SIEMENS

SIMATIC HMI

WinCC V8.0 WinCC: Configurations and Communication

System Manual

Print of the Online Help

03/2023

A5E52671436-AA

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 AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

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

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E52671436-AA
Ⓟ 03/2023 Subject to change

Table of contents

Multi-User Systems 1

1.1 Multi-user systems in WinCC

Content

WinCC can be used to configure client/server systems, thus allowing for system operation and monitoring features to be distributed to several clients and servers. In the case of large-scale systems, therefore, the loads applied to individual computers can be reduced and the performance increased.

This chapter shows you:

- Client/server scenarios that can be realized using WinCC.
- How to configure the server and clients in the client/server system.
- How to configure clients which provide views on several servers.
- How the client/server system behaves in runtime.
- How to configure a server project from a remote client.

1.2 Client/Server Systems in WinCC

Introduction

WinCC can be used to configure client/server systems having several clients and servers, allowing for large systems to be operated and monitored more efficiently.

By distributing the tasks for operating and monitoring processes amongst several servers, the utilization rate of each individual server is reduced, thus increasing the performance.
In addition, it is also possible to map systems which have a technologically or topologically complex structure using WinCC.

Client/server systems are used, for example:

- Large-scale systems where several operator and monitoring stations (clients) are required for the same tasks.
- If you want to distribute operator control and monitoring tasks over several operator stations, such as a central client for displaying all messages of a system.

Clients can be used for:

- Multi-user systems with one server for configuration:
  several clients access the project located on a server with process driver connection.
  In multi-user systems, it is not necessary to configure the clients as all data is provided by the server.
- Distributed systems with several servers to be configured:
  clients can access data on different servers with process driver connection.
  Within a distributed system, each client is individually configured, the necessary server data is imported on the clients and, if modified, can be automatically updated.
  The process data is provided by the servers.
- Remote configuration:
  a server project is configured from a client.
- Remote monitoring:
  a server project is monitored from a client.

Requirements for Configuring Client/Server Systems

- In order to configure client/server systems with WinCC, the "WinCC Server" option must be available on every WinCC server.
- The same WinCC version must be installed on all WinCC stations used in the WinCC system. These include, for example:
  - WinCC servers, redundant servers, archive servers, file servers
  - WinCC clients with their own project, WinCC clients without their own project
  - PCs on which WinCC options are installed, e.g. WebNavigator server, WebNavigator client, WinCC/Audit, WinCC/Calendar Options

  Mixed configurations with different WinCC versions on servers, clients or PCs with WinCC options are not released.
  Always install WinCC updates on all PCs in the WinCC system. If different update versions are installed on the clients or servers, no configuration changes are possible in the WinCC project, for example.
- All PCs in a client/server system must be connected to each other via a network (LAN).
  It is also possible to log on PCs in neighboring subnets, which are connected via a router, as clients or servers in the system.
- Remote access is disabled by default.
  To enable network access to the PC, activate the remote communication in the Simatic Shell settings.

Redundant Systems

To maintain system operation even in the case of faults, e.g. following failure of a server, configure redundant servers. For detailed instructions for the configuration of redundant systems, see the WinCC documentation "Redundant Systems".

The documentation contains information on the configuration of clients in redundant systems.

Client on a redundant server pair

- A WinCC project containing a client without local project may only be edited on a server (primary server or standby).
  No WinCC project may be opened on the partner server during this time. WinCC Explorer and Runtime must be closed.
- You can still configure the primary server or standby server as the preferred server for load distribution in Runtime.
- WinCC Explorer should remain closed on the clients while Runtime is activated.
  It is best to start Runtime using "Autostart". To do this, use the "AutoStart Configuration" tool of WinCC.
- Operator input is locked (user interface is "grayed-out") if WinCC Explorer remains open and you switch to another server computer.
  The WinCC editors cannot be started as long as the server from which the project was opened is not the current server of the client.

Note

Remote access to open WinCC project

Remote configuration is not permitted, if the project is open on both redundant systems.

In order to configure a redundant system, the standby computer must not have a WinCC project open.

In order to configure a redundant system in runtime, you must proceed as follows:

1. Deactivate the standby computer and close the project.
2. Configure the primary server either in Runtime, remotely or locally.
3. After completion of the online configuration, duplicate the project on the standby computer with the Project Duplicator in Runtime.
4. Open and activate the project again on the standby computer.

See also

Use of the OPC Interface in Client/Server Systems (Page 88)

Remote Configuration (Page 73)

System Behavior in Runtime (Page 65)

Client Configuration (Page 42)

Server Configuration (Page 27)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Typical configurations (Page 19)

Encrypted communication (Page 74)

1.3 Typical configurations

You can configure different client/server solutions as needed. In so doing, you have the option of using clients, web clients and thin clients.

Observe also the notes on quantity structures and performance in "Quantity structures and performance (Page 21)".

Using encrypted communication in distributed systems

Set up encrypted communication between computers in a multi-user system.

Specify PSK keys for the computers with "Simatic Shell". This means that only those computers in which the shared specified key is known prior to communication can communicate with each other. You can specify different environments with their own PSK keys for the same network.

Configuration of Client/Server Systems for Different Requirements

Clients

Depending on the configuration, clients in a client/server system can:

- Represent a view from a server on several clients (multi-user system)
- Display views of several servers on clients (distributed system)
- Configure a server project (remote) from a client
- Activate and deactivate a server project (remote) from a client

To configure several clients the "WinCC Server" option is required on each server.

WebClients

WebClients are installed in a client/server system, for example, when:

- Access to the system is required via narrowband connections
- Only temporary access to data is required
- Data access is necessary over large distances, e.g. via Internet

WebClients have the following advantages:

- Client computers with different operating systems can be implemented
- Simultaneous access to a server by several WebClients is possible
- Large quantity structures can be realized

In order to configure WebClients, a minimal installation of WinCC is required and the "WinCC WebNavigator" option.

ThinClients

ThinClients basically have the same main features as WebClients and the additional features:

- They can also be used on rugged client platforms

In order to configure ThinClients, the minimal installation of WinCC is required and the "WinCC WebNavigator" option.

See also

Client/server scenarios (Page 24)

Using Data from Different Servers (Page 59)

Displaying Pictures from Different Servers (Page 56)

How to Configure a Preferred Server (Page 51)

How to configure a standard server (Page 49)

Configuring the Import Package (Page 46)

Displaying Messages from Different Servers (Page 62)

Creating a New Project on the Client (Page 44)

Client Configuration (Page 42)

Creating a new project on the server (Page 29)

How to Register Clients in the Computer List (Page 31)

How to Configure the Package Export (Page 35)

Configuring Clients in the Server Project (Page 39)

Client/Server Systems in WinCC (Page 16)

Quantity structures and performance (Page 21)

Encrypted communication (Page 74)

1.4 Quantity structures and performance

Performance of WinCC projects in runtime.

The performance of a WinCC project is dependent on the system configuration, the quantity structures and the configuration. Each of these factors can influence things like the time it takes to change pictures and the archiving speed.

You can find information on the configuration of your system in "Typical configurations (Page 19)". Observe also the hardware requirements in the Installation Notes.

Notes on configuration

The following notes apply not just to multi-user systems but to every type of WinCC project.

However, you should take these notes into account, in particular, in multi-user systems because the quantity structures grow quickly there and configuration changes can have a stronger effect.

Note when configuring that the following factors affect the performance of a WinCC project:

- Number of web clients
  The number of tags to be processed in the project grows with each additional web client used: For each web client, the internal tags of the WinCC project are managed on the web server as tags that are local to the computer. Depending on the configuration, additional web clients multiply the number of managed tags.
- Number of picture windows in a process window
  It is recommended that a maximum of 100 simultaneously displayed picture windows be used.
- Number of nestings in the picture windows (levels)
  10 levels are permitted as the maximum nesting depth of picture objects.
- Scripts in process screens
  Ensure that the processing time of all scripts with the same cycle is not longer than the configured cycle time.
- Number of tags
  The server load in WinCC Runtime arises from the sum of all tags that are simultaneously registered on all clients or web clients.
- Number of monitors (Multi-VGA)
  If multi-VGA is used, the number of WinCC clients may be reduced.
  In this case also note potential performance limitations caused by the number and complexity of the process pictures.
  We recommend limiting the number of monitors to 60.
  Example: If you use four monitors each on all WinCC clients, a maximum of 15 WinCC clients are supported.

To analyze the performance of the WinCC project when reading, writing and archiving data, use the "@PRF_..." system tags.

Simultaneous start of several clients on a server

The simultaneous start of several clients connected to a server may lead to an overload situation. In this case, the clients go into timeout.

We recommend starting the clients in succession.

Possible numbers of clients and servers

Different quantity structures can be realized according to the type and number of client types used. Mixed systems are possible, meaning the parallel use of clients and web clients within a client/server system.

If you use only WinCC clients with a custom project, up to 50 parallel clients can access a server in a WinCC network. A WinCC client can access up to 18 servers in Runtime.

You can use a maximum of 36 servers in the form of 18 redundant server pairs.

If you use only web clients, you achieve quantity structures of up to 151 clients (1 client and 150 web clients).

Also take into consideration the effect of multi-VGA on the number of clients.

Configuration of mixed systems

When configuring a mixed system, observe the following rule of thumb to achieve the maximum quantity structures.

The following values are defined for the client types:

- Web client/thin client = 1
- Client = 2
- Client with "Configure remote" function = 4

The total of values of all clients should not exceed following values:

- WinCC server without operator function: 160 per server
- WinCC server with operator function: 16 per server

Example:

Configuration	Significance
3 Clients with the "Configure remote" function	3 x 4 = 12
5 Clients	5 x 2 = 10
138 web clients	138 x 1 = 138
Sum	**160**

Note

No mixed configuration with WinCC servers

The mixed configuration of WinCC servers that access other WinCC servers as clients is not approved.

See also

Client/Server Systems in WinCC (Page 16)

Use of the OPC Interface in Client/Server Systems (Page 88)

Remote Configuration (Page 73)

System Behavior in Runtime (Page 65)

Client Configuration (Page 42)

Server Configuration (Page 27)

Client/server scenarios (Page 24)

Typical configurations (Page 19)

1.5 Client/server scenarios

Introduction

Using WinCC you can implement different client/server scenarios depending on the application:

Multi-user systems

A multi-user system is typically configured for smaller systems in which a distribution of data to several servers is required.

A server with process driver connection is configured which is then responsible for all central functions and several operating stations (clients).

The individual operator stations can execute the same or different tasks.

WinCC clients: Behavior in Runtime

Depending on the operator authorization, clients have different functions:

- Only monitor the system.
- Both monitor and operate the system.
- Configure the server project remotely, e.g. as a maintenance computer.

The display on the clients depends on the configuration:

- All clients show the same view of the project when, for example, the process is to be operable from several points in the system.
- Clients display different views of the project, such as only messages or only process values.

User authorization is issued to define the functions that are available to an operator on a certain operating console.

Types of configuration

- WinCC clients without their own project:
  If the operator stations do not need their own project data, you configure the clients in the server project.
- WinCC clients with their own project:
  If the operator stations need their own project data, e.g. different start screens, you create client projects.

Installation

You can install the "WinCC Client" on both client types.

For this you need one of these two client licenses:

- "RT Client"
- "RC Client"

Installation of Microsoft SQL-Express is a requirement.

More information in the WinCC Information System: "Licensing > Licensing overview"

Distributed systems

Distributed systems implementing several servers are generally used in the case of large systems when particularly large quantities of data must be processed.

As a result of distributing tasks over several servers, the load applied to individual servers is relieved. This achieves a better system performance realize larger typical applications.

If distributed systems are configured in a WinCC system, the process tasks are distributed amongst the servers by means of the corresponding configuration according either to the process steps or functionally:

- In the case of a technological distribution, each server takes over a technically limited area of a system, e.g. a certain press or dry unit.
- In the case of a functional distribution, each server takes over a certain task, e.g. visualization, archiving, issuing alarms.

In Runtime, the clients in a distributed system can each display the data from up to 18 different servers or from redundant pairs of servers.

Each client in a distributed system is configured individually with basic pictures and a little local data.

The server data required for displaying the process data is transferred from the servers to the clients and can be updated automatically, if necessary.

File server

You can use a file server for Client-Server Systems in order to save all projects and administer them in a centralized system. This makes it easier to create periodic backup copies of all projects, for example.

The servers with process driver connection have access to the file server and can configure the projects on the file server.

The file server can be used for configuration only.

The file server can be adapted to specific demands, as necessary, by adding further hardware components. For example, you can ensure reliability with mirrored disks.

Central Archive Server

You can save process values and messages of all connected WinCC servers on a centralized archive server (for example, Process Historian).

You can display the saved process values and messages as usual in runtime in WinCC OnlineTrendControl or WinCC AlarmControl in the process picture.

Furthermore, you have direct access to archived process values and messages using defined interfaces, such as OLE DB. In this way, you can make important production data available throughout the company for analyzing purposes, for example.

Server-server communication

During communication between two servers, one server accesses data on another server.

One server can access the data from up to 18 other servers or redundant pairs of servers.

The server accessing the data behaves as a client in respect of the configuration and operation, except that a standard server cannot be configured.

Each server making access requires a WinCC server license.

When the system is being configured, the accessing server must be included in the quantity structure.

Standard server

In distributed systems, your data is allocated server prefixes from specific servers so that WinCC controls can display messages and process data.

In a distributed system, a standard server is configured for clients so that data, to which no unique prefix has been specified, can be requested from the standard server.

If no standard server is defined, an attempt is made to access the corresponding data locally. If there is no local data management (e.g. messages and archives), access is rejected and an error message issued.

Preferred server

If you use redundant servers in your multi-user system or distributed system, configure a preferred server among the clients.

A preferred server is the server in a redundant server pair which has priority for a client in a multi-user system.

The preferred server can be selected separately for each client in order to ensure the operability of the system.

See also

How to Configure a Preferred Server (Page 51)

How to configure a standard server (Page 49)

System Behavior in Runtime (Page 65)

Client Configuration (Page 42)

Server Configuration (Page 27)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

Typical configurations (Page 19)

1.6 Server Configuration

1.6.1 Server Configuration

Introduction

A server in a WinCC network can perform the following tasks:

- Connection to the process
- Acquisition of process values
- Archiving of alarms and process values
- Supplying the clients with data from the process
- Supplying the clients with configuration data

The tasks that individual servers perform can be distributed according to technological or functional aspects:

- Technological: Each server manages a specific section of the process/plant.
- Functional: Each server performs a specific task in runtime which is related to the entire process, e.g. the message system or archiving.

Note

Always configure the servers of a client-server system before the associated clients.

Configuration steps

Configuring a multi-user system

In a multi-user system, you can configure several clients that display the view of a server in runtime.

These "WinCC clients without their own project" receive their data exclusively from the server and do not have their own configuration.

To configure a server in a multi-user system, the following steps are required:

1. Create a new "Multi-user project" type on the server.
2. Configure the required project data (pictures, archives, tags ...) on the server.
3. Include clients that are to be configured or monitored remotely in the computer list on the server.
4. Assign operator authorizations for the clients that are to be configured remotely ("Configure remotely").
5. Activate automatic package import on the server.
6. Configure the properties of the clients in the server project (start picture, disable key combinations ...).

Configuring a distributed system

In a distributed system, you can configure clients with views on multiple servers.

These "WinCC clients with their own project" have their own projects with their own local data. Data updated from the server is transferred to the clients via the package export.

To configure a server in a distributed system, the following steps are required:

1. Create a new project of type "Multi-user project" on each server.
2. Configure the required project data (pictures, archives, variables ...) on the servers. Depending on the division (technological/functional), this can only be specific project data, e.g. only archives.
3. Include clients that are to be configured remotely in the computer list on the server.
4. Assign operator authorizations for the clients that are to be configured remotely.
5. Configure package export (manually or automatically).
6. Configure client projects on the clients.
7. Make server data (packages) available to clients.

See also

How to Configure a Preferred Server (Page 51)

Configuring Clients in the Server Project (Page 39)

How to Configure the Package Export (Page 35)

How to Configure Operator Authorization (Page 32)

How to Register Clients in the Computer List (Page 31)

Creating a new project on the server (Page 29)

Client Configuration (Page 42)

Client/server scenarios (Page 24)

Client/Server Systems in WinCC (Page 16)

1.6.2 Creating a new project on the server

Introduction

When creating a new project in WinCC, you can select from the project types below:

- Single-user project:
  A project for a standalone operating station which carries out all tasks in runtime (process driver connection, operating, monitoring, archiving, etc.).
  Not relevant for client/server systems.
- Multi-user project:
  A server project for a multi-user system or distributed system in which several clients and/or servers are configured.
- Client project:
  A project for one client within a distributed system which can display views on several servers.

Note

Changing the project type

An existing project can also be converted to a server project by modifying the project type later.

You change the settings under "General" in the "Properties - Project" area of the "Computer" editor.

Setting the port

For communication between server and client, the operating system dynamically selects a port in the range from 1024 to 65535.

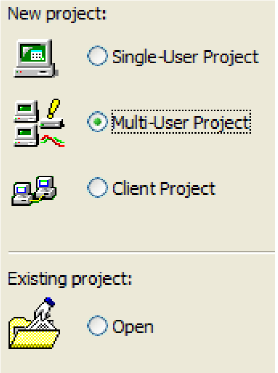
You can also specify a specific port in the communication settings of Simatic Shell.

Activate "Encrypted communication" and enter the desired port.

Procedure

The following procedure describes how to create a server project for a multi-user system or distributed system:

1. On a server in WinCC Explorer, select the menu command "File > New".
   The "WinCC Explorer" dialog opens:



New project:

○ Single-User Project

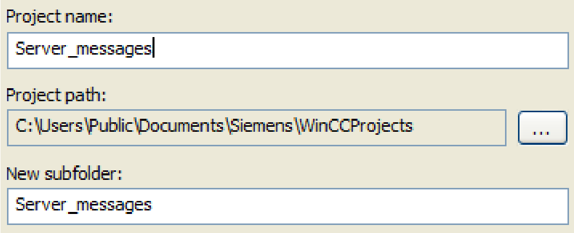
● Multi-User Project

○ Client Project

Existing project:

○ Open

2. Select "Multi-User Project" and click "OK".
   The "Create New Project" dialog appears.
3. Enter a project name and the name of a subdirectory if the directory name should be different from that of the project.
   By default, the following folder is used as the project path:
   - "Public documents\Siemens\WinCCProjects"



Project name: Server_messages

Project path: C:\Users\Public\Documents\Siemens\WinCCProjects

New subfolder: Server_messages

4. Click the "Create" button.
   The project is created and opened in WinCC Explorer.
   The current project is automatically a server project.

See also

Configuring Clients in the Server Project (Page 39)

How to Configure the Package Export (Page 35)

How to Configure Operator Authorization (Page 32)

How to Register Clients in the Computer List (Page 31)

Server Configuration (Page 27)

Client Configuration (Page 42)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

Typical configurations (Page 19)

1.6.3 How to Register Clients in the Computer List

Introduction

If you wish a client to access a server remotely or in runtime, this client must be registered in the computer list of the server.

Requirement

A multi-user project is created as multi-user system or distributed system.

The configuration computer is automatically a server in the client/server system.

Procedure

1. Open the "Computer" editor in the WinCC Configuration Studio.
2. Select the "Project" entry in the navigation area.
3. Click on the first empty box in the "Name" column on the "Computers" tab.
4. Enter the name of the client computer to be granted access to the current server.
5. Confirm the "Full access" authorization to the project directory.
   The computer is added to the list of computers registered in the project.
   If the client computer can be reached via the network, you can configure the autostart settings of the WinCC client under "Local settings".
6. Add all computers to be granted access to the current server in your client/server system.

Note

WinCC client: Changing computer name

To rename a client computer, delete the selected client computer from the list of computers.

Include the new client computer with the modified name as new computer in the computer list.

See also

Server Configuration (Page 27)

Configuring Clients in the Server Project (Page 39)

How to Configure the Package Export (Page 35)

How to Configure Operator Authorization (Page 32)

Creating a new project on the server (Page 29)

Client Configuration (Page 42)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

Typical configurations (Page 19)

1.6.4 How to Configure Operator Authorization

Operator Authorization in WinCC

In order for a client to open and process a server project remotely or in Runtime, you must configure the appropriate client operator authorizations in the server project.

The following operator authorizations are available on the server for this purpose:

- "Remote configuration":
  The can open a server project from a remote station and has full access to the project.
- "Remote activation":
  The client can place a server project in runtime.
- "Web Access - monitoring only":
  The Web client is authorized to monitor the plant.
  Such an operator authorization is not relevant for the configuration of other clients.

Configuration on the client

If a client has the authorization to configure a server project, the operator authorization can also be changed in the server project from the client.

The computers in the network are not notified when an operator authorization is changed.

The change takes effect when a new client logs on to a server.

Behavior

The operator authorization is requested on the client as soon as the client opens, activates or deactivates a project on the corresponding sever.

If the corresponding operator authorization is not available on the server, the project cannot be processed.

When the server project has been closed on the client, logging in is requested again when opened again.

Note

Operator authorization is linked to the user

The operator authorizations configured are user-related, not computer-related.

An operator authorization assigned is therefore valid for all operating stations with the same login.

Operator authorizations in the operating system

In order for clients to access the server project, the corresponding project folder must be enabled for network access on the server:

1. Open the "Computer" editor in the WinCC Configuration Studio.
2. In the "Properties - Project" area, deactivate the following option under "Options":
   – "Project directory is only shared for write-protected access."
3. Set up authorizations in the operating system with all rights required for the users who should have access to the projects.

Note

Windows operator authorizations

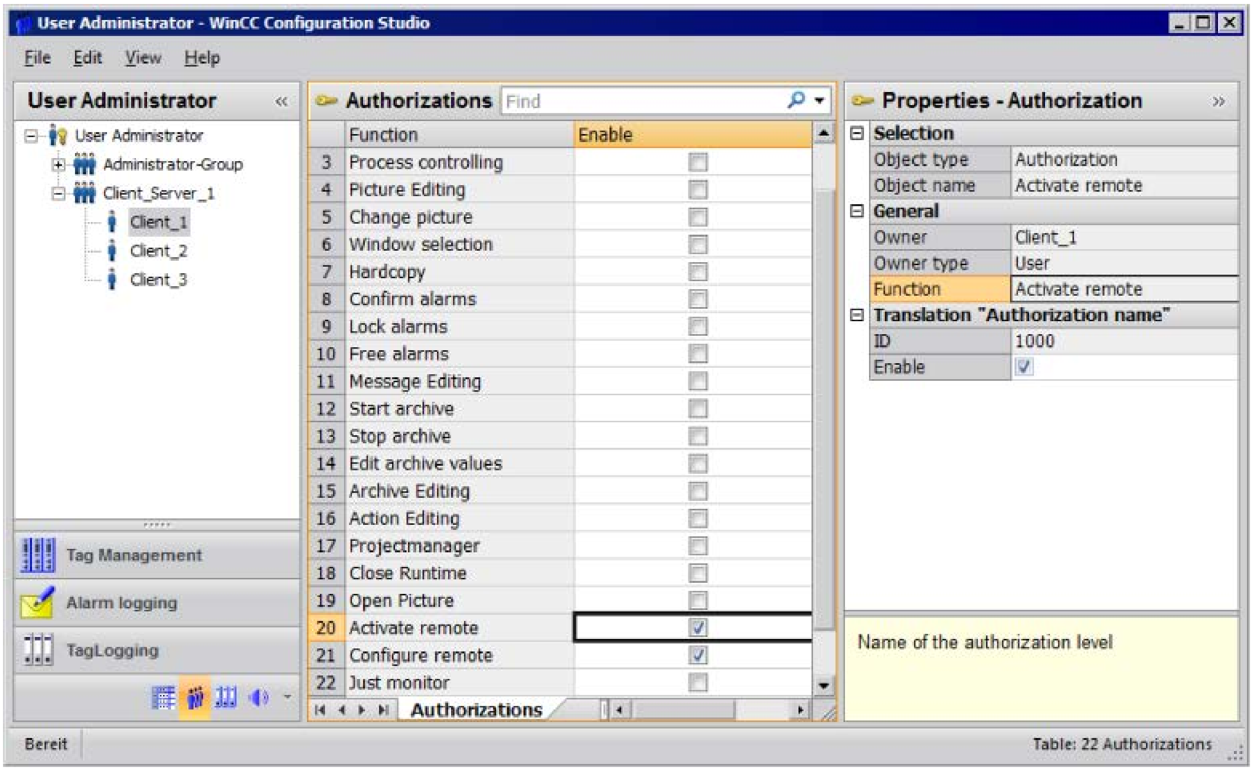
With regard to network security, different Windows operator authorizations can be assigned for the project directories enabled.

Detailed information on the assignment of operator authorizations is provided in the Windows documentation.

Procedure

1. Open the User Administrator in WinCC Explorer.
2. Select the user in the navigation area.

3. To give a user full access to the server project, enable the following authorizations:
   - "Remote activation"
   - "Remote configuration"



	Function	Enable
3	Process controlling	☐
4	Picture Editing	☐
5	Change picture	☐
6	Window selection	☐
7	Hardcopy	☐
8	Confirm alarms	☐
9	Lock alarms	☐
10	Free alarms	☐
11	Message Editing	☐
12	Start archive	☐
13	Stop archive	☐
14	Edit archive values	☐
15	Archive Editing	☐
16	Action Editing	☐
17	Projectmanager	☐
18	Close Runtime	☐
19	Open Picture	☐
20	Activate remote	☑
21	Configure remote	☑
22	Just monitor	☐

Selection	
Object type	Authorization
Object name	Activate remote
General	
Owner	Client_1
Owner type	User
Function	Activate remote
Translation "Authorization name"	
ID	1000
Enable	☑

Name of the authorization level

4. Close the User Administrator.

See also

Configuring Clients in the Server Project (Page 39)

How to Configure the Package Export (Page 35)

How to Register Clients in the Computer List (Page 31)

Creating a new project on the server (Page 29)

Server Configuration (Page 27)

Client Configuration (Page 42)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

1.6.5 How to Configure the Package Export

Principle

Packages are data packets with all current configuration data (tags, messages, archive, etc.) which are provided to all connected clients in a distributed system or multi-user system. The packages are exported from the server and imported to the clients.

The first ever export at the server and import at a client is done manually. All further update of the packages in server and client can be executed automatically. You can adjust the parameters when package update should occur, and the causes for initiating this update. For instance, you transfer the packages during commissioning manually to the clients to distribute the configuration data for the first time to the clients. To keep data on the clients updated you can then configure automatic package update on each modification of server data.

Note

If a project containing packages already created is copied onto another server, adapt the computer name to the copied project in the WinCC computer properties according to the new computer. If you regenerate packages in the copied project, you might have to update the name of the computer in the "Package Properties" dialog box.

When server-server-communication is selected, one server accesses the data of another server. The accessing server behaves at that moment like a client regarding the imported packages. So, in the description below, the details referring to the client are applicable to it.

The configuration data can be updated during normal operation either manually or automatically:

Manual Creation of Packages

In case of demand, new packages are manually created at the server. These are available to the clients for importing.

Automatic Package Update

With the function "Implicit Update", you can automate package export at the server as well as package import at the client.

The options displayed in the dialog "Configuration Implicit Package Update" on the server affects both the export of the packages from this server and the import of packages from other servers. You can see this in the columns "Import" and "Export" in the tables below.

