder "untrusted" during the connection attempt.

1. To confirm a certificate as trusted, move the corresponding "*.DER" file to the "trusted" folder.

You can also move certificates want to reject to the "untrusted" folder afterwards.

See also

Industry Online Support: STEP 7 (TIA Portal) - Documentation: Signatures and certificates (https://support.industry.siemens.com/cs/ww/en/view/109798671/143786688779)

14.3.6 How to configure a tag without optimized block access

Introduction

This section shows you how to configure a tag in WinCC without optimized block access to the address area in the automation system.

Requirement

- The property "Optimized block access" is deactivated for the data block in the TIA Portal.
- The "SIMATIC S7-1200, S7-1500 Channel" must be integrated into the project.
- A connection must be created in the "OMS+" channel unit.

Notes on the configuration of an 8-bit text tag

For an 8-bit text tag in the "SIMATIC S7-1200, S7-1500 Channel", WinCC only supports the S7 string type consisting of a control word and the user data of the string:

- To configure an 8-bit text tag in WinCC, enter the address of the control word that exists in the automation system (AS) memory before the user data.
  The first byte of the control word contains the customized maximum length of the string, the second byte the actual length.
- With respect to creating the data structure in the automation system memory, you must note that the length of the 8-bit text tag configured in WinCC is extended by 2 bytes of the control word.
  If the data structures of the 8-bit text tags are created directly one after the other in the memory, the subsequent data is overwritten.
- While reading, the control word is read along with the user data and the current length is evaluated in the second byte.
  Only the user data according to the current length contained in the second control byte is transferred to the 8-bit text tags of WinCC.
- While writing, the actual length of the string is ascertained ("0" characters) and the control byte with the current length is sent to the automation system along with the user data.

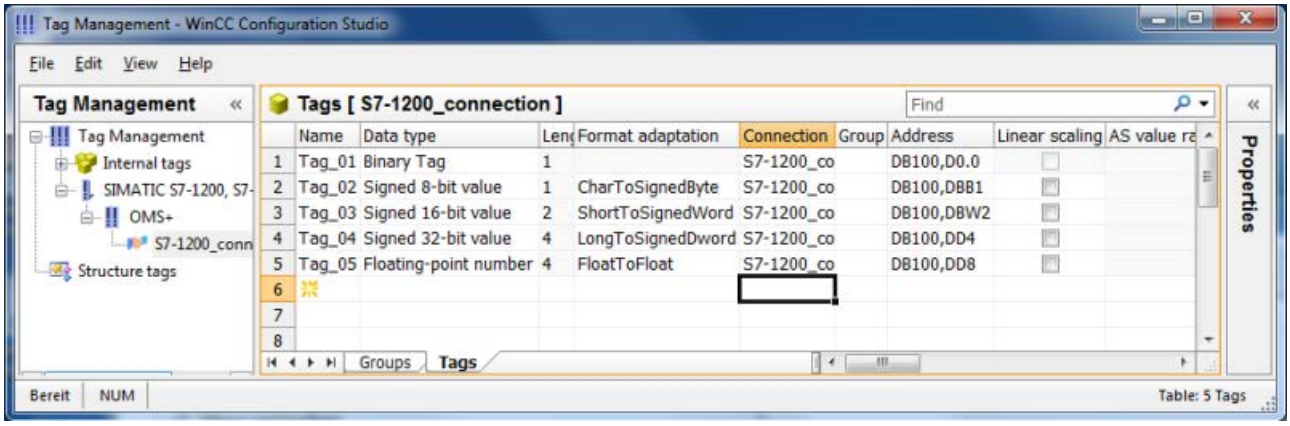
Procedure

1. Select the required connection.
2. Click the "Tags" tab below the table area.

3. Enter a name for the tag in the top free cell of the "Name" column.
   Configure the following settings in the table area or on the right-hand side in the "Properties - Tags" data area.
4. Select one of the supported data types.
5. Click ... in the "Address" column.
6. Enter the tag address.
7. Select the "Quality code" check box if the tag is with quality code and you wish to use it in WinCC.
   The code must also exist in the automation system.
   The check box can only be activated if the "DB" data area is selected.
8. Close the dialog by clicking "OK".

Result

Tags without optimized block access are configured in the Tag Management.



14.3.7 How to configure a tag with optimized block access

Introduction

This section shows you how to configure a tag in WinCC with optimized block access to the address area in the automation system.

You import the tags from the controller into your WinCC project.

Load online changes is not possible

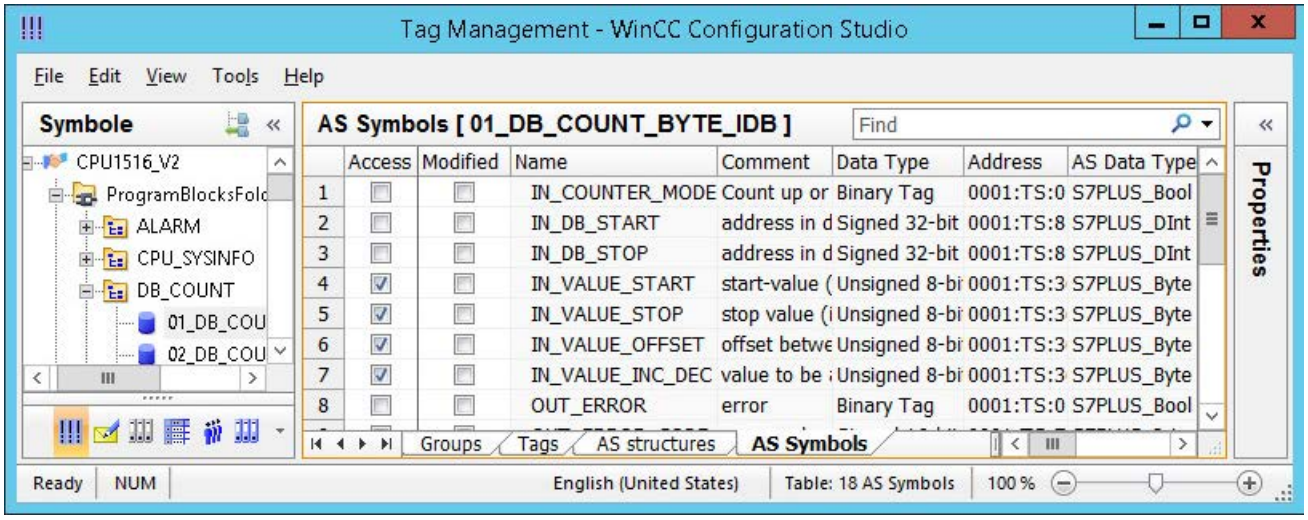
You cannot transfer tags that you have created as AS symbols via "Load to AS" in Runtime with load online changes.

Requirement

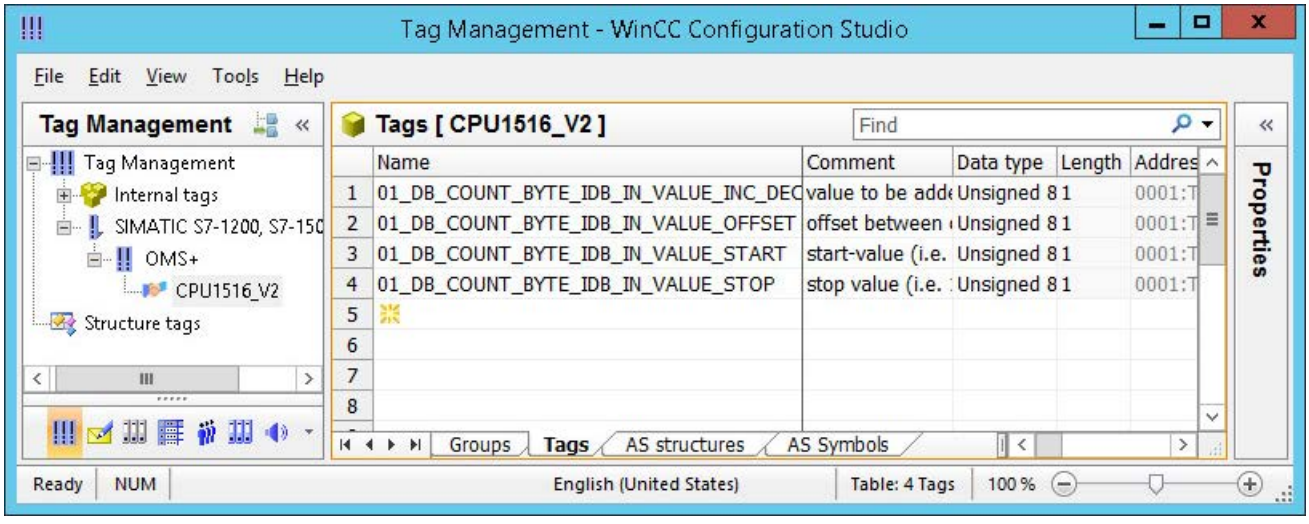
- The property "Optimized block access" is activated for the data block in the TIA Portal.
- The "SIMATIC S7-1200, S7-1500 Channel" must be integrated into the project.
- A connection must be created in the "OMS+" channel unit.
- The connection must be established in Runtime.

Procedure

1. Select the required connection.
2. Select "AS Symbols > Read from AS" from the shortcut menu of the connection.
   The available controller data is loaded and the "Symbols" view opens.
   The loaded data is displayed in the table area in the "AS Symbols" tab.
   If the loaded data also contains structures, the "AS structures" tab is displayed additionally.



3. The AS symbols are not automatically included in tag management.
   To transfer the required AS symbols to the "Tags" tab, activate the respective check box in the "Access" column.
   The selected tags are now contained in the tag management.



Editing AS symbols without connection to the controller

You can configure AS symbols offline independent of a connection to the controller.

To do so, you save the loaded AS symbols in a file.

1. Select the required connection.
2. Select "AS Symbols > Save to file" from the shortcut menu of the connection.

You can then load the AS symbols to the tag management in the offline project.

1. Select the required connection.
2. Select "AS Symbols > Load from file" from the shortcut menu of the connection.

You can find additional information about offline configuration under Providing an export from the TIA Portal project (Page 345).

Synchronizing WinCC tags with the controller

After loading from the controller or a file, the tag management checks the properties of the AS symbols.

Address, data type and tag name are compared with the properties of the AS symbol in the WinCC project.

- If the properties of a symbol do not match, the "Modified" field on the "AS Symbols" tab is activated.
  The respective property field is highlighted in red. The tooltip of the field contains additional details.
- If a WinCC tag is not found in the controller, the entire row of the connected AS symbol is highlighted in red.

This reaction occurs in the following cases, for example:

- The WinCC project was created with WinCC V7.3. Migrated projects do not yet contain all synchronized information.
- The address of the AS symbol has been changed in the controller, for example, due to configuration changes in the TIA Portal.
- The data type or the name of the AS symbol has been changed.
- The AS symbol has been deleted in the controller.

Importing tags again

To synchronize the properties, update the AS symbols used in the WinCC project.

Proceed as follows:

1. Select the modified AS symbol in the "AS Symbols" tab.
   To update a migrated project, select all lines.
2. Deactivate the "Modified" field.

The parameters of the AS configuration are read in again and used in the Tag Management.

Note

Before migrating a TIA Portal project: Updating AS symbols

When you upgrade a TIA Portal version, adhere to the following sequence:

1. Update all AS symbols that are used as WinCC tags.
2. Migrate the TIA Portal project.
3. Load the controller in the TIA Portal.
4. Update all AS symbols that are used as WinCC tags again.

This ensures that the assignment of the WinCC tags to the AS symbols is maintained in the WinCC project after loading.

Otherwise the tags may not be read, as the assignment is no longer up to date.

See also

How to export AS project data (Page 359)

How to configure AS structures (Page 355)

14.3.8 How to download AS symbols offline

14.3.8.1 Providing an export from the TIA Portal project

Introduction

You can configure the SIMATIC S7-1200, S7-1500 Channel offline:

1. Export the data records from an existing TIA Portal project.
2. Load the export file into the WinCC project.

To export data records from a TIA Portal project, use the "SIEMENS SIMATIC SCADA Export" tool.

The export tool can be downloaded from Industry Online Support:

- SiePortal: Download "SIMATIC SCADA Export for TIA Portal" (ID 109748955) (https://support.industry.siemens.com/cs/ww/en/view/109748955)

You can also find an application example for the tool in Industry Online Support:

- SiePortal: "SIMATIC SCADA Export" documentation for WinCC communication with S7-1500 / S7-1200 / ET 200SP (ID 101908495) (https://support.industry.siemens.com/cs/ww/en/view/101908495)

Exporting data records with "SIEMENS SIMATIC SCADA Export"

1. Install the tool beforehand.
2. In the TIA Portal project, select the "Export to SIMATIC SCADA" entry in the PLC context menu.

Result

A ZIP file with the project data is exported and is available for import into a WinCC project.

Import the exported data into the "SIMATIC S7-1200, S7-1500 Channel" communication channel.

More export options

Exports from the WinCC Tag Management with the following file formats are also supported for import:

Format	Contents	Description
*.bin	Binary data	• "Tag Management" view > Context menu of the connection: AS Symbols > Save to file Not supported by the "SIMATIC S7 Protocol Suite" channel.
*.sdz	Structured export	• "Symbols" view > Menu: Edit > Export Also exports the structure information from the navigation area.

See also

How to export AS project data (Page 359)

How to configure AS structures (Page 355)

"Secure Communication" (Page 336)

SiePortal: "SIMATIC SCADA Export" documentation for WinCC communication with S7-1500 / S7-1200 / ET 200SP (ID 101908495) (https://support.industry.siemens.com/cs/ww/en/view/101908495)

SiePortal: Download "SIMATIC SCADA Export for TIA Portal" (ID 109748955) (https://support.industry.siemens.com/cs/ww/en/view/109748955)

14.3.8.2 Import with the Load Assistant

Introduction

You can use the Load Assistant to configure the "SIMATIC S7-1200, S7-1500 Channel" offline.

Requirement

- The AS was compiled in the TIA Portal.
- The corresponding PLC configuration data is exported and is available as a ZIP file.
- The communications processor and associated hardware driver are installed in the WinCC project.
- A connection is created in the "SIMATIC S7-1200, S7-1500 Channel" communications channel.
- The "Tag Management" editor is open.

Note

Creating a backup

Create a backup of the project before starting the Load Assistant.

It is not possible to undo steps that were carried out by the Load Assistant.

Preparing the Load Assistant

During import, the Load Assistant offers the option of filtering a TIA Portal export file by structure or instance type. When importing, you can automatically create structure tags, archive tags, and messages.

Before using the Load Assistant when importing TIA Portal export files, create a filter file in XML format.

Structure of a filter file

- Definition of the XML version and the <AED> element:
```
<?xml version="1.0" encoding="utf-8"?>
<AED>
  <!--Insert ImportCritera-->
</AED>
```
- Define the import criteria in the <AED> element.
```
<ImportCriteria>
  <!--Insert ImportCriteria 1-->
</ImportCriteria>
<ImportCriteria>
  <!--Insert ImportCriteria 2-->
</ImportCriteria>
<ImportCriteria>
  <!--Insert ImportCriteria 3-->
</ImportCriteria>
...
```
- Import criteria contain a filter group and one or more processes.
- A filter group contains one or more filters.
  Several filters in a filter group are linked with OR.

- A filter contains structure types or instance names as filter criteria.
  Several filter criteria in a filter are linked with AND.
- Define processes for the automated creation of structure types, structure tags, archive tags, and messages.
  The following processes are available for selection:
  - Process for creating structure types
    ```
    <Process ID="0" Description="CreateStructuresType">
      <!--Create or update structure types-->
    </Process>
    ```
  - Process for creating structure instances
    ```
    <Process ID="1" Description="CreateStructuresInstances">
      <!--Create or update structure instances-->
    </Process>
    ```
  - Process for creating archive tags
    ```
    <Process ID="2" Description="CreateLogingTag">
      <!--Create Tag Logging-->
    </Process>
    ```
  - Process for creating messages
    ```
    <Process ID="3" Description="CreateMessages">
      <!--Create messages-->
    </Process>
    ```

Examples of import criteria

```
<?xml version="1.0" encoding="utf-8"?>
<AED>
<!--Following ImportCriteria creates mentioned StructuresTypes and all of its instances-->
<ImportCriteria>
  <FilterGroup>
    <Filter>
<!-- Import and create following structure types -->
      <StructureType>EnS_typeLmgtGeneralData</StructureType>
    </Filter>
    <Filter>
      <StructureType>EnS_typeLmgtStorageElement</StructureType>
    </Filter>
  </FilterGroup>
<!--Create following process (1) node to create instances of structure types mentioned
under FilterGroup/Filter. If this node is not created, no instances (tags) will be created
in WinCC-->
  <Process ID="1" Description="CreateStructuresInstances"></Process>
</ImportCriteria>
```

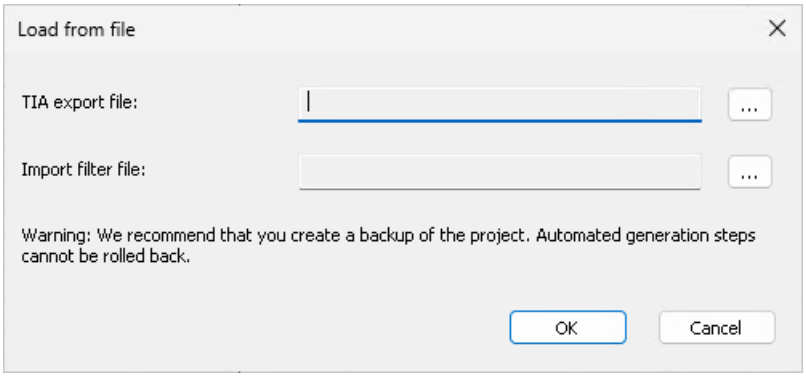
```
<!--Following ImportCriteria creates mentioned StructuresTypes, all of its instances and
archives-->
<ImportCriteria>
  <FilterGroup>
    <Filter>
<!-- Import and create following structure type-->
      <StructureType>EnS_typeEnergyBasic</StructureType>
<!--InstanceName will be used only in processes 2 and 3-->
      <InstanceName>*energyBasic.energy</InstanceName>
    </Filter>
    <Filter>
<!-- Import and create following structure type-->
      <StructureType>EnS_typeEnergyBasic</StructureType>
<!--InstanceName will be used only in processes 2 and 3-->
      <InstanceName>*energyBasic.power</InstanceName>
    </Filter>
  </FilterGroup>
<!--Create following process node to create instances of structure types mentioned under
FilterGroup/Filter. If this node is not created, no instances (tags) will be created in
WinCC-->
  <Process ID="1" Description="CreateStructuresInstances"></Process>
<!--Create following process (2) to create archives in Tag Logging editor for each
<InstanceName> like "*energyBasic.energy" and "*energyBasic.power of structure type
"EnS_typeEnergyBasic". <TagName> is used to make the tag name column unique in the editor.
Please note that, only creation of “Process value archives” with following mentioned
properties is supported. More than one process value archive can be configured only under
different <ImportCriteria>-->
  <Process ID="2" Description="CreateLogingTag">
<!--Create structure types and its instances (tags)-->
<!--Create Tag Logging-->
    <ProcessValueArchive>
      <ArchiveName>"EnS_EnergyArchive"</ArchiveName>
<!--if empty, use the found <InstanceName> name-->
      <ProcessTag>"@EnS_SystemTag"</ProcessTag>
<!--If not mentioned, by default tag supply is "System", No -> "Manual input"-->
      <SystemTagSupply>No</SystemTagSupply>
<!--If Empty, "Cyclical, Continous"-->
      <AcquisitionType>"After every change"</AcquisitionType>
<!--If Empty, "500 ms"-->
      <AcquisitionCycle>500 ms</AcquisitionCycle>
<!--If Empty, "500 ms"-->
      <ArchivingCycle/>
<!--Tag Instance name + modificators e.g remove/prefix/suffix some string from the instance
name-->
      <TagName Prefix="" Remove="energyBasic." Suffix=""></TagName>
    </ProcessValueArchive>
  </Process>
</ImportCriteria>
```

```
<!--Following ImportCriteria creates mentioned StructuresTypes, all of its instances and
archives-->
<ImportCriteria>
  <FilterGroup>
    <Filter>
<!-- Import and create following structure type-->
      <StructureType>EnS_typeEnergyBasic</StructureType>
<!--InstanceName will be used in processes 2 and 3-->
      <InstanceName>*energyBasic.energyCounter</InstanceName>
    </Filter>
  </FilterGroup>
<!--Create Tag Logging-->
  <Process ID="2" Description="CreateLogingTag">
    <ProcessValueArchive>
      <ArchiveName>"EnS_EnergyArchive"</ArchiveName>
<!--Tag Instance name + modificators e.g remove/prefix/suffix some string from the instance
name-->
      <TagName Prefix="" Remove="energyBasic." Suffix=""></TagName>
    </ProcessValueArchive>
  </Process>
</ImportCriteria>
<!--Following ImportCriteria creates mentioned StructuresTypes, all of its instances and
creates messages-->
<ImportCriteria>
  <FilterGroup>
    <Filter>
<!-- Import and create following structure type-->
      <StructureType>EnS_typeLmgtActuatorElement</StructureType>
<!--InstanceName will be used in processes 2 and 3-->
      <InstanceName>*manualCommand</InstanceName>
    </Filter>
  </FilterGroup>
<!--Create structure types or update-->
  <Process ID="0" Description="CreateStructuresType"></Process>
<!--Create instances  or update (tags) -->
  <Process ID="1" Description="CreateStructuresInstances"></Process>
<!--Create following process node (3) to create bit alarms for each <InstanceName> like
"*manualCommand" of structure type "EnS_typeLmgtActuatorElement". The found instance name
will be trigger tag for the created alarm. As of now only bit alarm with "error" class is
supported.-->
  <Process ID="3" Description="CreateMessages">
    <Message>
      <MessageBit>0</MessageBit>
      <MessageText>Manual command</MessageText>
    </Message>
  </Process>
</ImportCriteria>
```

```
<!--Following ImportCriteria creates mentioned StructuresTypes, all of its instances and
creates messages-->
<ImportCriteria>
  <FilterGroup>
    <Filter>
<!-- Import and create following structure type-->
      <StructureType>EnS_typeLmgtActuatorElement</StructureType>
<!--InstanceName will be used in processes 2 and 3-->
      <InstanceName>*status_inOutVariables_enableActuator</InstanceName>
    </Filter>
  </FilterGroup>
<!--Create messages-->
  <Process ID="3" Description="CreateMessages">
    <Message>
      <MessageBit>0</MessageBit>
     <MessageText>TRUE: Actuator is available for switching; FALSE: Actuator is ignored by
LMGT system</MessageText>
    </Message>
  </Process>
  <Process ID="1" Description="CreateStructuresInstances"></Process>
</ImportCriteria>
</AED>
```

Using the Load Assistant

1. Select "AS Symbols > Load Assistant" from the context menu of the connection.
   The Load Assistant is displayed.



2. Select the TIA Portal export file to be loaded.
3. Select the filter file.
   The controller data that matches the import criteria of the filter file is loaded.
   The configuration is imported and the "Symbols" view opens.

Information on how to proceed:

- "Editing imported AS symbols (Page 354)"

See also

Providing an export from the TIA Portal project (Page 345)

14.3.8.3 Configuring an import manually

Introduction

You can configure the SIMATIC S7-1200, S7-1500 Channel manually offline:

1. Export the data records from a TIA Portal project.
2. Load the export file into the WinCC project.

Supported export formats

The following file formats are supported for the import:

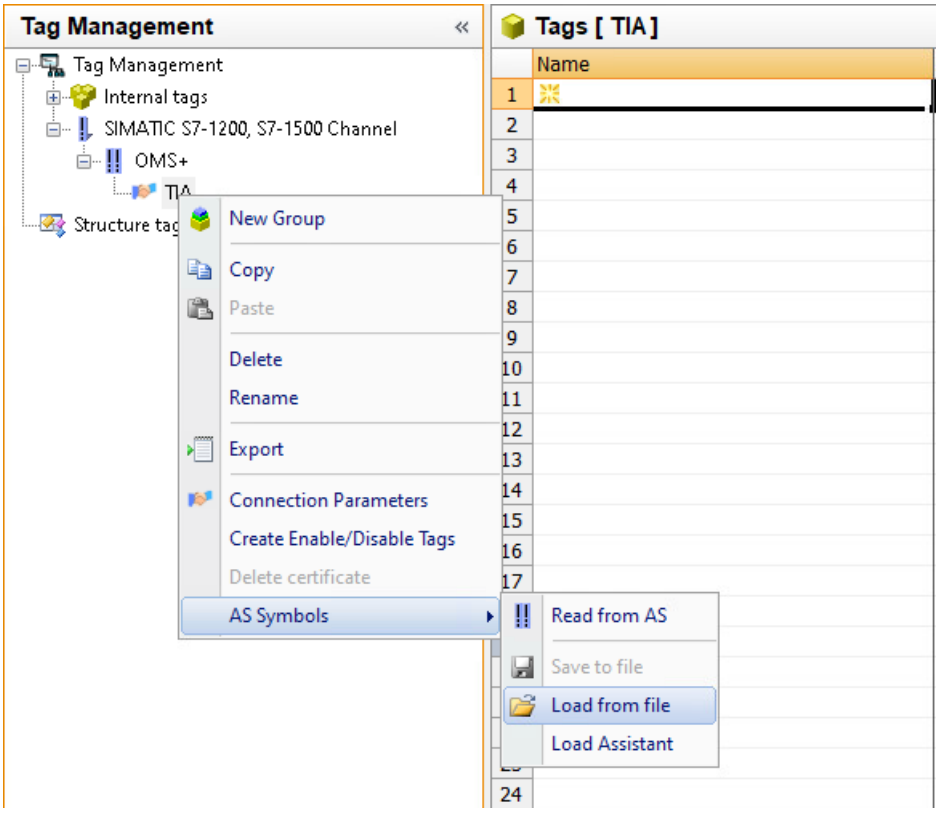
Format	Contents	Description
*.bin	Binary data	Export from the WinCC Tag Management: • "Tag Management" view > Context menu of the connection: AS Symbols > Save to file Not supported by the "SIMATIC S7 Protocol Suite" channel.
*.sdz	Structured export	Export from the WinCC Tag Management: • "Symbols" view > Menu: Edit > Export Also exports the structure information from the navigation area.
*.zip	TIA Portal export file	Export from the TIA Portal with the "SIEMENS SIMATIC SCADA Export" tool.