In the dialog, you have the following possibilities for WinCC

Setting for WinCC CS	Import	Export	Meaning
Update server data when project is opened	X		Client imports whenever the project is **opened**
Automatic update when notified	X		The client always imports on receipt of a notification when the following conditions are fulfilled: • The server setting "Notify after export" has been activated. • The project is not activated.

Setting for WinCC CS	Import	Export	Meaning
Monitor changes to the configuration data. • Generate server data when project is opened. • Generate server data when project is closed. • Generate Server data immediately when changes are made		X	Server exports the package • when the project is opened • when the project is closed • on every project data change
Notify after export		X	Server sends notification after package export This setting should be activated if the client setting "Automatic update when notified" should become effective.
Automatic import		X	Server reimports its own exported package • to make configurations independent of a special server with the symbolic computer name, e.g. "Tags". • with views of a special server for clients without their own projects.

Setting for WinCC RT	Import	Export	Meaning
Update server data when project is opened	X		Client imports whenever the project is **activated**
Automatic update when notified	X		The client always imports on receipt of a notification when the following conditions are fulfilled: • The client setting "Automatic update when notified" is activated. • The project is activated.

Note

Do not use automatic package export if project data must be changed frequently, e.g., during commissioning or during the use of configuration tools.

To configure the export package, use the server data editor in the WinCC Explorer.

Requirement

The server project must be open.

Procedure

Manual package export

1. Go to "Server data" in WinCC Explorer and select "Create" in the shortcut menu.
2. In the "Package Properties" dialog box specify the symbolic and the physical server names. This information identifies the origin of the package on the client.
   Define the physical and symbolic computer names of the server as soon as possible during configuration. If the symbolic computer name is changed, it must be adapted in all configuration data.
   The symbolic computer name is generally comprised of a combination of the project name and physical computer name.
3. Click "OK". The server data is created. This may take some time, depending on the size of the configuration.

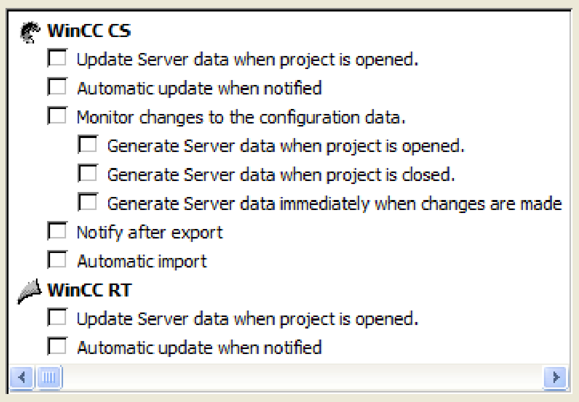
Results

The package with the server data is located in the WinCC Explorer in the list under "Server data". The packages are saved in the project directory under <project_name>\<computer\packages>\*.pck in your file system.

The clients can then import the packages.

Automatic package export

1. Go to "Server data" in the WinCC Explorer and select "Implicit update" in the shortcut menu.



WinCC CS
- □ Update Server data when project is opened.
- □ Automatic update when notified
- □ Monitor changes to the configuration data.
  - □ Generate Server data when project is opened.
  - □ Generate Server data when project is closed.
  - □ Generate Server data immediately when changes are made
- □ Notify after export
- □ Automatic import

WinCC RT
- □ Update Server data when project is opened.
- □ Automatic update when notified

2. Select the required options. Multiple selection is possible.
3. Click "OK" to confirm your choice.

Results

The packages with server data from the own server are generated at the moment you have selected, or packages already imported from other servers are updated, on closing the project, for instance.

Note

In WinCC projects that were created with the SIMATIC Manager, the "Server data" shortcut menu does not contain the options "Create..." and "Implicit Update...": This also applies to WinCC projects created in WinCC and subsequently imported into SIMATIC Manager by using the function "Import WinCC Object". This type of projects are also referred to as TIA projects.
If you copy a TIA project with WinCC Explorer and then edit the copy with WinCC Explorer, the shortcut menu of "Server data" includes the menu items "Create..." and "Implicit Update...".

Display of Generated Packages

When the packages have been generated, they are displayed in the WinCC Explorer data window as follows:

Keyboard, right: Loaded package

Keyboard, left: Package exported from the server

🖥: Loaded package without standard server

🖥: Loaded package with standard server

🖥: Server export package (not reimported)

🖥: Locally created package that was reimported in own project.

See also

Client Configuration (Page 42)

Configuring Clients in the Server Project (Page 39)

How to Configure Operator Authorization (Page 32)

How to Register Clients in the Computer List (Page 31)

Creating a new project on the server (Page 29)

Server Configuration (Page 27)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

Typical configurations (Page 19)

1.6.6 Configuring Clients in the Server Project

Principle

If you configure a multi-user system in which multiple clients display a view of exactly one server, you do not create separate projects for the clients.

You configure the behavior of the clients without their own project in the server project.

Server client configuration

A client without its own project only has a view of the server on which the client is configured.

The connection of this server to another server via server-server communication or to a central archive server is not allowed.

Internal tags on clients without their own project

The following particularities apply to internal tags on clients without their own project

- The "Computer-local" setting is relevant.
  With this setting you specify whether tag changes will be updated on a project-wide or computer-local basis.
  Internal tags are always updated project-wide on WinCC servers.
  On clients with their own project, internal tags are always updated on a computer-local basis
- If the "Computer-local" setting is enabled, the "Runtime persistence" setting has no effect.

Application of project changes when WinCC Explorer is grayed out

Project changes on the client are not applied when WinCC Explorer is grayed out.

Initial situation

- A WinCC editor is open in Runtime on a client without its own project.
- WinCC Runtime is deactivated on the server.

Behavior

Changes in the editor, e.g. a script change in a project function, are not applied.

Solution

You must not configure as long as WinCC Explorer is grayed out on the client without its own project.

Requirement

- The clients which should display the server data have been registered in the server's computer list.
  The clients only attempt to access one specific server.
- That server must not import packages form other servers.
- The server project is open on the server.

Procedure

1. Open the "Computer" editor in the WinCC Configuration Studio.
2. Select the client you want to configure under "Project" in the navigation area.
3. Activate the applications to be active on the client on the "Processes when starting WinCC Runtime" tab.
   Example: If you work with scripts, enable "Global Script Runtime".
4. In the "Properties - Computer" area under "Parameters", select the language in which runtime should be started on the client.
   It is possible, for example, to configure two clients which display the same data in different languages.
   Change the default runtime language of the client, if necessary.
5. Select a start picture for the client under "Graphics".
   The start picture can be individually selected for each client.
6. If necessary, configure further settings in the "Properties - Computer" area, e.g. hotkeys for operation in runtime under "Keys".
7. Configure the properties of the other clients in the WinCC project.
8. Select the "Implicit Update" entry in the pop-up menu of the "Server data" editor in the WinCC Explorer.
   The configuration dialog opens.
9. Activate the "Automatic import" setting and confirm with "OK".
10. Create the server package via the "Server data" shortcut menu.

See also

How to Configure the Package Export (Page 35)

How to Configure Operator Authorization (Page 32)

How to Register Clients in the Computer List (Page 31)

Creating a new project on the server (Page 29)

Server Configuration (Page 27)

Client Configuration (Page 42)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

Typical configurations (Page 19)

1.7 Client Configuration

1.7.1 Client Configuration

Introduction

A client configuration is only necessary when a distributed system is configured in which the clients can display the views on several servers. If a multi-user system is configured in which the clients only display data from one server, no client configuration is necessary. The clients receive all data and their runtime environment from the server project.

If a client/server system is configured which includes several servers, and clients display different views on several servers (distributed system), configure an individual client project for each client. In Runtime, each client can display views on up to 18 different servers or redundant pairs of servers, e.g. display messages from Server 1 and Server 2, display and write process values from Server 3, display pictures from Server 4, etc.

Clients in a distributed system can perform the following according to the respective operating authorizations on the server:

- Monitor the process.
- Monitor and operate the process.
- Remote configuration of projects on a server.
- Remote activation and deactivation of projects on a server.

Note

In order to display data from different servers, the server prefixes (i.e. the server names) must be unique within the distributed system.

Each client has its own configuration and stores a little administrative client-specific data locally in the client database, e.g.:

- Local tags
- User administrator data
- Data from the Text Library
- Project Properties
- User cycles

Note

All external data of the server configuration must also be available on the clients so that it can be displayed correctly in the client project. External data relates to ActiveX Controls which do not come from WinCC and external graphics which are integrated as OLE objects, for example.

Configuration Steps

1. Configuring server projects.
2. Creating and exporting server packages.
3. Configuring the package import on the client.
4. Configuring the client projects on the clients.

Note

If you deactivate Runtime on the server, you must also finish Runtime on the client in order to continue configuration.

See also

Configuring a Message Sequence Report for Messages from Several Servers (Page 63)

Displaying Messages from Different Servers (Page 62)

Using Data from Different Servers (Page 59)

Configuring a Picture Change on the Client (Page 58)

Displaying Pictures from Different Servers (Page 56)

Configuring the Start Picture of the Client (Page 54)

How to Configure a Preferred Server (Page 51)

How to configure a standard server (Page 49)

Configuring the Import Package (Page 46)

Creating a New Project on the Client (Page 44)

Server Configuration (Page 27)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

Typical configurations (Page 19)

1.7.2 Creating a New Project on the Client

Introduction

When creating a new client project in WinCC, select from the project types below:

- Single-user project:
  A project for a standalone operating station which carries out all tasks in runtime (process driver connection, operating, monitoring, archiving, etc.).
  Not relevant for client/server systems.
- Multi-user project:
  A server project for a multi-user system or distributed system in which several clients and/or servers are configured.
- Client project:
  A project for one client within a distributed system which can display views on several servers.

Note

Clients without their own project

If a multi-user system is configured in which several clients display a view of precisely one server, do not create local projects for the clients, but configure the client behavior in the server project.

Changing the project type

An existing project can also be converted to a client project by modifying the project type later.

You change the settings under "General" in the "Properties - Project" area of the "Computer" editor.

Setting the port

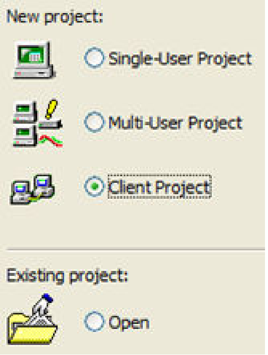
For communication between server and client, the operating system dynamically selects a port in the range from 1024 to 65535.

You can also specify a specific port in the communication settings of Simatic Shell.

Activate "Encrypted communication" and enter the desired port.

Procedure

1. On a client in WinCC Explorer, select the menu command "File > New".
   The "WinCC Explorer" dialog opens:



New project:

○ Single-User Project

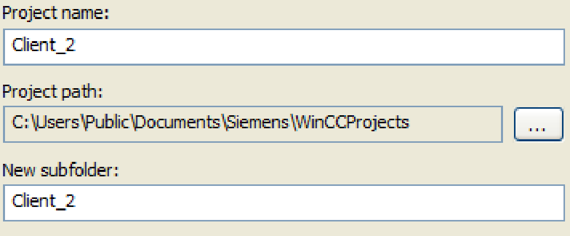
○ Multi-User Project

● Client Project

Existing project:

○ Open

2. Select "Client Project" and click "OK".
   The "Create New Project" dialog appears.
3. Enter a project name and the name of a subdirectory if the directory name should be different from that of the project.
   By default, the following folder is used as the project path:
   – "Public documents\Siemens\WinCCProjects"



Project name: Client_2

Project path: C:\Users\Public\Documents\Siemens\WinCCProjects

New subfolder: Client_2

4. Click the "Create" button.
   The project is created and opened in WinCC Explorer.

See also

Client/server scenarios (Page 24)

Configuring Clients in the Server Project (Page 39)

Configuring a Message Sequence Report for Messages from Several Servers (Page 63)

Displaying Messages from Different Servers (Page 62)

Using Data from Different Servers (Page 59)

Configuring a Picture Change on the Client (Page 58)

Displaying Pictures from Different Servers (Page 56)

Configuring the Start Picture of the Client (Page 54)

Configuring the Import Package (Page 46)

How to Configure a Preferred Server (Page 51)

How to configure a standard server (Page 49)

Client Configuration (Page 42)

Server Configuration (Page 27)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

Typical configurations (Page 19)

1.7.3 Configuring the Import Package

Introduction

For a client to display the process data of different servers in a distributed system, it needs the information of the corresponding data. Packages with the configuration data are created on the server in a distributed system for this purpose and the packages are made available to the clients. The client requires the packages from those servers whose data it wants to use.

Overview

The first ever package export at the server and import at a client is done manually. All further update of the packages in server and client can be executed automatically. You can set when the update is to take place and what triggers it.

Note

When server-server-communication is selected, one server accesses the data of another server. The accessing server behaves at that moment like a client regarding the imported packages. So, in the description below, the details referring to the client are applicable to it.

The server can reimport its own packages in order to configure tags, for example, independent of a special server with the symbolic computer.

To complete the package import, use the Server Data editor in WinCC Explorer. There are three ways to import packages:

Manual loading

Packages generated on the server are loaded on the client. The import process is triggered manually with the "Load" command. The first import of the packages must be done manually.

Manual Update

Packages already loaded on the client by the server are updated using the "Update" command.

Automatic Update

An implicit package update can be configured on the client, so that the new packages are automatically updated on the clients when a specific condition is met. However, the first import of the packages must be done manually.

Settings	Meaning
for WinCC CS • Update server data on opening project • Automatic update when notified	• Client imports whenever the project is **opened** • Server sends notification after package export, client imports whenever it is notified. This setting is effective only if the setting "Notify after export" is activated at the server for package export.
for WinCC RT • Update server data on opening project • Automatic update when notified	• Client imports whenever the project is **activated** • Server sends notification after package export, client imports whenever it is notified. This setting is effective only if the setting "Notify after export" is activated at the server for package export.

Requirement

- The packages have been created on the server.
- The client project is open.

Procedure

Manual loading

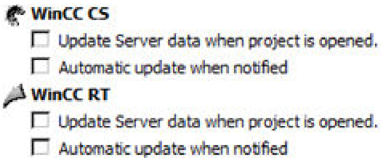
1. Open the client project on the client.
2. In WinCC Explorer select "Server data" and in the shortcut menu select "Load". The "Open File" dialog appears.
3. Select the package you want to load and click "OK".
   The packages are by default stored in the directory "...\\<Server-Project name>\<Computer name>\Packages\" under the name "<Project name_Computer name>*.pck". However, it is also possible to access packages stored on any data medium.
4. Click "Open". The data is loaded. If the corresponding server is not available, the appropriate fault entry appears on requesting the new package.

Manual Update

1. Open the client project on the client.
2. In WinCC Explorer select "Server data" and in the shortcut menu select the "Update" command.
3. The data is updated. If, in the case of a server-server communication, no packages form other servers are loaded, a fault message appears on the server.

Automatic Update

1. Open the client project on the client.
2. In WinCC Explorer select "Server data" and in the shortcut menu select "Implicit Update": The "Configuration Implicit Package Update" dialog appears.



WinCC CS
□ Update Server data when project is opened.
□ Automatic update when notified
WinCC RT
□ Update Server data when project is opened.
□ Automatic update when notified

3. Select the required options. Multiple selection is possible.
4. Confirm your selection with "OK". The server data is automatically updated on the client, e.g. on opening a project or following notification via the network. If the corresponding server is not available, no fault message appears on the client.

Note

If new packages are added or packages are deleted, while the project has been activated on the client, difficulties in representation can occur. You can remedy this situation by deactivating the client and then activating it again.

Display of the packages loaded

When the packages have been loaded, they are displayed in the WinCC Explorer data window as follows:

Keyboard, right: Loaded package

Keyboard, left: Exported, but not yet loaded package

🖥: Loaded package, without standard server

🖥: Loaded package, with standard server

See also

Configuring a Message Sequence Report for Messages from Several Servers (Page 63)

Displaying Messages from Different Servers (Page 62)

Using Data from Different Servers (Page 59)

Configuring a Picture Change on the Client (Page 58)

Displaying Pictures from Different Servers (Page 56)

Configuring the Start Picture of the Client (Page 54)

How to Configure a Preferred Server (Page 51)

How to configure a standard server (Page 49)

Creating a New Project on the Client (Page 44)

Client Configuration (Page 42)

Server Configuration (Page 27)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

Typical configurations (Page 19)

1.7.4 How to configure a standard server

Introduction

Configure a standard server for a client in a distributed system from which data should be requested if no unique server prefix (e.g. for tags) is specified.

If no standard server has been configured for a component, an attempt is made to access local client data (e.g. internal tags).

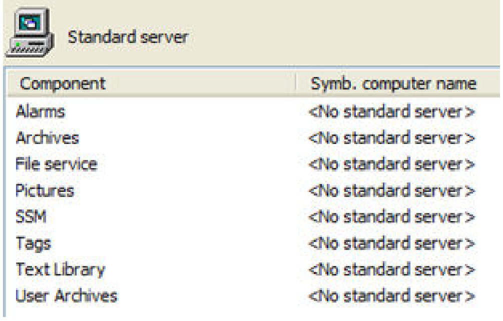
If there s no local data management on the client (e.g. messages and archives), access is rejected and an error message issued.

Requirements

A standard server can only be selected on the client after importing the corresponding packages.

Procedure

1. Select the "Server Data" entry on the client in WinCC Explorer.
2. Select the "Standard server..." item from the shortcut menu.
   The "Configure standard server" dialog appears.
3. Click the entry for the component required at the symbolic computer name.
   Select a server from the drop-down list box.
   The list contains symbolic computer names of all packages loaded on the client.



Standard server

Component	Symb. computer name
Alarms	<No standard server>
Archives	<No standard server>
File service	<No standard server>
Pictures	<No standard server>
SSM	<No standard server>
Tags	<No standard server>
Text Library	<No standard server>
User Archives	<No standard server>

4. The components listed in the dialog are dependent on the WinCC installation.
   If options have been installed, the component options (e.g. SSM - Split Screen Manager) can be listed in addition to the components displayed.
5. Click "OK" to confirm your choice.

When must a standard server be selected for a component?

Alarms

If operating messages should be generated on the client, a standard server for alarms must be specified beforehand. No Alarm Logging can be configured on the client itself and the messages must be issued on a server.

A client can retrieve user-defined selections of messages centrally from the default server.

Archives, Pictures, Text Library, User Archive, Tags

When a standard server has been configured on the client, data from these components for which no valid server prefix has been generated, is searched for on the defined standard server. If no standard server has been configured on the client, no server can be located for this data since there is no server prefix.

Setting a default server for archives, pictures, text library, user archives, and tags is only makes sense for for special applications. If you are not explicitly prompted to set a specific server by SIMATIC documentation or Customer Support, leave the setting on "No Standard Server" in the "Server Data" editor in the "Configure Standard Server".

Note

If a standard server is entered for tags on a WinCC client, no status information is shown in the tooltip in runtime for tag management.

Select standard server when using Basic Process Control

Alarms

For alarms, you always have to indicate a standard server.

Tags

Never indicate a standard server for tags.

SSM (Split Screen Manager)

Always indicate a standard server for the SSM component.

When trend groups are combined on a WinCC client, the trend groups are saved on the standard server and its redundant partner server. Other WinCC clients can indicate this server also as standard server for the SSM component. Thus the configured trend groups are be made available to these WinCC clients as well. If no standard server is configured on the WinCC client for the SSM component, the compiled trends are locally saved on this computer. Other WinCC clients cannot display these trend groups in WinCC OnlineTrendControl. It is principally be impossible to display these trend groups in the server project.

If screen compositions are configured on the WinCC client, they are saved on this server only if a standard server is indicated for the SSM component. If no standard server is indicated, the configured WinCC client screen compositions are saved locally and are not accessible for any other client. It is principally impossible to display these screen compositions in the server project.

If redundancy is configured on a server, the data of the trend groups and of the screen compositions is also synchronized on its redundant partner server. On redundancy changeover, all compiled trend groups and all screen compositions can be requested from the WinCC clients.

See also

Configuring the Import Package (Page 46)

Configuring a Message Sequence Report for Messages from Several Servers (Page 63)

Displaying Messages from Different Servers (Page 62)

Using Data from Different Servers (Page 59)

Configuring a Picture Change on the Client (Page 58)

Displaying Pictures from Different Servers (Page 56)

Configuring the Start Picture of the Client (Page 54)

How to Configure a Preferred Server (Page 51)

Creating a New Project on the Client (Page 44)

Client Configuration (Page 42)

Server Configuration (Page 27)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

Typical configurations (Page 19)

1.7.5 How to Configure a Preferred Server

Introduction

You configure the preferred server on a client of a distributed system or multi-user system if redundant servers are used.

A preferred server is the server in a redundant server pair which has priority for a client in a distributed system. The client receives data from the preferred server as long as it remains available.

The preferred server can be defined individually for each client so that the clients can be distributed among the redundant servers to ensure the permanent operability. If there is a network interruption to the configured server, the client switches over to the redundant partner server. When the server is available again, the client switches back to the preferred server.

By distributing the clients among the redundant servers, the load is distributed and the performance of the entire system is improved.

Note

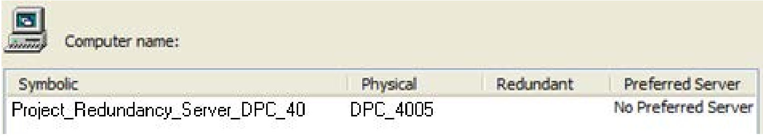
The configuration of redundant systems in WinCC is described under the topic "Redundant Systems".

Procedure

The preferred servers for the clients in distributed systems and multi-user systems are configured differently:

Configuring a preferred server for clients in a distributed system

1. Select the "Server Data" entry on the client in WinCC Explorer.
2. Select "Configuring" from the shortcut menu.
   The "Configure server data" dialog appears.
3. The list contains the symbolic and physical computer names of all servers from which packages are provided on the client. If a redundant server is available for a server, the physical computer name is specified. Select a server from the redundant server pairs as the preferred server.
   A redundant server pair in a distributed system has only one, common, symbolic name, by which the server is addressed.



Computer name:

Symbolic	Physical	Redundant	Preferred Server
Project_Redundancy_Server_DPC_40	DPC_4005		No Preferred Server

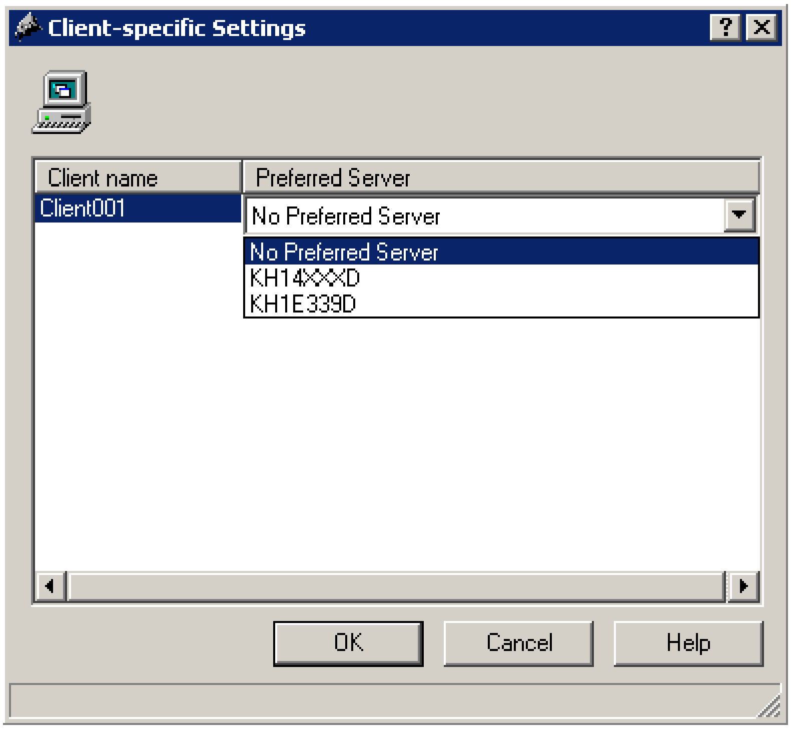
4. Conclude the input by clicking "OK".

Configuring a preferred server for clients in a multi-user system

The clients must be entered in the server's computer list.

1. Select the "Server Data" entry on the server in WinCC Explorer.
2. Select "Client-specific Settings" from the shortcut menu.
   The "Client-specific Settings" dialog appears.

3. A list of all clients entered in the server's computer list appears. Select the required client and select one of the two redundant servers from the "Preferred Server" column as the preferred server.



4. Conclude the input by clicking "OK".

Runtime behavior of the client

The client remains connected to the specified redundant server as the preferred server as long as it is available.

If the preferred server fails, the client switches to the redundant partner server. When the failed preferred server becomes available again, the client switches back to it.

See also

Configuring the Start Picture of the Client (Page 54)

Configuring a Message Sequence Report for Messages from Several Servers (Page 63)

Displaying Messages from Different Servers (Page 62)

Using Data from Different Servers (Page 59)

Configuring a Picture Change on the Client (Page 58)

Displaying Pictures from Different Servers (Page 56)

How to configure a standard server (Page 49)

Configuring the Import Package (Page 46)

Creating a New Project on the Client (Page 44)

Client Configuration (Page 42)

Server Configuration (Page 27)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

Typical configurations (Page 19)

1.7.6 Configuring the Start Picture of the Client

Introduction

Any picture in the distributed system can be used as the start picture of a client.

This can be the picture of the server, a local picture of the client, or any other picture.

The following procedure describes how to use a picture on the server as a start picture.

Requirement

The packages of the server whose picture is to be used as the start picture are imported on the client.

Procedure

1. Open the client project on the client.
2. Select the computer name in the "Computer" editor.
   The "Properties - Computer" area is displayed.
3. In the "Graphics" area, click in the "Start picture" field.
4. Enter the name of the server computer as the start picture and then the picture to be used in the form <Server name>::<Picture name>, for example:
   – Server1::StartPicture.pdl
5. It is also possible to search for pictures using the "Search" button.
   The selection dialog displays pictures of all server packages loaded on the client.
6. Complete your entry with OK.

See also

Configuring a Message Sequence Report for Messages from Several Servers (Page 63)

Displaying Messages from Different Servers (Page 62)

Using Data from Different Servers (Page 59)

Configuring a Picture Change on the Client (Page 58)

Displaying Pictures from Different Servers (Page 56)

How to Configure a Preferred Server (Page 51)

How to configure a standard server (Page 49)

Configuring the Import Package (Page 46)

Creating a New Project on the Client (Page 44)

Client Configuration (Page 42)

Server Configuration (Page 27)

Client/server scenarios (Page 24)

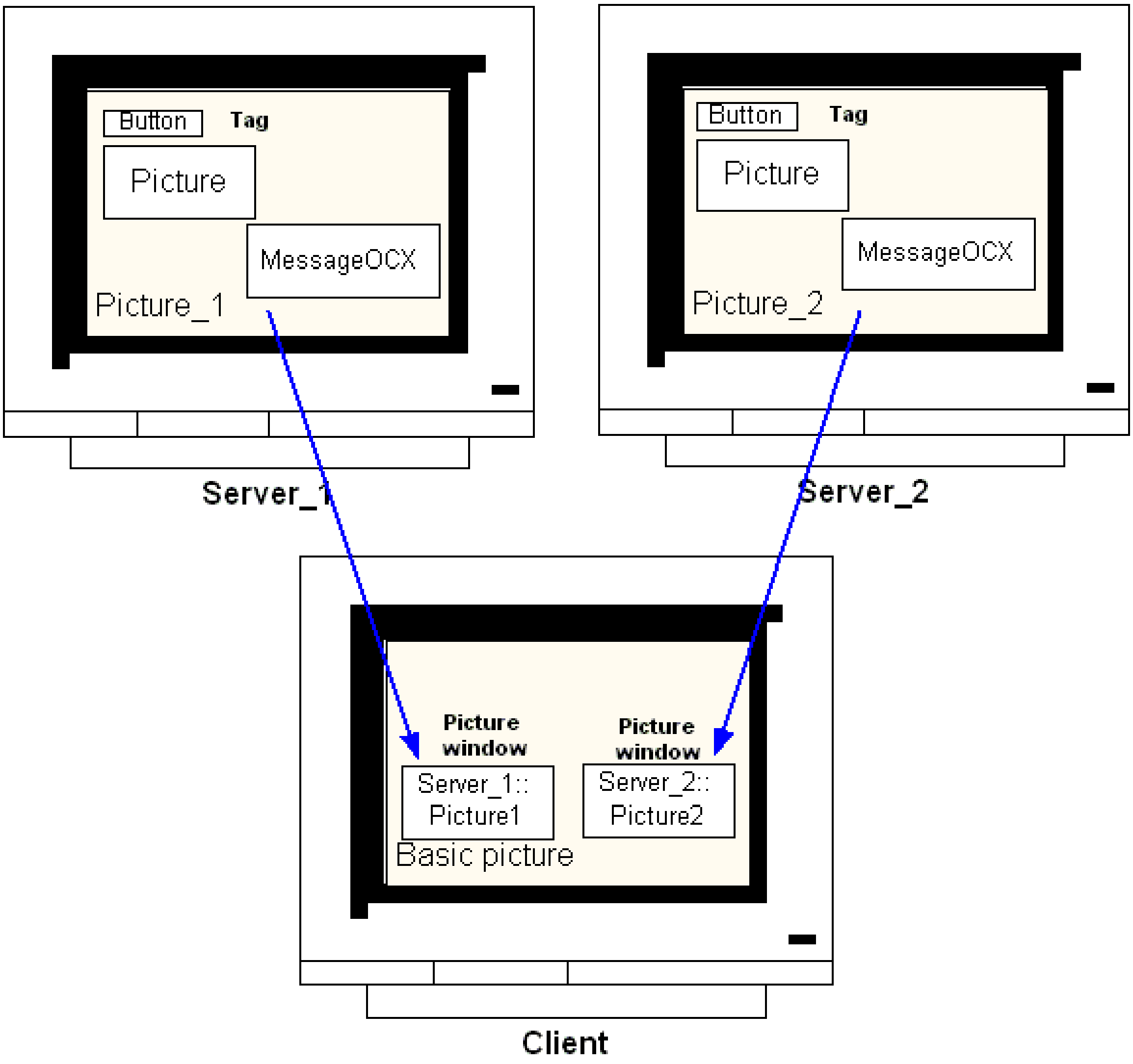
Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

1.7.7 Displaying Pictures from Different Servers

Principle

Pictures from different servers can be displayed in picture windows inside the basic screen configured on the client:



Data from a server can be accessed from each picture window. In order to integrate a server picture as a picture window in a client picture, the server prefix must precede the picture file name.

Note

The server prefixes must be unique within the distributed system.

Server pictures can be inserted in picture windows via a script (C or VBS) and via direct linking.

The pictures on the server must be adapted to the picture client's window size.

Requirement

The packages on the corresponding server must be imported on the client.

Procedure

1. Open the picture on the client to be inserted in the picture window.
2. From the standard pallet in Graphics Designer, select the "Picture Window" from the group of smart objects and insert it in the picture.
3. Open the Properties dialog by double-clicking the picture window.
4. From in the "Miscellaneous" group, double-click the "Properties" tab and select the "Picture Name" attribute in order to search for a picture.

   or:
   In the "Picture Name" attribute, double-click the "Static" column to enter a picture name directly in the form "<Server_prefix>::<Picture_name>".
5. Close the Properties dialog.

Note

If a server prefix is not automatically specified in the "Picture Name" attribute, the server prefix can also be entered via the "Server prefix" attribute. After double-clicking the "Server prefix" attribute, a selection list containing all servers appears whose packages are on the client.

See also

Configuring a Message Sequence Report for Messages from Several Servers (Page 63)

Displaying Messages from Different Servers (Page 62)

Using Data from Different Servers (Page 59)

Configuring a Picture Change on the Client (Page 58)

Displaying Pictures from Different Servers (Page 56)

How to Configure a Preferred Server (Page 51)

How to configure a standard server (Page 49)

Configuring the Import Package (Page 46)

Creating a New Project on the Client (Page 44)

Client Configuration (Page 42)

Server Configuration (Page 27)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

Typical configurations (Page 19)

1.7.8 Configuring a Picture Change on the Client

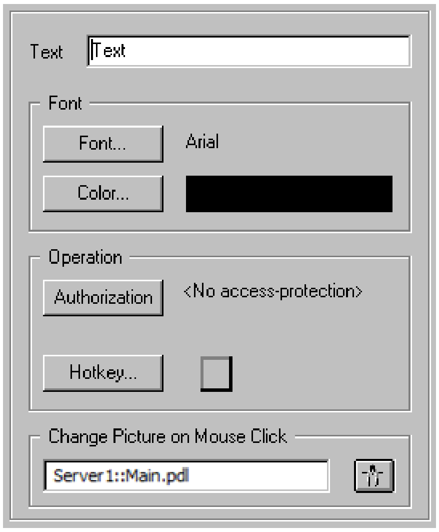
Introduction

It is possible to use a client in a distributed system to configure a picture change on a server picture by prefixing the server prefix to the target picture. There is no difference in configuration with WinCC whether configuring a normal" picture exchange or changing a basic picture.

Procedure

The following procedure describes an example of how to configure a button to initiate a change of picture on the server.

1. Open a picture of the client project in Graphics Designer.
2. Insert a button in the picture from the group of Windows objects.
   The Configuration dialog appears.
3. Enter the server prefix as the target under "Change Picture" and the picture name in the form "<server_prefix>::<picture_name>", e.g.:


Text: Text

Font: Arial

Operation: <No access-protection>

Change Picture on Mouse Click: Server1::Main.pdl

4. Close the dialog by clicking "OK".

Alternative Procedure

The picture change can also be configured in the "Properties" dialog of the button:

- Use the "Events "tab to configure a direct connection, for example, with a mouse click.
- Enter the picture name with server prefix as the constant of the direct connection.

See also

Server Configuration (Page 27)

Configuring a Message Sequence Report for Messages from Several Servers (Page 63)

Displaying Messages from Different Servers (Page 62)

Using Data from Different Servers (Page 59)

Configuring a Picture Change on the Client (Page 58)

Displaying Pictures from Different Servers (Page 56)

How to Configure a Preferred Server (Page 51)

How to configure a standard server (Page 49)

Configuring the Import Package (Page 46)

Creating a New Project on the Client (Page 44)

Client Configuration (Page 42)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

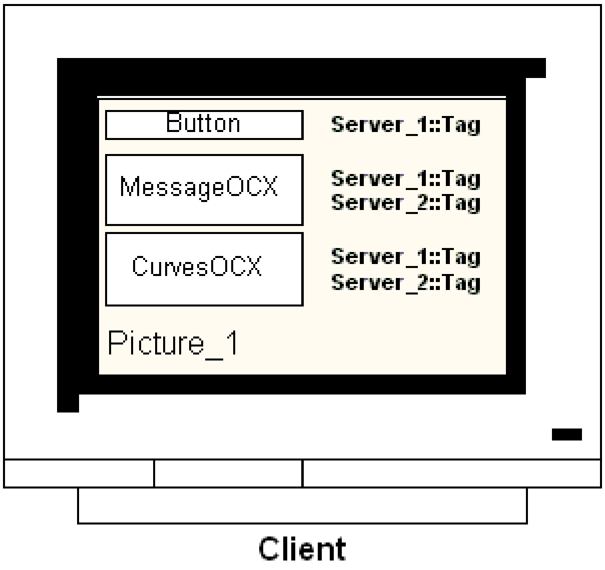
1.7.9 Using Data from Different Servers

Principle

You configure the basic picture of the client of a distributed system and all objects contained in it directly on the client.

You can access the data from multiple servers in each basic picture, for example:

- Two output fields:
  An output field for the process value of Server_1, which monitors the plant unit A.
  An output field for the process value of Server_2, which monitors another part of the plant
- Trend displays that compare data from different plant units/servers
- Message windows displaying the messages of multiple servers



You can also copy basic pictures configured on one client to other clients.

However, the packages of the servers addressed in the basic picture must be available on the target clients.

Note

All tags configured on the server and transferred to the client with a package are available on the client in the tag selection dialog.

To run them, C actions and C functions or VBS actions and VBS procedures from Global Script must be present on the client. Global C scripts and VBS scripts are not part of the packages.

Procedure

The following procedure shows you an example of how to display process data from two different servers in a trend view on the client.

1. Open the client project on the client.
2. Use Graphics Designer to configure the picture to be used as the basic picture.
3. Insert a WinCC OnlineTrendControl into the basic picture from the object palette, Controls tab.
   The "Properties of WinCC OnlineTrendControl" dialog appears.
4. Select "Online tags" as data source if you want to monitor the current process.
5. Activate the trends tab.
6. For the first trend, select a tag whose process values are to be displayed by pressing the "Selection" button under "Selection of Archives/Tags".

7. Enter the tag name in the following form:
   – "<ServerPrefix1>::<TagName>".

   Confirm with OK.
8. On the trends tab, press the "+" button to add a second trend.
9. Connect the second trend to a tag of the second server in the form:
   – "<Serverprefix2>::<TagName>".
10. Confirm your settings with OK.

Result

In runtime, two trends are displayed in the trend window on the client:

- Trend 1 shows data of server 1.
- Trend 2 shows data of server 2.

See also

Configuring the Import Package (Page 46)

Configuring a Message Sequence Report for Messages from Several Servers (Page 63)

Displaying Messages from Different Servers (Page 62)

Configuring a Picture Change on the Client (Page 58)

Displaying Pictures from Different Servers (Page 56)

How to Configure a Preferred Server (Page 51)

How to configure a standard server (Page 49)

Creating a New Project on the Client (Page 44)

Client Configuration (Page 42)

Server Configuration (Page 27)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

Typical configurations (Page 19)

1.7.10 Displaying Messages from Different Servers

General procedure

Messages from several servers can be displayed on a client in a distributed system as follows:

- Configure a message display for each server whose messages are to be displayed
- Specify several message servers as source in a message display

Note

If an Alarm Control is integrated in a basic picture of the client, the associated server picture is displayed as the basic picture on the client on executing the "Loop in Alarm" function. It is not possible to return to the original basic picture.

If an Alarm Control is integrated in a picture window of the client, the associated server picture is displayed in the "Loop in Alarm" picture window on executing the "Loop in Alarm" function. Return to the basic client picture by clicking the relevant button.

Procedure

1. Open the client project on the client.
2. Use Graphics Designer to configure the picture to be used as the basic picture.
3. Insert the WinCC Online Trend Control in the basic picture from the "Object Pallet", "Alarm Controls" tab. The "Properties of WinCC Alarm Control" dialog opens.
4. When the messages of all connected servers in this Alarm Control are to be displayed, select "Server Selection" and activate the "All Servers" check box.
5. If only the messages from a specific server are to be displayed, deactivate the "All Servers" check box and click the "Selection" button to select a WinCC server from the network .
6. Close the dialog by clicking "OK".

Note

In multi-user systems, you must ensure that contents displayed in the selection dialog on a client are named identically on all servers.

See also

Configuring a Message Sequence Report for Messages from Several Servers (Page 63)

Displaying Messages from Different Servers (Page 62)

Configuring a Picture Change on the Client (Page 58)

Displaying Pictures from Different Servers (Page 56)

How to Configure a Preferred Server (Page 51)

How to configure a standard server (Page 49)

Configuring the Import Package (Page 46)

Creating a New Project on the Client (Page 44)

Client Configuration (Page 42)

Server Configuration (Page 27)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

Typical configurations (Page 19)

1.7.11 Configuring a Message Sequence Report for Messages from Several Servers

Principle

If messages from different servers are displayed in a basic picture on the client, you can also output corresponding message sequence reports.

The messages from all servers are collected and output in the correct order.

WinCC provides a preconfigured layout and a print job for the message sequence report.

Procedure

1. Click on "Report Designer" in the WinCC Explorer.
2. Double-click on the language-neutral layout "@CCAlgRtSequence.RPl" in the data window.
The line layout editor opens.
3. Press the "Selection" button.
The "Report - Table Column Selection" dialog appears.
4. Add the servers whose messages should be logged in the message sequence report to the "Selected servers" list with the "Add Server" button.
Only servers whose packages have been imported to the client are shown.
5. With the arrow buttons, transfer the message blocks to be logged to the "Column sequence of the report" list.
6. Confirm your entry with OK.
7. Open the "@Report Alarm Logging RT Message sequence" print job in the WinCC Explorer.
8. If you have saved the layout under its own name, select your layout from the "Layout" list.
Select the "Line layout for line printer" option.
9. Activate the "Printer" option on the Printer Setup tab.
10. From the list of connected printers, select the printer on which the report should be output.
11. Confirm your entries with OK.
12. Open the "Computer" editor in the WinCC Configuration Studio.

13. Select the client computer under "Project" in the navigation area.
14. Activate the "Message sequence report" application on the "Processes when starting WinCC Runtime" tab.

See also

Creating a New Project on the Client (Page 44)

Configuring a Message Sequence Report for Messages from Several Servers (Page 63)

Displaying Messages from Different Servers (Page 62)

Configuring a Picture Change on the Client (Page 58)

Displaying Pictures from Different Servers (Page 56)

How to Configure a Preferred Server (Page 51)

How to configure a standard server (Page 49)

Configuring the Import Package (Page 46)

Client Configuration (Page 42)

Server Configuration (Page 27)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

1.8 System Behavior in Runtime

1.8.1 System Behavior in Runtime

Introduction

A client/server system in WinCC can be used to distribute the system configuration to several servers in order to reduce the load on the individual servers.

The data configured on the servers can be displayed by clients. A client can display data from up to 18 different servers or redundant server pairs in runtime.

Behavior of Editors in Runtime

Archives

If the archive system is activated on an operating station, the Tag Logging Runtime operates on the servers as archive server, on the clients as archive client:

- Only the archive server accesses the database and compiles and archives the process data.
- The clients receive archive data from the archive server.

The archive data can be displayed as a table or graphic on every client on which the Tag Logging Runtime runs.

The data to be displayed always comes from the archive server.

All operations on the client are transmitted to the server and the result of the processing is transferred back to the client.

Graphics

When a picture is called in by a client in runtime, Graphics Runtime initially searches for locally stored pictures:

- If no picture with the corresponding name is found locally, a search is made in the project folder on the server.
- If no picture is available, the corresponding message appears.

If a picture request requires an exchange with another editor (Alarm Logging, Global Script), the exchange is local.

A picture can be opened and processed by several operating stations in Runtime.

Note

Picture cache: Improving performance

To speed up the picture build up on the client, you can copy the respective pictures locally onto the client.

Configure the corresponding storage path on the client in the "Computer" editor:

- In the "Settings - Local Settings" area, in the "Picture cache" field, select the folder for the storage path.
- In the "Use cache" field, choose whether the local storage path should be used always or preferably.
  To disable the function, select the option "Never".

If a picture is modified in the server project, the data must be updated manually. Copy the modified picture back into the storage path on the client.

Messages

If messages are displayed on a client, the client receives the data displayed from the server.

The message server receives the configured data from the database.

Archive data and message lists can be displayed on every client.

The data to be displayed always comes from the message server. When new messages are received, the messages are archived in the message server.

When an operator station acknowledges a message, the acknowledgment is forwarded to the message server. The server enters the change of status in the archive and distributes the notification to all participating clients. The same process applies to the locking of messages.

If a message server is not available in Runtime, the corresponding message appears in the message window instead of the messages. When the server becomes available again, messages are displayed in the message window again.

Reports

The protocol system in WinCC does not detect Runtime in the real sense of the meaning.

Protocols and print jobs can be configured and executed at any time. Only print jobs which are to be display the archive or process data are dependent on Runtime.

The protocol system is automatically started on every client during the startup routine.

The server operates as a protocol server, the clients as protocol clients.

During the startup routine, the client log in on the server and receive the current information on the print jobs available and their status.

If a print job is started on a client, it obtains the related data from the server database. The print job is started locally. The protocol server receives the current data concerning the print job status from the client and transfers the information to the other clients.

Scripts

If an operating station activates a project locally, the server's project functions and standard functions are loaded locally.

User Administrator

The operator authorizations are checked by the runtime component of the User Administrator.

The user administrator Runtime component is automatically started on every computer when WinCC is activated.

If the login is changed, the current operating rights list is loaded from the local database.

Text Library

If the server project is activated, Text Library Runtime runs on the server as text server and on the clients as text client.

The data is always read from the server database.

Behavior in the event of system errors

If a server is not available, the clients poll the server cyclically until is has been started up again.

Data on the server cannot be displayed in the case of faults. All operable graphic objects, for example, are switched to inactive.

Note

System restart

If problems develop on a client concerning the running WinCC, you can restart the client to reconnect to the server without affecting the server.

The "Application Health Check" function

The "Application Health Check" function automatically monitors all important WinCC applications.

After detecting a software error, the lifebeat monitoring triggers the following actions:

- A process control message notifies users of the software error.
  A process control message cannot be triggered if the alarm server caused the failure.
- Redundant system:
  - In the system tag "@RedundantServerState" the server state changes to "Fault".
  - The connected clients switch over to the redundant partner server.

Note

Redundant system server restart after error

If the "Application Health Check" function detects a software error and client switching was initiated, the relevant server must be restarted.

Only after the server restart is it possible to reconnect clients to this server.

The archives are synchronized retroactively up to the point where the software error was detected.

See also

Starting Up the Server (Page 68)

Specifics of Communication for a Server with Several Network Cards (Page 70)

Shutting Down the Client (Page 72)

Shutting Down the Server (Page 72)

Starting Up the Client (Page 69)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

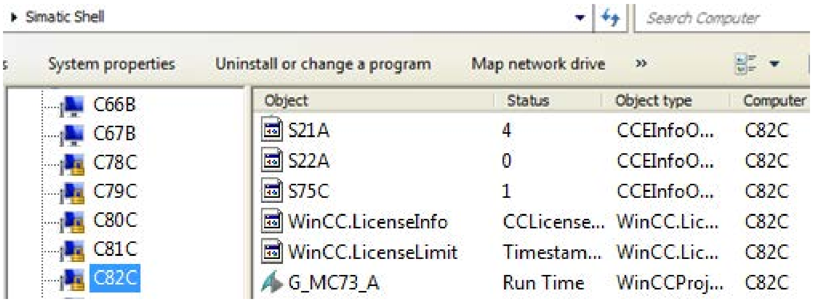
Client/Server Systems in WinCC (Page 16)

1.8.2 Starting Up the Server

Principle

Servers in a client/server system can be started up independently of the clients. As soon as a server has been started up, it makes its services available to the clients and retrieves information on all participants in the network.

You can view the current status of all servers in the data window of the "Simatic Shell" dialog. You open "Simatic Shell" via the Windows Explorer.



Object	Status	Object type	Computer
S21A	4	CCEInfoO...	C82C
S22A	0	CCEInfoO...	C82C
S75C	1	CCEInfoO...	C82C
WinCC.LicenseInfo	CCLicense...	WinCC.Lic...	C82C
WinCC.LicenseLimit	Timestam...	WinCC.Lic...	C82C
G_MC73_A	Run Time	WinCCProj...	C82C

If a server fails during normal operation, the data on the clients can no longer be updated and information is provided on the failed server.

Note

If a file server is used in the client/server system, the system is only ready for operation again when both the file server and the WinCC server have been started up.

Remote Activation

A server can also be started from another remote computer (client or server). The procedure for this is described in "Activate project".

See also

Specifics of Communication for a Server with Several Network Cards (Page 70)

How to Activate a Project (Page 85)

Shutting Down the Client (Page 72)

Shutting Down the Server (Page 72)

Starting Up the Client (Page 69)

System Behavior in Runtime (Page 65)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

1.8.3 Starting Up the Client

Principle

The clients in a client/server system boot independently of the servers.

When a client in a client/server system starts up, it receives all current information on the following via the WinCC servers known to it in the network, e.g.:

- Project names
- Server names and IP addresses
- Project status of the servers (configuration or Runtime)

The user can view the corresponding information in the list in the "Simatic Shell" dialog. When the status of a server changes, the "Simatic shell" is also updated.

Server not available.

If servers are not available, a corresponding error message is issued. In addition, graphic objects, for example, who receive their data from the server , are switched inactive.

Scripts can be used to configure the display of connection faults to the client.

See also

Specifics of Communication for a Server with Several Network Cards (Page 70)

How to Activate a Project (Page 85)

Shutting Down the Client (Page 72)

Shutting Down the Server (Page 72)

Starting Up the Server (Page 68)

System Behavior in Runtime (Page 65)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

1.8.4 Specifics of Communication for a Server with Several Network Cards

Introduction

If several network cards or SIMATIC NET SOFTNET drivers are installed on a WinCC server for the process connection and are operated with an active TCP/IP protocol, communication of the server with WinCC clients could be affected.

A possible cause could be that each network card or the SOFTNET driver in the server has its own IP address. Therefore, under certain circumstances, it is possible that when the server is logged on in the network, Windows attempts to establish a connection via the wrong IP address, e.g. via that of the SOFTNET driver. If the attempt to establish a connection fails, Windows marks the communication via this IP address as defective but does not attempt to establish a connection via another IP address available on the computer.

In this case, appropriate modifications must be made by the network administration.

Checking the sequence of the network cards

If several network cards are installed on the computer, the network card for the terminal connection must be in first place.

Check the sequence in the Windows Control Panel under "Network connections".

In the "Advanced" menu select the menu command "Advanced settings". The sequence is available in the "Advanced" dialog on the "Network cards and connections" tab in the "Connections" section.

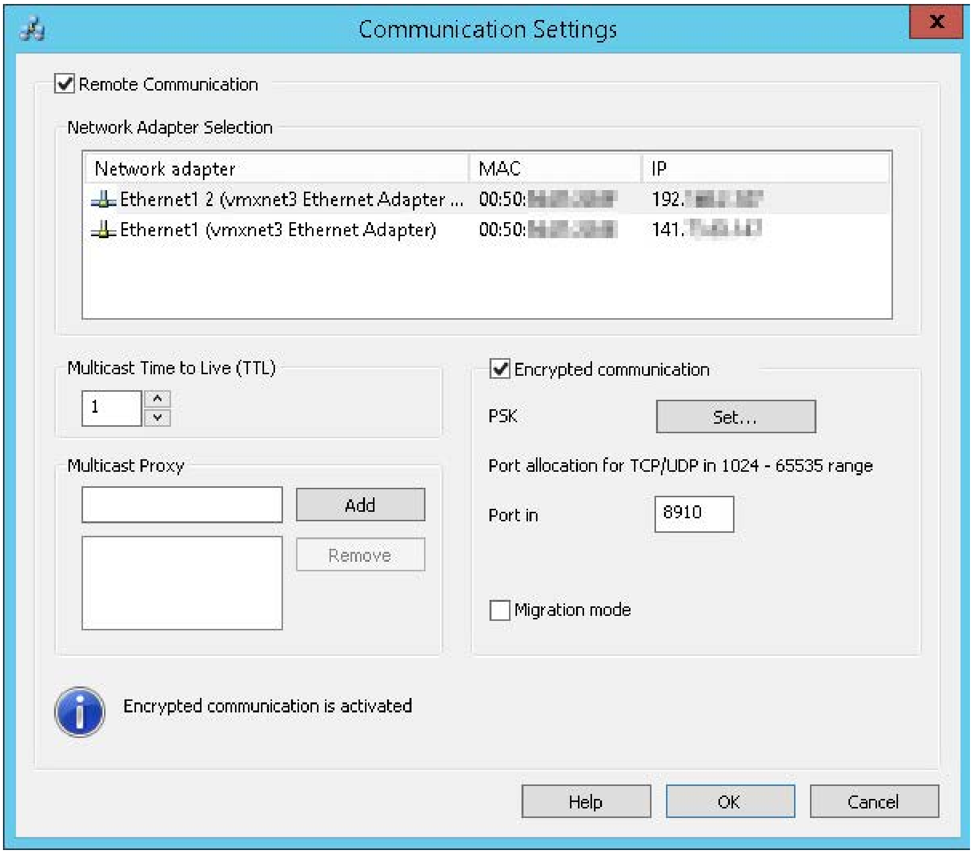
Diagnostics

Using the directory "Simatic Shell", you may check the configuration of the network card.

If you determine that a computer indicates an address with an incorrect, i.e. inaccessible network area, select a different network adapter.

Procedure

1. In the navigation window of Windows Explorer, click the "Simatic Shell" directory.
2. In the shortcut menu of the directory, select the "Settings..." dialog.



Communication Settings

☑ Remote Communication

Network Adapter Selection

Network adapter	MAC	IP
Ethernet1 2 (vmxnet3 Ethernet Adapter ...	00:50:[redacted]	192.[redacted]
Ethernet1 (vmxnet3 Ethernet Adapter)	00:50:[redacted]	141.[redacted]

Multicast Time to Live (TTL): 1

Multicast Proxy

☑ Encrypted communication

PSK

Port allocation for TCP/UDP in 1024 - 65535 range

Port in: 8910

□ Migration mode

Encrypted communication is activated

3. If you wish to change the network interface, click the desired network card in the "Network Adapter" area.

A check is also to be made in the configuration of the SOFTNET driver on the server whether the Windows utilities not required for the process connection can be deactivated.

If a connection can still not be established after checking these points, please contact Customer Support.

See also

How to Access Computers Outside a Subnet (Page 76)

Remote Configuration (Page 73)

Starting Up the Client (Page 69)

Starting Up the Server (Page 68)

System Behavior in Runtime (Page 65)

Client/Server Systems in WinCC (Page 16)

1.8.5 Shutting Down the Server

Principle

If a server in the client/server system is shut down, it can no longer provide the connected clients with process data. It is simultaneously logged off from the system and is marked as deactivated in the "Simatic Shell".

Remote Deactivation

A server can also be shut down from another remote computer (client or server). The procedure for this is described in "Deactivate project".

See also

Starting Up the Client (Page 69)

How to Deactivate a Project (Page 87)

Shutting Down the Client (Page 72)

Starting Up the Server (Page 68)

System Behavior in Runtime (Page 65)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

1.8.6 Shutting Down the Client

Principle

When a client in a client/server system is shut down, it is logged off from the system.

See also

How to Deactivate a Project (Page 87)

Shutting Down the Server (Page 72)

Starting Up the Client (Page 69)

Starting Up the Server (Page 68)

System Behavior in Runtime (Page 65)

Client/server scenarios (Page 24)

Quantity structures and performance (Page 21)

Client/Server Systems in WinCC (Page 16)

1.9 Remote Configuration

1.9.1 Remote Configuration

Clients provided with the corresponding operator authorizations can operate a server project remotely, e.g.:

- Remote configuration of a server project
- Activate a server project
- Deactivate a server project

You can find more information on remote access and on RDP in the WinCC Information System in the Release Notes under "Notes on WinCC > Remote access and Remote Desktop Protocol (RDP)".