Requirement

- The AS was compiled in the TIA Portal.
- The corresponding PLC configuration data is exported and is available as a ZIP file.
- The communications processor and associated hardware driver are installed in the WinCC project.
- A connection is created in the "SIMATIC S7-1200, S7-1500 Channel" or "SIMATIC S7 Protocol Suite".
- The "Tag Management" editor is open.

Procedure for "Load from file"

1. Select "AS Symbols > Load from file" from the context menu of the connection.



2. Select the desired data records to be loaded.
   - The available controller data is loaded.
   - The configuration is imported and the "Symbols" view opens.
   - The loaded data is displayed in the "AS Symbols" tab in the table area and is available for further processing.
   - If the loaded data also contains structures, the "AS Structures" tab is displayed additionally.
3. Create the WinCC tags.
   Use the imported AS symbols to configure the archive tags and messages.

Information on how to proceed:

- "Editing imported AS symbols (Page 354)"

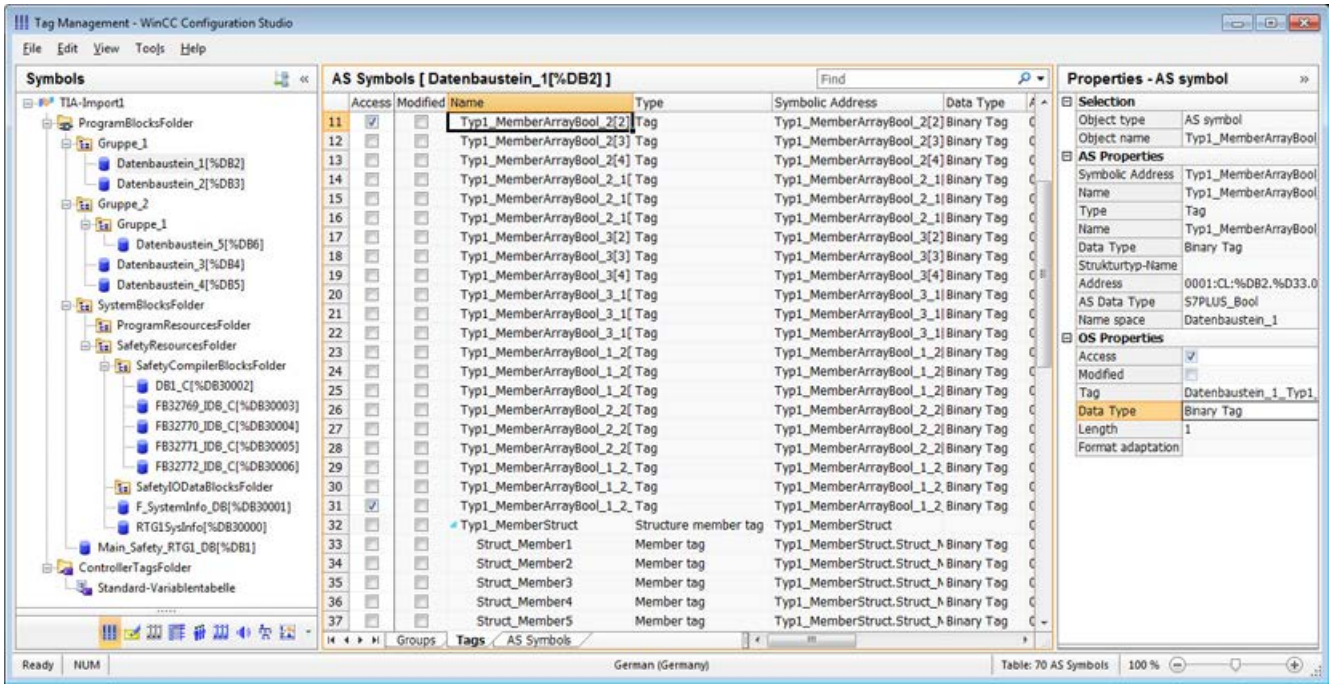
See also

Providing an export from the TIA Portal project (Page 345)

14.3.8.4 Editing imported AS symbols

Requirement

- AS symbols have been imported.
- The loaded data is displayed in the "AS Symbols" tab in the table area and is available for further processing.
  If the loaded data also contains structures, the "AS Structures" tab is displayed additionally.



Displaying AS symbols

Use the "Symbols" button to switch between the default view and the "Symbols" view in the Tag Management: 

The button is available only after the data records have been loaded.

After the editor is closed, the "AS Symbols" and "AS Structures" tabs are hidden once again.

Navigation area

The representation of the data in the structure tree corresponds to the hierarchy from the TIA Portal.

Table area

If a WinCC tag does not match the control tags, the boxes in of the "Modified" column are automatically selected.

Use this column to filter the control tags.

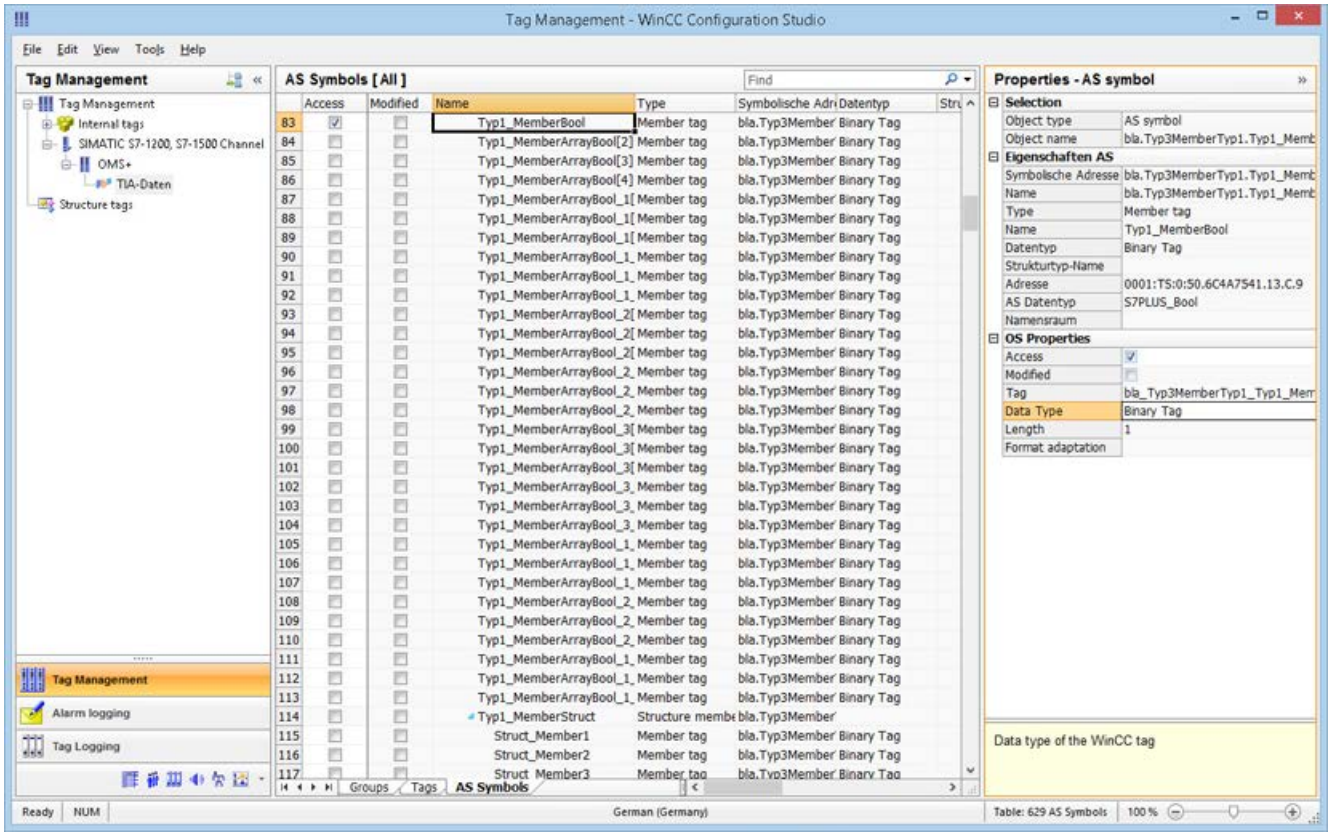
To create a WinCC tag from the imported control tag, select the corresponding check box in the "Access" column.

AS symbols in the Tag Management

You also have access to the AS symbols in the Tag Management via the "AS symbols" tab.

In contrast to the data block-related "Symbols" view, all available tags of the controller are displayed.

This view also shows tags that have already been configured, regardless of whether these tags are available on the AS.



See also

How to configure AS structures (Page 355)

Import with the Load Assistant (Page 346)

Configuring an import manually (Page 352)

14.3.9 How to configure AS structures

Introduction

If you load AS symbols, structured data types of the control system (UDT) are also imported. Structures of the "STRUCT" type are not taken into consideration.

The procedure depends on the communication channel:

- SIMATIC S7 Protocol Suite:
  - Load from file
- SIMATIC S7-1200, S7-1500 Channel
  - Load from file
  - Read from AS

Note

Behavior during loading

"Load from file" loads the absolute and symbolic address of a tag.

"Read from AS" only loads the symbolic address of a tag.

AS structures in Tag Management

The AS structures are displayed in the default view and in the "Symbols" view in the "AS structures" tab.

You have the following possibilities to use the AS structures in WinCC:

- Create a WinCC structure type for the AS structure tag.
  The structure is created as a structure type under "Structure tags" in the WinCC Tag Management.
  A structure type element is also created for each contained "Tag type member".
- Assign a WinCC structure type to the AS structure tag.
  Then select a structure type element of the selected structure type for each "Tag type member".

You change the name of the WinCC structure type and the structure type elements in the Tag Management. The assignment of the AS structure is automatically adjusted.

Note

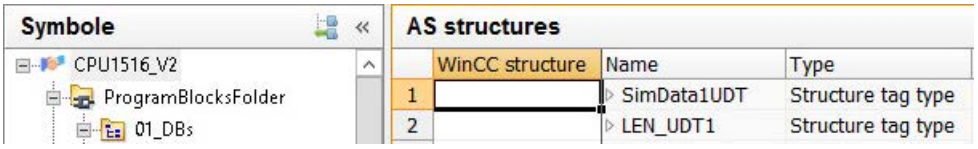
It is possible to configure nested AS structures as of WinCC V8.1.

Requirement

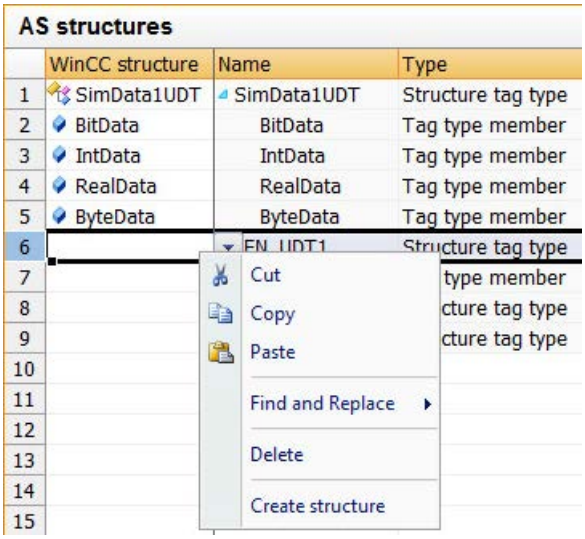
- You have access to the configuration data of the PLC by one of the following methods:
  - A connection to the PLC is established in Runtime.
  - The exported configuration data is available, for example, as a ZIP file.
- A connection is created in the "SIMATIC S7-1200, S7-1500 Channel" or "SIMATIC S7 Protocol Suite".

Procedure

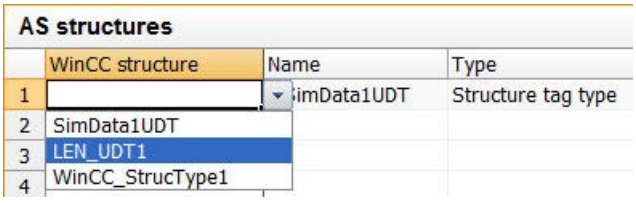
1. Load the AS symbols via "Read from AS" or "Load from file" or "Load Assistant".
   The loaded messages are displayed in the "Symbols" Tag Management view.
   The loaded structures are displayed in the "AS structures" tab.
   The structure names are transferred when loading from the AS.



2. Click "AS structures".
   To display the elements of a structure, click the arrow in front of the structure name.
3. Select the entire row of a structure and select the "Create structure" entry in the context menu.

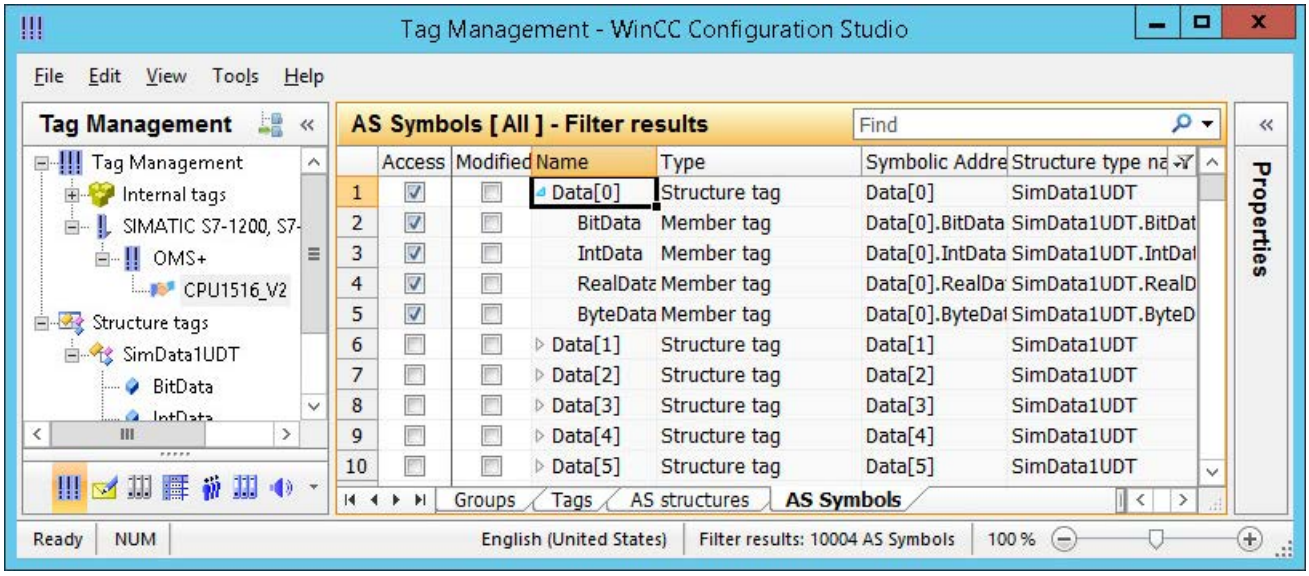


   Alternatively, select a structure type that has already been created in the WinCC Tag Management.
   Then assign a structure type element to the "Tag type member".



   A structure type is created in the WinCC Tag Management for each "Structure tag type" of the AS structures.
   A structure type element is created for each "Tag type member".
4. Select the "AS Symbols" tab in the "Tag Management" view.

5. To only have structure tags and member tags displayed, filter for the desired AS structure in the "Structure type name" column.



6. To access an AS structure tag in the WinCC Tag Management, enable the "Access" field.
   The contained member tags are automatically enabled.
   The AS structure tag is created as a structure tag in the WinCC Tag Management.

Result

Through the structure types and structure tags in WinCC Tag Management you have access to the AS structure tags.

In this way you can, for example, access AS structures in WinCC faceplate types and represent them in faceplate instances.

See also

Providing an export from the TIA Portal project (Page 345)

How to export AS project data (Page 359)

How to configure a tag with optimized block access (Page 341)

14.3.10 How to update the AS structures

Introduction

If you change an AS structure that has already been imported in the automation system, update the AS structure in WinCC. To do this assign the changed tag type members to the respective structure type elements.

Requirement

- You have access to the configuration data of the PLC by one of the following methods:
  - A connection to the PLC is established in Runtime.
  - The exported configuration data is available, for example, as a zip file.
- A connection is created in the "SIMATIC S7-1200, S7-1500 Channel" or "SIMATIC S7 Protocol Suite".

Procedure

1. Modify the existing structure type in the WinCC Tag Management according to the structure in the AS system, for example by deleting or adding the structure type element.
2. Load the updated AS symbols via "Read from AS" or "Load from file".
   The loaded messages are displayed in the "Symbols" Tag Management view.
   The loaded structures are displayed on the "AS structures" tab.
   The structure names are transferred when loading from the AS.
3. Click "AS Structures" and open the elements of the changed structure by clicking the arrow in front of the structure name.
4. For all changed elements assign a respective structure type element to the "Tag type member".
5. Select the "AS Symbols" tab in the "Tag Management" view.
6. To only have structure tags and member tags displayed, filter for the desired AS structure in the "Structure type name" column.
   The changed elements are identified with a "Modified" check mark in the field.
7. Deactivate the "Modified" field.
   The access to the modified tag type member is activated.

Result

You have updated the existing structure.

14.3.11 How to export AS project data

Exporting AS symbols

You use the export files for the offline configuration.

You can export AS project data to the following formats:

Communication channel	Exported data	Format of the export file
SIMATIC S7-1200, S7-1500 Channel	AS symbols and AS structures	Binary data: *.bin Structured export: *.sdz
SIMATIC S7 Protocol Suite	AS symbols and AS structures	Structured export: *.sdz

Requirement

- A connection is created in the "SIMATIC S7-1200, S7-1500 Channel" or "SIMATIC S7 Protocol Suite".
- You have loaded AS project data and configured it in WinCC.

Procedure: Exporting binary data

1. Select the connection in the Tag Management.
2. Select the "AS Symbols > Save to file" entry from the shortcut menu.
   The "Export" dialog opens.
3. Select the storage path and enter a file name.
   Close the dialog with the "Export" button.
   The configuration data is exported as a binary data set to a .bin file.

Procedure: Exporting structured data

1. Select the "Symbols" view in the Tag Management.
2. Select the "Edit > Export" menu command.
3. Select the storage path and enter a file name.
   Close the dialog with the "Export" button.
   The configuration data is exported to an *.sdz file.
   The structured export also contains the structure information from the navigation area.

See also

How to configure a tag with optimized block access (Page 341)

Providing an export from the TIA Portal project (Page 345)

How to configure AS structures (Page 355)

14.3.12 Redundant System S7-1500R/H

CPU redundancy

CPU redundancy means the CPUs are available twice. A SIMATIC memory card must be inserted in each of the CPUs.

During redundant operation, both CPUs execute the user program in parallel.

If one CPU fails, the other CPU maintains control over the process.

Primary CPU	CPU conducting process If the R/H system is in RUN-Redundant system state, the primary CPU performs the process.
Backup CPU	The following CPU The backup CPU synchronously executes the user program and assumes process control in the event of a failure of the primary CPU.

You can find additional information in manuals for S7-1500R/H at Product Support:

- S7-1500 > Redundant CPUs (https://support.industry.siemens.com/cs/products?dtp=Manual&mt=2001&mfn=ps&pnid=25152&lc=en-WW)
- "Redundant System S7-1500R/H" manual (https://support.industry.siemens.com/cs/ww/en/view/109754833) (10/2018)

Operating objectives

In practice, redundant automation systems are used to achieve higher availability or fail-safety:

- Fault-tolerant systems:
  Reduction of the probability of production downtimes through parallel operation of two systems.
- Fail-safe systems:
  Protection of life, environment and capital by safe shutdown to a safe idle position.

⚠ WARNING

Difference between fault-tolerant and fail-safe systems

S7-1500R/H is a fault-tolerant automation system, but not a fail-safe system.

You must not use the S7-1500R/H system to control safety-relevant processes.

Usage

In redundantly operated systems, the failure or malfunction of individual automation components must not affect the operation of the system.

Redundant systems S7-1500R/H are used in the following areas, for example:

- Tunnel
- Airports, e.g. baggage handling systems
- Subways
- Shipbuilding
- Wastewater treatment plants
- High-bay warehouse

Functions of the S7-1500 R/H CPUs

The S7-1500R/H redundant system is based on media redundancy (MRP) in the PROFINET ring.

The display of the CPU shows control and status information in different menus. Fast access to diagnostic messages minimizes downtimes of the system during servicing.

The CPUs support trace functions for all CPU variables for effective commissioning and optimization of drives and control loops.

Media redundancy (MRP)

The "Media Redundancy Protocol" (MRP) is a function for ensuring network and system availability.

The two CPUs in the redundant system must be located in a PROFINET ring that uses the MRP media redundancy protocol.

- S7-1500R uses the PROFINET ring to synchronize the two CPUs.
- S7-1500H uses the redundancy connections over fiber-optic cables to synchronize the two CPUs.
  The PROFINET ring via the PROFINET X1 interfaces of the H-CPUs is also mandatory for S7-1500H.

Additional information on media redundancy and system redundancy S2:

- "Redundant System S7-1500R/H" manual (https://support.industry.siemens.com/cs/ww/en/view/109754833): Section "System overview > S7-1500 R/H-CPUs"
- "PROFINET with STEP 7 V15" function manual (https://support.industry.siemens.com/cs/ww/en/view/49948856)

See also

Configuring redundant control systems (Page 362)

Software redundancy for S7-1500R/H (Page 365)

Find: S7-1500 > Redundant CPUs (https://support.industry.siemens.com/cs/products?dtp=Manual&mt=2001&mfn=ps&pnid=25152&lc=en-WW)

"Redundant System S7-1500R/H" manual (https://support.industry.siemens.com/cs/ww/en/view/109754833)

"PROFINET with STEP 7 V15" function manual (https://support.industry.siemens.com/cs/ww/en/view/49948856)

14.3.13 Configuring redundant control systems

Fully configure the CPU redundancy in STEP 7 and in the TIA Portal.

In WinCC, create only the corresponding connection to the IP address of the S7-1500 R/H CPU and create the script for the software redundancy.

IP addresses

Device IP addresses

In order for the interfaces of the CPUs and the IO devices to be accessible, the interfaces within the network require unique IP addresses, the device IP addresses.

Use the device IP address for the "Software redundancy" function.

MAC addresses

The CPUs have a unique MAC address for each interface and its ports.

The MAC addresses of the PROFINET ports are required for the LLDP protocol, e.g. for the "neighborhood detection" function.

The number range of the MAC addresses is continuous. The first and the last MAC address are printed on the type label on the right side of the CPU.

System IP addresses

The redundant system S7 1500R/H additionally supports system IP addresses:

System IP address X1	System IP address for the PROFINET X1 interfaces of two CPUs
System IP address X2	System IP address for the PROFINET X2 interfaces of two CPUs

You use the system IP addresses for communication with other devices, e.g. HMI devices, CPUs, PG/PC.

The devices always communicate with the primary CPU of the redundant system via the system IP address. This ensures that the communication partner can communicate with the new primary CPU (previously backup CPU) in the system state RUN-Solo after a failure of the primary CPU.

The system IP addresses are activated in STEP 7.

Virtual MAC addresses

Each system IP address includes a virtual MAC address.

The virtual MAC addresses of the two PROFINET interfaces must differ from each other and from the MAC addresses of the CPUs.

Information on configuration of the system IP addresses and the virtual MAC addresses:

- "Redundant System S7-1500R/H" manual (https://support.industry.siemens.com/cs/ww/en/view/109754833): Section "Configuration > Configuration procedure"

Advantages of system IP addresses / device IP addresses

System IP addresses have the following advantages over device IP addresses:

- The communication partner communicates specifically with the Primary CPU.
- The communication of the redundant system S7-1500R/H via a system IP address continues to function even if the primary CPU fails.

Software redundancy over device IP addresses has the following advantage:

- Redundant physical connection (network cable)

Further information about the IP addresses:

- "S7-1500R/H redundant system" manual (https://support.industry.siemens.com/cs/ww/en/view/109754833): Section "System overview > System and device IP addresses"
- "PROFINET with STEP 7 V15" function manual (https://support.industry.siemens.com/cs/ww/en/view/49948856)

HMI devices

HMI devices are used for process visualization and machine-level control.

For the redundant system S7-1500R/H, use the same HMI devices as for the standard S7-1500 system.

One or more HMI devices exchange data with the CPUs via the HMI communication, e.g. HMI Basic/Comfort/Mobile Panel.

Configuration of HMI devices

The connection of the HMI device to the redundant system depends on the respective application.

You can configure the following communication options in STEP 7:

- The HMI device communicates with the redundant system via the system IP address.
- The HMI device communicates via the device IP addresses with the respective R/H CPUs, e.g. for diagnostic purposes.

Additional information about using HMI devices:

- "Redundant System S7-1500R/H" manual (https://support.industry.siemens.com/cs/ww/en/view/109754833): Section "Using HMI devices"
- "SIMATIC S7-1500, ET 200MP, ET 200SP, ET 200AL, ET 200pro Communication" function manual (https://support.industry.siemens.com/cs/ww/en/view/59192925)

You can find an overview of all available HMI devices in the Industry Mall.

Configuration

In the WinCC Tag Management, you configure a single connection to the S7-1500 R/H controller under "SIMATIC S7-1200, S7-1500 Channel".

CPU redundancy

The system IP address is used to establish a connection to the primary CPU.

Fully configure the CPU redundancy in STEP 7 and in the TIA Portal.

This includes the following steps:

1. Create projects and R/H CPUs
2. Assign IP addresses (device IP addresses)
   The redundancy IDs of the CPUs are created and assigned in the project tree of STEP 7.
   The upper CPU in the tree always has the redundancy ID 1.
3. Assign system IP addresses (optional)

4. Set scan cycle monitoring time or use default value
5. Create IO devices
6. Connect the IO devices to the two redundant CPUs
7. Assume the MRP role of the CPUs of the S7-1500R/H redundant system
   STEP 7 automatically assigns the MRP role "Manager (auto)" for the PROFINET X1 interfaces of the two CPUs.
8. Define an MRP role for other ring nodes in STEP 7
   In the "Devices" table, you assign the MRP "Client" role to all other nodes of the ring.
9. Parameter assignment for nodes outside the STEP 7 project
   For nodes of the ring that are not in STEP 7, set the MRP "Client" role.

Additional information on configuration and commissioning:

- "Redundant System S7-1500R/H" manual (https://support.industry.siemens.com/cs/ww/en/view/109754833)

"Software redundancy" function

If you want to use software redundancy via device IP addresses, you must also configure a script for the redundancy controller in WinCC.