You can find current instructions for remote access in the FAQ 78463889:

- SiePortal: Remote access to WinCC stations (http://support.automation.siemens.com/WW/view/en/78463889)

Function of Simatic Shell

For configuration of remote access, the "Simatic Shell" dialog is available.

In the "Simatic Shell" dialog, you can view the enabled servers and computers with the WinCC projects available through the network.

These include all projects which run under a demo license.

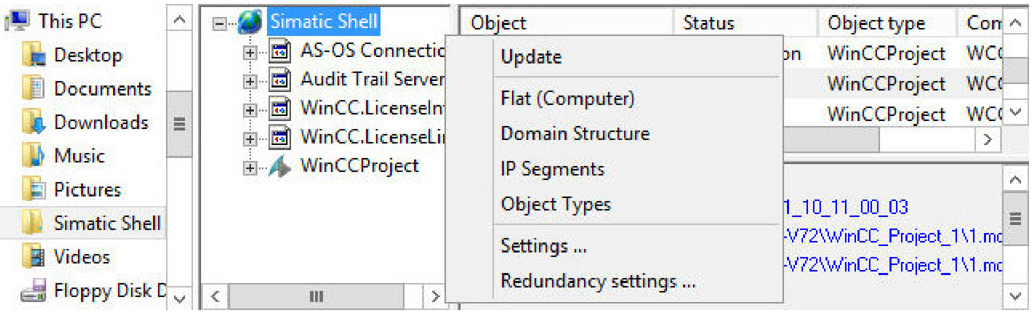
The "Simatic Shell" dialog

Open the "Simatic Shell" dialog via the Windows Explorer.

The PCs and WinCC projects can be displayed structured according to the following criteria:

Flat (computer)	All of the entries are displayed below one another.
Domain structure	The servers and their WinCC projects are grouped by domain.
IP segments	The servers and their WinCC projects are grouped by IP addresses.
Object types	The view is grouped by server types.

You sort the entries with a double-click on the column header.



Icon for network interruption

If the local network adapter is temporarily unavailable, e.g. after disconnecting the network cable, an exclamation mark is displayed above the entry for a short time: !, !

If required, update the "Simatic Shell" view to display the changed status of the connected PCs.

Configuring IGMP for multiple routers

The "Internet Group Management Protocol (IGMP)" network protocol is used on the terminal bus.

When you are using multiple computers, only one router may be active as "Querier". Note the following settings:

Setting	Configuration
IGMP Snooping	"ON"
IGMP Querier	Only one station must be activated with the setting "ON". Select the setting "OFF" for all other stations. If multiple stations are configured as querier, only the station with the lowest switch IP address is active.
Snooping switch IP	A separate, unique IGMP switch IP address must be configured for each station.

See also

How to Deactivate a Project (Page 87)

How to Activate a Project (Page 85)

How to Edit Server Project Pictures (Page 84)

How to Open a Project for Editing (Page 82)

Access to Projects from Several Clients (Page 80)

How to Access Computers Outside a Subnet (Page 76)

Client/Server Systems in WinCC (Page 16)

Encrypted communication (Page 74)

SiePortal: Remote access to WinCC stations (http://support.automation.siemens.com/WW/view/en/78463889)

1.9.2 Encrypted communication

When accessing a computer, always ensure that encrypted communication of the computers is established.

Only use the unencrypted communication option temporarily, for example, for migration purposes.

Encryption in WinCC

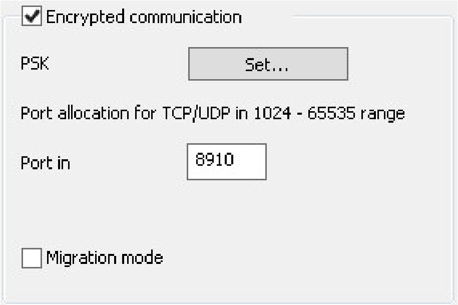
WinCC uses the "Security Support Provider" (SSP) interface from Microsoft.

With computers in a domain, the Microsoft "Remote Desktop Protocol (RDP)" encryption is used.

Outside of a domain, state-of-the-art symmetrical encryption is used.

Configuring encrypted communication

Open the communication settings in the "SIMATIC Shell" shortcut menu in Windows Explorer.



☑ Encrypted communication

PSK

Port allocation for TCP/UDP in 1024 - 65535 range

Port in 8910

□ Migration mode

If you use encrypted communication, connections are only established to computers for which the same PSK key was specified. You can only communicate with these computers. Connecting to unencrypted computers is not possible.

You can specify different environments with their own PSK keys for the same network.

Depending on the configuration of encrypted communication, only the relevant computers are shown in the Simatic Shell.

You can find information about configuration under "How to Access Computers Outside a Subnet (Page 76)".

Authentication: Self-signed certificates

Self-signed certificates are not supported in the communication between WinCC stations.

If only self-signed certificates can be found on the server, the configured PSK key is used for the communication.

Microsoft SQL Server for WinCC

You can find more information on encrypted communication in the following Microsoft article:

- Internet: "Configure SQL Server Database Engine for encrypting connections" (https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/configure-sql-server-encryption?view=sql-server-ver15)

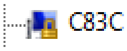
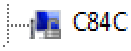
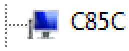
Migration mode

Migration mode is also available for upgrading during operation. This mode allows encrypted and unencrypted connections side by side in the network.

In migration mode, all computers in the network with encrypted and unencrypted connections are shown.

Use migration mode only as a temporary solution on the way to encrypted communication throughout the entire plant.

Computer symbols

 C83C	The computer only allows encrypted connections.
 C84C	The computer allows encrypted and unencrypted connections. (migration mode)
 C85C	The computer allows unencrypted connections. (migration mode or view with unencrypted communication)

See also

How to Access Computers Outside a Subnet (Page 76)

Remote Configuration (Page 73)

Client/Server Systems in WinCC (Page 16)

Typical configurations (Page 19)

Internet: "Configure SQL Server Database Engine for encrypting connections" (https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/configure-sql-server-encryption?view=sql-server-ver15)

1.9.3 How to Access Computers Outside a Subnet

You integrate computers in your network which are downstream from a router, for example, into the system via "Simatic Shell".

"Simatic Shell" is part of WinCC and is used for central maintenance and diagnostics of all computers integrated in the client-server system.

Principle

Using the settings in "Simatic Shell", you introduce a computer within your subnet as an "Agent" who distributes the information from other computers to the computers within the subnet.

If you enable encrypted communication, only those computers in which the shared specified key is known prior to communication can communicate with each other.

After you have logged on, all participating computers in the system can communicate even beyond network limits. Each computer added to an existing group is informed of the current status of all computers.

When the status of a computer changes, a message is issued to all participants, e.g.:

- If a computer has activated a project.
- If a computer is shut down.
- If a computer is started up and thus enters the group.

Firewall settings

To allow WinCC computers from different networks to communicate with one another, you need to adapt the following settings of the local Windows firewall.

For all WinCC-specific firewall rules, you need to expand the scope by the IP addresses of the computers from other networks or the complete IP scope of the other networks.

Procedure

1. In the Windows Control Panel, open the category "System and Security > Windows Firewall".
2. Click "Advanced settings".
   The "Windows Firewall with Advanced Security" dialog opens.
3. Under "Inbound Rules", select one by one all affected firewall rules, e.g. CCAgent, OPC UA Discovery, WinCC ProjectManager.
   The rules each have a group name in the "Group" column that begins with "SIMATIC", e.g. "SIMATIC Communication Services", "SIMATIC WinCC OPC".
4. Open the "Properties" dialog via the Rules shortcut menu.
5. In the "Scope" tab, add the IP addresses or IP scopes of the communication partners to "Remote IP address".

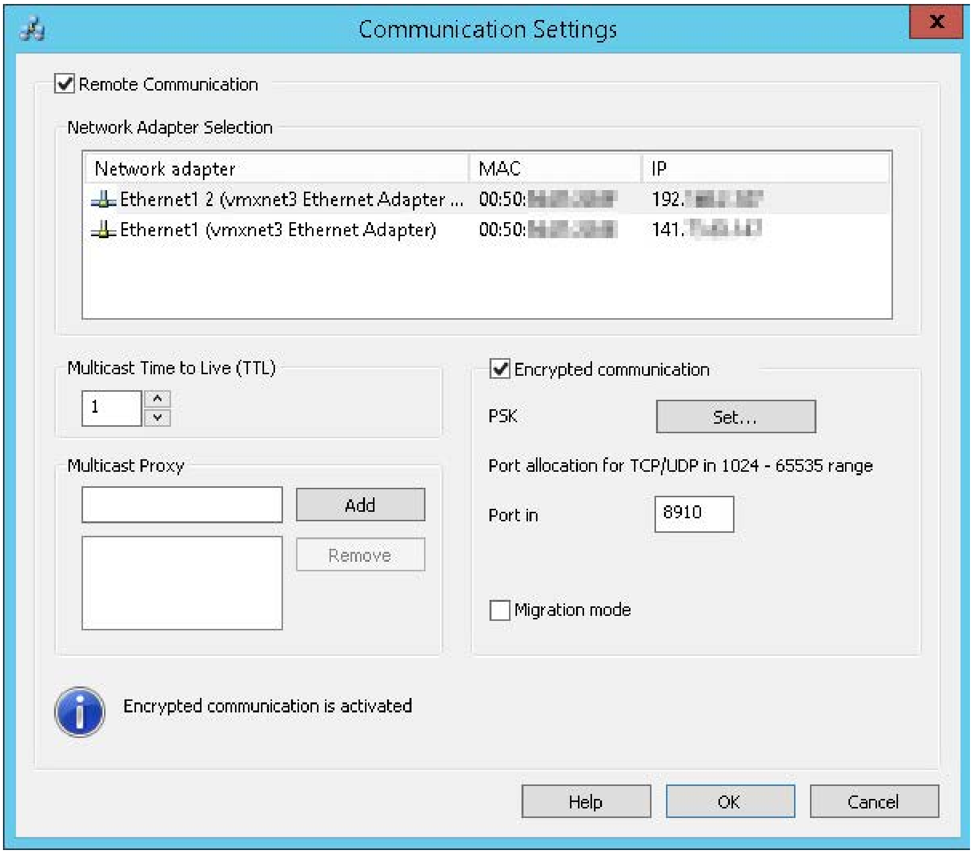
Requirement

- The "Remote Communication" option is enabled and the network adapter is configured.

Procedure

1. Open the Windows Explorer on the computer which accesses the WinCC computer as client.
2. Select the entry "SIMATIC Shell".
   The "Simatic Shell" window opens.

3. Select the "Settings" command from the "SIMATIC Shell" shortcut menu.
   The "Communication Settings" dialog opens.



Communication Settings

☑ Remote Communication

Network Adapter Selection

Network adapter	MAC	IP
Ethernet1 2 (vmxnet3 Ethernet Adapter ...	00:50:[redacted]	192.[redacted]
Ethernet1 (vmxnet3 Ethernet Adapter)	00:50:[redacted]	141.[redacted]

Multicast Time to Live (TTL): 1

Multicast Proxy

☑ Encrypted communication

PSK

Port allocation for TCP/UDP in 1024 - 65535 range

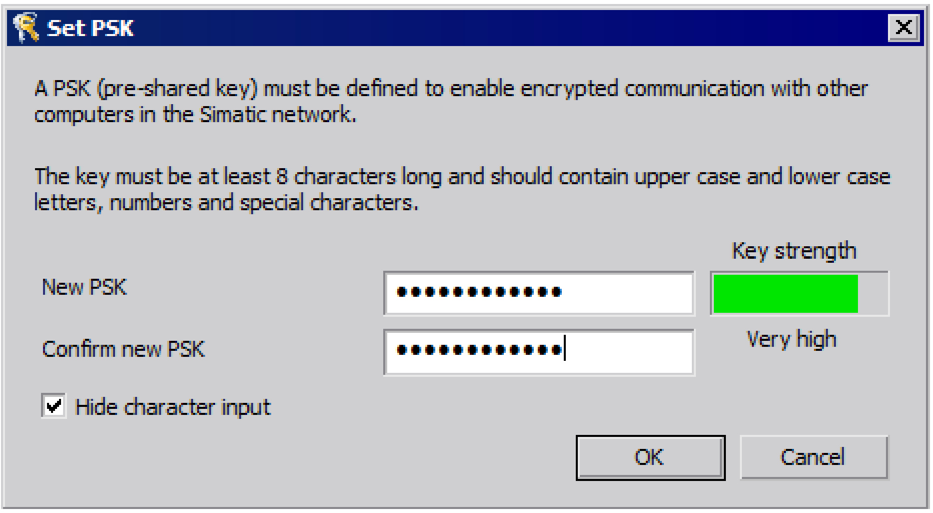
Port in: 8910

□ Migration mode

Encrypted communication is activated

4. Check the setting in the field "Multicast Life Time (TTL)".
   The value specifies the maximum number of transitions between various subnets (IP parameters TTL).
5. In the "Multicast Proxy" input field, enter the IP address for the computer intended as "Agent" for the subnet.
   This may be any computer in the subnet (client or server).
   To add the computer to the list of agents, click "Add".

6. To set up encrypted communication, select the "Encrypted communication" option.
   To enter the PSK key, click the "Specify" button.



Set PSK

A PSK (pre-shared key) must be defined to enable encrypted communication with other computers in the Simatic network.

The key must be at least 8 characters long and should contain upper case and lower case letters, numbers and special characters.

Key strength

New PSK ●●●●●●●●●●●● 🟩

Confirm new PSK ●●●●●●●●●●●●

Very high

☑ Hide character input

7. Enter characters with a high key strength for the key.
   The key must be at least 8 characters long and include numbers and symbols in addition to uppercase/lowercase letters.
   Confirm your settings with "OK".
8. If you do not want to use the available port assigned with the default setting, specify the assignment of the inbound port.
9. To allow encrypted and unencrypted connections side by side, select the "Migration mode" option.
   Only use this option temporarily, for example, when upgrading during operation.
10. Confirm your settings with "OK".

See also

Access to Projects from Several Clients (Page 80)

How to Deactivate a Project (Page 87)

How to Activate a Project (Page 85)

How to Edit Server Project Pictures (Page 84)

How to Open a Project for Editing (Page 82)

Remote Configuration (Page 73)

Client/Server Systems in WinCC (Page 16)

Encrypted communication (Page 74)

1.9.4 Access to Projects from Several Clients

Configuration options

Depending on the type of data, one or more clients can access the server project remotely.

A distinction is made between data stored in the server database (Alarm Logging, Tag Logging, Tags, User Admin, Text Library) and file-based data (pictures and graphics, reports, scripts).

Note

Editing of multiple clients off

Data from the server database can be edited by multiple clients at the same time.

Note, however, that the changes of the last saving client are always saved when several clients access the same data. For data from the server database, all data of the respective editor is always saved, even if only individual values have been changed.

For data stored in files, an already open file is blocked for further access.

More information: "Working with WinCC > Working with Projects > Creating and Editing a project > How to Use Multiuser Engineering"

Archives (Tag Logging)

Archives are stored in the server database.

The Tag Logging data can be changed in runtime.

The changes are distributed from the server to all clients involved.

Pictures

Pictures are stored as files on the server or file server.

When a client accesses a picture on the server, the picture is locked to other clients. Different pictures of a project can be opened by different clients.

Pictures can be changed in runtime and are available after saving the next time you select the picture.

You can also store pictures locally for editing. In this case you then have to synchronize them with the server manually.

Messages

Messages are stored in the server database.

The data of the message system can be changed in runtime.

The changes are distributed from the server to all clients involved.

Reports

Reports are stored centrally in the project directory of the server. The report data is divided into layouts (files) and print jobs (entries in the project database).

Only one client can configure the report system on the server at a time.

You can also store reports locally for editing. In this case you then have to synchronize them with the server manually.

It is not possible to make changes to the log system in runtime since reports can run independently of runtime.

Scripts

Scripts are stored centrally in the project directory of the server.

Project-specific scripts can be defined independently on a computer-specific basis. Scripts are stored in files. Actions of the Graphics Designer are stored in the picture.

During editing, the files (scripts or pictures) are locked to other clients.

If there is no connection to the server, you can also change scripts locally. In this case you then have to synchronize them with the server manually.

Scripts can be modified in runtime.

The server distributes the changes to all participating computers.

Note

Editing scripts locally

If a client configures a script without access to the server, the script is stored locally.

If you want the script to be available on the server, you must manually copy the script to the appropriate server directory.

Texts from the Text Library

Texts from the Text Library are stored in the server database.

The text objects are saved individually.

Texts can be changed in runtime.

The server distributes the changes to all participating computers. The update is carried out in the locally set configuration language.

Note

Access from different editors

Some WinCC editors, such as Alarm Logging and User Administrator, access the same database table of the Text Library during configuration.

You can therefore only edit these editors on one operator station at a time.

Tags

Tags are stored in the server database.

Note

Updating in runtime

If a client project is deactivated to change a tag, the changes do not take effect until all computers on which the project was active at the time of the change have been restarted.

User Administrator

The operator authorizations of the User Administrator are stored in the server database.

User Administrator data can be changed in runtime.

Participating computers are not notified.

The new data will be effective after the client has logged in again.

See also

How to Activate a Project (Page 85)

How to Deactivate a Project (Page 87)

How to Edit Server Project Pictures (Page 84)

How to Open a Project for Editing (Page 82)

How to Access Computers Outside a Subnet (Page 76)

Remote Configuration (Page 73)

Client/Server Systems in WinCC (Page 16)

1.9.5 How to Open a Project for Editing

Principle

A server project can be edited from a client in configuration or Runtime operation of the project.

Updating the data in Runtime is dependent on which data is configured.

The "Simatic Shell" dialog in Windows Explorer provides you with a list of all server projects within the network enabled for configuration, including more information on the selected server, e.g., which mode is currently active (configuration/Runtime).

Several clients can open and edit the same project simultaneously.

Note

Activating server project in Runtime

If you have opened a server project for processing through the client and execute the "Activate Runtime" command in WinCC, you must observe the following:

If you activate Runtime from a client in a multi-user system, only the client project is activated even if the server project is open.

To activate the server project, use the command "Activate Remote" in the "Simatic Shell" dialog.

The same applies to the "Deactivate Runtime" command.

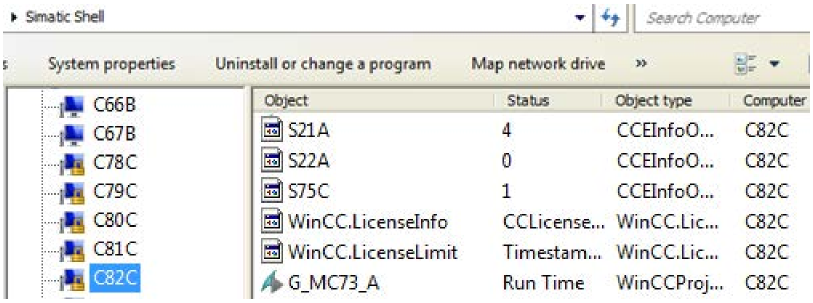
Requirement

In order to open a server project on a client for remote editing, the following conditions must be fulfilled:

- The user registered on the client has the operator authorization for "Configure remote" in the server WinCC project.
- The client has been entered in the server's computer list.
- The project has been enabled for network access

Procedure

1. In Windows-Explorer of the client, select the "Simatic Shell" entry.
   The "Simatic Shell" window is displayed.
   The navigation window contains all servers and projects currently available on the network.
   To open the "Simatic Shell" dialog as an independent window, select "Open in new window" in the shortcut menu of the "Simatic Shell" entry in Windows Explorer.



Object	Status	Object type	Computer
S21A	4	CCEInfoO...	C82C
S22A	0	CCEInfoO...	C82C
S75C	1	CCEInfoO...	C82C
WinCC.LicenseInfo	CCLicense...	WinCC.Lic...	C82C
WinCC.LicenseLimit	Timestam...	WinCC.Lic...	C82C
G_MC73_A	Run Time	WinCCProj...	C82C

2. Select a computer in order to display the projects specific to this computer.
3. In the project list, select the "Open" entry from the shortcut menu of the project you want to open.
   The login dialog opens.
4. Enter the user name and password for the current computer.
   Passwords are case-sensitive.
   In the dialog "WinCC Explorer - Server not available", click the "Start server locally" button.
   The project is opened on the client for configuration.

See also

How to Deactivate a Project (Page 87)

How to Activate a Project (Page 85)

How to Edit Server Project Pictures (Page 84)

Access to Projects from Several Clients (Page 80)

How to Access Computers Outside a Subnet (Page 76)

Remote Configuration (Page 73)

Client/Server Systems in WinCC (Page 16)

1.9.6 How to Edit Server Project Pictures

Principle

Pictures on a server can be opened, edited and saved on a remote client. If Runtime is active while editing, the modifications take effect the next time the picture is activated.

Since each picture is stored in an individual file, only one client can access a picture at a time. Access is blocked to the picture for other computers.

Requirement

- The project folder on the server must be enabled for network access.
- The user registered on the client has the operator authorization for "Configure remote" in the server WinCC project.

Procedure

1. In Windows-Explorer of the client, select the "Simatic Shell" entry. The "Simatic Shell" window appears.
   All currently available servers and projects in the network are displayed in a navigation window.
2. From the project list, select the project to be opened and select the "Open" command from the shortcut menu.
   A Login dialog appears. Enter the user name and password for the current computer. Passwords are case-sensitive.
   In the dialog "WinCC Explorer - Server not available", click the "Start server locally" button. The project is opened on the client for configuration.
3. Open the desired picture in Graphics Designer on the client.
4. Edit the picture and save it again in the project folder on the server.

See also

How to Deactivate a Project (Page 87)

How to Activate a Project (Page 85)

How to Open a Project for Editing (Page 82)

Access to Projects from Several Clients (Page 80)

How to Access Computers Outside a Subnet (Page 76)

Remote Configuration (Page 73)

Client/Server Systems in WinCC (Page 16)

1.9.7 How to Activate a Project

Principle

A client/server system not only provides the option of remote project configuration but also to activate and deactivate them remotely.

If you activate a server project from a client by using the "Simatic Shell" dialog, only the server project is activated. However, if you have opened a server project for processing and activate it in WinCC by using the "Start Runtime" button in the toolbar, only the client project is activated even if the server project is open.

Note

You can only activate Runtime if the project is on the local computer.

Requirements

In order to open a server project on a client for remote activation, the following conditions must be fulfilled:

- The user registered on the client has the operator authorization for "Activate remote" in the server WinCC project.
- The client has been entered in the server's computer list.
- The project has been enabled for network access.

Procedure

1. In Windows-Explorer of the client, select the "Simatic Shell" entry. The "Simatic Shell" window is displayed.
   All servers and projects available on the network as well as their current status are displayed.
2. Select the project to be activated.

3. Select the "Activate remote" command from the shortcut menu.
   A Login dialog appears.
4. Enter the user name and password for the current computer. The project is activated on the server.

Note

Passwords are case-sensitive.

How to activate an OS computer remotely from an Engineering Station

The following requirements generally apply to an OS project and remote activation:

- Enter the ES computer name as OS server in the WinCC project for the OS project.
- If you want to use another computer to remotely activate the OS project, this computer must be entered as client computer.

Because the computer name for server and client cannot be identical in a WinCC project, you must note the following procedure:

1. Change the ES computer name entered under "Server" in the computer list to a fictitious name in WinCC Explorer.
2. Close the project.
3. Open the project.
4. Add a new client in the computer list in the WinCC Explorer.
5. Enter the ES computer name under "Client" in the computer list.
6. Load the target system using SIMATIC Manager.
7. You can now remotely activate Runtime on the OS computer from the Engineering Station.

See also

How to Deactivate a Project (Page 87)

How to Edit Server Project Pictures (Page 84)

How to Open a Project for Editing (Page 82)

Access to Projects from Several Clients (Page 80)

How to Access Computers Outside a Subnet (Page 76)

Remote Configuration (Page 73)

Client/Server Systems in WinCC (Page 16)

1.9.8 How to Deactivate a Project

Principle

A client/server system not only provides the option of remote project configuration but also to activate and deactivate them remotely.

If you deactivate a server project from a client by using the "Simatic Shell" dialog, only the server project is deactivated. However, if you have opened a server project for processing and deactivate it in WinCC by using the "Stop Runtime" button in the toolbar, only the client project is deactivated even if the server project is open.

Requirements

In order to open a server project on a client for remote deactivation, the following conditions must be fulfilled:

- The user registered on the client has the operator authorization for "Activate remote" in the server WinCC project.
- The client has been entered in the server's computer list.
- The project has been enabled for network access

Procedure

1. In Windows-Explorer of the client, select the "Simatic Shell" entry. The "Simatic Shell" window is displayed.
   In the "Simatic Shell" dialog you can view the enabled servers and projects of your client/server system available through the network.
2. Select the project to be activated.
3. Select the "Deactivate remote" command from the pop-up menu. A Login dialog appears.
4. Enter the user name and password for the current computer. The project is deactivated on the server.

Note

Passwords are case-sensitive.

See also

How to Activate a Project (Page 85)

How to Edit Server Project Pictures (Page 84)

How to Open a Project for Editing (Page 82)

Access to Projects from Several Clients (Page 80)

How to Access Computers Outside a Subnet (Page 76)

Remote Configuration (Page 73)

Client/Server Systems in WinCC (Page 16)

1.10 Use of the OPC Interface in Client/Server Systems

Principle

OPC (OLE for Process Control) is a worldwide communication standard for components in the automation industrial sector.

Developed from Windows-based technology, the OPC provides an open interface which enables problem-free, standardized data exchange between PLCs, operating and monitoring systems and office applications from different manufacturers.

Note

Leading companies involved in the automation industry cooperate within the "OPC Foundation".

Additional information on the OPC Foundation is available on the Internet under the following address: "http://www.opcfoundation.org"

Using OPC in WinCC

Used within a distributed system, each WinCC server can monitor the entire system. A WinCC server, however, only assumes a specific range of tasks, for example, such as message editing or archiving.

The WinCC OPC servers enable OPC access to the WinCC Runtime data via the software interface. The WinCC OPC servers support the full functional scope complying with the corresponding OPC specifications.

As OPC client, any software can be implemented which is based on the respective OPC specification. In this way, the OPC client can be used, for example, to analyze various sources. Proprietary OPC clients may be created to best meet specific requirements.

In order to operate the WinCC OPC server mode, the Connectivity Pack license must be installed on the computer which is to be used as the WinCC OPC server. No Connectivity Pack License is required for the OPC DA sever.

The OPC interface is installed on the client and server with the WinCC installation.

The OPC servers from WinCC support the following specifications:

- OPC Data Access 2.05a, 3.00
- OPC XML Data Access 1.01
- OPC Historical Data Access 1.20
- OPC Alarm & Events 1.10
- OPC UA 1.03

Detailed information about the use of the OPC interface in WinCC is available in the WinCC Information System under "Communication".

See also

Client/Server Systems in WinCC (Page 16)

Functionality of OPC (Page 682)

File Server 2

2.1 Setting Up the File Server

Introduction

The WinCC file server is a server with minimum configuration of WinCC components.

You can save projects on the file server and manage them centrally.

This facilitates the regular creation of backup copies of all projects.