You can find additional information under: "Software redundancy for S7-1500R/H (Page 365)"

See also

Redundant System S7-1500R/H (Page 360)

Software redundancy for S7-1500R/H (Page 365)

"Redundant System S7-1500R/H" manual (https://support.industry.siemens.com/cs/ww/en/view/109754833)

"PROFINET with STEP 7 V15" function manual (https://support.industry.siemens.com/cs/ww/en/view/49948856)

"SIMATIC S7-1500, ET 200MP, ET 200SP, ET 200AL, ET 200pro Communication" function manual (https://support.industry.siemens.com/cs/ww/en/view/59192925)

14.3.14 Software redundancy for S7-1500R/H

Use the "Software redundancy" function to operate CPU redundancy with a physical connection of the CPUs. Depending on the redundancy configuration, you need one or two network cards or Siemens CPs.

Fully configure the CPU redundancy in STEP 7 and in the TIA Portal.

More information: "Configuring redundant control systems (Page 362)"

Principle of software redundancy

Addressing

Use the device IP addresses of the two redundant controllers instead of the system IP addresses.

System tags

Create the required system tags in the tag management for redundancy control and monitoring.

For each communication connection, you create system tags that contain the corresponding connection name:

- @<Connection name>@<System tag for the software redundancy>

The address of the system tag must point to a free area in the PLC.

Script for software redundancy

To monitor the software redundancy, create a script, in ANSI-C or VBScript, for example.

The script is called cyclically.

- Trigger cycle time: ≥ 10 seconds.

In this script you define the switching between primary CPU and backup CPU.

To run the script as a cyclic action in runtime, enable the "Global Script Runtime" application in the start-up list of the computer.

Example scenario

If the connection is interrupted, WinCC reads or sets the system tags used.

Initial situation

- The WinCC project is in Runtime.
- The connection "PLC1" to the primary CPU is active.
- The system tag "@<PLC1>@AlternativeAddress" contains the valid address of the backup CPU, for example:
  - AccessPoint=abc;IPAddress=111.111.111.111;

Connection fault

1. The loss of connection is determined via the tag "@<PLC1>@ConnectionState".
2. The connection is deactivated:
   - @<PLC1>@ForceConnectionState = 0
3. The connection parameters are changed:
   - @<PLC1>@UseAlternativeAddress = 1
4. The connection is re-established:
   - @<PLC1>@ForceConnectionState = 1
5. WinCC establishes the alternative connection to the backup CPU.

Diagnostics

Use the following system tags to evaluate the operating modes of the two controllers:

- @<PLC1>@OpStateConfiguredAddress
- @<PLC1>@OpStateAlternativeAddress

System tags for software redundancy

Create the following system tags in the WinCC tag management for connection to an S7-1500R/H controller:

Tag	Use	Value	Explanation
@<...>@ForceConnection-State	Establish / terminate redundant connection	1 / 0	Behavior when runtime is activated: • Start value = 1: The connection is established. • Start value = 0: The connection remains deactivated. Data type: Unsigned 32-bit value Access: read / write
@<...>@ConnectionState	Redundant connection status	1 / 0	Querying of the connection status: • 1: The connection is ready for operation. • 0: The connection is interrupted or terminated. The tags of the connection are not archived. Data type: Unsigned 32-bit value Access: read
@<...>@ConfiguredAddress	Connection to the primary CPU	String	Properties of the connection that is configured in the Tag Management Data type: Text tag 8-bit character set; length = 255 Access: read
@<...>@AlternativeAddress	Connection to the backup CPU	String	Properties of the alternative connection to the backup CPU The tag must have a start value, for example: • AccessPoint=abc;IPAddress=111.111.111.111; The value can be changed subsequently. Data type: Text tag 8-bit character set; length = 255 Access: read / write
@<...>@CurrentAddress	Current address	String	Properties of the currently used connection Data type: Text tag 8-bit character set; length = 255 Access: read
@<...>@UseAlternativeAddress	Use an alternative connection	1 / 0	Determines the currently used connection: • 1: Alternative connection to backup CPU • 0: Connection to the primary CPU Data type: Unsigned 32-bit value Access: read / write

Tag	Use	Value	Explanation
@<...>@RedundantCPUs	Redundancy active	1 / 0	Indicates whether the connected controller is redundant: • 1: Yes • 0: No, CPU is not redundant The tag must have a start value. The value can be changed subsequently. Data type: Unsigned 32-bit value Access: read / write
@<...>@OpStateConfiguredAddress	Operating mode of the primary CPU	0 / 4 / 6 / 8 / 22	Operating modes of the controller: • 0: not connected • 4: STOP • 6: STARTUP • 8: RUN • 22: SYNCUP Data type: Unsigned 32-bit value Access: read
@<...>@OpStateAlternativeAddress	Operating mode of the backup CPU		

Example script

```
'VBS381
Option Explicit
Function action

    Dim C1ConnectionState
    Set C1ConnectionState = HMIRuntime.Tags("@Connection1@ConnectionState")
    C1ConnectionState.Read

    Dim C1OpStateConfiguredAddress
    Set C1OpStateConfiguredAddress =
HMIRuntime.Tags("@Connection1@OpStateConfiguredAddress")
    C1OpStateConfiguredAddress.Read

    Dim C1OpStateAlternativeAddress
    Set C1OpStateAlternativeAddress =
HMIRuntime.Tags("@Connection1@OpStateAlternativeAddress")
    C1OpStateAlternativeAddress.Read

    Dim C1UseAlternativeAddress
    Set C1UseAlternativeAddress =
HMIRuntime.Tags("@Connection1@UseAlternativeAddress")
    C1UseAlternativeAddress.Read

    Dim C1ForceConnectionState
    Set C1ForceConnectionState =
HMIRuntime.Tags("@Connection1@ForceConnectionState")
    C1ForceConnectionState.Read


  'check if connection got disconnected
    If C1ConnectionState.Value = 0  Then
  ' force connection state = 0
        C1ForceConnectionState.Value = 0
        C1ForceConnectionState.Write, 1 'synchronous write

  'switch between plc1 and plc2
        Select Case C1UseAlternativeAddress.Value

            Case 0
                    C1UseAlternativeAddress.Value = 1
                    C1UseAlternativeAddress.Write, 1 'synchronous write
            Case 1
                    C1UseAlternativeAddress.Value = 0
                    C1UseAlternativeAddress.Write, 1 'synchronous write

        End Select

  'reconnect
        C1ForceConnectionState.Value = 1
        C1ForceConnectionState.Write, 1 'synchronous write


  'example for handling operating states: stop mode
```

```
  'check if plc is in stop mode; stop mode does not deliver connectionstate
= 0
    Elseif (C1OpStateConfiguredAddress.Value = 4 And
C1UseAlternativeAddress.Value = 0) Or (C1OpStateAlternativeAddress.Value =
4 And C1UseAlternativeAddress.Value = 1) Then

  'set force connection state = 0
        C1ForceConnectionState.Value = 0
        C1ForceConnectionState.Write, 1 'synchronous write

  'switch between plc1 and plc2
        Select Case C1UseAlternativeAddress.Value

            Case 0
                    C1UseAlternativeAddress.Value = 1
                    C1UseAlternativeAddress.Write, 1 'synchronous write
            Case 1
                    C1UseAlternativeAddress.Value = 0
                    C1UseAlternativeAddress.Write, 1 'synchronous write

        End Select

  'reconnect
        C1ForceConnectionState.Value = 1
        C1ForceConnectionState.Write, 1 'synchronous write

    Else

    End If

Set C1ConnectionState = Nothing
Set C1OpStateConfiguredAddress = Nothing
Set C1OpStateAlternativeAddress = Nothing
Set C1UseAlternativeAddress = Nothing
Set C1ForceConnectionState = Nothing

End Function
```

See also

Redundant System S7-1500R/H (Page 360)

Configuring redundant control systems (Page 362)

SIMATIC TI Ethernet Layer 4

15

15.1 WinCC channel "SIMATIC TI Ethernet Layer 4"

Introduction

The communication driver "SIMATIC TI Ethernet Layer 4" handles the link between a WinCC station and a SIMATIC TI505 automation system via Industrial Ethernet. The communication is handled with the ISO transport protocol.

This section shows you how to:

- Configure the data transfer with the "SIMATIC TI Ethernet Layer 4" channel.
- Configure a connection and a tag.

Channel units

It has two channel units to run a maximum of two CP 1613 A2. The functionality of the channel unit is identical. They differ only in the logical device names of the two CP 1613 A2.

The logical device name can be changed via the system parameters of the channel unit. Here, it is also possible to set the parameters for the ISO transport protocol.

The following application capabilities exist:

- Channel unit 505 Ethernet (CP 1413-1) for the communication modules for SIMATIC Industrial Ethernet (e.g. CP 1613 A2).
- Channel unit 505 Ethernet (CP 1413-2) for the communication modules for SIMATIC Industrial Ethernet (e.g. CP 1613 A2).

See also

System parameters of the channel unit (Page 379)

Configuring the tags (Page 375)

Configuring the channel "SIMATIC TI Ethernet Layer 4" (Page 373)

15.2 Data type of the tags

Introduction

Define the required tags for a logical connection. From the WinCC viewpoint, you can access the following data types:

- Binary tag
- Unsigned 8 bit value (is only supported by VMS addressing)
- Signed 8 bit value (is only supported by VMS addressing)
- Unsigned 16-bit value
- Signed 16-bit value
- Unsigned 32-bit value
- Signed 32-bit value
- Floating-point number 32-bit IEEE 754
- Raw data type

15.3 Configuring the Channel

15.3.1 Configuring the channel "SIMATIC TI Ethernet Layer 4"

Introduction

The following steps are required for configuring the channel "SIMATIC TI Ethernet Layer 4".

See also

System parameters of the channel unit (Page 379)

Configuring the tags (Page 375)

How to configure the connection (Page 373)

15.3.2 How to configure the connection

Introduction

The process connection via Industrial Ethernet is possible with the SIMATIC TI505 automation system.

Communication module CP 1434 TF is used in the automation system. The communication is handled with the ISO transport protocol.

The communication module CP 1613 A2 is used in the WinCC system.

Because communication takes place via the ISO transport protocol, it is not necessary to configure the logical connection in the local database.

For a logical connection, WinCC establishes one connection in the transport layer for reading (READ) and one for writing (WRITE). Only if both connections are established is the logical connection also indicated as being established.

Parameters for the READ function

When configuring the connection, parameters are defined for the READ function in WinCC. These are independent of the request used in the SIMATIC TI.

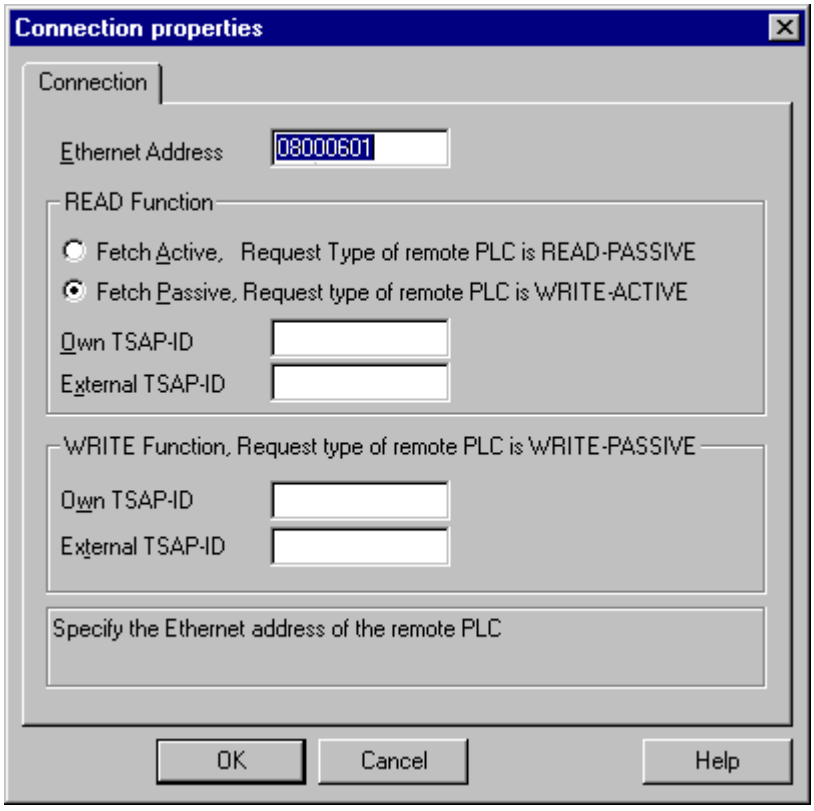
WinCC side	SIMATIC TI side
FETCH-Active (Request "READ-Active")	READ-Passive (Request "READ-Passive")
FETCH-Passive (Request "WRITE-Passive")	WRITE-Active (Request "WRITE-Active")

Parameters "Own TSAP-ID" and "External TSAP-ID" for the WRITE function

WinCC side	SIMATIC TI side
Request "WRITE Active"	Request "WRITE Passive"

Procedure

1. Select the required connection and select "Connection parameters" from the shortcut menu. The "Connection properties" dialog opens.



2. Enter the station address of the SIMATIC TI on the industrial Ethernet bus in the field "Ethernet Address".
3. Now, define the parameters for the READ function in the WinCC system.
   These are independent of the request used in the SIMATIC TI.
4. Then, enter the value in the allocated field "Own TSAP-ID" that was configured in the "Remote parameter" area as "TSAP" while configuring the CP 1434 TF.
5. Now, enter the value in the allocated field "External TSAP-ID" that was configured in the "Local parameter" as "TSAP" while configuring the CP1434 TF.
6. Define the parameters "Own TSAP-ID" and "External TSAP-ID" for the WRITE function accordingly.

See also

Configuring the tags (Page 375)

System parameters of the channel unit (Page 379)

Data type of the tags (Page 372)

15.3.3 Configuring the tags

15.3.3.1 Configuring the tags

Introduction

For a connection between WinCC and the AS via channel "SIMATIC TI Ethernet Layer 4", tags of different data types can be created in WinCC. The following describes how to configure a tag of these data types.

See also

How to Configure a Tag with Byte by Byte Access (Page 378)

How to configure a tag with bit-wise access (Page 377)

How to configure the address of a tag (Page 375)

15.3.3.2 How to configure the address of a tag

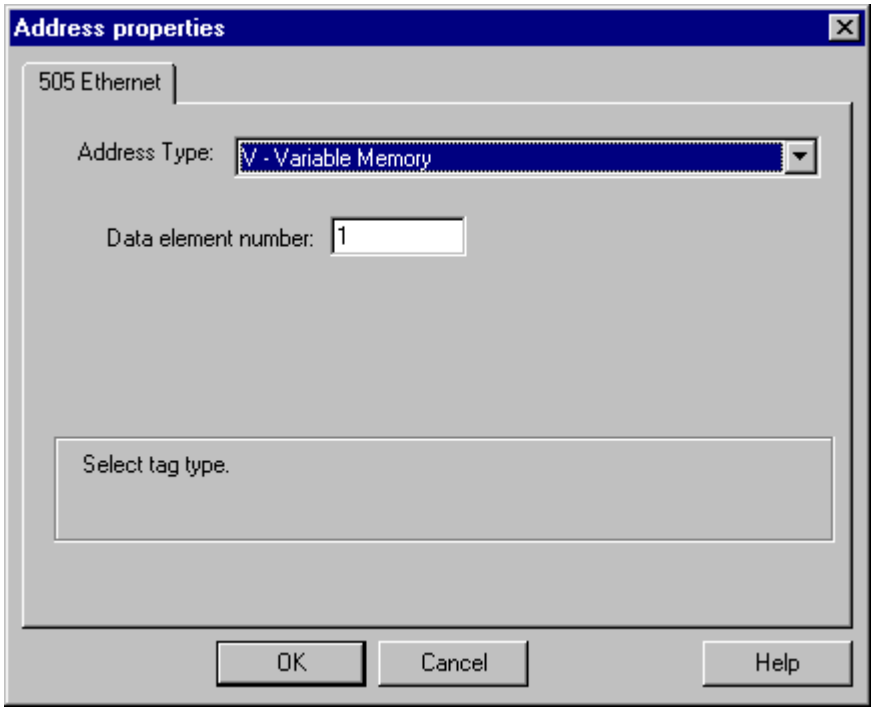
Introduction

The tag address is entered according to the address structure of the SIMATIC TI505.

Procedure

1. Select the tag
2. Select the desired data type in the "Data Type" field.
3. If it is a "Binary" or "8-Bit" tag, the "Bit/Byte tag" option is available in the "Properties" area. Tick the corresponding check box "Access a Bit/Byte", if data should be written to the AS memory.

4. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



5. Select the location of the tag in the automation system in the "Address type" field. Depending the selected address type, more definitions have to be made (e.g. "V-tag memory" for address type in the "Data element" field).
6. In the field "Read-Only Tag", you can specify that the tag cannot be written by WinCC.

Note

Structure tags are not supported.

A description of address types may be found in the SIMATIC TI505 Technical Documentation.

Write access to memory areas in the AS can only performed bit-wise or byte-wise in channel "TI Ethernet Layer 4". When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to define write access to individual bits or bytes. For this purpose, the addressed memory area is read from the AS for every single write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the AS's memory.

Note

Changes that have been made by the AS in a read data area are overwritten when writing back into the data area.

Depending on the type of tag, you can access the memory in the AS bit-wise or byte-wise.

15.3.3.3 How to configure a tag with bit-wise access

Introduction

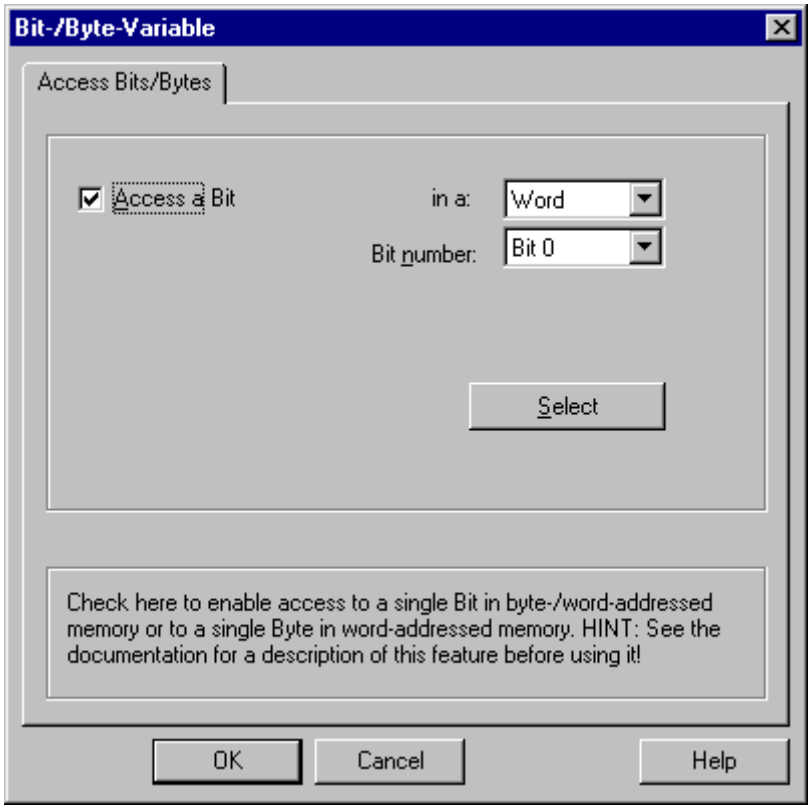
Write access to memory areas in the AS can only performed bit-wise or byte-wise in channel "TI Ethernet Layer 4". When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to define write access to individual bits or bytes. For this purpose, the addressed memory area is read from the AS for every single write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the AS's memory.

Note

Changes that have been made by the AS in a read data area are overwritten when writing back into the data area.

Procedure

1. Select the tag.
2. Set the "Binary tag" data type in the "Data Type" field.
3. Open the "Bit/Byte tag" dialog.
   For this purpose, click in the "Address" field and then on the ... button.
4. Select the "Access to a bit" check box and define the addressing for the bit.



5. Click the "Select" button. The "Address properties" dialog is opened.
6. Select the addressing type of the PLC memory in the "Address type" selection field.
7. From the list below, select the number of the element to be changed.

Note

A description of address types may be found in the SIMATIC TI505 Technical Documentation.

15.3.3.4 How to Configure a Tag with Byte by Byte Access

Introduction

Write access to memory areas in the AS can only performed bit-wise or byte-wise in channel "TI Ethernet Layer 4". When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to define write access to individual bits or bytes. For this purpose, the addressed memory area is read from the AS for every single write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the AS's memory.

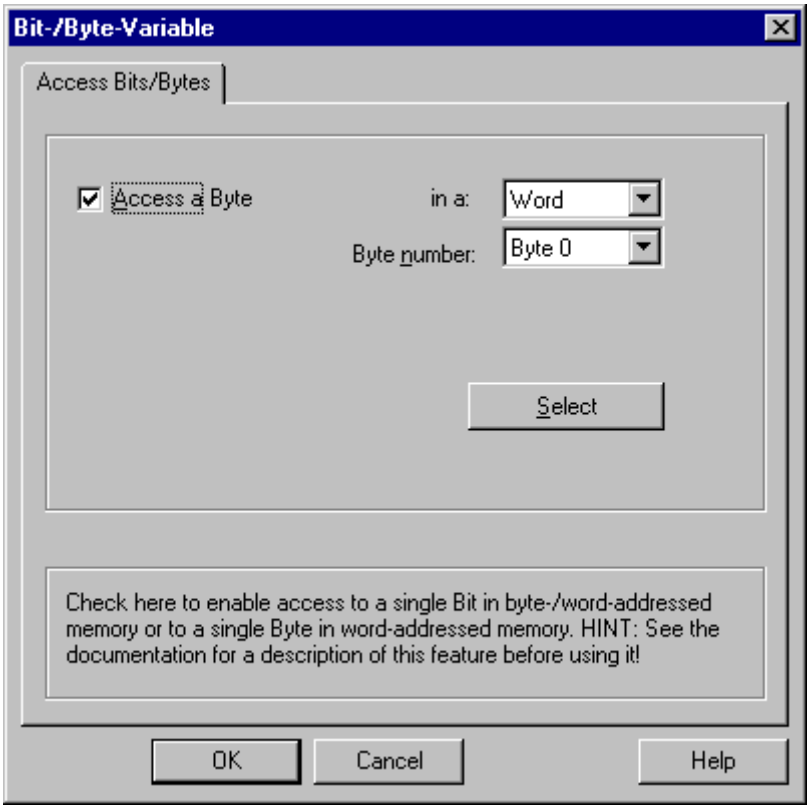
Note

Changes that have been made by the AS in a read data area are overwritten when writing back into the data area.

Procedure

1. Select the tag.
2. In the "Data Type" field, set the data type to "Unsigned 8-bit value" or "Signed 8-bit value".
3. Open the "Bit/Byte tag" dialog.
   For this purpose, click in the "Address" field and then on the ... button.

4. Select the "Access to a byte" check box and define the addressing for the byte.



5. Click the "Select" button. The "Address properties" dialog is opened.
6. Select the addressing type of the PLC memory in the "Address type" selection field.
7. From the list below, select the number of the element to be changed.

Note

A description of address types may be found in the SIMATIC TI505 Technical Documentation.

15.3.4 System parameters

15.3.4.1 System parameters of the channel unit

Introduction

If you require a configuration that deviates from the standard WinCC settings, you can make all the required changes using the "System parameters" dialog of the channel unit.

The following individual points can be changed:

- the device name
- the transport parameter

Device Name

Communication between WinCC and the automation system takes place via logical device names. These names are assigned during the installation of the communication module and are unit-specific. The device name represents the logical device name. The logical device name is initially defined as "/CP_H1_1:/SCP" as default.

Transport Parameter

Specific settings for the channel unit are made in the transport parameters, e.g. PDU size, setup attempts, etc.

Note

The system parameters apply for all CPs in the AS.

See also

How to change the transport parameter (Page 381)

How to Change the Device Name (Page 380)

15.3.4.2 How to Change the Device Name

Introduction

The process connection via Industrial Ethernet is possible with the SIMATIC TI505 automation system.

Communication module CP 1434 TF is used in the automation system. The communication is handled with the ISO transport protocol.

The communication module CP 1613 A2 is used in the WinCC system.

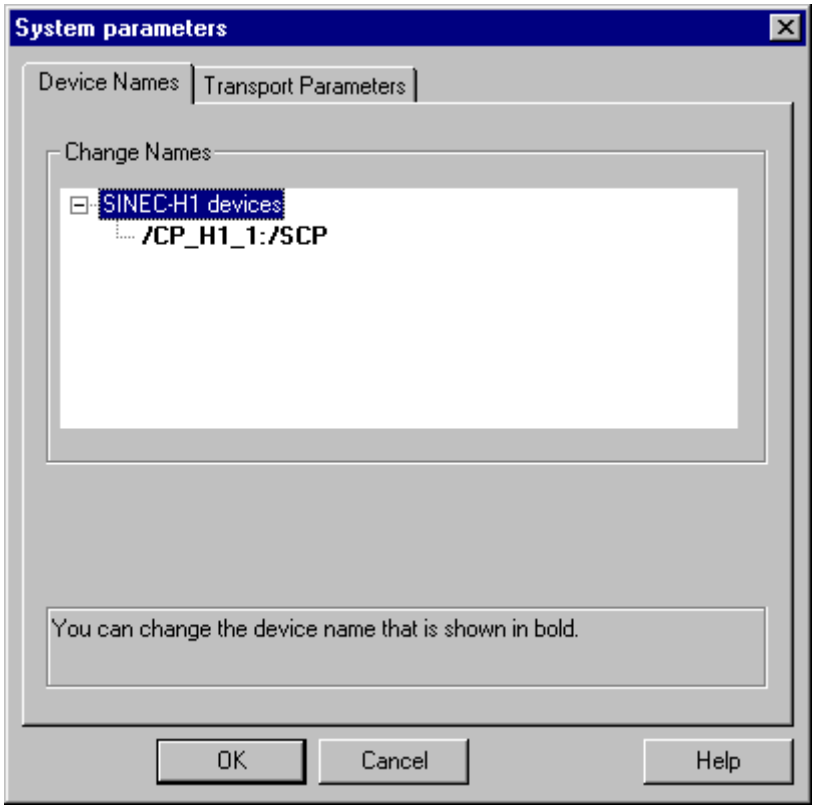
Because communication takes place via the ISO transport protocol, it is not necessary to configure the logical connection in the local database.

Requirements

- The channel "SIMATIC TI Ethernet Layer 4" must be integrated in the project.

Procedure

1. Select the channel unit and open dialog window "System parameters" with the context menu.
2. Select the "Device Names" Tab.



3. Now, you can select the logical device name shown in "bold" print with the mouse and change it with a mouse click in the name field.
   The logical device name is defined as "/CP_H1_1:/SCP" as default during the hardware driver installation.
   Only if you have defined another name there, which is not recommended, will you have to change the device name here as well.