Note

Use for configuration only

You use the file server exclusively for configuring.

Requirement

The requirements described in the Installation Notes apply to installing a WinCC Fileserver V8.

The following conditions also apply:

- The computer must be available in the network (LAN).
- If you want to use the file server, you need administrator rights.

Note

PC without WinCC basic installation

WinCC V8 and WinCC Fileserver V8 cannot be installed at the same time on a computer.

Installation

In order to set up a computer as a file server, you run the Fileserver Setup on the computer.

1. Start the WinCC installation DVD.
2. Select the installation type "Custom Installation".
3. In the "WinCC" group of the "Programs" dialog, select the entry "WinCC Fileserver".

The minimum installation for WinCC is installed on the computer.

Configuration

The projects are stored on the file server.

Enable access

To enable all project members to access the projects, you must share the corresponding drives or folders on the file server.

To share folders or drives, you must have Windows administrator rights.

Assign the shared folders or drives with unique drive letters on the configuration computers.

Project members can then open the projects on the file server like a local project.

WinCC ServiceMode 3

3.1 WinCC ServiceMode: Standard Project and Service Project

WinCC ServiceMode provides the option of operating WinCC Runtime as a service.

WinCC Runtime can also be active as a service when no interactive user is logged into the computer.

This section shows you:

- In which configurations the WinCC ServiceMode can be used
- How to configure a project as a service project
- How a service project is activated

Overview

Your can configure a WinCC project as a standard project or as a service project.

To operate a WinCC project in WinCC ServiceMode, you must configure it as a service project.

Standard project

In order to run WinCC Runtime, a user must be logged into the computer.

Interactive user inputs are possible.

Service project

WinCC Runtime can also be run on the computer when no interactive user is logged into the computer.

WinCC Runtime can also be operated with a logged-in user. In this case interactive user input is possible.

Note

WinCC is not executable when the system is being accessed

Changes to the processes and services of WinCC in the Control Panel and in the Windows Task Manager are not allowed.

The following changes are affected:

- Changes to the properties
- Manual accesses:
  - Starting
  - Exiting
  - Stop
  - Resume
  - Restart
- Priority change

There are dependencies between the individual processes and services.

Do not make any changes.

See also

Using a service project and restrictions (Page 96)

Mode of operation of a service project (Page 100)

How to activate a service project (Page 107)

3.2 Configurations for a service project

Overview

WinCC Runtime can run as a service project on the server in the following configurations:

- WinCC Server with Windows Server operating system
  WinCC clients with their own project
  WinCC clients without their own project
- WinCC WebNavigator server or dedicated Web server
  WinCC Web clients
- DataMonitor server or dedicated DataMonitor server
  DataMonitor clients

3.3 Using a service project and restrictions

Use

On the server, the project in WinCC service mode is operated as a service project.

WinCC Runtime starts as a service.

A service project is started automatically or manually.

Operation without logged on user

A service project can run without an interactive user being logged on to the computer.

If no interactive user is logged on, no interactive operation is possible.

Operation with logged on user

Interactive operation is not generally desired in service projects.

An interactive user can log on e.g. for service purposes. In this case, the user can activate the interactive operation of the service project.

Autostart

With automatic start, WinCC Runtime is automatically started when the server is turned on and the set project is activated.

The automatic start can be performed without an interactive user being logged on.

Manual start

With a manual start, the user must log on to the server and then activate the project.

When the user logs off the server, WinCC Runtime continues to be active.

User logon and logoff

While the service project is active, interactive users can log on and off the server at any time.

Limitations

A service project is subject to the following restrictions:

Scripts

Since an interactive user is not normally logged on to service projects, C scripts and VB scripts e.g. lead to problems in the following cases:

- If scripts require interactions, e.g. inputs.
- The scripts open message boxes.

There is no common data area for C scripting in the service mode.

Thus, for example, no global C variables can be exchanged between "Global Script" and the "Graphics Designer".

Additional programs or tasks

With a service project, you cannot add additional programs and tasks to the startup list.

Non-released components

OPC access via Connectivity Station is not released for a service project.

Diagnostics information for a service project

As a general rule, a user is not logged on to a server with an activated service project.

WinCC cannot show diagnostics information on the server. WinCC thus forwards diagnostics information to the clients.

You can find additional information on this in the WinCC Information System under "Working with WinCC > Working with projects > Appendix > WinCC diagnostics window and license information".

Note

Editing or migrating service projects

in order to edit or migrate a service project, you need to administer the ServiceMode user accordingly on the computer.

If the ServiceMode user is not available, the logged on Windows user must have been administered accordingly for editing or migrating the project.

See also

How to configure Autostart for a service project (Page 105)

WinCC ServiceMode: Standard Project and Service Project (Page 93)

3.4 WinCC status and control in the system tray

Introduction

WinCC shows the "SIMATIC WinCC" symbol in the Taskbar Notification Area, the so-called tray area.

This symbol provides information on the project status.

The WinCC project can be activated and deactivated via the symbol's shortcut menu.

Project status

The following table shows which project status goes with which "SIMATIC WinCC" symbol:

Icon	Status
	• WinCC is not active. • No project is open.
	WinCC changes the status: • WinCC opens a project. • WinCC activates a project. • WinCC deactivates a project. • WinCC closes a project.
	Project is open.
	The project is activated.
	Project is activated and the server has the "Fault" status.

Control Options via the Pop-up Menu

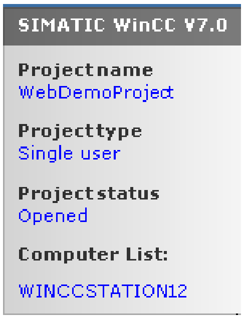
The shortcut menu of the "SIMATIC WinCC" symbol provides the following functions:

- Enabling runtime
- End Graphics Runtime
- Deactivate Runtime
- Close project
- Runtime start options (autostart configuration)
- Open diagnostics window
- WinCC license analysis

SIMATIC WinCC® Window

To open the "SIMATIC WinCC" window, click on the "SIMATIC WinCC" symbol.

Example: Window with Runtime activated



SIMATIC WinCC V7.0

Projectname
WebDemoProject

Projecttype
Single user

Projectstatus
Opened

Computer List:

WINCCSTATION12

The window shows the following information:

- Project name
- Project type
- Project status
- Computer List
  The local computer is represented in blue.

Computer List

The computer list contains all computers on the network.

If the project is activated, the connection status of all existing computers is shown.

The following table shows the icons of the connection status and their meaning:

Icon	Status
🟥	• No connection • Connection disconnected
🟦	• Local computer • Redundant partner server
🟩	Connected • With standby server • With master server, but standby server is the preferred server
🟩★	Connected • With master server • With standby server as preferred server

This view only provides information on the status of the PCs in the network.

To query the connection status for the controller, use the "Status of driver connections" function in WinCC Explorer or the system tag "@<Connection name>@ConnectionStateEx".

See also

How to activate a service project (Page 107)

3.5 Functionality and Prerequisites

3.5.1 Mode of operation of a service project

Introduction

This chapter describes the mode of operation of a WinCC service project.

Standard project

A standard project is started as follows:

- The user logs on to the system.
- The user starts WinCC Runtime or WinCC Runtime starts automatically.

WinCC Runtime remains active until one of the following cases occurs:

- The user exits WinCC Runtime.
- The user logs off from the system.
  In this case the system terminates WinCC Runtime.

Service project

With a service project, WinCC Runtime is started as a service. Depending on the setting, these services are started at the following times:

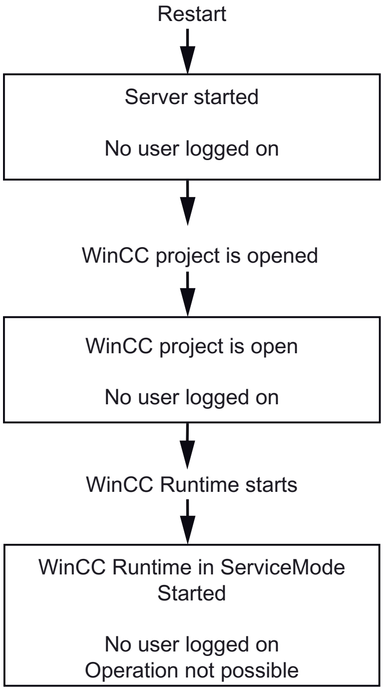
- Automatically after the operating system has started.
- After a user has logged on and started WinCC Runtime.

WinCC remains even if the user logs off again.

The WinCC Runtime data is still accessible.

A logged on user can activate runtime operation as required.

The following diagram shows the states between starting the server and automatic Runtime start with a service project.



See also

WinCC ServiceMode: Standard Project and Service Project (Page 93)

How to activate a service project (Page 107)

3.5.2 Requirements for running a service project

Interactive operation is not generally desired in service projects.

Scripts

Since an interactive user is not normally logged on to service projects, C scripts and VB scripts e.g. lead to problems in the following cases:

- If scripts require interactions, e.g. inputs.
- The scripts open message boxes.

Service project in a distributed WinCC scenario

You must set up a dedicated Windows user for a service project.

The Windows user configured for the service project must belong to the "SIMATIC HMI" user group.

For a WinCC multi-user system or distributed systems with server-server communication, you can use a local Windows user or a Windows domain user. The user must be created on all servers and clients as member of the "SIMATIC HMI" user group.

Local Windows user

- The user must be a member of the local "SIMATIC HMI" user group on all the computers in the network.
- The password for this user must be identical on all computers.

Windows domain user

- The user is a member of the local "SIMATIC HMI" user group on all computers.
- The user is a member of a group which is in turn a member of the local "SIMATIC HMI" user group.

Passwords

Requirement for an uninterrupted runtime operation of a WinCC service project:

- If you change the password in Windows, you must apply this change on all computers and in the configuration of the WinCC project.
- The password of the configured user cannot expire.

To ensure this, activate the following options when setting up the user:

- "Password never expires"
- To prevent accidental changes, also deactivate the "User cannot change password" option. To avoid inconsistencies, perform the password updates centrally as an administrator in each case.

Change user and password via command line

The "CCStartStop.exe" application enables access to the WinCC project via command lines in the Windows command prompt.

Use the "/su" parameter to update the ServiceMode user and the Windows password in the WinCC project.

Examples

Configure new ServiceMode user:

- `CCStartStop /su /domain:plant011 /user:operator02 /password:MYpa$$w0rd`
  The user "operator02" from the "plant011" user group is adopted as the ServiceMode user with the password "MYpa$$w0rd".

Update password of the configured ServiceMode user:

- `CCStartStop /su  /password:NEWpa$$w0rd`
  The password of the logged-in user is changed to the new password "NEWpa$$w0rd".

More information:

- "Working with WinCC > Working with Projects > Managing WinCC project via Windows prompt"

See also

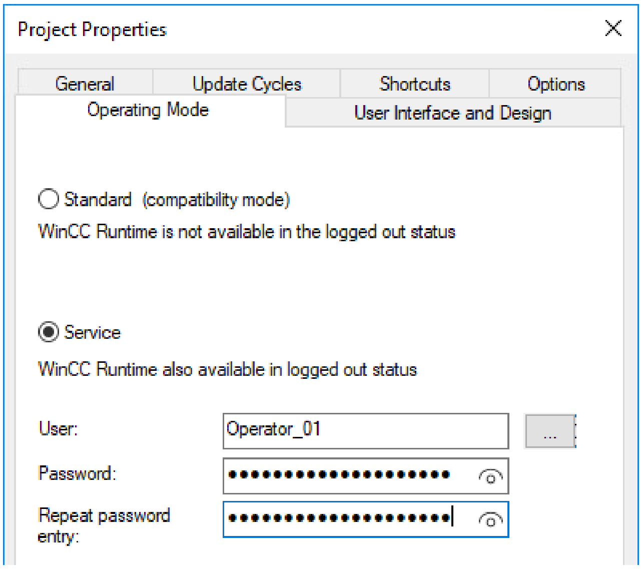
How to configure Autostart for a service project (Page 105)

3.6 Configuring WinCC ServiceMode

3.6.1 How to define a project as a service project

Introduction

Specify in the project properties whether the project is run as a standard project or a service project.



○ Standard (compatibility mode)
WinCC Runtime is not available in the logged out status

● Service
WinCC Runtime also available in logged out status

User: Operator_01
Password: ••••••••••••••••••••
Repeat password entry: ••••••••••••••••••••

Procedure - defining a service project

1. Select the project name in the navigation window of WinCC Explorer.
2. Select the "Properties" entry in the shortcut menu of the project.
   The "Project Properties" dialog opens.
3. Switch to the "Operating mode" tab.
4. Enable the "Service" option.
   WinCC shows a message that the project needs to be reloaded in order to convert the project.
5. Enter the user in the "User" field, under which the WinCC service project will run.
   For additional information on the required properties of this user: "Requirements for running a service project (Page 101)"
6. Enter the associated password in the "Password" field.
7. Confirm the password in the "Password" field.
8. Confirm your entries by clicking "OK".
9. Reload the project.

Converting a service project to a standard project

If you want to convert a service project to a standard project, check the "Standard" option on the "Operating mode" tab.

Converting a standard project to a service project

If you want to convert a standard project to a service project, check the "Service" option on the "Operating mode" tab.

Note

Converting a standard project to a service project not generally possible

A service project is subject to the restrictions. Note these before you perform a conversion.

You can find additional information under "Using a service project and restrictions (Page 96)".

See also

How to configure Autostart for a service project (Page 105)

3.6.2 How to configure Autostart for a service project

Setting up Autostart

WinCC activates the selected WinCC project when the computer is booted.

In the "AutoStart Configuration" dialog, select the desired service project as Autostart project.

Note

Project activation during service restart

When the service "SIMATIC WinCC CCProjectMgr" restarts, the Autostart project is also reactivated.

Changing the operating mode: Set up Autostart again

After converting the service project to a standard project and vice versa, you must reconfigure "Autostart".

Procedure - setting up autostart

1. In the "Siemens Automation" Windows program group, select the entry "AutoStart".
   The "AutoStart Configuration" dialog opens.
   The settings of the local computer are displayed.
2. Select the desired computer.
   You have the following options:
   - Enter the computer name.
   - Select a computer in the network path via "...".
   - To set up Autostart for the local computer, click "Local Computer".

   To display the current configuration of the selected computer, click "Read configuration".
3. Select the service project by clicking the ... button in the "Project" field.
   The project file and its full path are entered in the box.
   The project type is displayed under the path.
4. Configure the settings for the autostart behavior.
5. Activate the option "Autostart active".
   If the option is deactivated, autostart is not executed for the configured computer.
6. Confirm your settings with "Apply" and close with "OK".

Result

The next time you boot the computer, WinCC starts automatically and the selected project is opened.

"Disable operating system access at startup" option

If the option is activated, the start screen of WinCC is displayed immediately when the PC is booted. The Windows desktop is not visible.

However, the option is only activated in ServiceMode if a user is logged in when starting in ServiceMode. If no user is logged in, the behavior is always the same as starting with the option deactivated.

See also

How to define a project as a service project (Page 104)

Using a service project and restrictions (Page 96)

3.7 Service Project in Runtime

3.7.1 How to activate a service project

Introduction

WinCC creates the "SIMATIC WinCC" icon in the taskbar notification area, the so-called system tray:

You can use the shortcut menu of this icon to perform the following functions, among others:

- Enabling runtime.
- End Graphics Runtime.
- Deactivate Runtime.

For more information, visit "WinCC status and control in the system tray (Page 98)".

Requirements

The project must be saved as a service project.

You can find additional requirements under "Requirements for running a service project (Page 101)".

Automatic start in WinCC ServiceMode

If automatic start was correctly configured for the project with the "Autostart Configuration" tool, the following is carried out:

- The project is automatically activated as soon as the server is started up.

User input is not required.

Result

The project has been activated.

No user is logged on to the server.

Manual start in WinCC ServiceMode

The following procedure assumes that automatic start is not configured for the project.

1. Start the server.
2. Log on to the server.
3. Open the project.
4. Select the command "Activate Runtime" in the shortcut menu of the icon in the system tray. Alternatively, activate the project with the WinCC Explorer.

Result

The project is activated.

WinCC displays the icon .

To ensure that WinCC Runtime remains active when you log off from the server, only exit the WinCC Explorer. To do this, select the "Exit WinCC Explorer" entry in the dialog "Exit WinCC Explorer".

WinCC Runtime then remains active.

See also

Requirements for running a service project (Page 101)

WinCC ServiceMode: Standard Project and Service Project (Page 93)

Mode of operation of a service project (Page 100)

WinCC status and control in the system tray (Page 98)

3.7.2 Show to log into and off of an activated service project

Introduction

You can log into the service and log off again, while a WinCC project is in Runtime, in order to perform necessary work on the server.

Note

If updates are installed that require a restart, WinCC Runtime is ended.

Requirements

A service project is activated. No user is logged into the server.

Procedure

1. Log into the server.
2. Perform the desired actions.
3. Log back off of the server.

Results

You logged into and off of the server. WinCC Runtime is not affected.

3.7.3 How to activate the interactive operation for service purposes

Introduction

You can activate the interactive operation while a WinCC service project is in Runtime.

Requirements

A service project is active. The interactive operation is not activated.

You log in as a user who is a member of the "SIMATIC HMI" group.

Procedure - Activating the interactive operation

1. Log into the server.
2. Select the "Start Graphics Runtime" command from the pop-up menu of the icon in the tray area.
   WinCC releases the interactive operation. You can operate the WinCC project.

Procedure - Ending the interactive operation

1. Select the "End Graphics Runtime" command from the pop-up menu of the icon in the tray area.
   WinCC ends Graphics Runtime.
2. Log off if necessary.

Redundant Systems 4

4.1 Redundancy

Content

The WinCC option "WinCC/Redundancy" is used to configure a redundant system. The availability of WinCC and the system are enhanced by parallel operation of two interconnected servers and automatic switching of the servers in the event of a malfunction.

Overview

This documentation shows you the following:

- The requirements for a redundant system.
- How to create a redundant system in WinCC.
- How to configure the server for redundancy.
- How to configure the synchronization of the redundant archive.

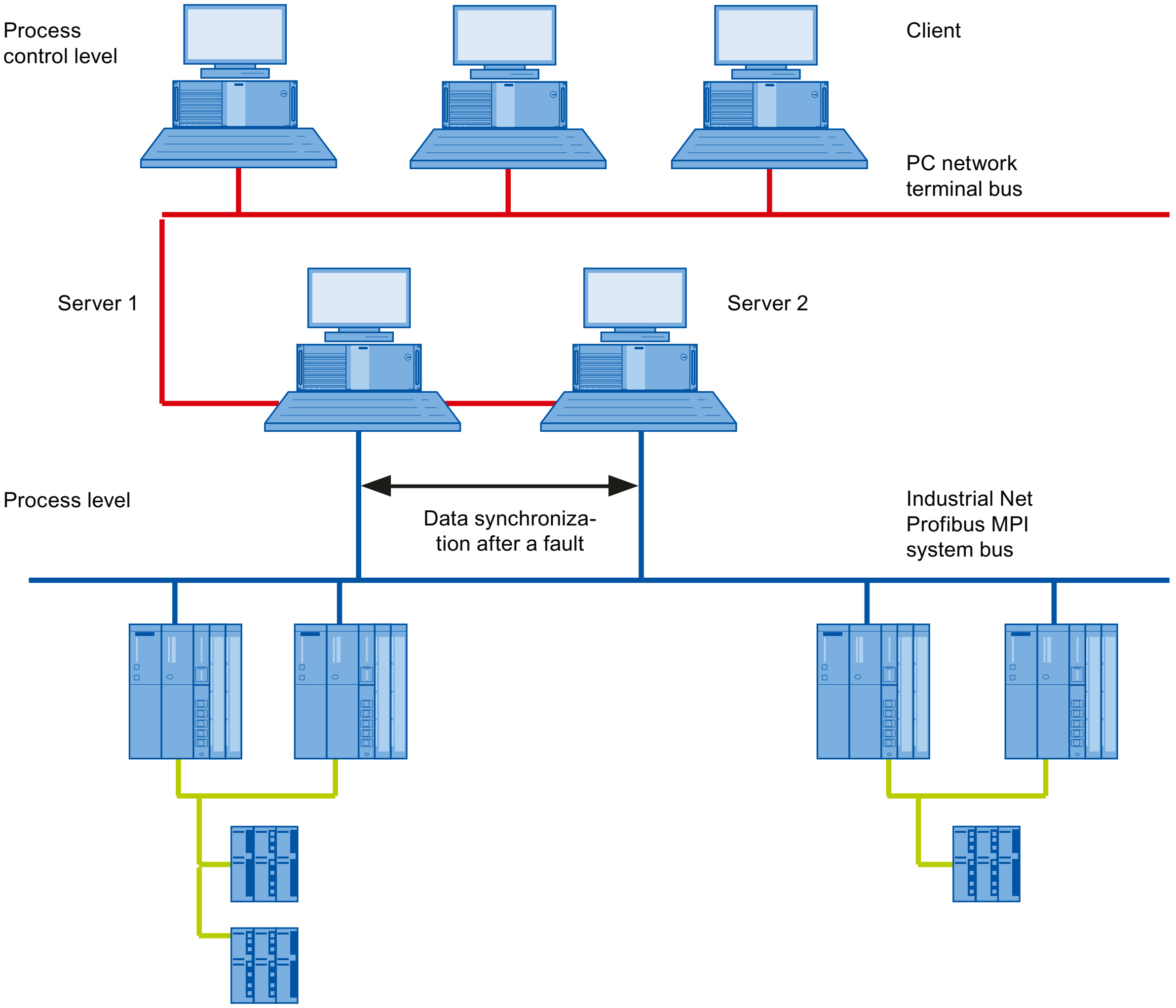
4.2 WinCC Redundancy

Introduction

A redundant WinCC project consists of two WinCC servers configured to perform the same functions and operating in parallel:

- A master server
- A standby server

The two servers are connected to the automation system, the clients and each other.



Overview of the functions of WinCC Redundancy

WinCC Redundancy provides the following functions:

- Automatic switching of clients if a server fails or the process connection fails.
- Automatic synchronizing of message archives, process value archives and user archives after a failed server has been restored or the process connection fault has been eliminated.

- Online synchronization of internal messages.
- Online synchronization of internal tags that support tag synchronization.
- Online synchronization of user archives.
- The "Project duplicator" for copying a project to the redundant server.
- Monitoring of WinCC applications via the "Application Health Check" function.
- Monitoring of the hardware and software of the local system via the "SelfDiagnosis" function.

The "Application Health Check" function

The "Application Health Check" function automatically monitors all important WinCC applications.

After detecting a software error, the lifebeat monitoring triggers the following actions:

- In the system tag "@RedundantServerState" the server state changes to "Fault".
- The connected clients switch over to the redundant partner server.
- A process control message notifies users of the software error.
  A process control message cannot be triggered if the alarm server caused the failure.

Note

Server restart after error

If the "Application Health Check" function detects a software error and client switching was initiated, the relevant server must be restarted.

Only after the server restart is it possible to reconnect clients to this server.

The archives are synchronized retroactively up to the point where the software error was detected.

The "SelfDiagnosis" function

The "SelfDiagnosis" function comprises the following tasks to ensure availability and stability of the redundant system:

- Monitoring and reporting local HW and SW problems
- Monitoring local system performance
- Monitoring the state of the data volume
- Server fail-over, if necessary

The following tasks are performed in case of malfunction:

- Restart of applications
- If necessary, the server state is set to "Fault" and the servers change over.
- A log entry is generated.
- A system alarm is triggered.

4.3 Requirements for redundant systems

Overview

The following prerequisites must be fulfilled for WinCC Redundancy:

- For redundant WinCC servers with multi-user operation, you can only use computers with server operating systems.
- The WinCC Redundancy option must be installed on both servers. The WinCC Redundancy license must be installed on the redundant servers.
- The two redundancy servers must be configured functionally identical.
- You may not configure any further PCs as redundant servers in addition to the two redundant servers.
- The servers have to be time-synchronized servers. Time synchronization of the entire system is recommended. The time synchronization can be configured with the "Time synchronization" option in WinCC.
- Messages and acknowledgments from the automation systems and clients must always have a time stamp in the frame (chronological messaging). This prevents duplicate entries. For example, use alarm blocks in the automation systems.
- Process values, messages and active message blocks from the lower-level automation systems have to be sent to both servers at the same time.
- One of the following additional connections must exist between the redundant servers:
  - Network adapter
  - Serial connection

  This additional connection ensures exact definition of the "Master" or "Standby" status. You configure the additional connection via the network card in the WinCC Explorer using the "Redundancy" editor. Use the TCP/IP protocol with the corresponding IP address. The IP address must not be in the same subnet as the terminal bus.

Note

Runtime behavior during commissioning of WinCC and activated WinCC Redundancy

During commissioning, WinCC Runtime is often activated and deactivated on the server computers. This repeated starting with an activated WinCC Redundancy causes the archives to be synchronized every time. This may result in a notable deterioration of the WinCC runtime behavior. We therefore recommend that you deactivate WinCC Redundancy during commissioning.

Uninterruptible power supply

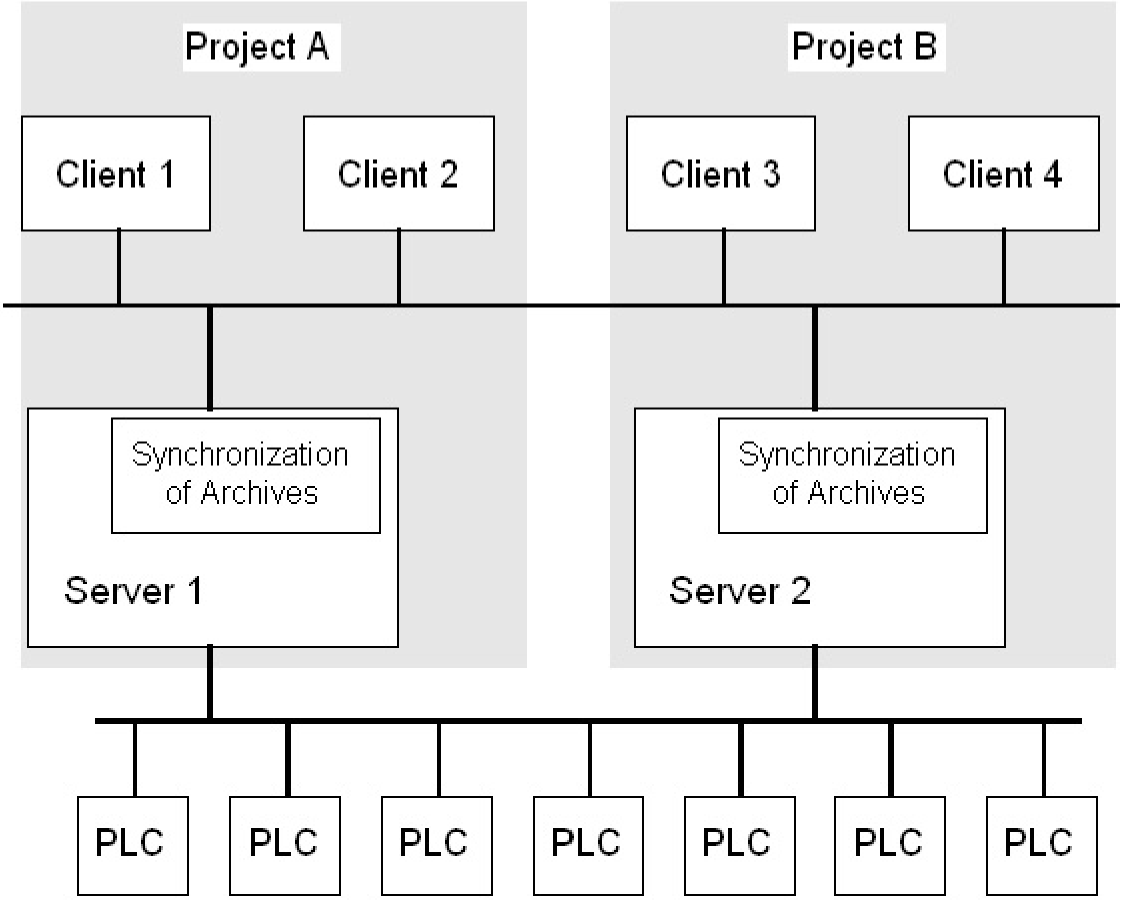
To safely exit WinCC in the event of a power failure, the use of an uninterruptible power supply (UPS) is recommended.