See also

How to change the transport parameter (Page 381)

15.3.4.3 How to change the transport parameter

Introduction

The process connection via Industrial Ethernet is possible with the SIMATIC TI505 automation system.

Communication module CP 1434 TF is used in the automation system. The communication is handled with the ISO transport protocol.

The communication module CP 1613 A2 is used in the WinCC system.

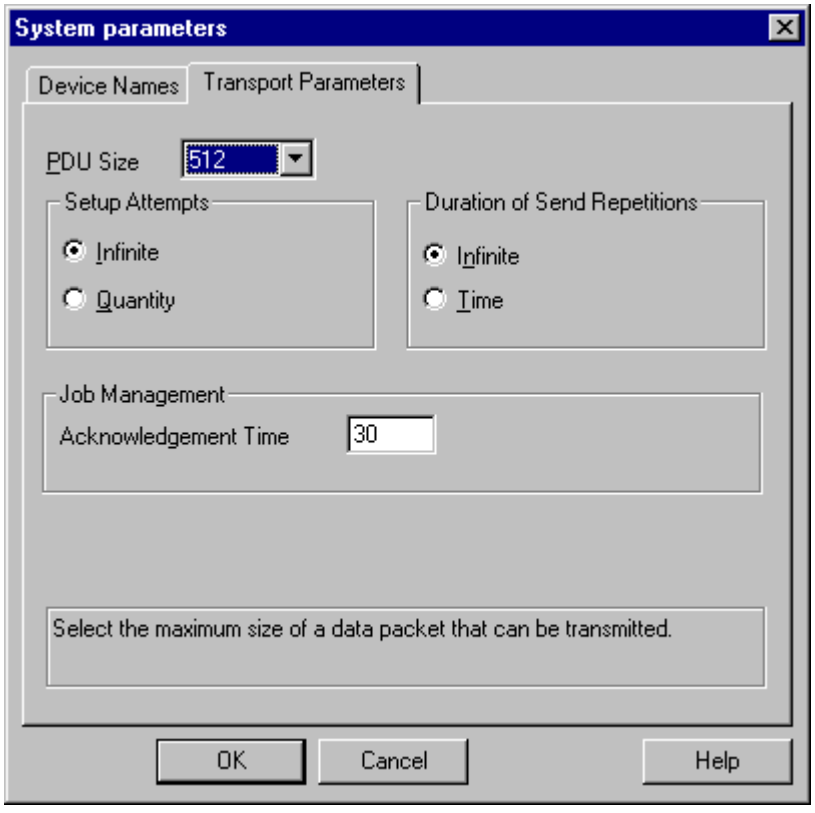
Because communication takes place via the ISO transport protocol, it is not necessary to configure the logical connection in the local database.

Requirements

- The channel "SIMATIC TI Ethernet Layer 4" must be integrated in the project.

Procedure

1. Select the channel unit and open dialog window "System parameters" with the context menu.
2. Select the "Transport Parameters" tab.



3. Set the value for "PDU size" to the value that was configured on the CP 1434 TF.
4. Define how often a connection establishment should be attempted in the "Setup Attempts" filed.
5. Select "Infinite" in the "Duration of Send Repetitions" area.
6. Enter value 30 in the "Ack. Time" field so that you are informed of the tag status after 30 seconds at the most, if the communication partner has not responded within this time (e.g. AS in "Stop" status).

See also

How to Change the Device Name (Page 380)

SIMATIC TI Serial

16

16.1 WinCC channel "SIMATIC TI Serial"

Introduction

The communication driver "SIMATIC TI Serial" is used for establishing a serial link between WinCC station and an SIMATIC TI505 automation device.

This chapter describes

- how to configure the data transfer with the "SIMATIC TI Serial" channel.
- how to configure a connection and a tag.

Channel units

The communication driver has one channel unit for controlling a COM port for the serial connection.

The following capability is available:

- Channel unit "505 Serial Unit #1" for serial communication, either via the TBP protocol or the NITP protocol.

Note

It is possible to run more than one logical connections (with different COM ports) through one channel unit.

See also

Configuring the tags (Page 389)

Configuring the "SIMATIC TI Serial" channel (Page 387)

16.2 Data type of the tags

Introduction

Define the required tags for a logical connection. From the WinCC viewpoint, you can access the following data types:

- Binary tag
- Unsigned 8 bit value (is only supported by VMS addressing)
- Signed 8 bit value (is only supported by VMS addressing)
- Unsigned 16-bit value
- Signed 16-bit value
- Unsigned 32-bit value
- Signed 32-bit value
- Floating-point number 32-bit IEEE 754
- Raw data type

16.3 Configuring the Channel

16.3.1 Configuring the "SIMATIC TI Serial" channel

Introduction

The following steps are required for configuring the channel "SIMATIC TI Serial".

See also

Configuring the tags (Page 389)

How to configure the connection (Page 387)

16.3.2 How to configure the connection

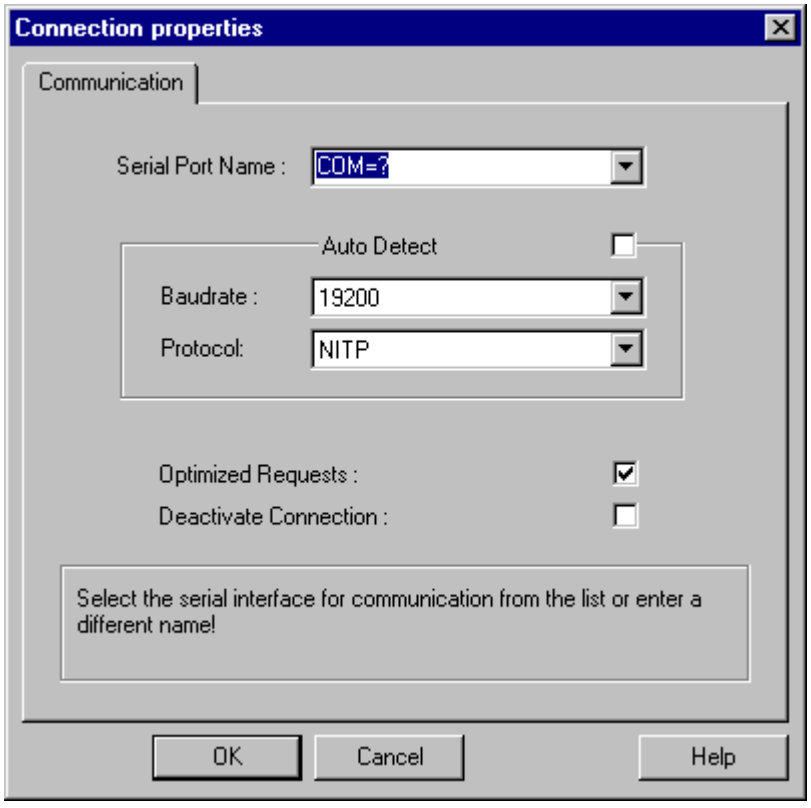
Introduction

The process connection using a serial connection is possible with the SIMATIC TI505 automation system. The serial interface on the CPU module is used in the automation system.

No additional communication module is required in WinCC. Communication can take place by means of the default COM ports available on the system.

Procedure

1. Select the required connection and select "Connection parameters" from the shortcut menu. The "Connection properties" dialog opens.



2. Select the communications interface (e.g. COM1, COM2 or a configured port) for the serial link in the "Serial port" field.
3. Select the field "Detect automatically" when the data transfer speed and the protocol used by the PLC are required to be detected automatically by the channel unit.
4. Set the data transfer rate and the protocol being used in the fields "Baud rate" and "Protocol".
5. By selecting the field "Optimized Requests," you can optimize data transfer to transfer several tags with one request.
6. If you select the "Deactivate connection" field, the logical connection is deactivated. This is often a good idea during commissioning to temporarily deactivate the connection.

See also

Configuring the tags (Page 389)

Data type of the tags (Page 386)

16.3.3 Configuring the tags

16.3.3.1 Configuring the tags

Introduction

For a connection between WinCC and the PLC via channel "SIMATIC TI Serial", tags of different data types can be created within WinCC. This is described in the following section.

See also

How to Configure a Tag with Byte by Byte Access (Page 392)

How to configure a tag with bit-wise access (Page 391)

How to configure the address of a tag (Page 389)

16.3.3.2 How to configure the address of a tag

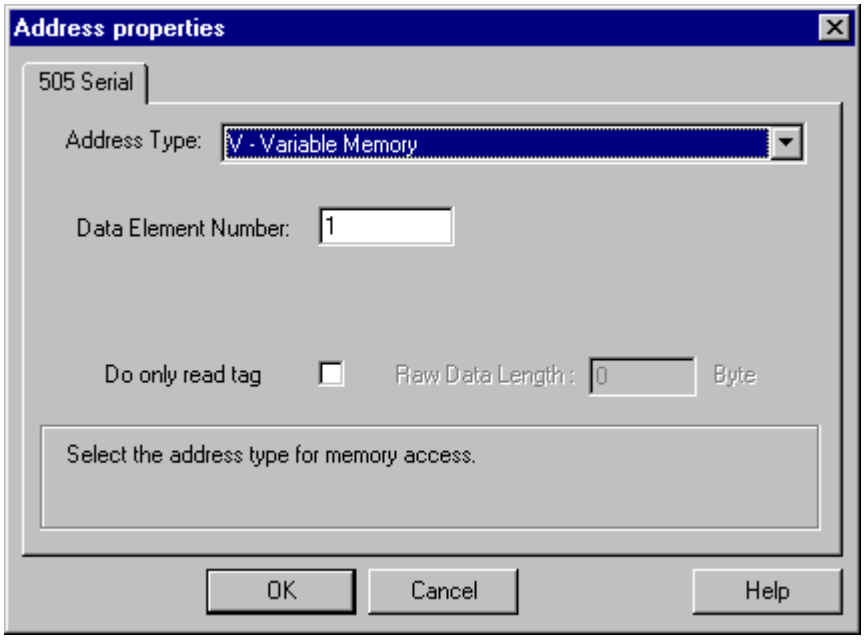
Introduction

The tag address is entered according to the address structure of the SIMATIC TI505.

Procedure

1. Select the tag.
2. Set the required data type in the "Data Type" field (e.g. signed 16-bit value).
3. If it is a "Binary" or "8-Bit" tag, the "Bit/Byte tag" option is available in the "Properties" area. Tick the corresponding check box "Access a Bit/Byte", if data should be written to the AS memory.

4. Open the "Address properties" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



5. Select the location of the tag in the automation system in the "Address type" field. Depending the selected address type, more definitions have to be made (e.g. "V-tag memory" for address type in the "Data Element Number" field).
6. In the field "Read-Only Tag", you can specify that the tag cannot be written by WinCC.

Note

Structure tags are supported in address areas V, K, X, Y and C.

A description of address types may be found in the SIMATIC TI505 Technical Documentation.

Write access to memory areas in the AS can only performed bit-wise or word-wise in channel "TI Serial". When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to define write access to individual bits or bytes. For this purpose, the addressed memory area is read from the AS for every single write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the AS's memory.

Depending on the type of tag, you can access the memory in the AS bit-wise or byte-wise.

16.3.3.3 How to configure a tag with bit-wise access

Introduction

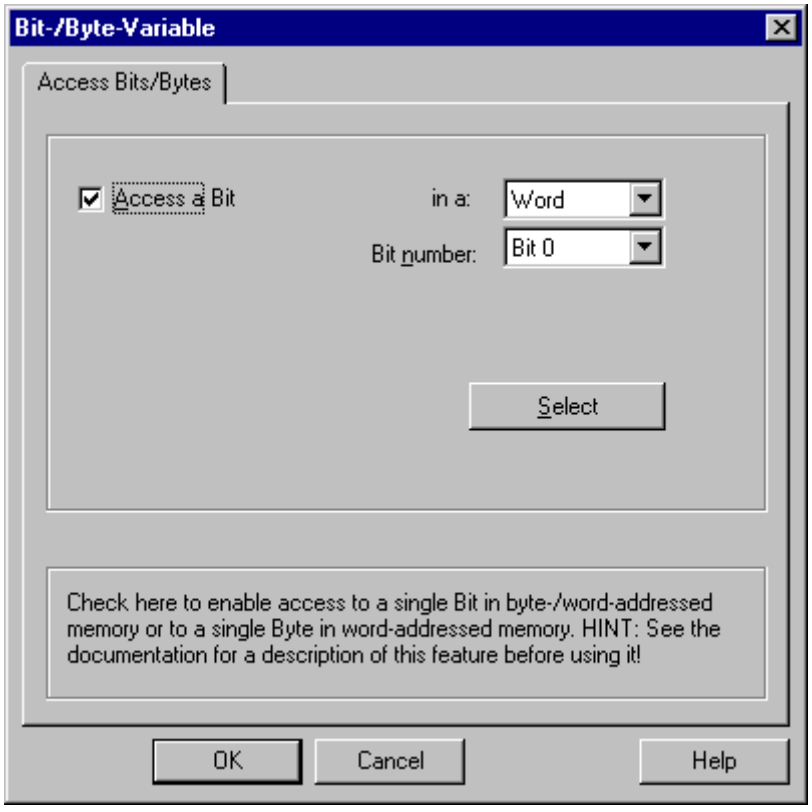
Write access to memory areas in the AS can only performed bit-wise or byte-wise in channel "SIMATIC TI Serial". When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to define write access to individual bits or bytes. For this purpose, the addressed memory area is read from the AS for every single write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the AS's memory.

Note

Changes that have been made by the AS in a read data area are overwritten when writing back into the data area.

Procedure

1. Select the tag.
2. Set the "Binary tag" data type in the "Data Type" field.
3. Open the "Bit/Byte tag" dialog.
   For this purpose, click in the "Address" field and then on the ... button.
4. Select the "Access to a bit" check box and define the addressing for the bit.



5. Click the "Select" button. The "Address properties" dialog is opened.
6. Select the addressing type of the PLC memory in the selection field.
7. Select the number of bit to be changed in the selection field.

Note

With the S5, flags, inputs and outputs can be addressed byte by byte; data blocks (DB, DX) are addressed word by word.

16.3.3.4 How to Configure a Tag with Byte by Byte Access

Introduction

Write access to memory areas in the AS can only performed bit-wise or byte-wise in channel "SIMATIC TI Serial". When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to define write access to individual bits or bytes. For this purpose, the addressed memory area is read from the AS for every single write request and the corresponding bits and/or bytes are modified. Afterwards, the data is written back to the AS's memory.

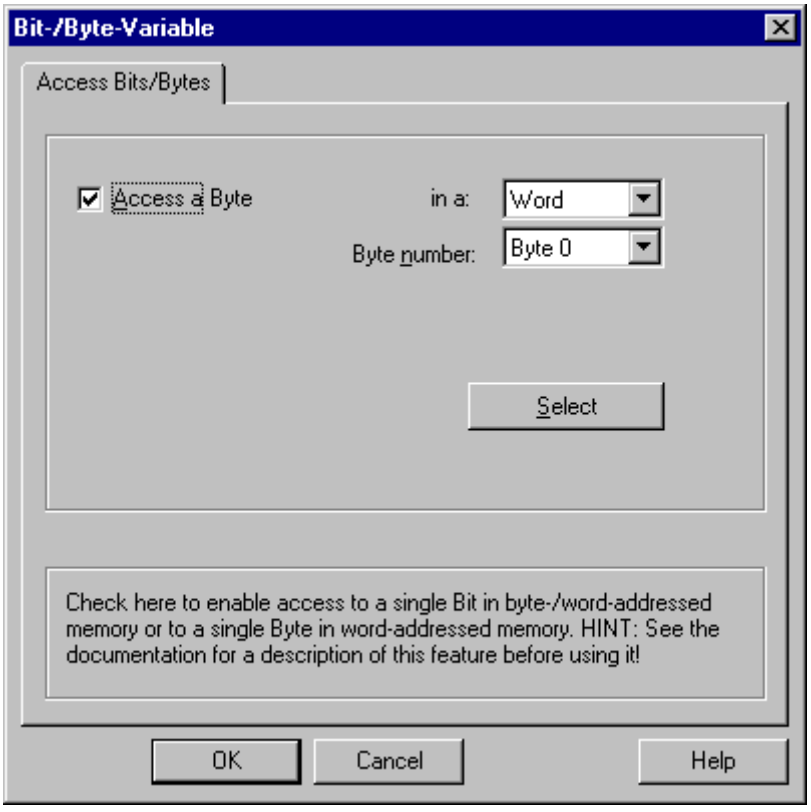
Note

Changes that have been made by the AS in a read data area are overwritten when writing back into the data area.

Procedure

1. Select the tag.
2. In the "Data Type" field, set the data type to "Unsigned 8-bit value" or "Signed 8-bit value".
3. Open the "Bit/Byte tag" dialog.
   For this purpose, click in the "Address" field and then on the ... button.

4. Select the "Access to a byte" check box and define the addressing for the byte.



5. Click the "Select" button. The "Address properties" dialog is opened.
6. Select the addressing type of the PLC memory in the selection field
7. Select the number of byte to be changed in the selection field.

SIMOTION 17

17.1 WinCC channel "SIMOTION"

Introduction

The "SIMOTION" channel connects a WinCC station with a SIMOTION automation system. The connection is established via Industrial Ethernet using the TCP/IP protocol.

SIMOTION is a system platform for automation and drive solutions with an emphasis on motion control applications and technology tasks.

The SIMOTION modular system consists of the SIMOTION SCOUT Engineering System and a common runtime system for various hardware platforms.

Export the project from SIMOTION SCOUT to access the data of a SIMOTION SCOUT project.. Then create a WinCC project from the exported files with the Simotion Mapper.

You can configure certain changes to the configuration of the SIMOTION automation system, e.g. change an IP address, later in WinCC. Further changes must be configured in SIMOTION SCOUT, exported again and transferred with the Simotion Mapper.

Note

You must be familiar with the SIMOTION SCOUT Engineering System and the configuration of WinCC to configure the "SIMOTION" channel in WinCC.

17.2 Overview of the supported data types

Introduction

The data type and the format adaptation to the data format in the automation system are determined in the configuration of a tag.

The table shows the data types supported by the channel and the application of format adaptations.

Supported data types

Data Types	Type conversion
Binary tag	No
Signed 8-bit value	Yes
Unsigned 8-bit value	Yes
Signed 16-bit value	Yes
Unsigned 16-bit value	Yes
Signed 32-bit value	Yes
Unsigned 32-bit value	Yes
Floating-point number 32-bit IEEE 754	Yes
Text tag, 8-bit font	No
Raw data type	No

17.3 Configuring the channel

17.3.1 Configuration of the "SIMOTION" channel

Introduction

This chapter describes how to configure the "SIMOTION" channel.

Note

You must be familiar with the SIMOTION SCOUT Engineering System and the configuration of WinCC to configure the "SIMOTION" channel.

Proceed as follows to configure the "SIMOTION" channel:

1. Export the SIMOTION SCOUT project from SIMOTION SCOUT.
2. Create WinCC project with the Simotion Mapper.
3. Open WinCC project.
4. Configure system parameters.

Further information about the diagnosis of the channel, the connection and the tags can be found in the "Diagnosis 'SIMOTION' channel" chapter.

See also

How to export a SIMOTION SCOUT project (Page 397)
How to create a WinCC project with Simotion Mapper (Page 398)
How to change the connection parameters (Page 401)
How to change the tag address (Page 402)
System Parameters of the Channel Unit (Page 403)
Diagnosis "SIMOTION" channel (Page 407)

17.3.2 How to export a SIMOTION SCOUT project

Introduction

This section describes how to export tags and message definitions from SIMOTION SCOUT.

Requirements:

- You are familiar with the SIMOTION SCOUT Engineering System.
- You have access to the SIMOTION SCOUT project to be exported.

Procedure

1. Open the SIMOTION SCOUT project to be exported in SIMOTION SCOUT.
2. Select "Export OPC Data" under "Tools."
3. Select version "SIMATIC NET V6.4", the desired scope and at least the "OPC-Alarm/Event" option for the export.

   Note

   Simotion Mapper does not process other export versions than "SIMATIC NET V6.4".

4. Select the destination directory.
5. Select the communication interface.
   The project is exported.
6. Enter the routing information if you are using routing.

The SIMOTION SCOUT project is exported. The "OPC_Data.sti" and "OPC_AE.xml" files are saved in the destination directory.

17.3.3 How to create a WinCC project with Simotion Mapper

Introduction

This section describes how to create a WinCC project from the exported SIMOTION SCOUT project with the Simotion Mapper.

Note

If a WinCC project was already created for an older version of the SIMOTION SCOUT project, only the SIMOTION parameters are changed in a transmission. All other configuration settings in the WinCC project (such as archiving) remain the same.

Requirements:

- You have access to the export files "OPC_Data.sti" and "OPC_AE.xml" of the SIMOTION SCOUT project.
- You have access rights to the WinCC installation directory.

Note

In the SIMOTION SCOUT programming environment, the tags to be exported can be filtered using the watch tables. Use the watch tables to keep the number of tags in the WinCC project low. More information on the watch tables can be found in the SIMOTION SCOUT online help.

Procedure

1. Launch the "SimotionMapper.exe" program in the WinCC installation directory.
2. Click "Open". Navigate to the directory with the files "OPC_Data.sti" and "OPC_AE.xml". The data is read and displayed in Simotion Mapper.
3. In the Simotion Mapper Explorer, select the groups and tags you need in your WinCC project.
4. Select "Create new WinCC project".
5. If you want to change the "WinCC connection name", click on the name displayed and enter the new name.
6. Specify the "First TA message number" for the technological alarm. The value must be selected in such a way that it does not lead to collisions with messages of other communication channels. The default value is 100.
   The Simotion Mapper creates one message and a total of six template messages for each Simotion connection starting from the number specified.
7. Click "Start mapping". Select the destination folder for the WinCC project.
   The WinCC project is created. The progress bar indicates the progress of the procedure.
8. Close Simotion Mapper.

The WinCC project is created and can now be opened and edited in WinCC.

Note

You may have to set the system parameters of the "SIMOTION" channel in WinCC to use the created WinCC project.

17.3.4 How to change a WinCC project with Simotion Mapper

Introduction

This section describes how to add an exported SIMOTION SCOUT project to an existing WinCC project with the Simotion Mapper. In this way, you can use the same Simotion project several times in one WinCC project, for example.

Note

If a WinCC project was already created for an older version of the SIMOTION SCOUT project, only the SIMOTION parameters are changed in a transmission. All other configuration settings in the WinCC project (such as archiving) remain the same.

Requirements:

- You have access to the export files "OPC_Data.sti" and "OPC_AE.xml" of the SIMOTION SCOUT project.
- You have access rights to the WinCC installation directory.

Note

In the SIMOTION SCOUT programming environment, the tags to be exported can be filtered using the watch tables. Use the watch tables to limit the number of tags in the WinCC project. More information on the watch tables can be found in the SIMOTION SCOUT online help.

Procedure

1. Open the WinCC project to be edited.
2. Launch the "SimotionMapper.exe" program in the WinCC installation directory.
3. Click "Open". Navigate to the directory with the files "OPC_Data.sti" and "OPC_AE.xml". The data is read and displayed in Simotion Mapper.
4. In the Simotion Mapper Explorer, select the groups and tags you need in your WinCC project.
5. Select "Add to the open project".
6. If you want to re-add a group or tag that has already been created, you must change the "WinCC connection name" by clicking on the name displayed.
7. If you do not want to transfer any messages, groups or tags for a connection, unselect "WinCC connection name".
8. Specify whether tags should be overwritten.
9. Specify the "First TA message number" for the technological alarm. The value must be selected in such a way that it does not lead to collisions with messages of other communication channels. The default value is 100.
   The Simotion Mapper creates one message and a total of six template messages for each Simotion connection starting from the number specified.

   Note

   Please do change any "First TA message number" that has already been mapped. If you do, you may experience unpredictable message behavior.

10. Click "Start mapping". Select the destination folder for the WinCC project.
    The SIMOTION SCOUT project is added to the open WinCC project. The progress bar indicates the progress of the procedure.
11. Close Simotion Mapper.

The WinCC project was expanded by the SIMOTION SCOUT project and saved with your settings.

17.3.5 How to change the connection parameters

Introduction

In this section, you will learn how to change the connection parameters of the SIMOTION network address.

Note

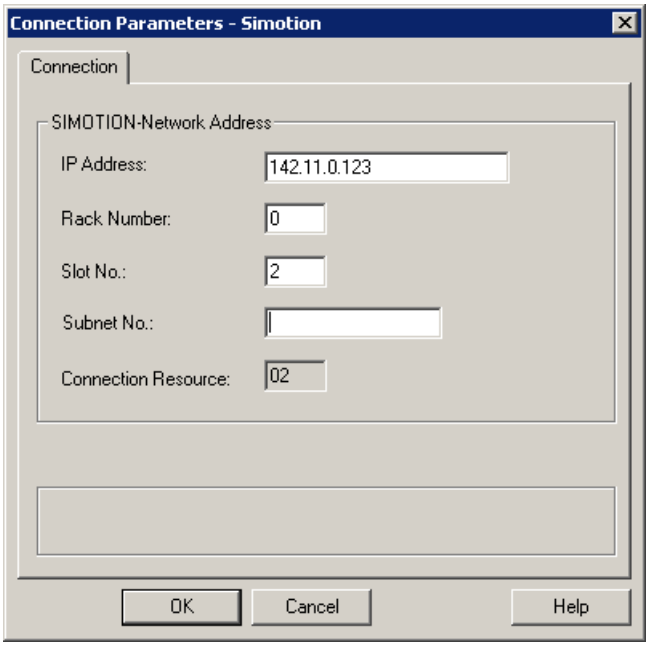
Change only the connection parameters listed here. Do not create new connection for the "SIMOTION" channel. Incorrectly set connections may result in control errors in the PLC. Configure new connections according to the description in the section "Configuration of the "SIMOTION" channel (Page 397)".

Requirements

- The SIMOTION communication driver is integrated in the WinCC project.
- A connection must be created in the "SIMOTION" channel unit.

Procedure

1. Open the directory structure for the "SIMOTION" communication driver in the "Tag Management" editor.
2. Select the entry "Connection parameters" from the shortcut menu of a connection of the "Simotion" channel unit.
   The "Connection parameters - SIMOTION" dialog opens.



3. Change the connection parameters for the SIMOTION network address in the respective fields.
4. Close each open dialog box by clicking "OK."

17.3.6 How to change the tag address

Introduction

This section describes how to change a tag address in the "SIMOTION" channel.

Note

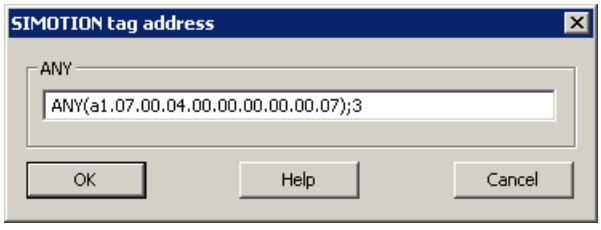
You must have very good knowledge of the use of ANY pointers to change the tag address in the "SIMOTION" channel. No communication connection may be established if the tag address is entered incorrectly.

Requirements

- The "SIMOTION" channel in integrated into the WinCC project.
- A connection with tags has been created in the "SIMOTION" channel unit.

Procedure

1. Open the "SIMOTION tag address" dialog.
   For this purpose, click in the "Address" field and then on the ... button.



2. Change the tag address.

17.3.7 System parameter configuration

17.3.7.1 System Parameters of the Channel Unit

Introduction

If you require a different configuration than the WinCC default settings, make these settings in the "System Parameters" dialog box.

You can change the following system parameters:

- Logical device name
- The channel uses cyclic read services in the AS

Logical device name

WinCC and the PLC communicate by means of logical device names that are assigned when the communications processor is installed in the PLC.

The channel uses cyclic read services in the AS

The PLC cyclic read services group the tags that are to be read cyclically into individual requests and transfer these to the PLC. The PLC sends the requested data the first time on receipt of the request and then again each time the cycle time elapses.

When cyclic read services are enabled, you can use the change-driven transfer function. If the PLC supports change-driven transfer, the data are then transferred only when values are changed.

See also

How to Configure the System Parameters (Page 403)

How to Change the Logical Device Name (Page 405)

17.3.7.2 How to Configure the System Parameters

Introduction

This section shows how to configure the system parameters of the "SIMOTION" channel.

The "System Parameters" dialog comprises two tabs:

- "SIMOTION" tab
- "Unit" tab

Note

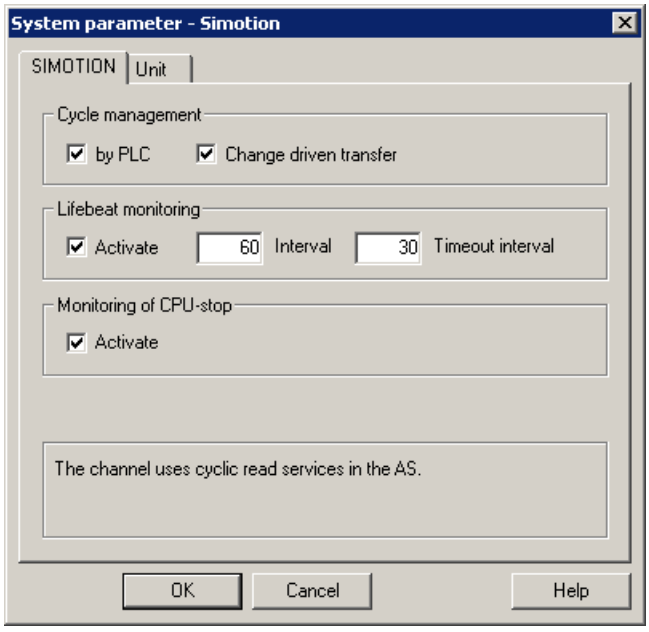
When the project is copied to another computer, the settings in the "Unit" tab are retained. The settings on the "SIMOTION" tab are deleted on the other hand.

Requirements

- The "SIMOTION" channel in integrated into the WinCC project.

Procedure

1. Select the "SIMOTION" channel in the variable management. Open the "System Parameters" dialog box in the shortcut menu of the "Simotion" channel unit.
2. Select the "SIMOTION" tab.

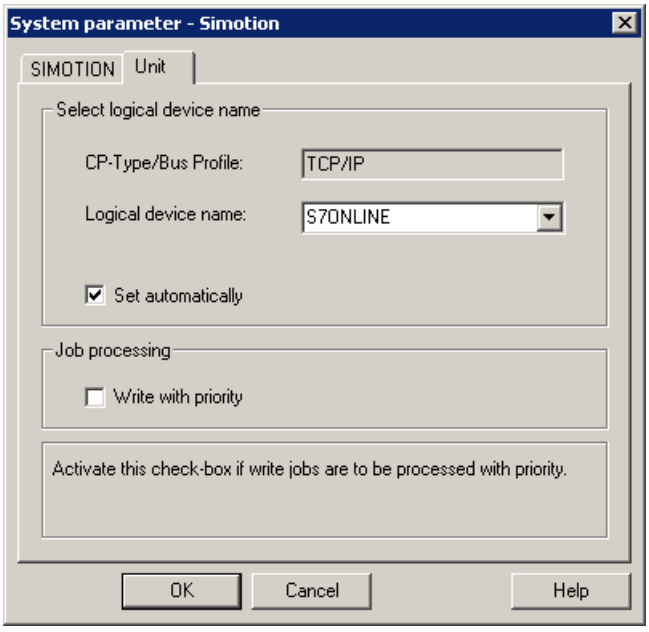


3. To enable cyclic reading of tags and change-driven transfer, select "by PLC" and "Change-driven transfer."

Note

The "cycle management", "lifebeat monitoring" and "stop monitoring" functions are not supported by the integrated SINAMICS servo control. The "SIMOTION" channel therefore ignores corresponding settings for connections to SINAMICS servo controls. The channel determines whether the AS supports the respective function when establishing the connection.

4. Select the "Lifebeat monitoring" function if required.
   Determine the interval in seconds for sending lifebeat monitoring messages.
   Determine the monitoring time in seconds for monitoring the response to a lifebeat monitoring message.
5. Enable "CPU Stop Monitoring" if you want WinCC to signal a fault in the communication when the SIMOTION CPU is in the stopped state.
6. Select the "Unit" tab.
   "S7ONLINE" is displayed as a default for "logical device name". You must change the device name if a different name was selected when installing the used communications processor.



7. To set the device name automatically at the start of runtime, select "Set automatically."
8. To give write jobs higher priority than read jobs during processing, select "Write with priority."
9. Close the dialog by clicking "OK."

Note

Setting changes only take effect after WinCC is restarted.

17.3.7.3 How to Change the Logical Device Name

Introduction

WinCC and SIMOTION communicate through logical device names. These logical device names are assigned when the communications processor is installed.

Requirements

- The "SIMOTION" channel in integrated into the project.
- A connection has been created in the "SIMOTION" channel unit.

Procedure

1. Select the SIMOTION channel in Tag Management.
2. Open the "System Parameters" dialog box in the shortcut menu.
3. Select the "Unit" tab.
4. Enter the device name in the "Logical device name" field. You can select an entry from the list or enter a new name.
   The device names are determined by the "Set PG/PC interface" tool. You call the tool in the system control. Only the currently set device name is displayed if it is not installed.
   If you specify a different logical device name, a message is displayed.
   Only enter a name if the communications processor being used on the target station is not installed on the configuring system.
5. Close the dialog by clicking "OK."

Note

Setting changes only take effect after WinCC is restarted.

17.4 Diagnosis "SIMOTION" channel

17.4.1 Diagnosis possibilities of the "SIMOTION" channel

The following possibilities exist for the diagnosis and error detection of the "SIMOTION" channel and its tags.

Checking the Communication Processor Configuration

After checking the access point, the communication processor can be tested with the "Set PG/PC interface" application. The communication processor can be checked under SIMATIC NET in the same way.

Checking the Configuration of the Connection and Tags

There may be errors in the configuration of the system and connection parameters. An incorrect tag addressing may also be responsible for wrong tag values.

Diagnosis of the Channel with "Channel Diagnosis"

You can query the status of the channel and the connection in runtime with "Channel Diagnosis". Errors are displayed by "Error Codes".

Diagnosis of the Channel Tags

You can query the current value, the current quality code and the last change time of the tag in runtime in the tag management.

17.4.2 Description of Log File Entries

Introduction

The channel enters important status changes and errors in the logfile. The entries support the analysis of communication faults.

Every entry in the file contains a date and time stamp with the following flag names and description.

Example of a logbook entry:

2009-10-28 12:10:11,467 INFO Log starting ...

2009-10-28 12:10:11,483 INFO | LogFileName :
D:\SIEMENS\WINCC\Diagnosis\Simotion_01.LOG

2009-10-28 12:10:11,483 INFO | LogFileCount : 3

2009-10-28 12:10:11,483 INFO | LogFileSize : 1400000

2009-10-28 12:10:11,483 INFO | TraceFlags : fa000000

2009-10-28 12:10:11,498 INFO SIMOTION channel DLL started!

2009-10-28 12:10:11,498 INFO SIMOTION channel with own cycle creation!

2009-10-28 12:10:11,967 INFO Connection "D445": StartRegisterEvVariable for dwVariableCount = 89

2009-10-28 12:10:11,967 INFO Connection "D445": RegisterEvVariable for Variable "@D445@CheckSum"!

...

2009-10-28 12:10:11,983 INFO Connection "D445": EndRegisterEvVariable

2009-10-28 12:10:12,436 INFO S7DOS release: @(#)TIS-Block Library DLL Version R8.0.0.0-REL-BASIS

2009-10-28 12:10:12,436 INFO S7DOS version: V8.0 / 0

2009-10-28 12:10:12,436 INFO SIMOTION version: V6.0 / Sep 15 2009 / 08:06:43

2009-10-28 12:10:12,436 INFO SIMOTION channel unit "Simotion" activated!

2009-10-28 12:10:12,451 ERROR Cannot connect to "SINAMICS_Integrated": Errorcode 0xFFDF 42C2!

2009-10-28 12:10:12,451 ERROR Cannot connect to "D445": Errorcode 0xFFDF 42C2!

Description of the Most Important Entries for the "INFO" Flag

Message text	Meaning
LogFileName : C:\ Siemens\ WinCC\ Diagnose\ "channel_name".LOG	Name of the log file with path
LogFileCount : "n"	Number of log files of the channel
LogFileSize : "x"	Size of the individual log files in bytes
TraceFlags : c4000000	Displays the flags used by the Trace function as a hexadecimal number
SIMOTION channel DLL started!	Start message
SIMOTION channel DLL terminated!	End message

Description of the Most Important Entries for the "ERROR" Flag

Message text	Meaning
Cannot connect to <connectionname>: Errorcode 0x0000 7<xxx>!	Communication error Communication to SIMOTION could not be established immediately after activating WinCC. <connectionname> = Name of connection <xxx> 1...fff The channel has received all other error codes as the result of a function call of S7DOS, a lower layer or from the AS.
Connectionerror <nnn> <connectionname>: Errorcode 0x0000 7xxx!	Communication error Communication to SIMOTION could not be established after activating WinCC. The connection was broken. <nnn> = Number of connection terminations for this connection <connectionname> = Name of connection <xxx> 1...fff The channel has received all other error codes as the result of a function call of S7DOS, a lower layer or from the AS.
Channel API error: errorstring	Channel API error The channel passed the error string 'errorstring' to WinCC Explorer. The error string is displayed in an information box depending on the error relevance. See API error texts for a description of the error strings.
Max. count of API errors reached - API logbook deactivated	Channel API error Depending on the error and function, errors can occur cyclically on the API. To avoid filling the logbook file with these error messages, a maximum of 32 messages are output for an API error.
Cannot write storage data! Cannot read storage data / use default data Storage data illegal or destroyed / use default data! No storage data / use default data!	General Channel Error Messages
Devicename in unit "unitname" changed from "old devicename" to "new devicename"	Initialization message

Note

The error codes of the "SIMOTION" channel correspond to those of the "SIMATIC S7 Protocol Suite" channel. You will find the description of the error code in the "Error codes for connection fault" chapter in the help for the "SIMATIC S7 Protocol Suite" channel.

In addition, SIMOTION reports the error code 0x000 7301 if the consistency check failed. The reason for the error message is that the data exported from SIMOTION SCOUT for the AS entered checksum do not match the checksum in the connected device.

See also

Error codes during connection disturbances (Page 328)

SinumerikNC

18

Introduction

The "SinumerikNC" channel is used for communication between a WinCC station and CNCs of the type "SINUMERIK 840D" and the included "SINUMERIK S7-300".

Communication is via the Ethernet IP protocol.

Communication via MPI (Multi-Point Interface) is supported. Simultaneous use of both protocols is not possible.

Channel units

The "SinumerikNC" channel features the channel units "SINUMERIK 840D sl" and "S7-300-PLC".

Information on the configuration of a channel unit for S7-300 can be found in the documentation for the "SIMATIC S7 Protocol Suite" communication channel:

- "MPI" channel unit:
  - "MPI" channel unit (Page 253)
  - How to configure a "MPI" channel unit connection (Page 253)
- "TCP/IP" channel unit:
  - How to configure a "TCP/IP" channel unit connection (Page 265)
  - "TCP/IP" channel unit (Page 264)

Configuration of the "SINUMERIK 840D sl"

You can find comprehensive information on configuration of the CNC controller of the type "SINUMERIK 840D sl" on the Internet in the "SiePortal Knowledge Base":

- SiePortal: Getting Started SINUMERIK 840D sl STEP 7 Toolbox (https://support.industry.siemens.com/cs/ww/en/view/109801199)
- SiePortal: Configuration Manual SINUMERIK 840D sl STEP7 Toolbox (https://support.industry.siemens.com/cs/ww/en/view/109801245)

See also

SiePortal: WinCC V8.0 SINUMERIK (Entry ID 109816693) (https://support.industry.siemens.com/cs/ww/en/view/109816693)

18.1 Supported Data Types

Introduction

Define the required tags for a logical connection with a connected control. These data types of the SINUMERIK NC are supported by the "SinumerikNC" channel:

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

Global User Data (GUD) tags of the SINUMERIK NC are supported and can be reached in Runtime.

18.2 Configuring the channel

Introduction

WinCC needs a logical connection for communication of WinCC with the automation system (AS).

This section describes the communication with the "Sinumerik 840D sl" channel unit. All connection-specific parameters are defined during the setup.

TCP/IP protocol

If using TCP/IP, you must specify the IP address of the AS for the logic link.

The IP address consists of four numerical values, separated by dots.

The numerical values must be within the range of 0-255.

Note

Timeout behavior

Interrupted connections are not detected immediately when using the TCP/IP protocol.

The check-back message can take up to a minute.

18.2.1 How to configure a connection to a SINUMERIK 840D sl

Introduction

This section shows you how to configure a connection for the "Sinumerik 840D sl" channel unit.

Channel unit "S7-300-PLC"

Information on the configuration of a channel unit for S7-300 can be found in the documentation for the "SIMATIC S7 Protocol Suite" communication channel:

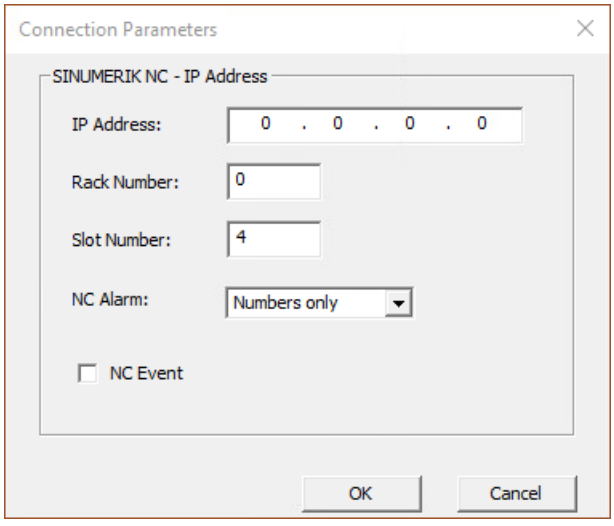
- "MPI" channel unit:
  - "MPI" channel unit (Page 253)
  - How to configure a "MPI" channel unit connection (Page 253)
- "TCP/IP" channel unit:
  - "TCP/IP" channel unit (Page 264)
  - How to configure a "TCP/IP" channel unit connection (Page 265)

Requirements

- The communication driver for the "SinumerikNC" is installed and integrated into the project.

Procedure

1. In the navigation area of the Tag Management, select the channel unit "SINUMERIK 840D sl" in the tree of the "SinumerikNC" communication driver.
2. Select the entry "New Connection" in the shortcut menu of the channel unit.
3. Enter the name of the connection.
4. Select the entry "Connection parameters" from the shortcut menu of the connection.
   The "Connection Parameters" dialog opens.



5. Enter the IP address of the NCU (Numerical Control Unit).
6. Enter the number of the rack on which the NCU is installed.
7. Enter the number of the slot.
   Default: Slot 4.
8. If alarms of the SINUMERIK 840 are to be displayed, select the entry "NC alarm" under "Text and Symbols".
9. Close the dialog with "OK".

See also

How to configure SINUMERIK S7-300 alarms (Page 420)

18.2.2 Configuring the tags

Introduction

For a connection between WinCC and the automation system (AS) via "SinumerikNC" channel, tags of different data types can be created in WinCC.

The following sections describe how to configure the tags.

18.2.2.1 How to configure a tag

Introduction

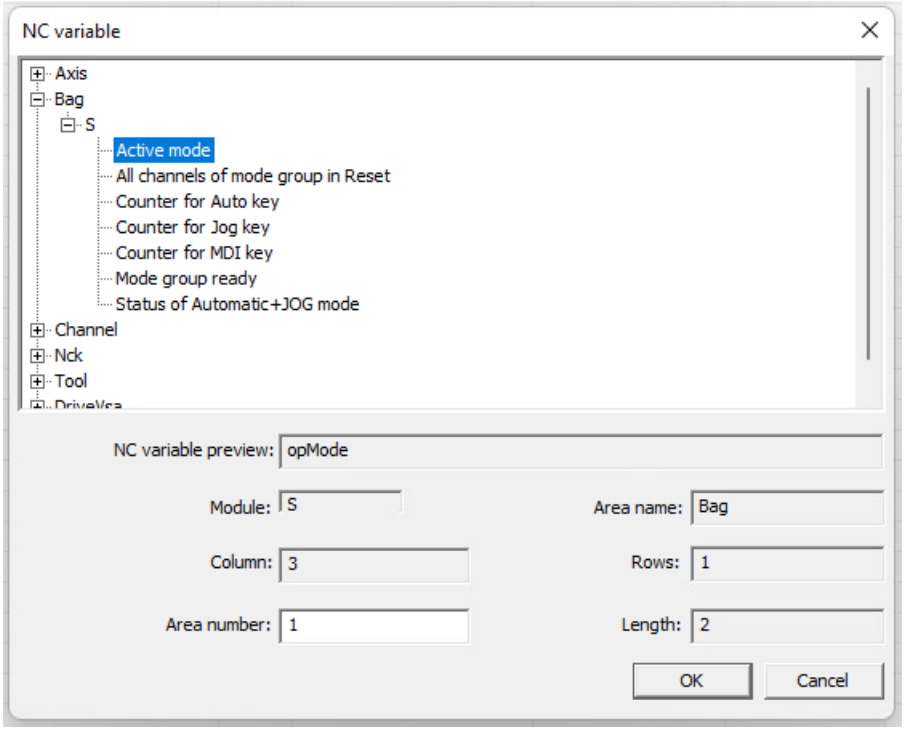
This section shows you how to configure a tag access for the address range in the automation system (AS).

Requirements

- The "SinumerikNC" channel is integrated into the project.
- A connection is created.

Procedure

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. Select the desired data type in the "Data type" field.
5. In the "Format adaptation" field, select one of the specified adaptations.
6. Click in the "Address" field and then the ... button.
   The "NC variable" dialog is displayed.



7. Select the corresponding NC variable in the tree view.
   You can find information on NC variables here: SINUMERIK 828D NC variables (https://support.industry.siemens.com/cs/ww/en/view/109823592/163245562251).

   Note

   After the address has been selected, the data type is automatically adapted.

   Change the column and the row if required.
8. Specify the other properties of the tag, for example, limit values in the "Properties - Tag" area.

18.2.2.2 Configuring GUD tags

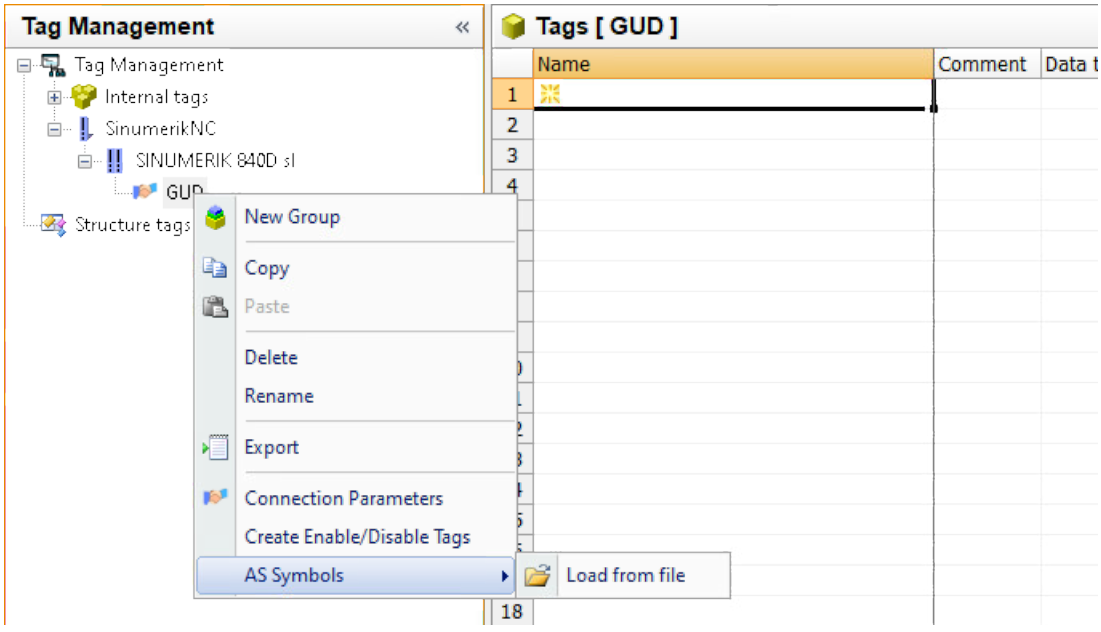
GUD (Global User Data) tags of a SINUMERIK are exported via an export file from the SINUMERIK and imported into WinCC.

Requirement

- A file with exported GUD tags is available.
- The file has the file extension .def.
- The same file is available on the NC device.

Procedure

1. From the context menu of the SINUMERIK connection, select "AS Symbols > Load from file".



   The dialog for selecting a file is displayed.
2. Select the DEF file with the exported GUD tags.

3. Click "Download".
   The available controller tags are loaded and the "Symbols" view opens.
   The loaded tags are displayed in the table area in the "AS Symbols" tab.
4. To choose the required tags, select the corresponding check boxes in the "Access" column.
5. Click to return to the Tag Management.
   The selected GUD tags are available in the Tag Management.

18.2.2.3 Configuring PLC tags

PLC tags of a SINUMERIK are exported via an export file from the SINUMERIK and imported into WinCC. Since a SINUMERIK uses two CPUs (NCU and PLC), a second connection is required.

To export data records from a TIA Portal project, use the "SIEMENS SIMATIC SCADA Export" tool.

The tool for the various TIA Portal versions is available for download in Industry Online Support:

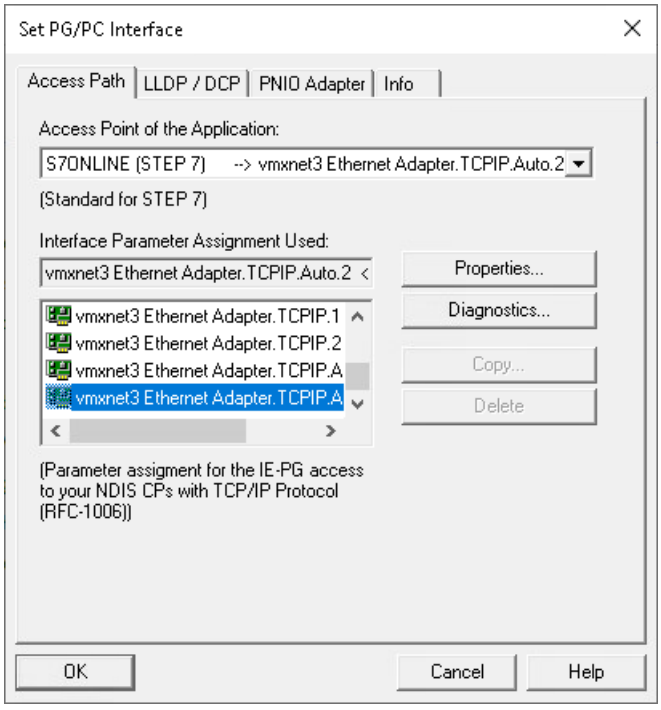
- Industry Online Support: "SIMATIC SCADA Export" documentation (https://support.industry.siemens.com/cs/ww/en/view/101908495)
- Industry Online Support: Download "SIMATIC SCADA Export for TIA Portal" (https://support.industry.siemens.com/cs/ww/en/view/109748955)

Requirement

- The "SinumerikNC" channel is integrated in the project.
- The PLC tags are exported from TIA Portal in a zip file.

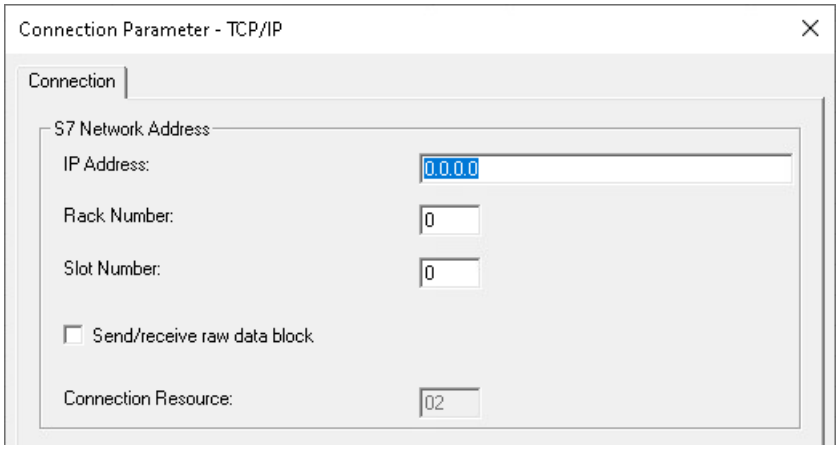
Procedure

1. Ensure that the access point of the application under "Set PG/PC interface" in the Windows system requirements is configured correctly.