4.4 How Redundancy works

Introduction

Both servers have equal rights and work independently of each other. Both are available to the user. If one of the servers fails, an equal redundant server will always be available.

The following picture shows the archiving and the archive synchronization of parallel-redundant servers.



Identification of the redundant servers

One of the two servers is configured as the default master.

The system variable "@RM_MASTER" is set to "1" in runtime for this server. If the status of the tag changes, for example, due to a computer failure, the clients switch over to the "standby" computer. The previous "Standby" computer is now the master.

The servers monitor each other in runtime to allow for an early detection of a failing partner server.

An additional connection via network adapter or serial connection between the servers is used to monitor the status. The connection leads to an improvement in the communication between the redundant partner servers. This increases the availability of the redundancy. The additional connection is not used to synchronize the archives.

WinCC archiving in normal operation

The servers usually run completely parallel in runtime. Each server computer has its own process driver connection and has its own data archives.

The process data and messages are sent by the automation systems to both redundant servers and are processed by both redundant servers accordingly.

User archives, internal messages and internal tags can be continuously synchronized online.

The two servers communicate via LAN with the TCP/IP protocol to synchronize the archives.

Failure of a server

If one server fails, the clients are automatically switched from the failed server to the redundant partner server. This ensures that all clients are always available for monitoring and operating the processes.

During the failure, the active server will continue to archive all messages and process data of the WinCC project.

After the failed server comes back online, the contents of all message archives, process value archives and user archives are automatically copied to the returned server. This will fill the archive data gaps of the failed server.

Note

Redundancy failure: At least 69 seconds

For technical reasons, the downtime must last at least 69 seconds for the automatic synchronization of the two systems.

Factors triggering the client switch

The switch of the clients from the default (master) server to the partner server during a server failure is performed automatically by the system.

The following factors cause a switch of servers:

- Network connection to server failed
- Server failure
- Malfunction of process connection
- The "Application Health Check" function has detected a defective WinCC application and triggers a switchover.
- The project is deactivated.

If the redundancy option for client switchover in case of a fault in the process connection is activated, the number of defective logical connections to the "Master" server and the redundant partner server is cyclically determined.

If the "Master" server has more defective logical connections than the redundant partner server, a client logged on to the "Master" server will be switched over to the redundant partner server.

Once the error in the process link has been eliminated, a client is switched back to the preferred server to which it was originally connected.

Monitoring of the process link will not be started until both redundant servers are in runtime.

Note

Software error on the server

In the event of a software error on the server it is possible for connected clients not to be switched over to the redundant partner but for the system to be blocked.

Factors triggering archive synchronization after the server returns

The synchronization of the archives between the servers will be initiated after the following errors have been corrected:

- Process connection error. You can, however, deactivate the process connection monitoring.
- Network connection failure to the partner server.
- Server failure.
- Project is not activated.

Synchronization after the server returns

After the failed server has returned, WinCC Redundancy transfers the missing data to the failed server.

This applies to the message archives, process value archives, user archives and internal tags.

For message archives and process value archives, all segments that were active during the downtime are transferred. Two equivalent servers are available once again after the transfer.

The archive synchronization is implemented as a background function and runs parallel to the process management and archiving of WinCC. Therefore the operation and observation of the system is guaranteed at all times.

Comparing internal tags

The internal tags must have the property "Tag synchronization".

Internal tags are compared on partner computers as soon as one of the tags is modified on one of the redundant servers.

The internal tags also include the system tags whose name starts with the "@" character, such as "@RM_Master". You may not configure an online synchronization for system tags.

Synchronization after process connection error

If you have activated the process connection monitoring, synchronization of all archives is started automatically after the fault between a server and the automation systems has been eliminated.

If process connection monitoring has been activated, the respective server carries out lifebeat monitoring on all configured connections. A server detects that the process connection to an automation system is faulty when the addressed automation system fails to send an acknowledgment back to the server.

If a network fault to one or more of the automation systems was found, synchronization of all archives of the automation systems belonging the project is carried out. The archives of the automation systems that have not failed are also synchronized. If this option is deactivated, the runtime loads on the servers are prevented.

Because an error in the network of automation systems is not recognized when the monitoring of network connections is deactivated, no archive synchronization will take place.

Online synchronization

Direct server-to-server synchronization is supported:

- With alarm logging for:
  - Internal message tags
  - Messages without tag connection
  - System operation messages
  - "Batch" messages
- For user archives
- For internal tags with tag synchronization

Comparing blocked messages

When a failed server is restored, currently blocked messages are searched and synchronized through a general query of the automation systems.

If a message is blocked passively on only one server, the blocking information is synchronized.

4.5 Configuring the redundant system

4.5.1 Guide to setting up a redundant system

Introduction

Here you get an overview of how a redundant WinCC system is set up. You can find general information on the structure of a client-server system in the WinCC Information System section entitled "Distributed Systems".

Entering the servers in Windows

The two redundant servers must recognize each other on the network. In addition, users/passwords must be identical on the redundant servers. You have to set up the users with Administrator or User rights. Users have to be members of the "SIMATIC HMI" user group.

Configuring the project on the server

The following is determined during the configuration of the WinCC redundancy:

- The standard master.
- The partner server.
- The switchover behavior of the clients.
- The type of archive synchronization.

Before duplicating the project, create the server package by using the "Server data" editor in WinCC Explorer. Create a server package preferably on the standard server.

Note

Only configure the user archives for the synchronization that you really need. The greater the number of user archives to be synchronized, the longer the synchronization process will take and the greater the system loads will be.

Duplicating the WinCC project

To have a functionally equivalent WinCC project on the redundant partner server, duplicate the project from the default server using the "Project Duplicator". The master server and the standby server then have the same project settings.

Note

Before duplicating, make sure there is sufficient memory on the computer on which the project is duplicated. If you are duplicating an existing project, this project may not be open.

Configuring the standby server

To monitor the status of the redundancy, you still have to set the additional connection to the master server on the standby server in the "Redundancy" editor.

Configuring the clients

To use WinCC Redundancy on the clients, follow these steps in the "Serverdata" editor:

- Create the package of the default server
- Set the preferred server and activate the automatic updating of packages.

Activating the redundancy servers

1. Activate initially the configured Master server.
2. Next start up the connected clients.
3. When the clients are active, activate the second server and its connected clients.

The first synchronization is now carried out. The downtime for this synchronization encompasses the interval between activating the first and second server.

Note

Please note during startup of redundant servers that the first server must be started completely prior to activating the redundant partner. During initial startup of servers, no clients must be active.

Once you have completely deactivated a redundant server pair, you must adhere to a specific sequence during reactivation. Activate the server first which was the last server to be deactivated. Once this server has been completed started, you can activate the redundant partner.

Deactivating a redundant server

Please note that prior to deactivating a redundant server, the second server must be functional and operating without errors.

Archive synchronization must be completed prior to deactivation as indicated by the corresponding process control message.

Note

Data losses may occur if you deactivate the second server before the archive synchronization of the first server was completed. This is particularly important in case of frequent switching between activation / deactivation of the servers during commissioning.

See also

How to configure the redundant servers (Page 122)

How to configure the synchronization of user archives (Page 125)

4.5.2 Configuring an identical function

Process data archives and message archives

Tag Logging and Alarm Logging must be configured in a functionally identical way for the redundant servers.

The two servers must have identical archives, whereby additions can be made in the form of additional measuring points or archives.

The extensions are not included in synchronization. You have to coordinate the extensions on the partner server yourself.

WinCC synchronizes the following archives that are located on hard disks:

- Process value archives
- Compressed archives
- Message archives

The synchronization of main memory archives is not performed.

User archives

The user archives require the same structure on both servers.

The configuration of user archives that are going to be synchronized must be identical in terms of their properties as well as field and record structure.

To ensure that functionally equivalent WinCC projects are running on the redundant partner servers, duplicate the project using the "Project Duplicator" after every change.

Note

Synchronization of Changed Configuration Data Not Possible via Load Online Changes

Changes to user archive configuration data, such as deleted fields in the archive, cannot be transferred to a redundant server pair with Load Online Changes.

User administration (User Administrator)

Changes in the user management are not synchronized automatically.

This also applies to the configuration in Runtime via the WinCC UserAdminControl.

If you want to configure changes to the user management, you have the following options:

- Configure the changes on the engineering station.
  Transfer the changes to the redundant servers.
- Configure the changes identically on both redundant servers.

See also

WinCC Redundancy (Page 112)

4.5.3 How to configure the redundant servers

Introduction

You use the "Redundancy" editor in WinCC Explorer to configure redundant servers and synchronization of the archives.

Requirement

- The two redundancy servers must be configured with identical functionality.

Simatic Shell: Configuring and testing the connection to the redundant partner

Use the shortcut menu of the "Simatic Shell" to open the "Redundancy settings" dialog.

This dialog offers an alternative way to configure the connection to the redundant partner server.

When it is opened, existing settings from the "Redundancy" editor are adopted.

Testing connection settings

Use the "Extended check of the network connectivity (Terminal bus)" to test the connection to the redundant partner and to individual clients:

- Availability of the "Default gateway"
- Availability of the stations
  Enter the names or IP addresses of the PCs whose connection you want to test.
  Separate the names and IP addresses with a semicolon without spaces, for example:
  "hostsv1;hostsv2;123.456.78.9"

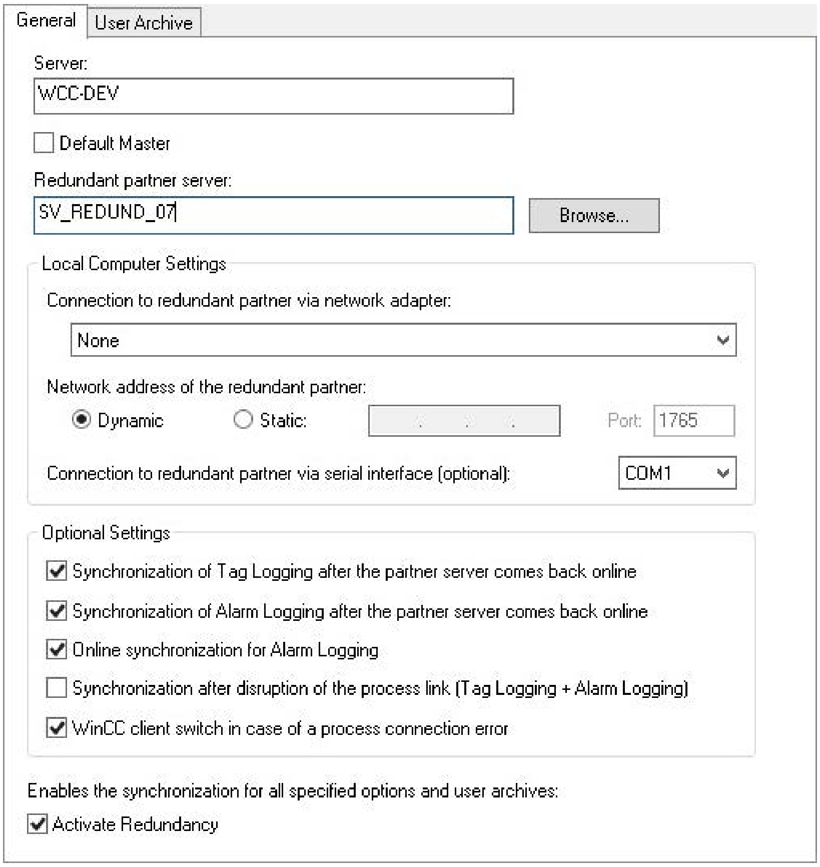
To test the availability, click the "Check" button.

The "Terminal bus info" dialog contains the results for the selected PCs:

- Name or IP address
- Availability via Ping:
- Response time

Procedure

1. Open the "Redundancy" editor in WinCC Explorer.
   Change to the "General" tab.
   The "Server" field contains the name of the current computer for which you are configuring WinCC Redundancy.



Server: WCC-DEV

□ Default Master

Redundant partner server: SV_REDUND_07

Local Computer Settings

Connection to redundant partner via network adapter: None

Network address of the redundant partner:
● Dynamic ○ Static: . . . Port: 1765

Connection to redundant partner via serial interface (optional): COM1

Optional Settings

☑ Synchronization of Tag Logging after the partner server comes back online

☑ Synchronization of Alarm Logging after the partner server comes back online

☑ Online synchronization for Alarm Logging

□ Synchronization after disruption of the process link (Tag Logging + Alarm Logging)

☑ WinCC client switch in case of a process connection error

Enables the synchronization for all specified options and user archives:

☑ Activate Redundancy

2. To configure WinCC Redundancy, select the "Activate Redundancy" option at the bottom of the dialog.
3. To activate the server as master by default at startup, select the "Default Master" option.
   If this option is not selected, the server starts as the standby server.

NOTICE
Only one redundant server can be the "Default Master"
Ensure that the "Default Master" option is only activated for one of the two redundant partner servers.
Configuring both partner servers as default masters may cause problems with redundancy switchover of clients.

4. Enter the computer name of the partner server.
   Alternatively, click "Browse".

5. Specify for the status monitoring whether there is a connection to the redundant partner via a network adapter.
   Connection via a network adapter is to be preferred to serial connection.
   To enter a fixed network address and the port of the redundant partner, select the "Static" option.
   If you want to use a serial connection, select an interface.
   Alternatively, configure the connection settings via the "Simatic Shell":
   - Open the "Redundancy Settings" dialog in the Microsoft Windows Explorer via the shortcut menu of the "Simatic Shell" folder.
   - Select the serial interface, the network adapter, and the network address.
   - If needed, test the connection settings via the extended network connectivity check.
6. To define the synchronization behavior on return or failure, select the desired options:
   - Synchronization of Tag Logging after the partner server comes back online
   - Synchronization of Alarm Logging after the partner server comes back online
   - Online synchronization for Alarm Logging:
     Synchronization of operator messages, messages without tag connection, and messages with internal message tags
   - Synchronization after disruption of the process link (Tag Logging + Alarm Logging):
     The process connection monitoring starts an automatic archive synchronization after the disruption between a server and the automation systems has been eliminated.
   - WinCC client switch in case of a process connection error:
     Clients connected to the server switch to the redundant partner server.
     The scenario is described under "Client switchover in the event of a process connection error (Page 131)".
7. Click "OK" to save the settings.
8. To apply the settings in Runtime, restart Runtime.
   If you do not restart Runtime, only the changes in the following options take effect immediately:
   - Synchronization of Tag Logging after the partner server comes back online
   - Synchronization of Alarm Logging after the partner server comes back online
   - Online synchronization for Alarm Logging
   - Synchronization after disruption of the process link (Tag Logging + Alarm Logging)

   Changes to the other options only take effect after restarting Runtime.

See also

Client switchover in the event of a process connection error (Page 131)

How to configure the synchronization of user archives (Page 125)

WinCC Redundancy (Page 112)

WinCC Redundancy system messages (Page 146)

Guide to setting up a redundant system (Page 119)

Failure scenarios (Page 135)

4.5.4 How to configure the synchronization of user archives

Introduction

User archives can be processed by operations, independent programs or automation systems.

For redundant systems configure the automatic synchronization of the user archives.

Note

Changed configuration data: No synchronization via Load Online Changes

Changes to user archive configuration data, such as deleted fields in the archive, cannot be transferred to a redundant server pair with Load Online Changes.

Runtime restart after change to archive synchronization

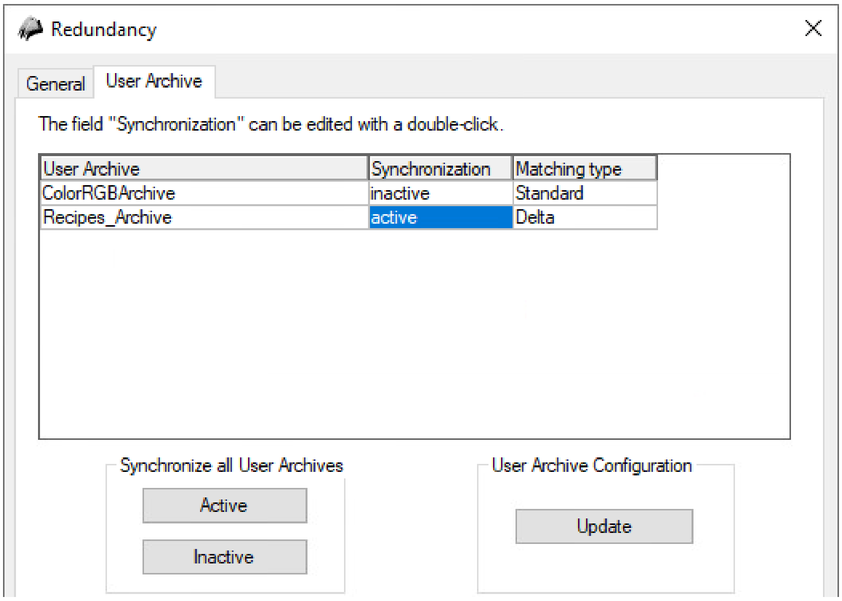
Changes to the archive synchronization with the User Archive only take effect after runtime has been activated again.

Requirement

- The configuration of the user archives must be identical on the two redundant servers. Use the Project Duplicator to this purpose.

Procedure

1. Open the "Redundancy" editor in WinCC Explorer
   In the "User Archive" column on the "User Archive" tab, all configured user archives are displayed in rows.



The field "Synchronization" can be edited with a double-click.

User Archive	Synchronization	Matching type
ColorRGBArchive	inactive	Standard
Recipes_Archive	active	Delta

2. Activate or deactivate the synchronization of the individual user archives by double-clicking the "Synchronization" column.
   The settings have to be identical on both partner servers.

3. The two buttons at the "Synchronization of all User Archives" field are used to activate or deactivate the synchronization of all displayed user archives.

4. If the configuration of the user archives in the "User Archive" editor has changed after the "Redundancy" editor was called, click the "Update" button.
   The current configuration of the user archives is applied.

5. If the project contains very large user archives with more than 100 000 data records, you can use delta synchronization to improve performance during redundancy synchronization.
   For the usual configuration limits, continue to use the default setting.
   To activate the "Delta" option, double click in the "Matching type" column.
   Requirements for delta synchronization:

   – Check that delta synchronization is also activated on the redundant partner server.
     The "Matching type" option must be configured to match on both servers.

   – Configure the user archives only on the respective primary server (master).
     After every change in the WinCC project, use the WinCC Project Duplicator to generate the redundant partner project.

6. Click "OK" to save your settings.

7. To ensure that a functionally equivalent WinCC project is running on the redundant partner server, duplicate the project using the "Project Duplicator".

Editing user archives in parallel

Boundary conditions for the parallel insertion of data records in redundant user archives:

- Records can only be added to a previously failed server if the synchronization is made after the return.
  If the synchronization is not complete, you will get an error message in the script or in the user archive control.
- Even during the online synchronization, some time will pass before the record has been synchronized in the redundant archive.

Note

Failure of both servers: Restart sequence

If both redundant servers have failed or both computers are shut down, you must first start the server computer that was used last.

If you do not adhere to this sequence, changes could be lost.

See also

How to configure the redundant servers (Page 122)

WinCC Redundancy (Page 112)

Guide to setting up a redundant system (Page 119)

WinCC Redundancy system messages (Page 146)

WinCC Redundancy system tags (Page 142)

Failure scenarios (Page 135)

4.5.5 How to Duplicate a Project for Redundant Servers

Introduction

The two redundant servers must be set up with the same hardware and software functions.

After completing the WinCC configuration and after every change in the WinCC project, use the WinCC Project Duplicator to generate the redundant partner project.

The Project Duplicator performs the following:

- Copying of all associated project data, such as pictures, scripts and archives to the redundant partner.
- Configuring all the required settings on the target computer, if the computer is already configured for the use of WinCC Redundancy.

You must change computer-specific settings manually afterward.

Note

To transfer a project to a redundant server, you cannot use the Windows Explorer.

You can save minor changes using the Load Online Changes function in SIMATIC Manager and then transfer them to the servers in runtime.

Principle

Select the project you want to duplicate in the Project Duplicator:

- Specify the target computer and folder in which the project is duplicated.
  The project folder is created in this target folder.
- You cannot duplicate a project on the local computer.
  You always duplicate a project on another computer in the network to which you have access rights.

Depending on the status of the project, you can duplicate the configuration data and the runtime data into the selected folder:

Project Status	Configuration Data	Runtime Data
Project closed	+	+
Project open and deactivated	+	-
Project in Runtime	+	-

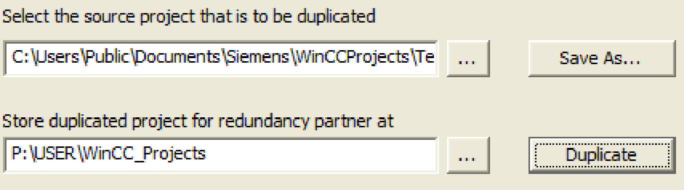
You can only duplicate the entire project and the entire folder structure. You cannot exclude any data or folders from the duplicate operation.

Requirements

- The WinCC Redundancy option is installed on both computers.
- The target folder for the duplication is created on the target computer and is made available for access.
- You have access rights for the target folder.
- The target computer has enough free space on the hard disk.
- The correct WinCC version must be installed on the target computer. The computer must be started.
- Runtime is deactivated on the target computer.
- The project is closed on the target computer.

Procedure

1. In the "Siemens Automation" Windows program group, select the entry "Project Duplicator".
   The WinCC Project Duplicator is opened.



Select the source project that is to be duplicated

C:\Users\Public\Documents\Siemens\WinCCProjects\Te

Store duplicated project for redundancy partner at

P:\USER\WinCC_Projects

2. Enter the project you want to duplicate in the "Select the source project that is to be duplicated" box.
   Enter the path and the <PROJECT>.MCP project file directly or search by clicking the ... button.
3. Enter the path where the duplicated project will be stored in the "Store duplicated project for redundancy partner at" box.
   Enter the path and the <PROJECT>.MCP project file directly or search by clicking the ... button.
4. Click the Duplicate button.
   The "Copy" window is opened. During duplication, the Project Duplicator displays the files and folders with a progress bar. Use the "Cancel" button to stop duplication.
   After duplicating the "Notes on the Project Duplicator" window is opened.
   WinCC indicates the settings that you still need to check.

   Note

   If you duplicate an open WinCC project on the source computer, no progress bar will be displayed.

5. Close the Project Duplicator with the Close button.
6. Check the settings in the duplicated project and change them if necessary, e.g.:
   - Computer name
   - Settings in the Redundancy Editor
   - Settings in other editors
   - Autostart configuration when autostart is configured in the WinCC project

Duplicating a project with project-based access protection

SIMATIC STEP 7 must be installed in order to transfer a WinCC project with project-based access protection to a redundant server.

When you click the "Duplicate" button in the "WinCC Project Duplicator" dialog, you have to enter the password for the STEP 7 project.

If SIMATIC STEP 7 is not installed or you enter the wrong password, the Project Duplicator aborts with an error message.

4.5.6 How to duplicate a redundant project at runtime

Introduction

If you edit a redundant project, you can also update the project on the redundant server during operation.

You can save minor changes with the Load Online Changes function and then transfer them to the servers in runtime. You should also refer to the documentation on the topic of "Load Online Changes".

Duplication using the Project Duplicator

Some configurations cannot be saved by the Load Online Changes function. In this case, you must generate a duplicate of the project to the redundant server using the Project Duplicator.

Note

No Redundancy

For changes during normal operation, you must deactivate one of the partner servers. During this time, no redundancy is available.

Requirements

- The target folder has been created.
- You have access rights for the target folder.
- The redundant server on which the copied project will be stored has enough free hard disk space.

Procedure

This section describes how to use this function in a redundant system with the two servers Server1 and Server2 as an example.

1. Exit Runtime on the redundant Server1 and close the project.
2. Make the configuration changes on Server2 in Runtime and save the changes.
3. Start the Project Duplicator on Server2.
4. Use the "Duplicate" button to duplicate the project on Server1 to the target folder of the project deactivated under "1." and overwrite the project.
5. Open the project on Server1.
6. Check the settings.
7. Start Runtime and wait for the redundancy synchronization.

4.6 Scenarios for WinCC Redundancy

4.6.1 Client switchover in the event of a process connection error

Overview

A redundant system consists of two functionally identical servers. One server is the "Master" server and the other is the redundant partner server.

The servers have the following status in the undisturbed operating state:

- The master server has the status "master".
- The redundant partner server has the status "standby".

Clients are connected to the respective preferred server or to the master server if no preferred server has been specified.

As soon as both servers are in Runtime, the processes coupling monitoring is activated. WinCC Redundancy determines cyclically the number of defective logical connections of the "Master" server and the redundant partner server.

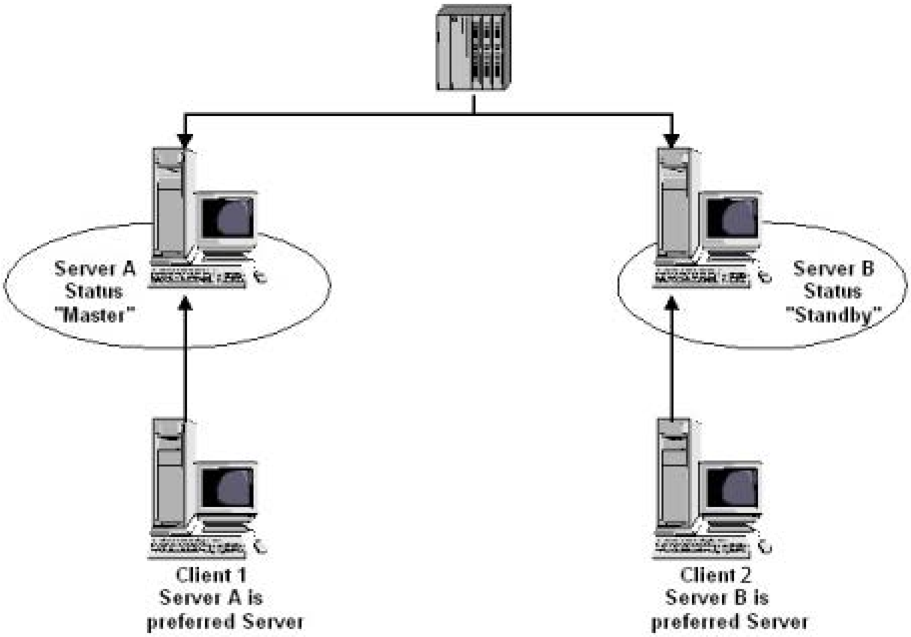
If the "Master" server has more defective logical connections than the redundant partner server, the status of the server is set to "Fault" in the "@RedundantServerState" system tag. The clients are switched over to the redundant partner server, which now has the "Master" status.