2. Select the "TCP/IP" connection type in the communication driver tree.
3. In the channel unit shortcut menu, select "System parameters".
   The "System parameters - TCP/IP" dialog opens.
4. Switch to the "Unit" tab:
5. Assign the logical device name "S7Online".
6. Close the dialog with "OK".
7. Select the entry "New Connection" in the shortcut menu of the channel unit.
8. Enter the name of the connection.

9. Select the entry "Connection parameters" from the shortcut menu of the connection.
   The "Connection Parameters" dialog opens.



10. Enter the IP address of the PLC.
11. Enter the number of the rack on which the PLC is installed.
12. Enter the number of the slot.
13. Close the dialog with "OK".
14. From the shortcut menu of the SIMATIC connection, select "AS Symbols > Load from file".
    The dialog for selecting a file is displayed.
15. Select the ZIP file exported from TIA Portal with the PLC tags.
16. Click "Download".
    The available controller tags are loaded and the "Symbols" view opens.
    The loaded tags are displayed in the table area in the "AS Symbols" tab.
17. To choose the required tags, select the corresponding check boxes in the "Access" column.
18. Click to return to the Tag Management.
    The selected PLC tags are available in the Tag Management.

Result

NC variables and PLC tags can be used in pictures to show data.

Alarms of the SINUMERIK can be displayed with the AlarmControl.

18.2.3 How to configure alarms on the SINUMERIK 840D sl

You have the option to configure NC alarms. Alarms are displayed in Runtime in the WinCC AlarmControl.

The following options are available:

- None: No alarms displayed in the WinCC AlarmControl.
- Only numbers: The alarm number is displayed in the WinCC AlarmControl.
- Text and numbers: The alarm number and alarm text are displayed in the WinCC AlarmControl.

Note

NC alarms can currently only be configured in the TIA Portal. The alarms cannot be configured in SIMATIC Manager.

Requirement

- In the "Startup" tab of the "Computer properties" dialog, the "Alarm Logging Runtime" check box is selected.
- The "Sinumerik NC" and "SIMATIC S7 Protocol Suite" communication drivers are installed. The CNC controllers have two CPUs (NC and PLC). Therefore, two connections need to be configured.
- A connection is configured under the channel unit "SINUMERIK 840D sl".

Procedure

1. Select the connection under the "SINUMERIK 840D sl" channel unit.
2. Select the entry "Connection parameters" from the shortcut menu of the connection. The "Connection Parameters" dialog opens.
3. Under "NC Alarm", select how alarms are to be transferred: "Numbers only" or "Text and numbers".
4. To apply NC events, select the "NC Event" option.

Result

Alarms are displayed in Runtime in the WinCC AlarmControl.

See also

How to configure a connection to a SINUMERIK 840D sl (Page 413)

How to configure SINUMERIK S7-300 alarms (Page 420)

18.2.4 How to configure SINUMERIK S7-300 alarms

Alarms are displayed in Runtime in the WinCC AlarmControl.

Requirement

- In the "Startup" tab of the "Computer properties" dialog, the "Alarm Logging Runtime" check box is selected.
- The "SinumerikNC" communication driver is installed.
- A connection is configured under the channel unit "S7-300-PLC".
- PLC alarms are available as a TIA Portal export (*.zip) or as a SINUMERIK PLC alarm file (*.ts).

Procedure

1. Select the connection under the "S7-300-PLC" channel unit.
2. Select the entry "Connection parameters" from the shortcut menu of the connection.
   The "Connection Parameters" dialog opens.
3. Enter the IP address and rack/slot number.
   Close the dialog with "OK".
4. Add a tag in Tag Management:
   - Name: [name of the connection]#RawEvent
   - Data type: Raw Data Type
5. Click on the "Address" field.
6. Select "Event" as the raw data type.
   Close the dialog with "OK".
7. Navigate to "Alarm logging".
8. Click "AS messages".
   The connection is displayed.
9. Select "Load from file" in the shortcut menu of the connection.
10. Navigate to the path where the export file is located.
11. Select the export file (*.zip or *.ts).
12. Click "Download".
    Configured PLC alarms are available under AS messages.
13. Enable the required alarms.
14. Click on the "AS messages" node.
    Corresponding IDs are generated for each enabled PLC alarm.
15. This is how you configure the WinCC AlarmControl.

Result

Alarms are displayed in Runtime in the WinCC AlarmControl.

See also

How to configure alarms on the SINUMERIK 840D sl (Page 419)
How to configure a connection to a SINUMERIK 840D sl (Page 413)

18.3 Executing system functions

18.3.1 Supported system functions

The following system functions are available for configuration of the channel:

- LogonNC
- LogoffNC
- ChangeNCPassword
- ConfigureNCMachineData
- ResetNC
- AcknowledgeNCCancelAlarms
- SetNCUserFrame

18.3.2 LogonNC

The function transfers a password to the SINUMERIK NC.

Parameter	Explanation
Connection	Connection name
Password	Password for the logon at the SINUMERIK NC

Example VB script:

```
Dim output
LogonNC "Connection", "Password", output
Hmiruntime.Trace output
```

18.3.3 LogoffNC

The function logs off the connection to SINUMERIK NC. The password used is canceled.

Parameter	Explanation
Connection	Connection name

Example VB script:

```
Dim output
LogoffNC "Connection", output
Hmiruntime.Trace output
```

18.3.4 ChangeNCPassword

The function transfers a password to the SINUMERIK NC for a password level. The existing password for this level is overwritten.

Parameter	Explanation
Connection	Connection name
Password	The new password for the logon at the SINUMERIK NC
Level	NC password level: 0 = system 1 = manufacturer 2 = service 3 = user

Example VB script:

```
Dim output
ChangeNCPassword "Connection", "Password", "Level", output
Hmiruntime.Trace output
```

18.3.5 ConfigureNCMachineData

The function enables all machine data with classification NEW_CONF. The CLASS parameter allows for a detailed classification. Currently only the value 1 is supported.

Parameter	Explanation
Connection	Connection name
Class	Classification of machine data which are activated. Variable or constant: 1, 2, 3

Example VB script:

```
Dim output
ConfigureNCMachineData "Connection", "Class", output
Hmiruntime.Trace output
```

18.3.6 ResetNC

Die Function initiates a restart of the SINUMERIK NC.

Parameter	Explanation
Connection	Connection name

Example VB script:

```
Dim output
ResetNC "Connection", output
Hmiruntime.Trace output
```

18.3.7 AcknowledgeNCCancelAlarms

The function acknowledges all pending alarms of a connection.

Parameter	Explanation
Connection	Connection name

Example VB script:

```
Dim output
AcknowledgeNCCancelAlarms "Connection", output
Hmiruntime.Trace output
```

18.3.8 SetNCUserFrame

This function activates the data for the work offset for a channel.

During entering these tags are first stored in a temporary memory of the NC. With the SetNCUserFrame function the newly entered values are fully activated and can then be read out.

With this function only one work offset can be enabled. If several work offsets are set one after the other and then SetNCUserFrame is called, only the parameters of the last set work offset are activated.

Set all parameters of a work offset and then call the SetNCUserFrame. Then write the value of a further work offset.

Parameter	Explanation
Connection	Connection name
Channel	Channel that is activated. Variable or constant: 1 ... 31

Example VB script:

```
Dim output
SetNCUserFrame "Connection", "2XY", output
Hmiruntime.Trace output
```

XY is determined by the selected channel.

System Info

19

19.1 "System Info" Channel

Contents

The "System Info" channel is used to evaluate system information such as the time, date, disk capacity and provides functions such as timers and counters.

This chapter will show you

- configure the channel, connection and tags
- display system information in a process picture
- use system information to trigger and display a message
- display system information graphically
- display the system information from several servers in a multi-user system

19.2 WinCC System Info Channel

Principle

The System Info channel is used to evaluate system information such as the time, date, disk capacity and provides functions such as timers and counters.

Possible applications are:

- Display of the time, date and day of the week in process pictures
- Triggering of events through evaluation of system information in scripts
- Display of the CPU load in a trend
- Display and monitoring of the available drive space on different servers of a client system
- Monitoring of the available disk capacity and triggering of a message

The channel requires no hardware, since it directly accesses the system information of the computer on which it has been installed. In order for the channel to function, a connection must be set up. Additional connections are possible, but not required for the proper operation.

For more information regarding the diagnosis of channels and tags, refer to "Communication Diagnostics".

Note

Licensing

The process tags required for the System Info channel need no licenses. Thus, the tags are not entered in the license count.

User Rights

If you have no administrator rights, you must be a power user and member of the "Performance Monitor User" group in order to use the System Info channel.

Tags for the connection status

The tags @<connection_name>@ForceConnectionStateEx and @<connection_name>@ConnectionStateEx are not supported by the "System Info" channel.

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 item number:

- A5E00391327

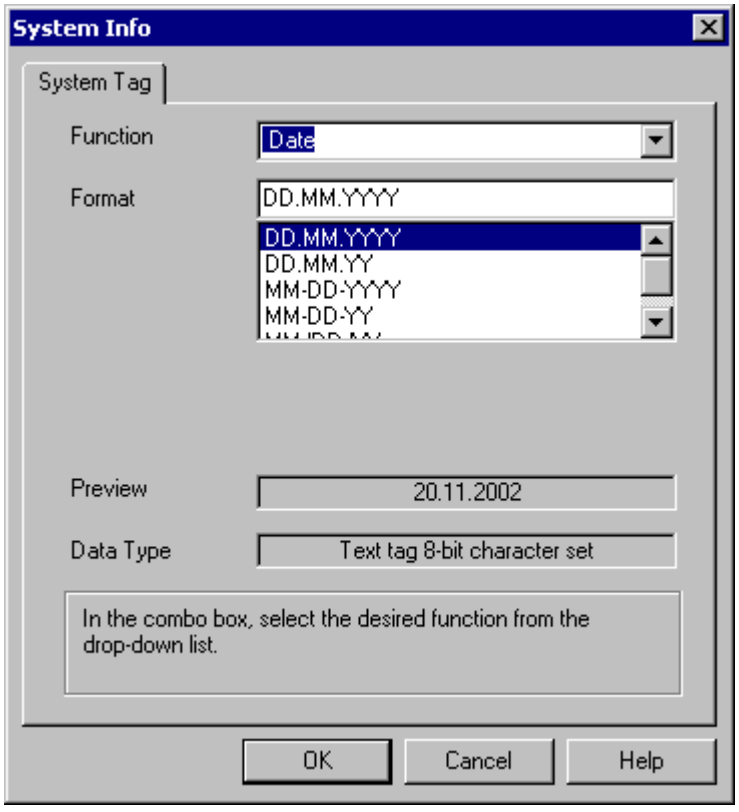
See also

Use in Multi-User and Client Systems (Page 449)
How To Call Up and Evaluate System Information (Page 437)
How to Configure the System Info Channel (Page 436)
Differences to Other Software Components (Page 435)
Overview of the Supported System Information (Page 430)

19.3 Overview of the Supported System Information

Introduction

In the "Function" field in the "System Info" dialog, you can specify the system information to be assigned to a WinCC tag. The display format is set in the "Format" field.



Overview of the supported system information of the "System Info" channel

Function	Data type	Format	Preview
Date	Text tag 8-bit character set	DD.MM.YYYY DD.MM.YY MM-DD-YYYY MM-DD-YY MM/DD/YY	21.10.1999 21.10.99 10-21-1999 10-21-99 10/21/99
Day	Unsigned 16-bit value	DD	1...31
Month	Unsigned 16-bit value	MM	1...12
Year	Unsigned 16-bit value	YYYY	2000

Function	Data type	Format	Preview
Day of the week	Unsigned 16-bit value Text tag 8-bit character set	Text: 1 for Monday to 7 for Sunday Text: Mon,Tue,Wed, Thu,Fri,Sat,Sun	1...7 Mon ... Sun
Time	Text tag 8-bit character set; length = 10 bytes Length = 12 bytes	HH:MM:SS HH:MM HH:MM AM,PM HH:MM:SS AM,PM	23:45:37 23:45 23:45 PM 23:45:37 PM
Hour	Unsigned 16-bit value	HH	0...23
Minute	Unsigned 16-bit value	MM	0...59
Second	Unsigned 16-bit value	SS	0...59
Millisecond	Unsigned 16-bit value	MSC	0...999
Counter	Signed 32-bit value	ZZZZ	0...9999
CPU load	Floating-point number 32-bit IEEE 754	Total load in % Idle time in % Process in %	0...100%
Timer	Signed 32-bit value	TTTT	0...9999
Free main memory	Floating-point number 32-bit IEEE 754	Free in kB Free in % Free in bytes	0...n kB 0...100% 0...n B
Free disk capacity (local disks)	Floating-point number 32-bit IEEE 754	Free in MB Free in %	0...n MB 0...100%
Printer monitoring	Unsigned 32-bit value	Filled capacity of spooler disk Printer status Job status Free spooler disk area in kB Free PRT_OUT- Disk areas in kB Size of spooler directory in kB Size of PRT_OUT directory in kB	0...n % 0...n 0...n 0...n kB 0...n kB 0...n kB 0...n kB
Status of the export file	Floating-point number 32-bit IEEE 754	Used in kB Used in % Available in kB	0...n kB 0...100% 0...n B

Counter

This function can be used in scripts for specific test purposes.

Timer

When this function is selected, the "System Info" dialog is expanded by the fields "Limits from" and "to".

After every second, the timer is incremented or decremented. The direction in which the changes are made is determined by the start and end values in the fields "Limits from" and "to". If the start value is smaller than the end value, the timer is incremented. If the start value is greater than the end value, the timer is decremented.

If a value is entered in the I/O field linked to the timer in Runtime, the start and current timer values are set to this value. Example: Timer configured from 0 to 60. If "0" is written in Runtime, the timer is reset.

After the deactivation, the original start value is reapplied.

CPU load

For the formats "Total load in %" and "Idle load in %" in connection with multiprocessor PCs, the CPU number must be entered, beginning with "0".

For the format "Process in %", the instance number needs to be specified as well.

Example

Three processes are running for "CCWriteArchiveServer":

- TagLogging slow
- TagLogging fast
- AlarmLogging

The corresponding format is specified as follows:

- CCWriteArchiveServer#0
- CCWriteArchiveServer#1
- CCWriteArchiveServer#2

The name of the process is specified without file extension (.exe).

Free disk capacity

The system can only determine the space available on a local hard disk or diskette.

Printer monitoring

With the "Printer status" and "Job status" formats, the server name must be entered in the "Printer" field. The printers in use must support this status information to be able to use this system information.

In order to analyze the printer status, please observe the following:

- The port monitor is responsible for the transmission of the printer status to the spooler. Depending on the selected printer port, different port monitor DLLs are installed. For the port monitors supplied with Windows, only "TCPMON.DLL" is capable of transmitting the printer status using the TCP/IP port. "LOCALMON.DLL" using the LPT port does not communicate the printer status.
- The printer status is determined only after a print job has been submitted, but not during polling of the status at the port.

In the case of the "Free PRT_OUT drive space" and "PRT_OUT Directory Size" formats, the channel automatically determines the path in the "Directory" field.

Error codes for the "Printer status" format

Status	Error code
PRINTER_STATUS_PAUSED	0x00000001
PRINTER_STATUS_ERROR	0x00000002
PRINTER_STATUS_PENDING_DELETION	0x00000004
PRINTER_STATUS_PAPER_JAM	0x00000008
PRINTER_STATUS_PAPER_OUT	0x00000010
PRINTER_STATUS_MANUAL_FEED	0x00000020
PRINTER_STATUS_PAPER_PROBLEM	0x00000040
PRINTER_STATUS_OFFLINE	0x00000080
PRINTER_STATUS_IO_ACTIVE	0x00000100
PRINTER_STATUS_BUSY	0x00000200
PRINTER_STATUS_PRINTING	0x00000400
PRINTER_STATUS_OUTPUT_BIN_FULL	0x00000800
PRINTER_STATUS_NOT_AVAILABLE	0x00001000
PRINTER_STATUS_WAITING	0x00002000
PRINTER_STATUS_PROCESSING	0x00004000
PRINTER_STATUS_INITIALIZING	0x00008000
PRINTER_STATUS_WARMING_UP	0x00010000
PRINTER_STATUS_TONER_LOW	0x00020000
PRINTER_STATUS_NO_TONER	0x00040000
PRINTER_STATUS_PAGE_PUNT	0x00080000
PRINTER_STATUS_USER_INTERVENTION	0x00100000
PRINTER_STATUS_OUT_OF_MEMORY	0x00200000
PRINTER_STATUS_DOOR_OPEN	0x00400000
PRINTER_STATUS_SERVER_UNKNOWN	0x00800000
PRINTER_STATUS_POWER_SAVE	0x01000000

Error codes for the "Job status" format

Status	Error code
JOB_STATUS_PAUSED	0x00000001
JOB_STATUS_ERROR	0x00000002
JOB_STATUS_DELETING	0x00000004
JOB_STATUS_SPOOLING	0x00000008
JOB_STATUS_PRINTING	0x00000010
JOB_STATUS_OFFLINE	0x00000020
JOB_STATUS_PAPEROUT	0x00000040
JOB_STATUS_PRINTED	0x00000080
JOB_STATUS_DELETED	0x00000100
JOB_STATUS_BLOCKED_DEVQ	0x00000200
JOB_STATUS_USER_INTERVENTION	0x00000400
JOB_STATUS_RESTART	0x00000800

Note

The error codes of the "Printer status" and "Job status" formats correspond to the values in the Visual C-referenced file "Winspool.h" .

19.4 Differences to Other Software Components

Introduction

Some of the System Info channel system information can also be evaluated or displayed by WinCC using ActiveX controls.

Once fundamental difference to ActiveX controls can be seen in the fact that the System Info channel system information is assigned to a WinCC tag. The continued evaluation (e.g. of messages, limit values) can be carried out repeatedly and is then configured individually. The ActiveX controls are intended for use in the specified applications and may also be used for multi-user or client systems.

For the following system information, differences between the ActiveX control and the channel exist:

Time

The ActiveX control "WinCC Digital/Analog Clock Control" is used to display the time in WinCC. This control also supports an analog display of the time. The control does not require the System Info channel even when if it is used in the process control system options. Using the control, it is possible to display a WinCC client's time in its process picture. This is not possible with the System Info channel, since this always displays the server's system time.

Free disk capacity

The ActiveX control "IX Diskspace" is supplied to display the disk space available in WinCC. This ActiveX control can also display the space available on network drives and supports other configuration options such as the setting of multiple limit values directly in the control.

19.5 Configuring the Channel

19.5.1 How to Configure the System Info Channel

Introduction

This section illustrates how to configure the System Info channel.

Procedure

1. In the navigation area of the tag management, select the entry "Add new driver" in the shortcut menu of node "Tag Management".
2. Select the "System Info" driver. The channel is created and the communication driver is displayed in the tag management.
3. Select the associated System Info channel unit and call up the shortcut menu. In this shortcut menu, select "New Connection".
4. Enter the name of the connection.
5. Click the "Tags" tab below the table area.
6. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
   If you want to use examples for this channel, continue with the topic "How to Configure a Tag".
7. Select the desired data type in the "Data Type" field.

19.6 Examples of Evaluating and Displaying System Information

19.6.1 How To Call Up and Evaluate System Information

This section uses examples to illustrate how system information can be displayed and evaluated in a variety of ways.

See also

How to Display the Printer Status in a Status Display (Page 445)

How to Configure a Message Regarding Free Disk Capacity (Page 442)

How to Display the CPU Load in a Trend Window (Page 441)

How to Display the Free Disk Capacity in a Bar Graph (Page 440)

How to Display the Time in an I/O Field (Page 438)

How to Configure a Tag in the System Info Channel (Page 437)

19.6.2 How to Configure a Tag in the System Info Channel

Introduction

This section illustrates how to configure tags in the System Info channel. These tags are used in the examples.

Requirements

Install the "SystemInfo.chn" channel.

Table of the Data Types Used

The table below shows the tag types and formats used in the System Info channel.

Example	Tag name	System information (function)	Format	Data type
I/O field	Sysinfo_Time	Time	Hours:Minutes:Seconds (HH:MM:SS)	Text tag 8-bit character set
Bar, message	Sysinfo_Drive_C	Free drive capacity (drive: C)	Number 0-100% (free as %)	Floating-point number 32-bit IEEE 754

Example	Tag name	System information (function)	Format	Data type
Trends	Sysinfo_CPU	CPU load	Number 0-100% (total load as %)	Floating-point number 32-bit IEEE 754
Printer status	Sysinfo_Printer-state	Printer monitoring	Number 0-n (hex) (printer status)	Unsigned 32-bit value

Procedure

1. In the shortcut menu of the associated System Info channel unit, select the entry "New Connection" and create a connection named "Testinfo".
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. Open the "System Info" dialog.
   For this purpose, click in the "Address" field and then the ... button.
5. Apply the function suitable for the example and the display format from the table.
   The “Data type” field is adapted automatically.
6. Close the dialog.

See also

How to Configure a Message Regarding Free Disk Capacity (Page 442)
How to Configure the System Info Channel (Page 436)
How to Display the Printer Status in a Status Display (Page 445)
How to Display the CPU Load in a Trend Window (Page 441)
How to Display the Free Disk Capacity in a Bar Graph (Page 440)
How to Display the Time in an I/O Field (Page 438)

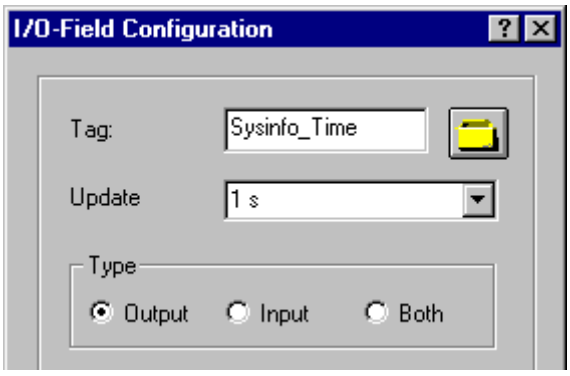
19.6.3 How to Display the Time in an I/O Field

Requirements

Configure a "Sysinfo_Time" tag with data type "Text tag 8-bit character set". This tag must be assigned the "Time" system information with display format "HH:MM:SS".

Procedure

1. Start Graphics Designer and open a picture.
2. 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.



3. In the "Tag" field, enter the name "Sysinfo_Time".
4. Set the update to "1 s".
5. Set the field type to "Output". Close the dialog.
6. Click "Properties" in the I/O field's shortcut menu to open the "Object Properties" dialog.
7. On the "Properties" tab, select "Output/Input". Set the "Data Format" attribute to "String".
8. Close the dialog and save the picture.
9. Click the appropriate button in the Graphics Designer toolbar to activate Runtime.

Note

The update cycle should be chosen with careful consideration, as it affects the load on the computer. Therefore, updating a time display every 250 ms is detrimental to the system performance.

See also

How to Start Runtime (Page 448)

How to Insert an I/O Field (Page 448)

How to Configure a Tag in the System Info Channel (Page 437)

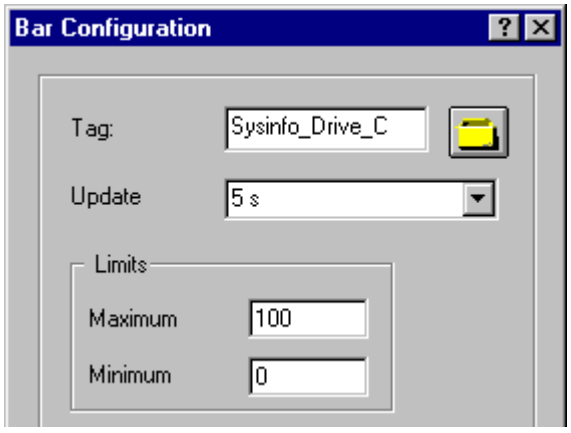
19.6.4 How to Display the Free Disk Capacity in a Bar Graph

Requirements

Configure a tag "Sysinfo_Drive_C" with data type "Floating-point number 32-bit IEEE754". This tag must be assigned the "Free Disk Space" system information for drive "C" with display format "Free capacity in %".

Procedure

1. Start Graphics Designer and open a picture.
2. Insert a bar graph into the picture. For this purpose, select the object "Bar" from "Smart Objects" in the object palette. The "Bar Configuration" dialog is opened.



3. In the "Tag" field, enter the name "Sysinfo_Drive_C".
4. Set the update to "5 s".
5. Set the maximum value to "100" and the minimum value to "0". Close the dialog.
6. Click on "Properties" in the bar graph shortcut menu to open the "Object Properties" dialog.
7. On the "Properties" tab, select "Axis". Set the attribute "Decimal Places" to "0".
8. Close the dialog and save the picture.
9. Click the appropriate button in the Graphics Designer toolbar to activate Runtime.

Note

The update cycle should be chosen with careful consideration, as it affects the load on the computer. Therefore, updating an available drive space display every second is detrimental to the system performance.

See also

How to Start Runtime (Page 448)

How to Insert a Bar Graph (Page 447)

How to Configure a Tag in the System Info Channel (Page 437)

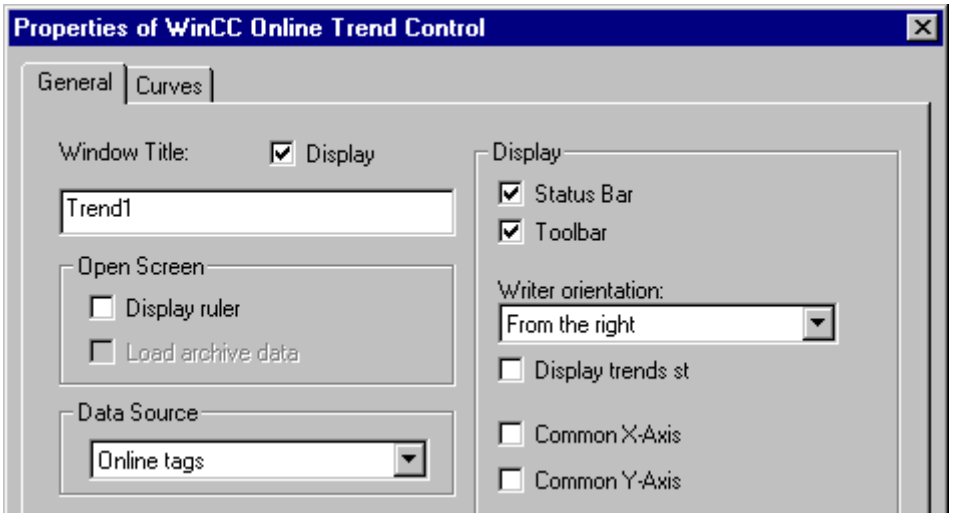
19.6.5 How to Display the CPU Load in a Trend Window

Requirements

Configure a tag named "Sysinfo_CPU" with data type "Floating-point number 32-bit IEEE754". This tag must be assigned the "CPU Load" system information with display format "Total load in %".

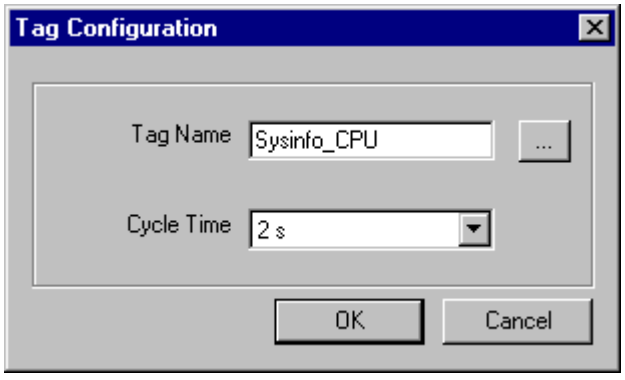
Procedure

1. Start Graphics Designer and open a picture.
2. Insert a trend display into the picture. You can accomplish this by selecting the "WinCC Online Trend Control" object from the "Controls" object palette. The "Properties of WinCC Online Trend Control" dialog is opened.



3. On the "General" tab, enter the name "Trend1" in the "Window Title" field.
4. In the "Data Source" field, select "Online Tags".
5. Click the "Trends" tab and then the "Selection" button to open the "Tag Configuration" dialog.

6. Enter "Sysinfo_CPU" as the name of the tag and then select a cycle time of "2 s". Close the dialog.



7. Close the "Properties of WinCC Online Trend Control" dialog and save the picture.
8. Click the appropriate button in the Graphics Designer toolbar to activate Runtime.

Note

The update cycle should be chosen with careful consideration, as it affects the load on the computer. Therefore, updating a CPU load display every 500 ms is detrimental to the system performance.

See also

How to Start Runtime (Page 448)

How to Configure a Tag in the System Info Channel (Page 437)

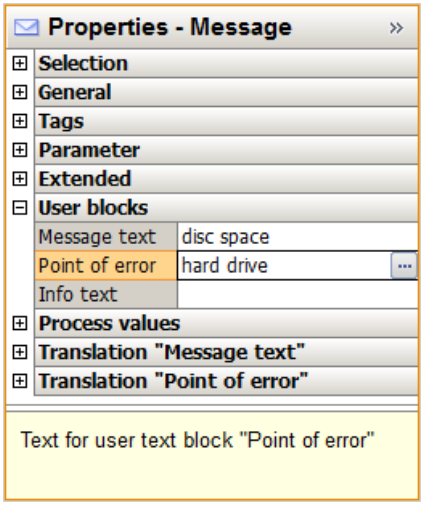
19.6.6 How to Configure a Message Regarding Free Disk Capacity

Requirements

Configure a tag "Sysinfo_Drive_C" with data type "Floating-point number 32-bit IEEE754". This tag must be assigned to the "available Drive Space" system information for drive "C" with display format "Free mem in %".

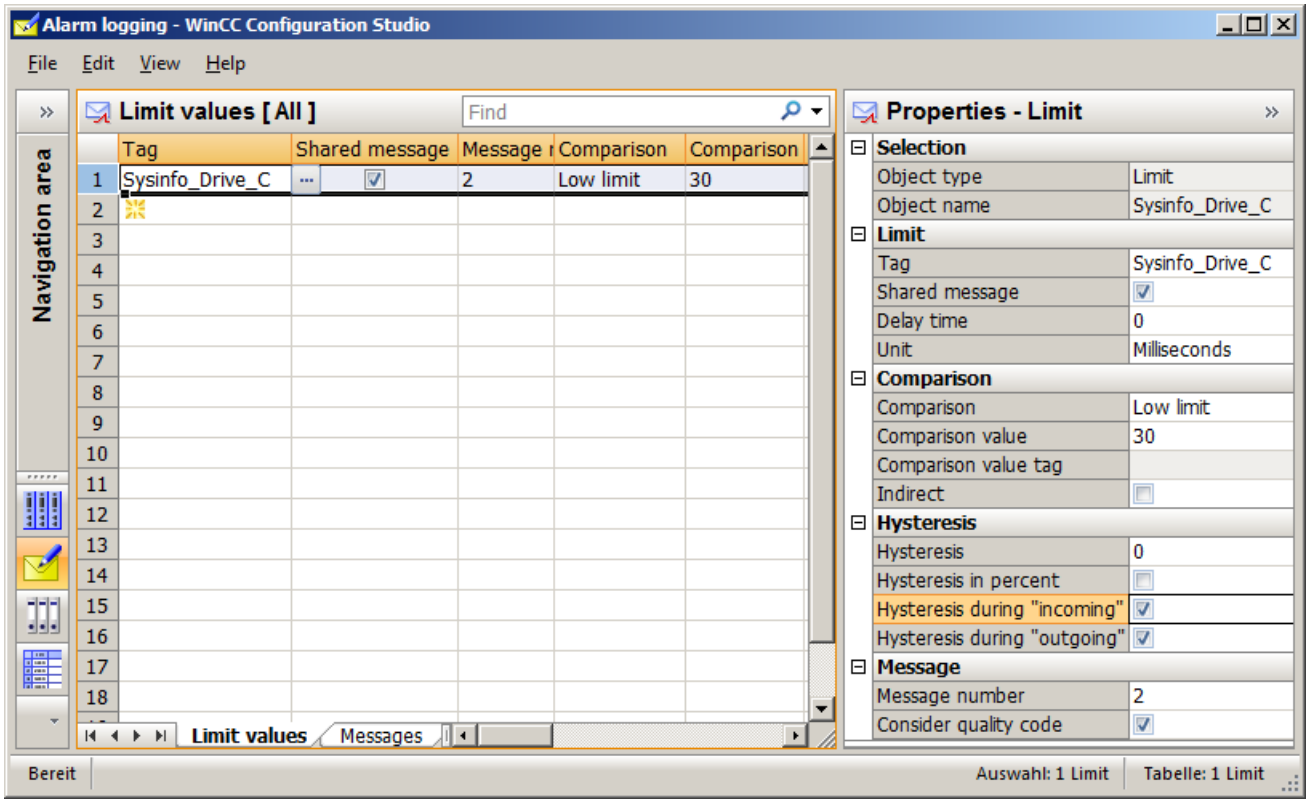
Procedure

1. Open the "Alarm Logging" editor and create a new message.
2. In the "Properties" area, define:
   - the text "Memory space" as "Message text"
   - the text "Hard drive" as "Point of error"



3. In the Alarm Logging navigation area, select the "Limit monitoring" node.
4. Create a new limit value for the tag "Sysinfo_Drive_C" in the "Limits" data window.
5. Activate the option "Shared message".

6. In the "Message Number" field, enter the number of the newly created message.
7. In the “Limits” data window extend the entry "Sysinfo_Drive_C” by clicking the arrow symbol.
   – Select “Low limit” in the newly inserted line.
   – Enter the value "30" in the “Comparison value" box.



See also

How to Configure a Tag in the System Info Channel (Page 437)

How to Display a Message regarding the Available Disk Capacity (Page 444)

19.6.7 How to Display a Message regarding the Available Disk Capacity

Requirements

- A tag "Sysinfo_Drive_C" with data type "Floating-point number 32-bit IEEE754". The tag must be assigned to the "Free Disk Space" system information for drive "C" with the display format "Free memory in %".
- A message text and the lower limit value for the limit value monitoring of this tag are configured.
- "Alarm Logging Runtime" must is set in the computer's startup parameters.

Procedure

1. Start Graphics Designer and open a picture.
2. Insert a message window into the picture. Select the "WinCC Alarm Control" object from the "Controls" object palette and place it in the picture.
3. Activate the check boxes “Message Text” and “Point of Error” in the “Existing message blocks” field on the “Message blocks” tab of the “Properties” dialog.
4. Move the existing message blocks “Message Text” and “Point of Error” into the “Elements of the message line” field of the “Message lists” tab".
5. Close the dialog and save the picture.
6. Click the appropriate button in the Graphics Designer toolbar to activate Runtime.

See also

How to Check the WinCC Startup Parameters (Page 447)

How to Start Runtime (Page 448)

How to Configure a Tag in the System Info Channel (Page 437)

19.6.8 How to Display the Printer Status in a Status Display

Introduction

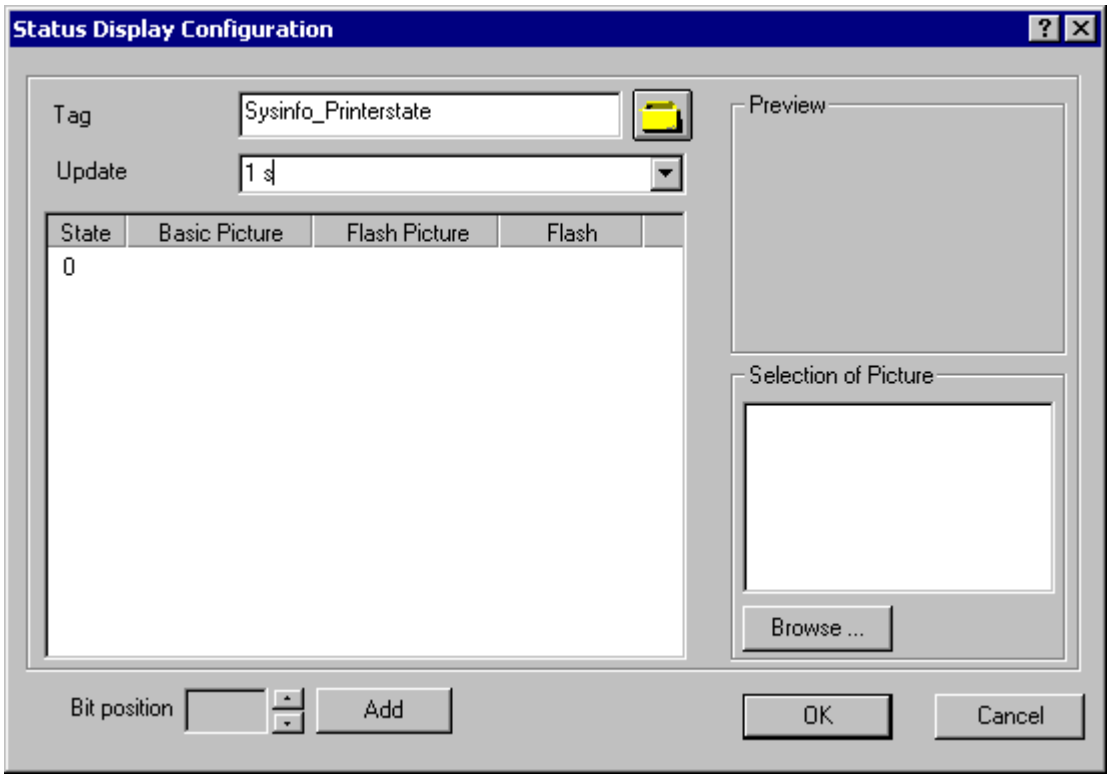
This example shows a possible evaluation of the "Printer monitoring" system information based on the "Printer status" display format. The printer or job status could also be evaluated by configuring messages that are triggered by single bits in the printer or job status.

Requirements

- Configure a "Sysinfo_Printerstate" tag with data type "Unsigned 32-bit value". This tag must be assigned to the "Printer monitoring" system information with the "Printer status" display format.
- The printers in use must support these status displays to be able to use this system information.

Procedure

1. Start Graphics Designer and open a picture.
2. Insert a status display into the picture. Select the "Status display" object from the object palette under "Smart Objects". The "Status Display Configuration" dialog is opened.



3. In the "Tag" field, enter the name "Sysinfo_Printerstate".
4. Set the update to "1 s".
5. Select the value "0" in the "Status" column. Assign this status an icon from the "Picture Selection" area which, for example, represents a printer. Select the desired icon, drag it to the "0" line with the mouse and drop it in the "Basic Picture" column. If no picture or a picture other than the desired picture is shown in the "Picture Selection" area, a selection dialog can be opened by clicking the "Browse..." button.
6. If you wish you can add additional bit positions with the "Add" button and assign another picture to these statuses.
7. Close the dialog and save the picture.
8. Click the appropriate button in the Graphics Designer toolbar to activate Runtime.

In Runtime, a printer ready for use is displayed with the icon assigned the status "0" in step 5. No picture is shown if there is an error in the printer or if you have not run step 6. If you assigned icons to the other bit positions in step 6, they are shown accordingly.

Note

The update cycle should be chosen with careful consideration, as it affects the load on the computer. Therefore, updating a time display every 250 ms is detrimental to the system performance.

See also

How to Configure a Tag in the System Info Channel (Page 437)

How to Start Runtime (Page 448)

19.6.9 How to Check the WinCC Startup Parameters

Procedure

1. Open the "Computer" editor in the WinCC Configuration Studio.
2. Select the computer name in the navigation area.
   The "Processes when starting WinCC Runtime" and "Additional applications" tabs are displayed in the data area.
3. Activate the required runtime applications on the "Processes when starting WinCC Runtime" tab.
4. To add more applications to the startup list, switch to the "Additional applications" tab
5. In the "Application" column, click the "..." button in the first empty box and select the desired application.

19.6.10 How to Insert a Bar Graph

Procedure

1. Start Graphics Designer and open a picture.
2. In the "Standard" object palette under "Smart Objects", select the object "Bar".
3. Insert a bar graph into the picture. To do this, point the mouse on the location in the working area where you want the bar graph to be placed. While keeping the mouse button pressed, drag the object to the desired size.

4. After you release the mouse button, the "Bar Configuration" dialog is opened. In this dialog, enter the name of a WinCC tag and set the update specifications and limit. Additionally, you can use the "Bar Direction" to set the orientation of the displayed bar.
5. Close the dialog.

19.6.11 How to Insert an I/O Field

Procedure

1. Start Graphics Designer and open a picture.
2. In the "Standard" object palette under "Smart Objects", select the object "I/O Field".
3. Insert the I/O field into the picture. To do this, position the mouse on the location in the working area where you want the I/O field to be placed. While keeping the mouse button pressed, drag the object to the desired size.
4. After you release the mouse button, the "I/O Field Configuration" dialog is opened. In this dialog, enter the name of a WinCC tag and set the update and field type settings. Additionally, you can also select the "Font" to be used to display the value.
5. Close the dialog.

19.6.12 How to Start Runtime

Requirements

A startup picture must be defined before Runtime is activated.

Procedure

1. Save and close all files that may be open in an editor.
2. Select WinCC Explorer.
3. Activate the project by clicking the "Activate" button in the toolbar or by selecting "Activate" in the "File" menu.

19.7 Special Functions

19.7.1 Use in Multi-User and Client Systems

19.7.1.1 Use in Multi-User and Client Systems

Introduction

In multi-user and client systems, the System Info channel can be used to process the system information from a server on a client system. In a client system, it is thus possible for a single WinCC client to monitor several servers.

See also

Monitoring the system information of several servers on a WinCC client (Page 449)

19.7.2 Example of monitoring system information from multiple servers

19.7.2.1 Monitoring the system information of several servers on a WinCC client

Introduction

In this example, two servers are monitored by a single WinCC client. The monitored system information such as available disk space and CPU load is displayed in a process picture on the WinCC client.

This requires the following configurations:

Configuration of first server

Configuration of second server

Import of tags on the WinCC client

Configuration of the process picture on the WinCC client

Activation of the project

Requirements

The server and the WinCC client must be connected through a Windows network.

See also

How to Activate the Project (Page 454)

How to Configure the Process Picture on the WinCC Client (Page 453)

How to Import the Tags to the WinCC Client (Page 452)

How to Configure the Second Server (Page 451)

How to Configure the First Server (Page 450)

19.7.2.2 How to Configure the First Server

Introduction

This section presents the configuration of the first server, which is necessary for this example.

1. Configure the tags of the "System Info" channel to display available drive space and CPU load.
2. Generation of a package.

Table of the Data Types Used

The tag names and formats used in this "System Info" channel example are listed in the following table.

Tag	Function	Data type	Format
Sysvar_1_Drive_C	Free drive capacity	Floating-point number 32-bit IEEE 754	0-100% (free in %)
Sysvar_1_CPU	CPU load	Floating-point number 32-bit IEEE 754	0-100% (total load in %)

Procedure

1. Create a multi-user project named "Testinfo_1" on the first server. Install the "System Info" driver in the project.
2. In the shortcut menu of the associated "System Info" channel unit, select the entry "New Connection" and create a connection named "Connection1".
3. Click the "Tags" tab below the table area.
4. Click in the top free cell of the "Name" column.
   Enter the name "Sysvar_1_Drive_C" for the tag.
5. Set the "Data Type" to "Floating-point number 32-bit IEEE 754".
6. Open the "System Info" dialog.
   For this purpose, click in the "Address" field and then on the ... button.
7. Set the "Function" field to "Available Drive Space", the "Drive" to "C" and "Format" to "Free capacity in %". Close the dialog.
8. Click in the top free cell of the "Name" column.
   Enter the name "Sysvar_1_CPU" for the tag.

9. Set the "Data Type" to "Floating-point number 32-bit IEEE 754".
10. Open the dialog "System Info".
    For this purpose, click in the "Address" field and then on the ... button.
11. Set the value in the "Function" field to "CPU Load" and the value in "Format" to "Total load in %". Close the dialog.
12. Create a package. Proceed by selecting "Server data" in the navigation window and opening the shortcut menu. Select the menu item "Create". Acknowledge the message stating that the package was created.

See also

How to Configure the Second Server (Page 451)

19.7.2.3 How to Configure the Second Server

Introduction

This section presents the configuration of the second server, which is necessary for this example.

1. Configure the tags of the "System Info" channel to display available drive space and CPU load.
2. Generation of a package.

Table of the Data Types Used

The tag names and formats used in this "System Info" channel example are listed in the following table.

Tag	Function	Data type	Format
Sysvar_2_Drive_C	Free drive capacity	Floating-point number 32-bit IEEE 754	0-100% (free in %)
Sysvar_2_CPU	CPU load	Floating-point number 32-bit IEEE 754	0-100% (total load in %)

Procedure

1. Create a multi-user project named "Testinfo_2" on the second server. Install the "System Info" driver in the project.
2. In the shortcut menu for the associated "System Info" channel unit, select the entry "New Connection" and create a connection named "Connection2".
3. Click the "Tags" tab below the table area.
4. Click in the top free cell of the "Name" column.
   Enter the name "Sysvar_2_Drive_C" for the tag.
5. Set the "Data Type" to "Floating-point number 32-bit IEEE 754".
6. Open the "System Info" dialog.
   For this purpose, click in the "Address" field and then on the ... button.

7. Set the "Function" field to "Available Drive Space", the "Drive" to "C" and "Format" to "Free capacity in %". Close all open dialogs.
8. Click in the top free cell of the "Name" column.
   Enter "Sysvar_2_CPU" as the name of the tag.
9. In the connection shortcut menu, select "New Tag".
10. Set the "Data Type" to "Floating-point number 32-bit IEEE 754".
11. Open the dialog "System Info".
    For this purpose, click in the "Address" field and then on the ... button.
12. Set the value in the "Function" field to "CPU Load" and the value in "Format" to "Total load in %". Close all open dialogs.
13. Create a package. Proceed by selecting "Server data" in the navigation window and opening the shortcut menu. Select the menu item "Create". Acknowledge the message stating that the package was created.

See also

How to Import the Tags to the WinCC Client (Page 452)

19.7.2.4 How to Import the Tags to the WinCC Client

Introduction

This section presents the configuration of the WinCC client, which is necessary for this example.

1. Loading the package of the first sever project.
2. Loading the package of the second sever project.

Requirements

This example requires the use of two server project packages.

Server	Project	Package
1	Testinfo_1	Testinfo_1_<computer_name>
2	Testinfo_2	Testinfo_2_<computer_name>

Procedure

1. Create a client project named "mc_info" on the WinCC client.
2. In the server data shortcut menu, select "Load". The "Open" dialog is opened.
3. Select the computer on which the first server project "Testinfo_1" is located.
4. Select the package "Testinfo_1_<computer_name>.pck" in the "<project_name> \ <computer_name> \ Packages" directory.

5. Click the "Open" button and acknowledge the message after the package has opened.
6. Load the package "Testinfo_2_<computer_name>.pck" on the second server. For this purpose, repeat steps 2 to 5 with the appropriate settings and names for the second project from the "Requirements" table.

See also

How to Configure the Process Picture on the WinCC Client (Page 453)

19.7.2.5 How to Configure the Process Picture on the WinCC Client

Introduction

This section illustrates the configuration of the WinCC client, which is required in this example to display the server system information in a process picture on a WinCC client.

1. Configuration of the system information display of the first server
2. Configuration of the system information display of the second server

Requirements

This example requires that the server project packages are loaded in the client project.

Package	Project	Tag
Testinfo_1_<computer_name>	Testinfo_1	Sysvar_1_Drive_C
Testinfo_1_<computer_name>	Testinfo_1	Sysvar_1_CPU
Testinfo_2_<computer_name>	Testinfo_2	Sysvar_2_Drive_C
Testinfo_2_<computer_name>	Testinfo_2	Sysvar_2_CPU

Procedure

1. On the WinCC client, start Graphics Designer and create a picture named "p_serverinfo".
2. 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.
3. Click the button for tag selection. The "Tags" dialog is opened.
4. Select the tag "Sysvar_1_Drive_C" of the first server project "Testinfo_1". For this purpose, open the directory structure under the package "Testinfo_1_<computer_name>". Close the dialog.
5. In the "I/O Field Configuration" dialog, set the update to "5 s".
6. Set the field type to "Output". Close the dialog.
7. Insert a second I/O field into the picture and configure it for the tag "Sysvar_1_CPU" of the same project. For this purpose, repeat the steps 2 to 6 with the appropriate settings taken from the "Requirements" table.

8. Repeat steps 2 to 7 to configure the tags of the second server project "Testinfo_2".
9. Close the dialogs and save the picture.

Note

The update cycle should be chosen with careful consideration, as it affects the load on the computer. Therefore, updating a date display every second is detrimental to the system performance.

See also

How to Activate the Project (Page 454)

19.7.2.6 How to Activate the Project

Introduction

This section shows how to activate the projects on the servers and the WinCC client.

1. Create a startup picture in the server project "Testinfo_1" and activate the project.
2. Create a startup picture in the server project "Testinfo_2" and activate the project.
3. Define the startup picture in the project "mc_info" on the WinCC client and activate the project.

Procedure

1. In the navigation window in the project "Testinfo_1" on server 1, select Graphics Designer and use the shortcut menu to create a new picture.
2. Set this picture as the startup picture. To do this, select "Set as startup picture" from the shortcut menu.
3. Click the "Activate" button in the toolbar to activate the project.
4. In the navigation window in the project "Testinfo_2" on server 2, select Graphics Designer and use the shortcut menu to create a new picture.
5. Set this picture as the startup picture. To do this, select "Set as startup picture" from the shortcut menu.
6. Click the "Activate" button in the toolbar to activate the project.
7. On the WinCC client in the navigation window in the project "mc_info", select Graphics Designer. The process picture "p_serverinfo" is displayed in the data window.
8. Set this picture as the startup picture. To do this, select "Set as startup screen" from the shortcut menu.
9. Click the "Activate" button in the toolbar to activate the project.

WinCC Unified Channel

20

20.1 Channel "WinCC Unified Channel"

Introduction

The "WinCC Unified Channel" channel enables communication as part of the Unified Collaboration between WinCC and WinCC Unified.

You can use shared tags, for example, to interconnect picture objects, for dynamization, and in scripting or for triggering messages.

A detailed description of collaboration is available in the documentation for the WinCC Unified Engineering System in the section "Using distributed systems".

More information:

- Manual: SIMATIC HMI WinCC Unified Engineering (https://support.industry.siemens.com/cs/ww/en/view/109813308/141159929739).

Channel units

The "WinCC Unified Channel" channel has a "Unified Tag Access" channel unit.

Licensing

To use the WinCC Unified Channel under WinCC, you need the Connectivity Pack license. The connected tags do not count in the project PowerTag counting.

In WinCC Unified, the station requires a Unified Collaboration license.

More information: Collaboration license (https://support.industry.siemens.com/cs/ww/en/view/109798671/143737848971).

See also

WinCC Unified documentation (https://support.industry.siemens.com/cs/ww/en/view/109813308/141159929739)

Collaboration license (https://support.industry.siemens.com/cs/ww/en/view/109798671/143737848971)

20.2 Supported Data Types

Define the required tags for a logical connection with a connected controller.

The following data types are supported by the "WinCC Unified Channel" channel:

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

Structure tags are not supported. Structure tag elements are read in flat as individual data points during the import.

20.3 Configuring the channel

20.3.1 Requirements

Unified Collaboration certificate

To use the channel, WinCC requires a Unified Collaboration certificate.

1. Create a certificate for the WinCC system with the WinCC Unified Certificate Manager, which is also used to manage the certificates of the WinCC Unified stations.
2. Export the device.
   - To do this, select "Export device > To PC ..." in the device shortcut menu in the Certificate Manager.
   - Assign a password.
3. Open the WinCC Certificate Manager on the WinCC system.
4. To import the configuration, click "Open configuration".
   To import, you need the previously assigned password.
   The device is displayed together with the certificate in the Certificate Manager.
5. Select "Install" from the shortcut menu of the imported certificate.
   The imported Unified Collaboration certificate appears under "Installed Certificates".

Collaboration

- In the WinCC Unified Engineering System, "Collaboration" is activated, configured, and active on the Unified Runtime systems.
- The collaboration data of the remote system configured on the connection must match that in the WinCC Unified Engineering System:
  - system ID
  - IP address
  - collaboration name

Note

Duplicating a WinCC project

Modify the collaboration data after duplicating a project.

The collaboration data must be unique for all collaboration systems involved.

20.3.2 How to configure the channel

To be able to exchange data via collaboration, WinCC requires a logical connection.

All connection-specific parameters are defined during the setup of the connection.

Basic procedure

The following steps are required for configuring the "WinCC Unified Channel":

1. Configuring the channel
2. Configuring a connection
3. Configuring tags

Note

Unique names: System ID / Collaboration name

The system ID and collaboration name must be unique throughout the system among all participating collaboration systems.

If the project has been copied or duplicated, adjust the configuration accordingly. The system ID and collaboration name must also be unique on redundant systems.