Normal operating state

The system is made up of the following computers:

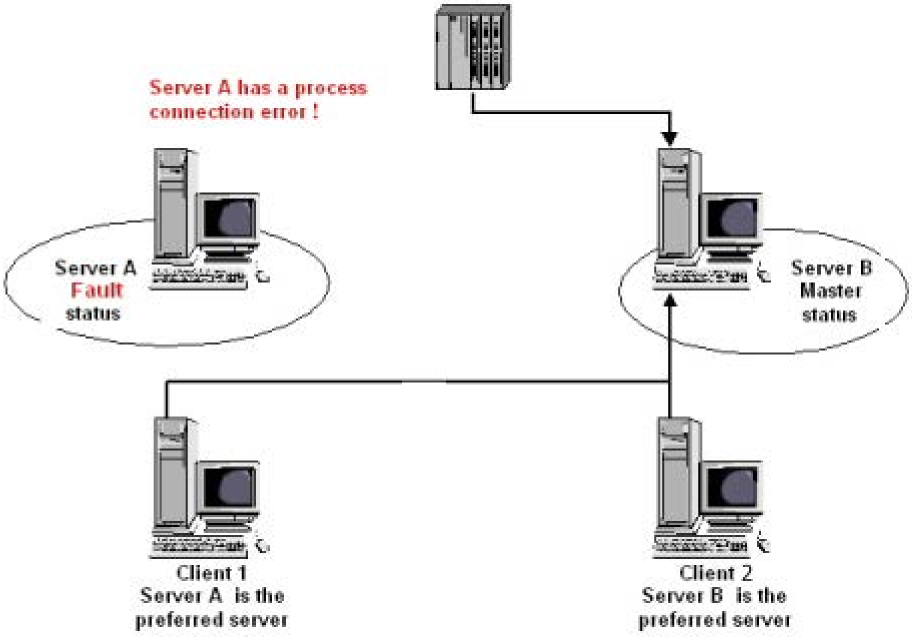
- Redundant Server A
- Redundant Server B

- Client 1 with preferred server A
- Client 2 with preferred server B



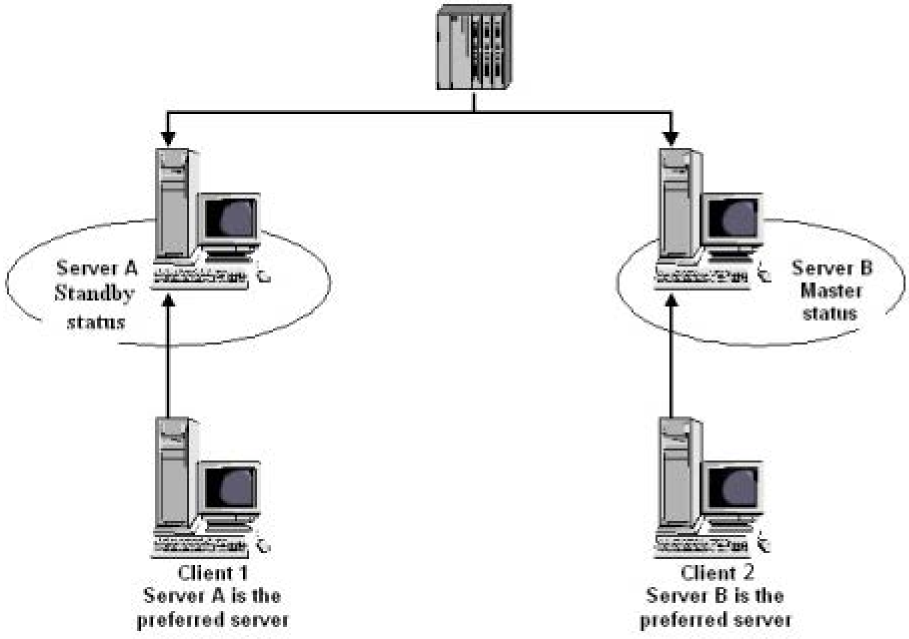
Process connection error on server A

There is a process link error on server A. The error is not present on server B. The number of defective logical connections on server A is greater than on server B. Server A therefore receives the "Fault" status. As a result, Client 1 switches over to the redundant server B.


Server A has a process connection error !
Server A Fault status
Server B Master status
Client 1 Server A is the preferred server
Client 2 Server B is the preferred server

End of the process link error

When the process link error on server A has been cleared, server A then has the status "Standby". Client 1 then switches over to Server A because the server is entered as its preferred server. Client 2 remains connected to Server B as its preferred server.



Note

The OPC couplers are not monitored. Therefore, no client switching in case of an error of the OPC couplers takes place.

This restriction does not apply to the OPC UA.

See also

Failure scenarios (Page 135)

Guide to setting up a redundant system (Page 119)

WinCC Redundancy (Page 112)

How to configure the redundant servers (Page 122)

How to configure the synchronization of user archives (Page 125)

4.6.2 Failure scenarios

4.6.2.1 Failure scenarios

Introduction

We use some failures that occur in reality to illustrate how WinCC Redundancy works.

1. Scenario 1: Project on server computer not in Runtime (Page 136)
2. Scenario 2: Connection Fault to Partner Server (Page 138)
3. Scenario 3: Faulty Network Connection to Client (Page 139)
4. Scenario 4: Faulty Process Connection (Page 140)
5. Scenario 5: Software Error (Page 141)

WinCC Redundancy will recognize the current error itself or react to error messages with the following actions:

- Saving times of events.
- Archive synchronization.
- Changing the "Master" and "Standby" identifiers.
- Switching clients.
- Triggering messages.

Startup of the server PCs

When the server PCs are starting up, the redundancy component establishes whether the partner server is already active.

- If the partner server is already active, the "Standby" status is set in the server computer.
- If the partner server is not active during startup, the "Master" status is set in the server computer.

WinCC redundancy system tags

The status of the server computer is saved in the "@RM_MASTER" system tag.

Status of server computer	"@RM_MASTER" status
Master	1
Standby	0

The "@RM_MASTER_NAME" tag contains the name of the server system that has the "Master" status, e.g., "Server1".

The "@RedundantServerState" tag displays the redundancy status for each redundant server, e.g., "Standby".

Redundancy only sets the above tags. Both servers are always completely equal.

Scripts or other applications can evaluate these tags. Only the "@RM_MASTER" tag can be changed.

An overview of the system tags in available under WinCC Redundancy system tags (Page 142) .

Exchanging status information

The status of the redundancy is controlled via a separate connection. The connection can be established as follows:

- Using a network adapter
- Using the serial interface

Connection via a network adapter is to be preferred to serial connection.

Note

Note that the archive synchronization is performed via the terminal bus. The archive synchronization is not executed via the status connection.

4.6.2.2 Scenario 1: Project on server computer not in Runtime

Introduction

This scenario shows the behavior of WinCC Redundancy when the project on Server2 was deactivated.

The following actions will be triggered:

- Server1 stores the downtime with date and time of Server2.
- Server1 will report the failure of Server2 through a system message.
- If Server1 is the standby server, Server1 takes over the role of the master server. The "@RM_MASTER" tag is set and the "@RM_MASTER_NAME" and "@RedundantServerState" tags are changed.
- The clients connected to Server2 switch over to Server1.

Server2 comes back online

The downtime means that there is a gap in the archives of Server2. This gap will be filled by the following measures:

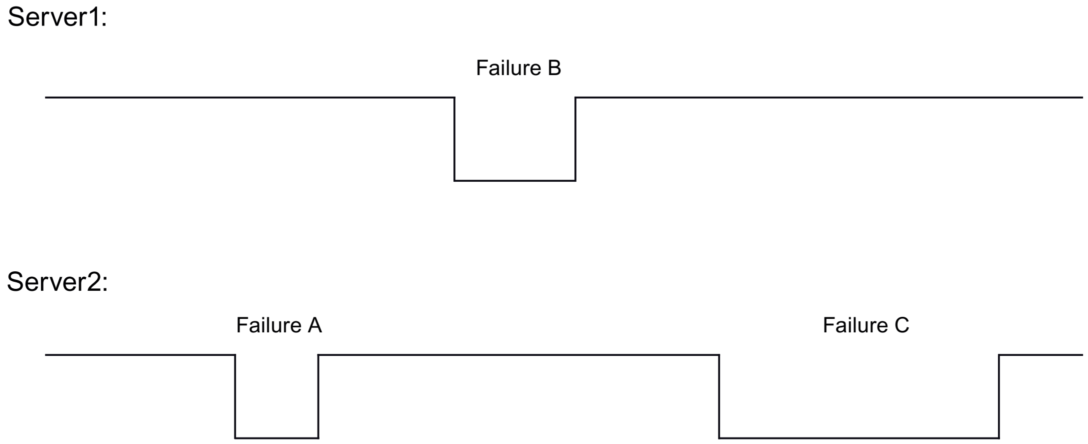
- Server1 stores the return time with date and time of Server2.
- Server1 reports the return of Server2 through a system message.

- A redundancy synchronization for the following archive from Server1 is executed on Server2.
  - Message archives
  - Process data archives
  - User archives
- With Server1 "@RM_MASTER" remains set, with Server2 "@RM_MASTER" is reset. "@RM_MASTER_NAME" and "@RedundantServerState " remain unchanged at both servers.
- Clients, which are configured with Server2 as their preferred server, switch back to Server2.

Compared to online synchronization, archive synchronization after a server failure can take longer. The duration of the synchronization depends on the number of records to be synchronized and the computer and network load.

Alternating Failure of the Server

If failures alternate between the two servers, they are synchronized one after the other. After the synchronization, all data is available in both archives.



If the synchronization was configured, a synchronization is always performed.

Failure A

Server1 transfers all values to Server2.

Failure B

Server2 transfers all values to Server1.

Failure C

Server1 transfers all values to Server2.

All these processes run automatically in the background, independently of the process value archiving and message archiving from the lower-level automation systems taking place at the same time.

4.6.2.3 Scenario 2: Connection Fault to Partner Server

Introduction

This scenario shows the behavior of redundancy in the case of a connection failure to the partner server.

Prior to the occurrence of this event, both servers run in Runtime without failures.

The described connection failure occurs if, for example, the network connection at Server1 is pulled.

Initial Situation 1

When the connection fails, Server 1 is the master server and Server 2 the standby server.

Connection failure occurs

The following reactions are triggered when the connection fails:

- Server2 becomes the master server and saves the time of the failure with date and time.
- Server2 displays a system message stating that the partner server has failed and Server2 is now the master server.
- Tags "@RM_MASTER", "@RM_MASTER_NAME" and "@RedundantServerState" are adapted accordingly on both servers.

Connection is restored

During the connection failure, the messages of Alarm Logging and the user archives were not synchronized.

The following measures are carried out:

- Master Server2 stores the time of the return.
- Server2 displays by way of a system message the return of the partner server.
- Redundancy synchronization from master server to standby server.
- Through online synchronization of the Alarm Logging, the following is reported from Server1 to Server2 and display on Server1 as a system message:
  - An error has occurred in the redundant operation.
  - Server1 has switched to "Standby" status.
  - Return of Server1.
- The "@RM_MASTER", "@RM_MASTER_NAME" and "@RedundantServerState" tags remain unchanged on the two servers.

Initial Situation 2

When the connection fails, Server 1 is the standby server and Server 2 the master server.

Connection failure occurs

The following reactions are triggered when the connection fails:

- Server2 remains the master server and saves the time of the failure with date and time.
- Server2 displays the failure of the partner server by means of a system message.
- Server1 changes to the internal state "Fault".
  The clients with Server1 as the preferred server switch over to Server2.
- The "@RM_MASTER", "@RM_MASTER_NAME" and "@RedundantServerState" tags remain unchanged on the two servers.

Connection is restored

During the connection failure, the messages of Alarm Logging and the user archives were not synchronized.

The following measures are carried out:

- Server2 stores the time of the return.
- Server2 displays by way of a system message the return of the partner server.
- Redundancy synchronization from master server to standby server.
- Through online synchronization of the Alarm Logging, the following is reported from Server1 to Server2 and display on Server1 as a system message:
  - An error has occurred in the redundant operation.
  - Return of Server1.
- The "@RM_MASTER", "@RM_MASTER_NAME" and "@RedundantServerState" tags remain unchanged on the two servers.

4.6.2.4 Scenario 3: Faulty Network Connection to Client

Introduction

In this scenario, there is a disturbance in the network connection between Server2 and the "CL5" client belonging to Server2. Server1 is the master server.

The following reaction is triggered:

- Client "CL5" automatically switches over from disturbed Server2 to running Server1.

End of the network disturbance to the client

The following reactions are triggered at the end of the network disturbance:

- The "@RM_MASTER", "@RM_MASTER_NAME" and "@RedundantServerState" tags remain unchanged on the two servers if Server1 was already the master server before the failure.
- The client "CL5" switches back to the preferred server, Server2.

See also

WinCC Redundancy system tags (Page 142)

Scenario 2: Connection Fault to Partner Server (Page 138)

Scenario 1: Project on server computer not in Runtime (Page 136)

Failure scenarios (Page 135)

Scenario 4: Faulty Process Connection (Page 140)

4.6.2.5 Scenario 4: Faulty Process Connection

Introduction

In this scenario, there is a fault on the process connection on Server2 due to an interrupted network connection to the automation systems.

Failure of a connection to an automation system

The connection failure to an automation system is only recognized in WinCC Redundancy if the connection to a server is faulty.

An interruption in the connection of an automation system to both servers is not a failure in terms of redundancy, for example, the failure of an automation system..

Reaction to an Error

If WinCC recognizes a failure, the following actions will be triggered:

- The disturbance of the process link is reported on Server2.
- Server1 receives a message that partner Server2 has failed.
- Server1 saves the time of the error on Server2 with date and time.
- If you have configured the "Client change with disturbance in the process connection" option in the "Redundancy" editor, the clients connected to this server are switched over to the partner server.
- With Server1, the "@RM_MASTER" tag is set to "Master", with Server2 to "Standby". The "@RM_MASTER_NAME" and "RedundantServerState" tags are adapted accordingly. The "@RedundantServerState" tag is set to "Fault" at Server2.

End of the process link error on Server2

Provided process connection monitoring has been activated, the gap in the archive of Server2 will be filled by the following measures:

- Server1 stores the return time of Server2.
- A redundancy synchronization is carried out from Server1 to Server2, since no faults were found for process connection on Server1. The data of all automation systems are synchronized, including the data of the automation systems without faults.

- With Server2, the "@RedundantServerState" tag is changed from "Fault" to "Standby".
- The correction of the process link error on Server2 is announced by a system message.

4.6.2.6 Scenario 5: Software Error

Introduction

In this scenario, an error occurs on Server2 in software that is being monitored. At this time of the failure, Server2 has the "Master" status and Server1 the "Standby" status. Several clients are connected to both servers.

If the "Application Health Check" function detects an error in the WinCC software, the following actions are initiated:

- "Application Health Check" reports the fault to WinCC Redundancy. The status of Server2 is set to "Fault" in the "@RedundantServerState" tag. The "@RM_Master" tag is set to "Standby".
- With Server1 "@RM_Master" is set to "Master". "@RM_MASTER_NAME" and "RedundantServerState" are adapted accordingly.
- The clients connected to Server2 switch over to Server1.
- A process control message informs users of the software error if the the alarm server itself has not caused the error.

Measures at the end of the software error on Server2

Deactivate the affected Server2 project. Restart Server2. When the project is activated on Server2, the archives are automatically synchronized.

- With Server2 "@RedundantServerState" is set to "Standby". Server1 remains the "Master".
- Server1 stores the return time of Server2 with date and time.
- Reconnecting to this server is now possible. Archive synchronization is only performed retroactively to the moment when the software error of Server2 was detected.

See also

WinCC Redundancy system tags (Page 142)

Scenario 3: Faulty Network Connection to Client (Page 139)

Scenario 2: Connection Fault to Partner Server (Page 138)

Scenario 1: Project on server computer not in Runtime (Page 136)

Failure scenarios (Page 135)

Scenario 4: Faulty Process Connection (Page 140)

4.6.3 WinCC Redundancy system tags

WinCC redundancy system tags

System tags are created in the internal "Performance" and "Redundancy" tag groups for diagnostics of the redundancy status.

Redundancy tags

If you open the "Redundancy" editor in the WinCC Explorer and close it again with "OK", the system tags are created by WinCC Redundancy.

The "@RM_MASTER" and "@RM_MASTER_NAME" system tags are used for Master/Standby control of the two redundant servers and for client switchover.

You can read the system tags via other applications or scripts.

You can only change the "@RM_MASTER" tag.

Performance tags

The performance tags @PRF_REDUNDANCY_... represent states of the redundant system.

When a WinCC project is created, the system tags are created in the "Performance" tag group.

More information:

- "Working with WinCC > Working with Projects > Making Settings for Runtime > System diagnostics with performance tags"

Overview of the redundancy tags

System tags	Meaning
@LocalMachineName	Contains the local computer name.
@RedundantServerState	Redundancy status of the server: 0: Undefined status or start value 1: Server is the primary server (master) 2: Server is Standby 3: Server is in "FAULT" status 4: Server is standalone or no redundant operation
@RM_MASTER	Tag value=1: Identifies the primary server. If the server becomes the standby server, "@RM_MASTER" is reset to "0". You can change the value of the tags, for example, via scripts.
@RM_MASTER_NAME	Name of the primary server
@RM_SERVER_NAME	Name of the server to which a client is connected

System tags	Meaning
@RM_UA_ONL_"Archiv name"	Is used for diagnostics: • 1: A user archive has changed. • 0: Is set on the standby server when online matching of the user archive is successfully completed. Primary server (Master): The value remains "1" since no feedback mechanism from the standby server exists. For an identical value on both servers, activate tag synchronization. A separate tag with the corresponding archive name is inserted for each user archive.
@RM_Offline_UA_Name	Is used for diagnostics. The tag contains the name of the user archive that has just been synchronized.

Overview of performance tags

The performance tags @PRF_REDUNDANCY_... represent states of the redundant server that are also evaluated in the "RedundancyControl" diagnostic tool.

System tag	Description
@PRF_REDUNDANCY_IS_SYNCHRONIZED	Synchronization status: • 0: Redundant applications are not synchronized. • 1: Redundancy synchronization of all applications is completed. The status can be influenced by WinCC and other applications logged on for redundancy, e.g. SIMATIC BATCH.
@PRF_REDUNDANCY_VALIDATION	Evaluation points of the server. The validation value determines which server becomes the primary server. The validation value depends, for example, on the connection and Runtime status. With correctly configured redundancy, the validation value is the same on both redundant servers. If the validation values differ, the server with the higher value becomes the primary server. Typical values: • 37: The server status is good. – Runtime is active. – Redundant connection via serial interface • 35: The server status is good. – Runtime is active. – Redundant connection via LAN • < 35: The server has the internal "FAULT" status. Check the connection status or the status of the server. The "FAULT" status is set in the case of a critical operating state, for example, when a server application is no longer responding. If a server takes on the "FAULT" status, the partner server becomes the primary server. Sample calculations: • If Runtime is disabled on the server, validation is reduced by 4 points. • If the terminal bus cannot be reached, validation is reduced by 20 points.
@PRF_REDUNDANCY_PARTNER_VALIDATION	Evaluation points of the redundant partner server With correctly configured redundancy, this value is the same on both redundant servers.
@PRF_REDUNDANCY_AS_COUNT	Number of AS connections on the server With correctly configured redundancy, this value is the same on both redundant servers. The following conditions cause a redundancy switchover: • The validation values on the redundant servers are the same. • The number of AS connections is different. In this case, the server with more AS connections becomes the primary server.

System tag	Description
@PRF_REDUNDANCY_PARTNER_AS_COUNT	Number of AS connections on the redundant partner server With correctly configured redundancy, this value is the same on both redundant servers.
@PRF_REDUNDANCY_CURRENT_STATE	Redundancy status of the server: • 0: Undefined status • 1: Server is the primary server • 2: Server is Standby • 3: Server is in "FAULT" status • 4: Server is standalone or no redundant operation
@PRF_REDUNDANCY_PARTNER_CURRENT_STATE	Redundancy status of the redundant partner server
@PRF_REDUNDANCY_FAULT_POSTPONED	Tag value=1: The server has the "FAULT_POSTPONED" status. The internal state of the local server is "FAULT", but the partner server cannot assume the "Master" state. Redundancy switchover is not possible. The cause may be a running redundancy synchronization, for example. As soon as the conditions for redundancy switchover are met, the server changes to "FAULT" status. The "@PRF_REDUNDANCY_CURRENT_STATE" tag assumes the value "3".
@PRF_REDUNDANCY_PARTNER_FAULT_POSTPONED	Tag value=1: The redundant partner server has the "FAULT_POSTPONED" status.
@PRF_REDUNDANCY_SWITCHOVER_COUNT	Number of redundancy switchovers since the activation of Runtime or since the last reset via "@PRF_REDUNDANCY_SWITCHOVER_COUNT_RESET".
@PRF_REDUNDANCY_SWITCHOVER_COUNT_PERIOD	Number of redundancy switchovers in a defined period Default setting: • Time period: 1 calendar day • The value is reset at 0:00 AM each day.
@PRF_REDUNDANCY_SWITCHOVER_COUNT_RESET	The reset tag resets the value of the following performance tag: • @PRF_REDUNCANCY_SWITCHOVER_COUNT

"RedundancyControl" diagnostic tool

For comprehensive redundancy diagnostics, you can use the "RedundancyControl" diagnostic tool.

You can find more information in the application example "Redundancy in WinCC V7.x and WinCC Professional":

- https://support.industry.siemens.com/cs/ww/en/view/109772627 (https://support.industry.siemens.com/cs/ww/en/view/109772627)

See also

WinCC Redundancy system messages (Page 146)

How to configure the synchronization of user archives (Page 125)

https://support.industry.siemens.com/cs/ww/en/view/109772627 (https://support.industry.siemens.com/cs/ww/en/view/109772627)

4.6.4 WinCC Redundancy system messages

Overview

WinCC Redundancy provides a series of system messages.

To use the system alarms, activate the "Use" column in the "Alarm logging" editor for the system messages.

The following system messages can be output by WinCC Redundancy:

Number	WinCC message text / description
1012200	REDRT: Partner station failure WinCC was terminated on the partner server.
1012201	REDRT: Partner station back online WinCC was restarted on the partner server.
1012202	REDRT: Projects are not functionally identical
1012203	REDRT: Archive synchronization failed
1012204	REDRT: Internal error in Redundancy
1012205	REDRT: Problem with partner connection There is a problem with the connection to the partner server.
1012206	REDRT: Connection to the partner has been reestablished The connection to the partner server has been restored.
1012207	REDRT: Partner server WinCC not started During startup, it is determined that WinCC has not been started.
1012208	REDRT: Archive synchronization launched This message is output at the beginning of an archive synchronization.
1012209	REDRT: Archive synchronization finished This message is output at the end of an archive synchronization.
1012210	REDRT: Tag Logging is being synchronized
1012211	REDRT: Tag Logging synchronization finished
1012212	REDRT: Alarm Logging is being synchronized
1012213	REDRT: Alarm Logging synchronization finished
1012214	REDRT: User Archive is being synchronized.
1012215	REDRT: User Archive synchronization finished
1012216	REDRT: Archive synchronization canceled Synchronization was interrupted due to another failure.

Number	WinCC message text / description
1012217	REDRT: Partner server project not activated System detected during startup that WinCC is not running on the partner server or is not in Runtime.
1012218	SWITCH: Client was switched automatically Client has been switched automatically to the partner server
1012219	SWITCH: Client was switched manually Client was manually to the partner server
1012220	UA: Synchronization is enabled for all user archives If the redundancy synchronization is activated for all archives, the message is output during the import.
1012221	UA: Synchronization is not enabled for all user archives The message is output during the import if the redundancy synchronization is deactivated for at least 1 archive.
1012226	REDRT: Partner server project has been activated System detected during startup that WinCC is activated on the partner server.
1012227	REDRT: Error: Partner server is not a server System detected during startup that the configured partner server is not a server.
1012228	REDRT: CAS: Archive synchronization launched '@2%s@'
1012240	REDRT: Error <error description> in <application name> causes switchover. Switchover was triggered by Application Health Check due to an error in the above application.
1012241	REDRT: Switchover to status <Status designation> Notification of a status change.
1012244	REDRT: Overload during Alarm Logging online update Too many messages to be synchronized.
1012245	REDRT: Loss of serial connection
1012246	REDRT: Serial connection reestablished
1012247	REDRT: <Machine name on which the message was created>: OS Server (Master) <machine name> OS Server (Standby) <machine name> Redundancy error. Depending on the failure scenario, the master server and the standby server or one of the two servers report the redundancy error. Redundancy is endangered.
1012248	REDRT: <Machine name on which the message was created>: OS Server (Master) <computer name> OS Server (standby) <computer name> Redundancy recovered
1012349	RedundancyControl: Loss of connection via network card with MAC address <Address>. The connection to the partner server via the redundant LAN is interrupted or disturbed.
1012350	RedundancyControl: Connection via network card with MAC address <Address> has been reestablished. The connection to the partner server via the redundant LAN has been restored.
1012351	RedundancyControl: System blockage detected. Switch to Fault status.
1012352	RedundancyControl: System blockage detected. Restart your computer as soon as possible.
1012354	RedundancyControl: Status changed to FAULT. However, server isolation is not activated.

Number	WinCC message text / description
1012355	RedundancyControl: Status changed to FAULT. However, server isolation is locked by <Name>. Reason: <Cause>
1012356	RedundancyControl: Status in FAULT changed => server is isolated
1012357	RedundancyControl: Status changed to FAULT. However automatic restart is not activated.
1012358	RedundancyControl: Status changed to FAULT. However automatic restart is locked. The network adapter is disconnected and DHCP is released.
1012359	RedundancyControl: Restart of the computer disabled by <Name>. Reason: <Cause>
1012360	RedundancyControl: Restart of the computer aborted. The last restart took place less than <number of seconds> s ago.
1012361	RedundancyControl: Restart of the computer aborted. After <number of restarts> restarts no further restarts are permitted for <number of seconds> s.
1012362	RedundancyControl: Rebooting computer in <number of seconds> s
1012700	Self-diagnostics: Value <Value> of station <Station name> is invalid.
1012701	Self-diagnostics: Value <Value> of station <Station name> violated the HIGH error limit.
1012702	Self-diagnostics: Value <Value> of station <Station name> violated the LOW error limit.
1012703	Self-diagnostics: Value <Value> of station <Station name> violated the HIGH warning limit.
1012704	Self-diagnostics: Value <Value> of station <Station name> violated the LOW warning limit.
1012705	Self-diagnostics: Value <Value> of station <Station name> no longer violates the error limit.
1012706	Self-diagnostics: Value <Value> of station <Station name> is OK.
1012707	Self-diagnostics: Station <Station name> causes @2%s@.
1012708	Self-diagnostics: Value <Value> of station <Station name> is invalid.

See also

WinCC Redundancy system tags (Page 142)

WinCC Certificate Manager 5

5.1 Introduction to the WinCC Certificate Manager

WinCC supports the use of CA-based certificates (CA = certificate authority) in Runtime.