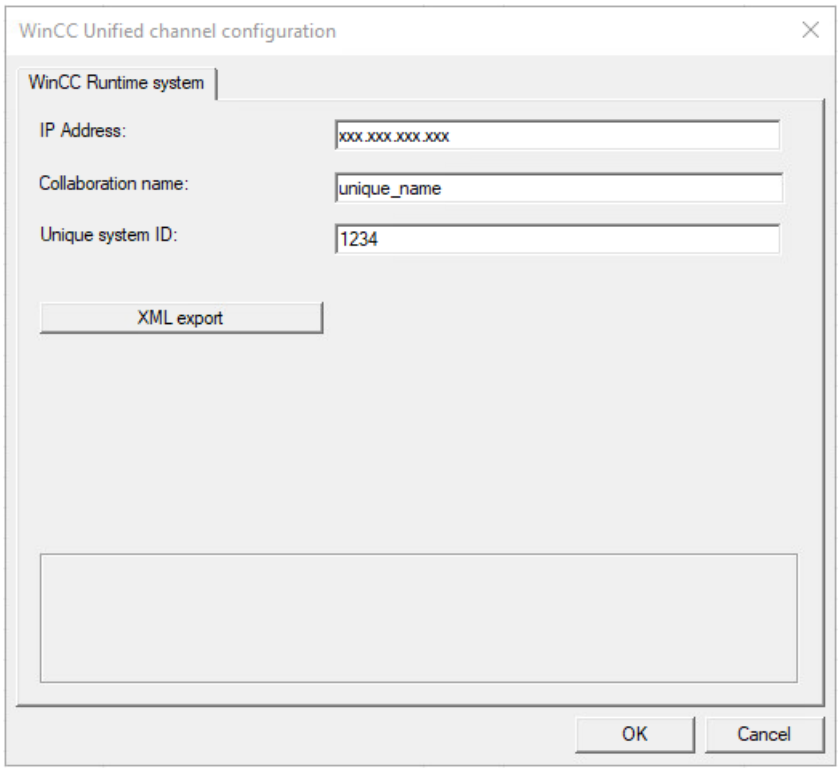
Requirements

- The communication driver for the "WinCC Unified Channel" is installed and integrated into the project.

Procedure

1. In the navigation area of the Tag Management, select the channel unit "Unified Tag Access" in the tree of the "WinCC Unified Channel" communication driver.
2. In the channel unit shortcut menu, select "System parameters".

3. The dialog for entering the system parameters opens.



4. Enter the IP address of the WinCC system.
5. Specify the unique "Collaboration name".
6. Specify the unique system ID.

20.3.3 How to configure a connection

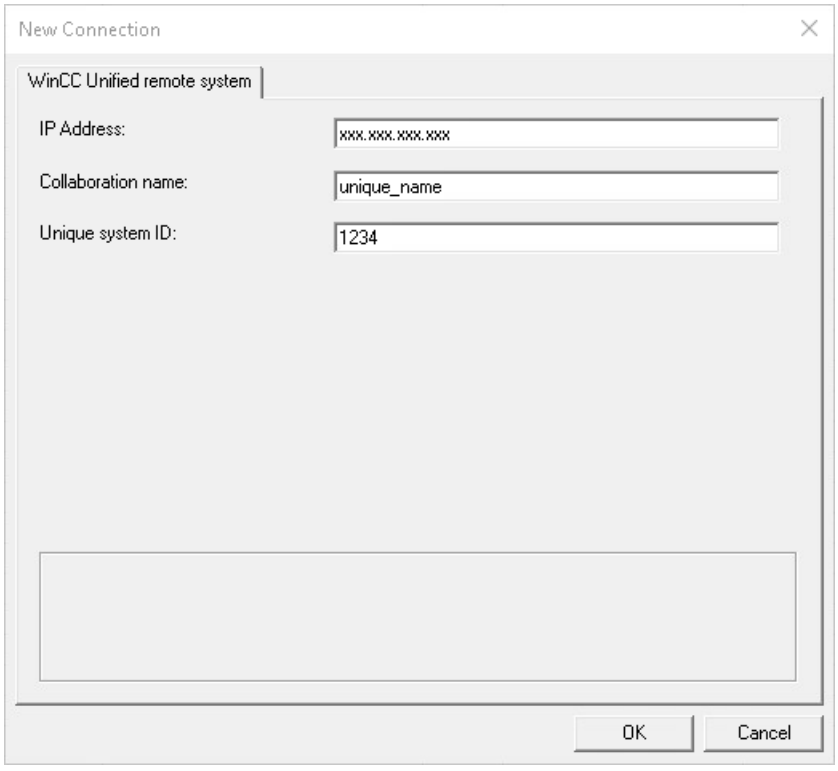
Requirements

- The communication driver for the "WinCC Unified Channel" is installed and integrated into the project.
- The channel is configured.

Procedure

1. In the navigation area of the Tag Management, select the channel unit "Unified Tag Access" in the tree of the "WinCC Unified Channel" communication driver.
2. Select the entry "New Connection" in the shortcut menu of the channel unit.
3. Enter the name of the connection.

4. Select the entry "Connection parameters" from the shortcut menu of the connection. The dialog for entering the connection parameters opens.



   The data entered here must match the data you entered in WinCC Unified under "Runtime Settings > Remote Access > Collaboration".

5. Enter the IP address of the HMI device.
6. Enter the unique collaboration name.
7. Specify the unique system ID.

Note

Unique names

The system ID and collaboration name must be unique throughout the system among all participating collaboration systems.

20.3.4 Configuring the tags

Collaboration allows you to share data with other systems.

Tags are read into WinCC by import. You use imported tags like other tags in pictures, controls or scripts.

The tags can be archived by WinCC.

Note

No structured tags can be used

Structure tags are read in flat without preserving the structure.

Version compatibility

You can import tags as of version WinCC Unified V17.

Session local tags

System tags configured on a Unified PC as session-local are not supported by the "WinCC Unified Channel".

20.3.4.1 How to configure tags

Introduction

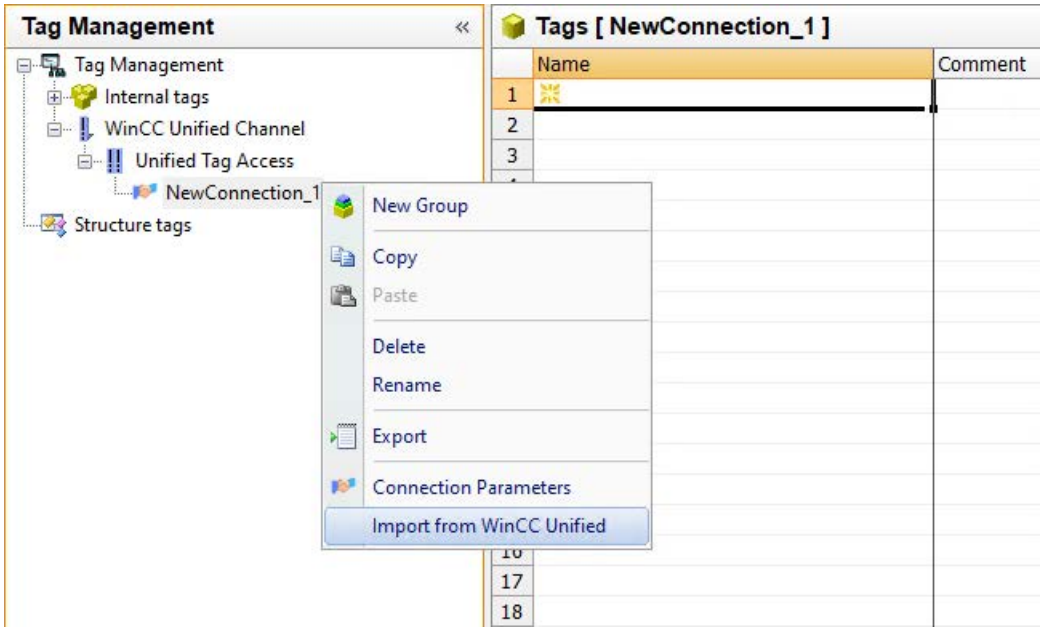
This section describes how to configure the sharing of tags as part of collaboration.

Requirements

- Collaboration is enabled and configured in WinCC Unified.
- The "WinCC Unified Channel" channel is configured and a connection created in WinCC.

Importing tags

1. In the shortcut menu of the connection, select the entry "Import from WinCC Unified".



The tags of the collaboration device configured in WinCC Unified are imported and are available in WinCC.

Creating tags

1. Select the connection for which a tag is to be configured.
2. Click the "Tags" tab below the table area.
3. Click in the top free cell of the "Name" column.
   Enter the name for the tag.
4. Select the desired data type in the "Data type" field.
5. In the "Type conversion" field, select one of the specified adaptations.
6. Specify the other tags of the property, for example, limit values in the "Properties - Tag" area.

Index

"

"Slot PLC" channel unit, 261
    Configuring the connection, 261

5

505 Ethernet (CP 1413-x)
    Channel unit, 371

A

Adding a tag, 77, 84
Adding new drivers, 193
Address properties, 88, 196, 197, 198, 203
    Example for S5 Profibus FDL, 211
Address type
    Mitsubishi Ethernet, 48
    Omron Ethernet IP, 67
Addresses of tags, 375
    TI Ethernet Layer 4, 375
AR_SEND function in WinCC
    Data block structure and parameters, 280
AR_SENDAR_SEND-FunctioninWinCC, 279
AR_SEND-Function in WinCC
    Configuring variants for an archive tag, 306
    Configuring Variants for Multiple Archive Tags, 308
    Example for Data Block Structure, 289, 290, 291, 296, 298, 300, 302, 305
    Overview of the properties of the variants, 285
    Variants for an Archive Tag, 288
    Variants for Multiple Archive Tags, 292
    Variants for Multiple Archive Tags (optimized), 304
AS symbol, 241, 248, 359
    AS structure, 245, 355, 358
    Export, 346
    Import, 241
    Importing with the Load Assistant, 346
    Manual import, 352
AS511, 213

B

Bar
    Example of System Info, 440
Bit by bit access, 170, 267
    S5 AS511, 220
    S5 Ethernet Layer 4, 170
    S5 Profibus FDL, 196
    S5 Serial 3964R, 229
    S7 Protocol Suite, 267
    TI Ethernet Layer 4, 377
    TI Serial, 391
Block, 208
    Cyclic block, 208
Byte array, 202
Byte by byte access, 171, 267
    S5 AS511, 221
    S5 Ethernet Layer 4, 171
    S5 Profibus FDL, 197
    S5 Serial 3964R, 231
    S7 Protocol Suite, 268
    TI Ethernet Layer 4, 378
    TI Serial, 392

C

Certificate, 112
Change-driven transfer, 273
Channel, 163, 223, 233
    Configuring SIMATIC S5 PROFIBUS FDL, 193
    Example for S5 PROFIBUS FDL, 208
    Mitsubishi Ethernet, 39
    Omron Ethernet IP, 63
    SIMATIC S5 Ethernet Layer 4, 163
    SIMATIC S5 Programmers Port AS511, 213
    SIMATIC S5 Serial 3964R, 223
    SIMATIC S7 Protocol Suite, 233
    SIMATIC S7-1200, S7-1500 channel, 329
    SIMATIC TI Ethernet Layer 4, 371
    TI Serial, 385
Channel unit, 193, 235
    505 Ethernet (CP 1413-x), 371
    Connection parameters, 373
    FDL (CP5412/A2-1), 193
    Industrial Ethernet, 250
    Mitsubishi Ethernet, 39
    MPI, 253

Named Connections, 255
Omron Ethernet IP, 63
PROFIBUS, 258
S5 Transport (CP1413-x), 166
S5 Transport (TCP/IP), 166
S5-AS511, 215
Selection for SIMATIC S7 Protocol Suite channel, 235
SIMATIC S7-1200, S7-1500 channel, 329
System parameters, 380
TCP/IP, 264
Channel Unit, 163
505 Serial Unit #1, 385
S5-AS511, 213
S5-RK512 (3964R), 223
Slot PLC, 261
Soft PLC, 263
System parameters, 175, 199, 272, 379
Channel unit FDL (CP5412/A2-1), 187, 193
Channel Unit Industrial Ethernet, 250
Configuring the connection, 251
Client certificate, 112
Communication type, 204
Configuring, 204
Communications drivers, (See channel)
Configure bit by bit access
Tag, 377
Configure byte by byte access
Tag, 378
Configuring
Tag, 375
TI Ethernet Layer 4, 373
Configuring a tag
TI Ethernet Layer 4, 375
Configuring a tag with bit by bit access
TI Ethernet Layer 4, 377
Configuring a tag with byte by byte access
TI Ethernet Layer 4, 378
Connection, 250
Active connection, 187
Configuring, 194, 251, 253, 256, 258, 261, 263, 265
Creating a connection, 77
Passive connection, 187
Connection Monitoring, 82
Connection parameters, 166, 193, 194, 204, 206, 250
Channel unit, 373
S5 AS511, 213, 215
S5 Ethernet Layer 4, 163
S7 Protocol Suite, 233
SIMATIC S5 Serial 3964R, 223
TI Ethernet Layer 4, 371, 373
TI Serial, 385
Connection properties, 194, 204, 206
Counter, 430
CP1434 TF, 380, 381
CPU load, 430
Example of System Info, 441
Cyclic Read Services, 273

D

Data area, 189
Data transmission, 204
Configuring a passive data transfer, 206
Configuring an active data transfer, 204
Data type, 81, 101, 239, 330, 430
Data types used, 430, 437
Mitsubishi Ethernet, 40, 48
Omron Ethernet IP, 64, 67
OPC UA, 101
S5 AS511, 214
S5 Profibus FDL, 189
S5 Serial 3964R, 224
S7 Protocol Suite, 239
SIMATIC S7-1200, S7-1500 Channel, 330
Supported WinCC data types, 81
Supported WinCC data types overview, 81
TI Ethernet Layer 4, 372
Data Types, 165
S5 Ethernet Layer 4, 165
TI Ethernet Layer 4, 372
Date, 430
Daylight saving time/standard time, 280
Device Name, 175, 199, 379
change, 200
TI Ethernet Layer 4, 379
Diagnosis
Channels, 407
Disk capacity, 430
Example of System Info, 440
Free disk capacity, 430
Free dive capacity, 435

E

Export tool, (See: Siemens SIMATIC SCADA Export)

F

Filter criteria, 84

I

I/O field
Example for S5 Profibus FDL, 211
Example of System Info, 438
Instance certificate, 112

L

Life beat monitoring for channel S5Seral3964R
Life beat monitoring, 225
Logical device name
Changing, 276
Logical Device Name, 272

M

Media redundancy, 362
MELSEC FX3U series, (See Mitsubishi Ethernet)
MELSEC iQ-R series, (See Mitsubishi Ethernet)
MELSEC Q series, (See Mitsubishi Ethernet)
Message
Example of System Info, 442, 444
Import, 241
Importing with the Load Assistant, 346
Manual import, 352
Message text, 442
Mitsubishi Ethernet, 39
Address types, 48
Channel units, 41, 42, 44, 45, 47
Configuring a tag, 41, 48, 50
Configuring connections, 41, 42, 44, 45, 47
Data types, 40, 48
Supported controllers, 41
Tag, 50
MPI channel unit, 253
Configuring the connection, 253
Multi-user system, 449
Use of channel, 449

N

Named Connections channel unit, 255
Configuring the connection, 256
NAT router, 111
New connection, 84

O

Offline configuration, 241, 345
Load Assistant, 346
Manual, 352
Omron Ethernet IP, 63
Address types, 67
Channel units, 65, 66
Configuring a tag, 65, 67, 68
Configuring connections, 65, 66
Data types, 64, 67
Supported controllers, 65
Tag, 68
OPC
Access to tag, 83
Adding a tag, 77, 84
Address properties, 88
Configuring access to a WinCC tag with the OPC Item Manager, 84
Configuring access to a WinCC tag without the OPC Item Manager, 88
Configuring an OPC channel on a WinCC OPC DA client, 84
Configuring structure, 90
Configuring structure tag, 90
Creating a connection, 77
Error handling in event of disturbed OPC communication, 93, 156
Filter criteria, 84
Functionality of the WinCC OPC DA Client, 82
New connection, 84
OPC in WinCC, 75
OPC Item Manager, 77
OPC UA, 99
OPC1, 88
OPCServer.WinCC-(DPC_4001), 84
Overview of OPC Item Manager functions, 77
Querying of ProgID, 77
Structure properties, 90
Supported WinCC data types, 81
Supported WinCC data types overview, 81
Tag selection, 77
Using structures on a WinCC OPC DA client, 90, 92
WinCC as an OPC DA client, 75, 93
WinCC as OPC UA client, 156
WinCC as OPC XML client, 75
OPC channel
Configuring an OPC channel on a WinCC OPC DA client, 84

OPC communication, 93, 156
Error handling in the event of disturbed OPC DA communication, 93
Handling errors in OPC UA communication, 156
OPC DA client, 82
Functionality of the WinCC OPC DA Client, 82
OPC Item Manager, 77, 84
Overview of OPC Item Manager functions, 77
OPC UA
Access to tag, 104
Data type, 101
WinCC as OPC UA client, 99
OPC UA client, 99
OPC1, 88
OPCServer.WinCC-(DPC_4001), 84

P

Point of error, 442
Printer monitoring, 430
Example of System Info, 445
Process picture
Example of System Info, 453
PROFIBUS channel unit, 258
Configuring the connection, 258
ProgID, 77
Querying of ProgID, 77
Programmable controller, 191

R

Raw data tag
as byte array, 311
Configuring as a byte array, 312
for BSEND/BRCV functions, 314
Raw Data Tag, 202, 310
as byte array, 202
Configuring, 203
Configuring for BSEND/BRCV functions, 317
Redundancy
CPU redundancy, 360
redundant PLC, 360, 363
Software redundancy, 366

S

S5 AS511, 215
Configuring, 215
Configuring a tag, 217
Configuring a tag with bit by bit access, 220
Data Types, 214
Defining a tag with byte by byte access, 221
Tag address, 217
S5 Ethernet Layer 4, 163
Configuring, 166
Configuring a tag, 168
Configuring a tag with bit by bit access, 170
Configuring a tag with byte by byte access, 171
Configuring a tag with word by word access, 172
Connection parameters, 166
Data Types, 165
Device Name, 175, 176
Error codes during connection disturbances, 179
iNA960 messages, 182
Internal error codes and constants, 179
Raw Data Tag, 174
SCI messages, 183
System parameters of the channel unit, 175
Tag address, 169
Transport Parameter, 175
Transport Parameters, 178
S5 Profibus FDL, 187
Access to a S5 tag of data type ARRAY OF BYTE, 189
Access to a S5 tag of data type BIT, 189
Access to a S5 tag of data type BYTE, 189
Access to S5 tags, 189
Active connection, 187
Adding new drivers, 193
Address properties, 196, 197, 198, 203
Block, 208
Changing the device name, 200
Changing write/read monitoring time, 201
Channel unit FDL (CP5412/A2-1), 187
Configuration of data handling block, 208
Configuring a passive data transfer, 206
Configuring a raw data tag, 203
Configuring a tag, 196
Configuring a tag with bit by bit access, 196
Configuring a tag with byte by byte access, 197
Configuring a tag with word by word access, 198
Configuring an active data transfer, 204
Configuring channel SIMATIC S5 Profibus FDL, 193
Configuring SIMATIC S5 Profibus FDL, 193
Configuring the communication type, 204
Configuring the connection, 194
Connection parameters, 193, 194, 204, 206
Connection properties, 194, 204, 206
Cyclic block, 208
Data area, 189
Data type, 189
Device Name, 199
Example, 208, 211

Particularities of the channel, 191
Passive connection, 187
Raw data tag as byte array, 202
Service Access Point, 187
Special function, 202
Startup block, 208
supported data type, 189
System parameters, 200, 201
System parameters of the channel unit, 199
Tag parameters, 193
Tag Properties, 196, 197, 198, 203
Type conversion, 191
Write/Read monitoring time, 199
S5 Serial 3964R, 223
Configuring, 225
Configuring a tag, 227
Configuring a tag with bit by bit access, 229
Configuring a tag with byte by byte access, 231
Data type of the tags, 224
S5 tag, 189
Access to a S5 tag of data type ARRAY OF BYTE, 189
Access to a S5 tag of data type BIT, 189
Access to a tag of data type byte, 189
Access to S5 tags, 189
S7 Plus, (See: SIMATIC S7-1200, S7-1500 channel)
S7 Protocol Suite, 233
AR_SEND variant for an archive tag, 288
AR_SEND variant for multiple archive tags, 292
AR_SEND variant for multiple archive tags (optimized), 304
Changing logical device names, 276
Channel unit, 235
Channel Unit, 250
Configuring, 240
Configuring a raw data tag as a byte array, 312
Configuring a tag with bit by bit access, 267
Configuring a tag with byte by byte access, 268
Configuring a tag with word by word access, 270
Configuring a text tag, 271
Configuring an AR_SEND variant for an archive tag, 306
Configuring an AR_SEND variant for multiple archive tags, 308
Configuring software redundancy, 324
Configuring system parameters, 274
Configuring the connection, 251, 253, 256, 258, 261, 263, 265
Configuring the raw data tag for the BSEND/BRCV functions, 317
Cyclic Read Services, 273
Data block structure and parameters in AR_SEND function, 280
Data exchange with FB S7ProtocolSuite:AR_SEND variants, 279
Data type, 239
Delete Software Redundancy in WinCC, 327
Error codes for connection fault, 328
Example for Data Block Structure AR_SEND, 289, 290, 291, 296, 298, 300, 302, 305
Overview of the properties of the AR_SEND variants, 285
Raw Data Tag, 310
Raw data tag as byte array, 311
Raw data tag for BSEND/BRCV functions, 314
Software redundancy, 319
Software Redundancy - Connection-specific internal tags, 320
supported data type, 239
System parameters, 272
S7 Protocol Suite"; "Selection of channel unit, 235
S7+, (See: SIMATIC S7-1200, S7-1500 channel)
S7-1200 controller, 329
S7-1500 controller, 329
S7-1500 R/H-CPU, 360, 363, 366
SCADA Export, 242, 346
Server certificate, 112
Service Access Point, 187
Siemens SIMATIC SCADA Export, 242
SIEMENS SIMATIC SCADA Export, 346
SIMATIC S5 Profibus FDL, 187
Raw Data Tag, 202
S5 Profibus FDL, 187
Special function, 202
SIMATIC S5 Programmers Port AS511, 213
SIMATIC S5 Serial 3964R, 223, 225
SIMATIC S7 Protocol Suite, 233
SIMATIC S7-1200, S7-1500 channel, 329
Import, 241, 352
Importing with the Load Assistant, 346
SIMATIC S7-1200, S7-1500 Channel
Configuration, 331, 333, 334
Configuring the connection, 334
Data type, 330
SIMATIC SCADA Export, 242, 346
SIMATIC TI Ethernet Layer 4, 371
Channel, 371
Transport Parameter, 379
SIMOTION channel, 395
Configuring, 397, 398, 399, 401, 402
Data types, 396
Diagnosis, 407
System parameters, 403, 405

Sinumerik NC
Channel units, 413
Configuring connections, 413
Soft PLC channel unit, 263
Configuring the connection, 263
Software redundancy, 319, 366
Software Redundancy
configure in WinCC, 324
Connection-specific internal tags, 320
delete in WinCC, 327
Special function, 202
Startup block, 208
Status display
Example of System Info, 445
Structure
Configuring structure, 90
Using structures on a WinCC OPC DA client, 90
Using WinCC OPC client, 92
Structure properties, 90
Structure tag, 90
AS structure, 245, 355, 358
Configuring structure tag, 90
Structure type
AS structure, 245, 355, 358
Supported system information, 430
Overview, 430
System Info, 428
Bar configuration, 440
Configuring, 436
Data types used, 437, 450
Differences to other software components, 435
Example, 437, 444, 450, 451, 453, 454
Example:, 438, 440, 441, 442, 445, 452
Free disk capacity, 430
I/O field configuration, 438
Message text, 442
Overview of supported system information, 430
Overview of the supported system information, 430
Point of error, 442
Status display configuration, 445
Tag to be monitored, 442
System Info System information
Use in multi-user system, 449
System parameters, 175, 272, 403, 405
Channel unit, 380
Channel Unit, 379
Configuring, 274, 276
S5 Ethernet Layer 4, 163
S5 Profibus FDL, 187
SIMATIC S7 Protocol Suite, 233
TI Ethernet Layer 4, 371, 379, 380, 381

T

Tag, 168, 437
Adding a tag, 77
Configure bit by bit access, 170, 196, 229, 267, 377, 391
Configure byte by byte access, 171, 197, 231, 268, 378, 392
Configuring, 168, 196, 375, 389
Configuring OPC Item Manager, 84, 88
Configuring tag addresses, 228
Configuring word by word access, 172, 198, 270
Example of System Info, 437, 452
Export, 248, 359
Import, 241
Import filter, 347
Importing with the Load Assistant, 346
Manual import, 352
OPC Item Manager, 84
Tag selection, 77
Tag address, 168
S5 Ethernet Layer 4, 163
TI Serial, 385
Tag Properties, 196, 197, 198, 203
Example for S5 Profibus FDL, 211
Tag to be monitored, 442
TCP/IP channel unit, 264
Configuring the connection, 265
Text Tag, 267
Configuring, 271
TI Ethernet Layer 4, 371
Addresses of tags, 375
Bit by bit access, 377
Byte by byte access, 378
Configuring, 373
Configuring a tag, 375
Configuring a tag with bit by bit access, 377
Configuring a tag with byte by byte access, 378
Connection parameters, 371, 373
Data type, 372
Data Types, 372
Device Name, 379
System parameters, 371, 379, 380, 381
Transport Parameter, 379
TI Serial, 385
Addresses of tags, 389
Configuring, 387
Configuring a tag, 389
Configuring a tag with bit by bit access, 391
Configuring a tag with byte by byte access, 392
Connection parameters, 387

Data Types, 386
Tag address, 389
TIA Portal
AS structure, 245, 355, 358
Import, 241
Importing with the Load Assistant, 346
Tag import filter, 348
Time, 430, 435
Example of System Info, 438
Time stamp, 280
Timer, 430
Transport Parameter, 175
SIMATIC S5 Ethernet Layer 4, 175
SIMATIC TI Ethernet Layer 4, 379
TI Ethernet Layer 4, 379
TTY interface, 213
Type conversion, 191

V

Variable, 267

W

WinAC Basis, 263
WinAC Pro, 261
WinCC
as OPC DA client, 75, 93
as OPC UA client, 99, 156
as OPC XML client, 75
OPC in WinCC, 75, 93, 156
OPC UA in WinCC, 99
Word by word access, 267
S7 Protocol Suite, 270
Write/Read monitoring time, 199, 201
change, 201