With the "WinCC Certificate Manager" application, you can create a WinCC certificate authority and the WinCC certificates needed for your PCs, distribute the certificates to the PCs, and install them there.

The Certificate Manager also supports you in establishing the trust relationship between Runtime and its communication peers.

Note

Use of an external WinCC certificate authority or an intermediate certificate authority is not supported.

Note

Use of self-signed certificates

For security reasons, the use of CA-based certificates is recommended. As an alternative, it is possible to use self-signed certificates.

Functionality of the WinCC Certificate Manager

- Central creation and management of certificates in the network
- Creation of a certificate authority with:
  - Private key
  - Public key (root certificate)
  - CRL file (CRL = Certificate Revocation List)
- Creation of the application certificates of PCs
- Renewing existing certificates
- Encrypted export of the application certificates and the root certificate for manual distribution to the PCs
- Encrypted import and installation of certificates on the PCs
- Encrypted export and import of the root certificate, the CRL file and the private key as well as all device certificates for data backup and recovery
- Export of the root certificate and its CRL file for distribution to external communication peers of the PC
- Export of an updated CRL file for distribution to the PCs and their external communication peers

Available application certificates

With WinCC Certificate Manager, you can create the following CA-based application certificates for WinCC PCs:

- WebUX | WebNavigator certificate
- OPC UA Server certificate
- OPC UA Client certificate
- OPC UA Tags Importer certificate
- REST Service certificate

5.2 Basics

5.2.1 Certificate authority

To issue the required WinCC application certificates, you need a WinCC certificate authority. You create the certificate authority with the "WinCC Certificate Manager" tool.

The PC on which you generate the certificate authority and application certificates is the certificate authority device.

Total configuration of the certificate authority

The total configuration of the certificate authority includes:

- The private key
  The certificate authority uses it to sign the application certificates. The signature guarantees the authenticity of the certificates.
  The private key remains on the certificate authority device.
- The public key/root certificate (CA certificate) and a CRL file (Certificate Revocation List). Root certificate and CRL file are distributed to the PCs and their external communication peers.

  Note

  This help uses the runtime root certificate.
- The PCs which were added to the certificate authority and your application certificates.

CA container

The CA container (Certificate Authority container) includes:

- The root certificate
- The CRL file
- The application certificates of the PCs added to the certificate authority

5.2.2 Required certificates

You need the following certificates for using CA-based certificates in WinCC Runtime:

WinCC certificates

- The root certificate of the WinCC certificate authority and the CRL file
- The WinCC application certificates issued for the PCs by the certificate authority

You create and distribute these certificates with WinCC Certificate Manager on the certificate authority device.

You import and install these certificates on the PCs with WinCC Certificate Manager.

WinCC Certificate Manager supports you in establishing the trust relationship between the PCs and their communication peers.

WinCC root certificate and CRL file

When you create the WinCC certificate authority, the WinCC root certificate and its CRL file are also automatically created.

WinCC application certificates

You need the following application certificates for the PCs:

- If PC is a web server: WebUX | WebNavigator certificate

  Note

  A web server (IIS) must also be installed on the PC.

- If PC is an OPC UA server: OPC UA server certificate
- If PC is an OPC UA client:
  - OPC UA client certificate
  - To import the variables of an OPC UA server: OPC UA Tags Importer certificate
- PC uses the REST interface: REST Service certificate

Certificates of the OPC UA communication peers

- The OPC UA application certificate of the communication peer
- The root certificate of the certificate authority that issued this certificate

You can create these certificates as described in the user help of the communication peer.

See also

Making certificates available (Page 160)

Establishing a trust relationship (Page 175)

5.2.3 Creating the trust relationship

A successful CA-based communication requires that a device trusts the application certificate of its communication peer. That is automatically the case if the root certificate of an application certificate has been installed on the device in the certificate store in the folder with the trustworthy authorities.

For WinCC web server communication, the web clients must trust the root certificate of the WinCC web server.

For OPC UA communication, the WinCC PC must trust the root certificate of its OPC UA communication peer, and vice versa.

Note

Application certificates from the same certificate authority

It is sufficient to install the root certificate of a certificate authority on a device once in the certificate store. The device then trusts all application certificates that have been issued by this certificate authority.

Two WinCC PCs with the same certificate authority whose certificate configuration has been installed will automatically trust each other in the following cases:

- In OPC UA communication
- When one PC is the web server and the other is the web client:
  - When using a WebNavigator client
  - When using Chrome and Edge as the WebUX client

See also

Establishing a trust relationship (Page 175)

5.2.4 Password requirements

WinCC Certificate Manager has the following requirements for passwords:

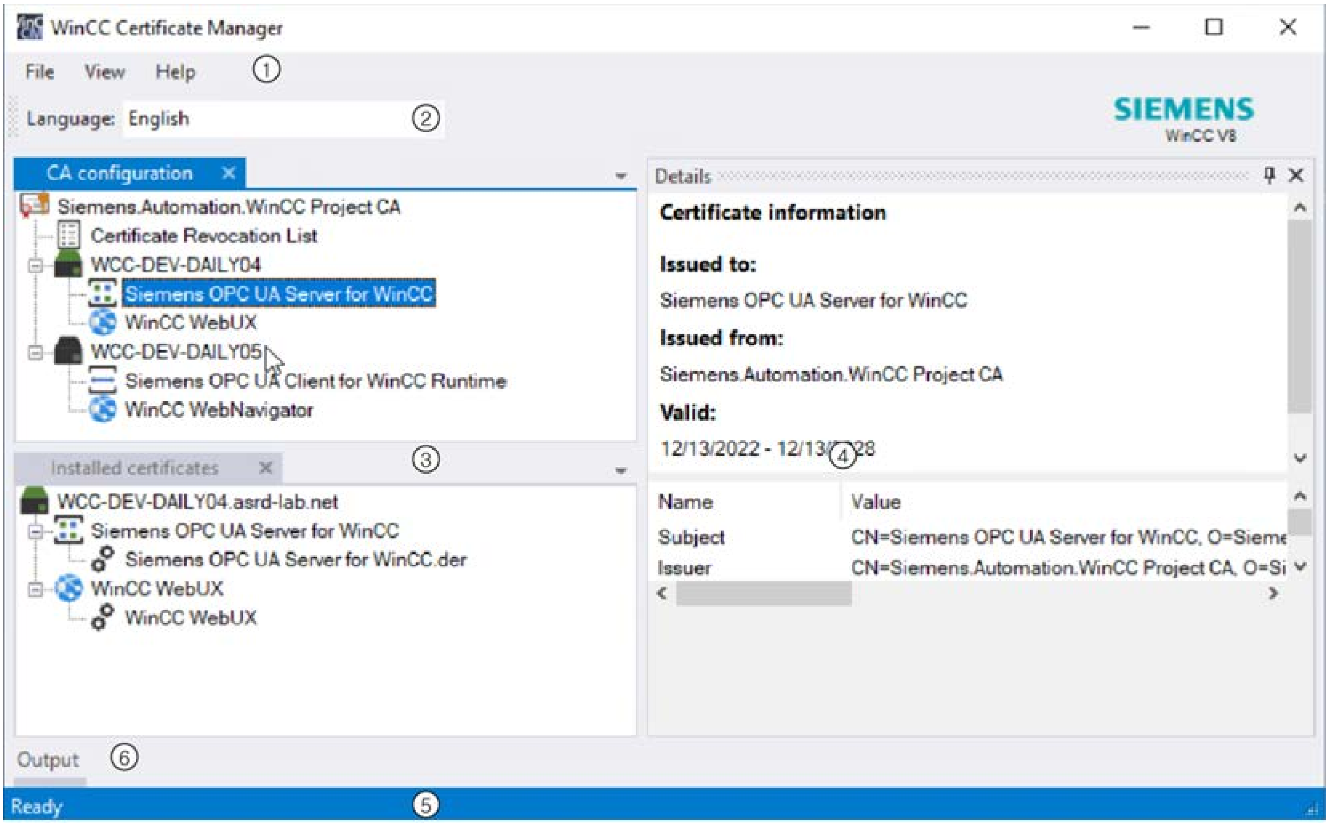
- Length: At least 8 characters
- In each case at least one uppercase letter, one lowercase letter, one number and one special character

5.3 Interface of the Certificate Manager

5.3.1 Structure of the user interface

Overview

The layout of the user interface of WinCC Certificate Manager is as shown here:



Siemens.Automation.WinCC Project CA
- Certificate Revocation List
- WCC-DEV-DAILY04
  - Siemens OPC UA Server for WinCC
  - WinCC WebUX
- WCC-DEV-DAILY05
  - Siemens OPC UA Client for WinCC Runtime
  - WinCC WebNavigator

WCC-DEV-DAILY04.asrd-lab.net
- Siemens OPC UA Server for WinCC
  - Siemens OPC UA Server for WinCC.der
- WinCC WebUX
  - WinCC WebUX

Certificate information

Issued to:

Siemens OPC UA Server for WinCC

Issued from:

Siemens.Automation.WinCC Project CA

Valid:

12/13/2022 - 12/13/2028

Name	Value
Subject	CN=Siemens OPC UA Server for WinCC, O=Sieme
Issuer	CN=Siemens.Automation.WinCC Project CA, O=Si

① Menu bar

② Toolbar

③ Work area with the "CA configuration" and "Installed certificates" tabs

④ "Details" area (fixed)
The "Details" area shows you detailed information about the certificate selected in the work area.

⑤ Information bar

⑥ "Output" area (hidden)
The "Output" area logs operator control actions.

You can customized the interface to suit your requirements.

Menu bar

Menu	Description
"File > Exit"	Closes Certificate Manager.
"View"	Configure which Certificate Manager interface elements you see. Open or close the following user interface elements: • "Output" area • "Details" area • "CA configuration" tab • "Installed certificates" tab
"Help"	"Info Certificate Manager" Opens a dialog with information about the installed software version.

Toolbar

Button	
▾	To change the user interface language

Tab of the working area

See "CA configuration" tab (Page 155) and "Installed certificates" tab (Page 157).

See also

Customize surface (Page 157)

5.3.2 "CA configuration" tab

On a certificate authority device

On a certificate authority device, you create and configure the certificate authority in the "CA configuration" tab:

- You can create the certificate authority and its root certificate.
- You can add PCs.
- You can create application certificates for the PCs.
- You can carry out exports:
  - To deploy the certificates
  - To back up data

- You can recreate certificates.
  You have the following options:
  - Recreating a root certificate
  - Updating a CRL file
  - Recreating the certificate configuration of a PC
  - Recreating an individual application certificate of a PC
- If the certificate authority device is used as WinCC PC: Install the application certificates of the PC on the device.

Note

Content of the tab

After starting Certificate Manager, you will see the same data that the certificate authority had when you last closed Certificate Manager:

- If no data has been generated yet, you will see the nodes "Open configuration ..." and "Create certificate authority ..."
- If data has already been generated, you will see the root certificate and its CRL file, as well as the configured Unified devices and their application certificates.
  You can edit them.

On a PC

On a PC that is not used as a certificate authority, you perform the following actions in the tab:

- You import the certificate configuration of the PC.
- You install the complete certificate configuration or an individual application certificate.
- You delete installed certificates.

Note

Content of the tab

After launching Certificate Manager, you will see the nodes "Open new configuration ..." and "Create certificate authority ..."

After opening a new configuration, you see the root certificate of the certificate authority and its CRL file as well as the PCs configured at the certificate authority and their application certificates.

You can only install the certificate configuration of the local PC. The certificate configuration of the other PCs is displayed for information only. You cannot change the certificate configuration of the other PCs.

Closing Certificate Manager also closes the configuration.

5.3.3 "Installed certificates" tab

In the "Installed certificates" tab, you see which application certificates have been installed on the local PC.

You can uninstall certificates by deleting them.

5.3.4 Customize surface

The display and arrangement of the WinCC Certificate Manager surface of can be configured:

	Close / Open	Move	Undock / dock	Fix / Unfix	Show / Hide
"Details" area	✓	✓	✓	✓	✓
"Output" area	✓	✓	✓	✓	✓
Tab of the working area	✓	✓	-	-	-

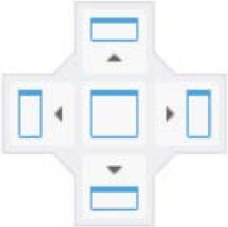
Closing and opening

To close a tab or area, click the "X" button. Alternatively, a tab or area can be disabled in the "View" menu.

To reopen a tab or area, enable it in the "View" menu.

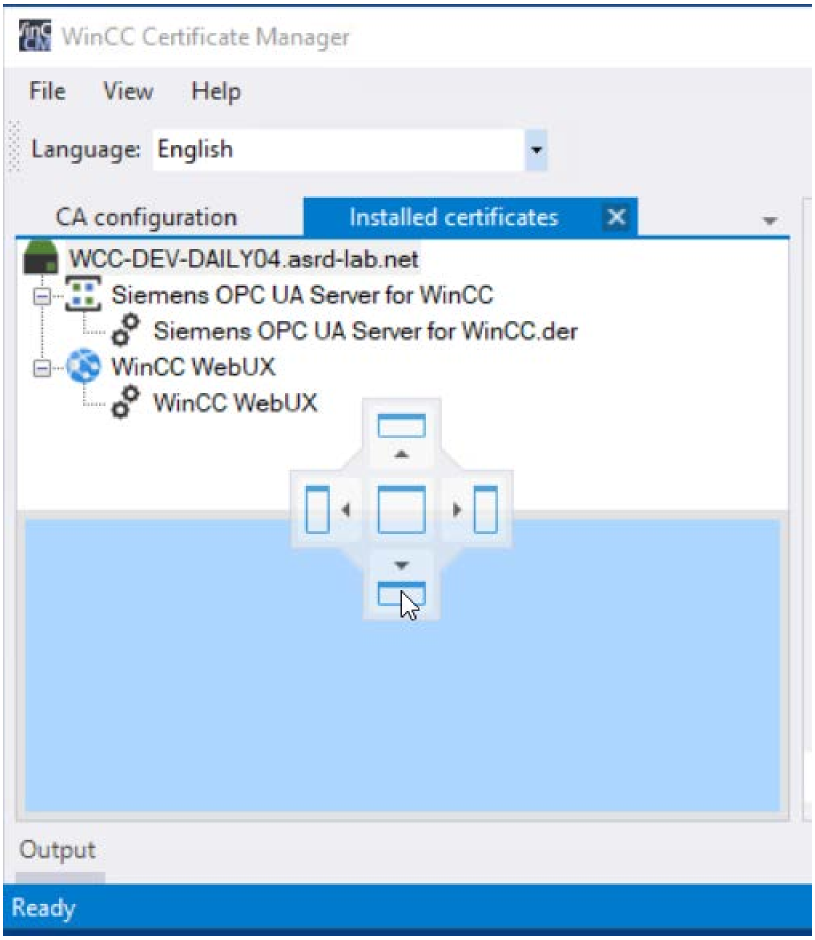
Move

1. Drag the title bar of the tab or area with the left mouse button pressed.
   Possible insertion positions are displayed in the interface:

   The available insertion positions depend on whether you are moving a tab or an area, and on which elements of the application window are already displayed.

2. To see a preview of the new arrangement, move the mouse cursor to one of the positions and keep the mouse cursor pressed:

   
   WinCC Certificate Manager
File
View
Help
Language:
English
CA configuration
Installed certificates
WCC-DEV-DAILY04.asrd-lab.net
Siemens OPC UA Server for WinCC
Siemens OPC UA Server for WinCC.der
WinCC WebUX
WinCC WebUX
Output
Ready

3. Release the mouse cursor over the desired insertion position.

The tab or area is moved.

Undocking and docking

When you move the header of the "Details" or "Output" area, the area is undocked from the application window and displayed as a standalone window. You can move the window freely.

To dock the area back to the application window, move it to one of the suggested insertion positions.

Fixing and unfixing

The following button fixes or unfixes the "Details" and "Output" areas:

Representation of the button	Status	Changing setting
📌	Fixed The area is displayed even if it does not have the focus.	Click the button to switch the setting.
📌	The area is hidden as soon as it loses focus.	

Showing and hiding

Requirement

The "Details" and "Output" areas are not fixed.

Procedure

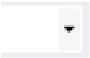
To show an area, click on its text. The area is displayed.

It is automatically hidden when you click the mouse cursor outside the area.

5.3.5 Changing the user interface language

Procedure

1. Click the button with the arrow in the toolbar:



2. Select the desired language.

Result

The user interface language changes.

5.4 Making certificates available

Procedure

To use CA-based certificates in WinCC Runtime, proceed as follows:

1. Select which PC is used as a certificate authority device.
2. Create the certificate authority on this PC (Page 162).
   This creates the root certificate, the CRL file, and the private key.
3. Add the PCs to the certificate authority (Page 164).
4. Add the required application certificates to the PCs (Page 168).
5. Export the certificate configuration of the PCs (Page 170).
6. For each PC:
   - Import the certificate configuration to the PC (Page 171).
   - Install the entire certificate configuration of the PC, or individual application certificates, on the PC (Page 173).

   Note

   Installing the root certificate

   When you install the certificate configuration or an individual application certificate, the root certificate and the CRL file are also always installed on the PC and classified as trustworthy.
7. Establish the trust relationship between the PCs and their communication peers (Page 175).

Additional options

Certificate Manager also offers you the following options:

- Recreating application certificates, for example, because their validity period is expiring (Page 183)
- Exporting application certificates as public certificates (Page 182)
- Creating a data backup of the certificate authority (Page 186)

See also

Creating a certificate authority and root certificate (Page 162)

Adding devices (Page 164)

Add or delete application certificates (Page 168)

Exporting a certificate configuration (Page 170)

Importing a certificate configuration (Page 171)

Installing the certificate configuration or individual certificates (Page 173)

Exporting root certificate and CRL file (Page 177)

Installing a root certificate at the WebUX/WebNavigator client (Page 179)

Backing up the certificate authority (Page 186)

Exporting application certificates as public certificates (Page 182)

Establishing a trust relationship (Page 175)

Recreating certificates (Page 183)

5.5 Creating a certificate authority and root certificate

Requirement

You have not yet created a WinCC certificate authority.

Procedure

1. Open WinCC Certificate Manager on the certificate authority device.
2. Select the "CA configuration" tab.
3. In the work area, double-click "Create new certificate authority".
4. Enter the properties of the root certificate in the "New certificate authority" dialog. The fields are freely editable.
   Mandatory fields:
   - "Name"
   - Password fields for the private key

   If necessary, select a different cryptographic key length and runtime for the certificate.
5. Click "Create".

Result

- The private key is generated.
- The root certificate is generated.
- An empty CRL (Certificate Revocation List) file is generated.
- In the "CA configuration" tab, a node for the root certificate is created and below it one for the CRL file.

Note

The private key is only available on the certificate authority device. The certificate authority uses it to sign the application certificates of the PCs.

The root certificate and CRL file belong to the certificate configuration of the PCs. They are exported or imported when the certificate configuration is exported or imported. They are installed automatically and classified as trusted when the certificate configuration is installed on the PC.

Next steps

- Add the PCs to the certificate authority.
- To distribute the root certificate and its CRL file without the certificate configuration of the PCs, e.g. to external communication partners, export the root certificate and the CRL file.

Deleting certificate authority and root certificate

NOTICE

Data loss prevention

Only delete the certificate authority and root certificate in the following cases:

- After you have saved the certification authority.
- When you no longer need the certificate authority and its data.

Procedure

1. Open Certificate Manager on the certificate authority device.
2. Select the "CA configuration" tab.
3. Right-click the root certificate and select "Delete".

Result

The certificate authority and its entire configuration are deleted by the device.

Note

If the certificate configurations were already installed on the PCs, the certificates are still installed there. Uninstall the certificates on the PCs.

See also

Adding devices (Page 164)

Exporting root certificate and CRL file (Page 177)

Password requirements (Page 153)

Add or delete application certificates (Page 168)

Uninstalling application certificates (Page 174)

Backing up the certificate authority (Page 186)

Installing the certificate configuration or individual certificates (Page 173)

Making certificates available (Page 160)

5.6 Adding devices

Requirement

A certificate authority was created on the certificate authority device in WinCC Certificate Manager.

Procedure

1. On the certificate authority device, open Certificate Manager.
2. Select the "CA configuration" tab.
   You see the root certificate and its CRL file as well as all PCs which have already been added to the certificate authority as well as their application certificates.

3. Right-click the root certificate and select "Add device ...".
4. Enter the device name of the PC, its IP address or both in the "New device" dialog.
   This specification is written to the certificates when the PC's application certificates are added. They serve as validation.

   Note

   Recommendation

   Enter the device name and IP address.

   Required inputs

   - Use the fully qualified domain name (FQDN) as the device name for devices in a domain. This prevents validation errors when accessing web pages.
   - For devices that have a WebUX | WebNavigator certificate, enter the information used to generate the address of the identity provider and the web page of the web server.
     Example: If the IP address is used, entry of the IP address is mandatory. Entry of the device name is optional, but recommended.
   - For devices that are used as OPC UA servers or OPC UA clients, enter the device name.
   - For devices with dynamic IP addresses, only enter the device name.

   Note

   Entry of multiple IP addresses

   To enter multiple IP addresses in the "IP" field, use ";" as a separator. Enter the IP address of the device as the first IP address (own IP).

   The IP addresses are added to the Subject Alternative Name of the certificate.

   Example:
   An HMI device is an OPC UA server and has an NAT router. The OPC UA clients communicate with the server via the NAT router. Enter the private IP address of the OPC UA server HMI device (own IP) and the public IP address in Certificate Manager.

   Note

   Allowed device names

   The host name or FQDN are allowed as the device name.

   The name "localhost" must not be used and will be replaced automatically with the device name of the local device by Certificate Manager.

   Note

   It is only possible to subsequently change this setting by deleting the device from the CA infrastructure and adding it again. As a result, you need to add, distribute and install the device's application certificates again.

Result

A node for the device is generated in the "CA configuration" tab.

Icons of the device nodes:

🟩	The local machine (if added)
⬛	Other devices

Next step

Add the required application certificates to the device.

See also

Add or delete application certificates (Page 168)

Creating a certificate authority and root certificate (Page 162)

Deleting devices (Page 167)

Subsequent change of IP address (Page 187)

Making certificates available (Page 160)

5.7 Deleting devices

Requirement

- A certificate authority was created on the certificate authority device in WinCC Certificate Manager.
- A device has been added to the certificate authority.

Procedure

1. Open Certificate Manager on the device.
2. Select the "CA configuration" tab.
3. Right-click the device and select "Delete".

Result

The device and its application certificates are deleted from the certificate authority.

Note

Deleting does not affect the certificate configuration installed on the device.

If required, uninstall the certificates from the device with the WinCC Certificate Manager.

See also

Adding devices (Page 164)

Uninstalling application certificates (Page 174)

5.8 Add or delete application certificates

Requirement

A device was added to the certificate authority in the WinCC Certificate Manager.

Procedure

1. On the certificate authority device, open Certificate Manager.
2. Select the "CA configuration" tab.
3. Add the required application certificates to the device.
   Proceed as follows for each certificate:
   - Right-click the device and select "Add <certificate type> ..."
   - Enter the properties of the certificate in the "New certificate" dialog.
     If necessary, select a different cryptographic key length and runtime for the certificate.

     Note

     Validity

     The maximum validity period is limited to 27 months for WebUX | WebNavigator certificates. Longer validity periods are not accepted by some browsers.

     Note

     Use the "Fully qualified domain name" as the name for the WebUX | WebNavigator certificate.
   - Click "Create".

Result

The certificate configuration of the device is completed.

Next step

Export the certificate configuration.

Note

If you use the certificate authority device as the Runtime PC, install the certificate configuration of the PC or individual application certificates directly. The export is not required.

Deleting an application certificate

Procedure

1. On the certificate authority device, open Certificate Manager.
2. Select the "CA configuration" tab.
3. Under the desired device, right-click the application certificate and select "Delete".

Result

The application certificate is deleted.

Note

Deleting does not affect the certificate configuration installed on the PC.

If required, uninstall the certificate on the PC with the Certificate Manager.

Additional options

You can recreate an application certificate, for example, because the validity period is expiring.

You can export an application certificate as a public certificate.

See also

Exporting a certificate configuration (Page 170)

Installing the certificate configuration or individual certificates (Page 173)

Adding devices (Page 164)

Recreating application certificates (Page 183)

Exporting application certificates as public certificates (Page 182)

Required certificates (Page 151)

Uninstalling application certificates (Page 174)

Making certificates available (Page 160)

5.9 Export, import and installation for PC

5.9.1 Exporting a certificate configuration

The certificate configuration of a PC consists of its application certificates as well as the root certificate and the CRL file.

Note

Export and import of the certificate configuration are not required when you use the certificate authority device as a Runtime PC and only want to provide the changed certification configuration of this device.

In this case, you install the certificate configuration or individual application certificates directly.

Introduction

The export of the certificate configuration of a PC is required in the following cases:

- After adding the device and configuring its application certificates for the first time
- After adding, deleting or recreating application certificates
- After the recreation of the root certificate
- After updating the CRL file

Export options

You have the following options:

- Export CA container
  The certificate configurations of all PCs are exported to a shared file.
  You import the same file on each PC. Afterwards, you can only install the certificate configuration of the respective device on each PC.
- Export PC
  Only the certificate configuration of the PC you have selected is exported.

Tip for an efficient procedure

Recommended procedure:

- If the certificate configuration of several PCs was changed: Export the CA container.
- If the certificate configuration of an individual PC was changed: Export the certificate configuration of this device.

Requirement

You have completed the certificate configuration of the desired PC or PCs in the Certificate Manager.

Procedure

1. On the certificate authority device, open Certificate Manager.
2. Select the "CA configuration" tab.
3. Follow these steps:
   To export the certificate configuration of all PCs:
   – Right-click on the root certificate module.
   – Select "Export > CA container ...".

   To export the certificate configuration of a single PC:
   – Right-click on the PC.
   – Select "Export device > To PC".
4. Enter and repeat a password in the following dialog to protect the export file.
5. Click "Export".
6. Click on "Save" and select the storage location and the file name.

Result

The certificate configuration of the PC or the certificate configurations of all PCs is/are saved encrypted with the specified password in a secure storage file.

Next step

Import the certificate configuration on the PC(s).

See also

Importing a certificate configuration (Page 171)

Password requirements (Page 153)

Making certificates available (Page 160)

5.9.2 Importing a certificate configuration

The certificate configuration of a PC consists of its application certificates as well as the root certificate and the CRL file.

Note

Export and import of the certificate configuration are omitted when you use the certificate authority device as a Runtime PC and only want to provide the changed certification configuration of this device.

In this case, you install the certificate configuration or individual application certificates directly.

Requirement

- The CA container or the certificate configuration of the PC has been exported on the certificate authority device.
- The PC whose certificate configuration you want to import has access to the storage location of the export file.

Procedure

1. Open WinCC Certificate Manager on the PC.
2. Select the "CA configuration" tab.
3. Double-click "Open configuration ...".
4. Select the export file.
5. Enter the password selected during export.
6. Confirm your entries.

Result

The configuration file is loaded to the "CA configuration" tab. The content of the tab depends on the options selected during the export:

- CA container exported:
  You will see the certificate configurations of all devices of the certificate authority.
  You can install only the certificates of the local device. The display of the other devices is for information purposes. You cannot change their configuration.
- Certificate configuration of the device exported:
  Display and installation are limited to the certificate configuration of the local device.

Note

Exiting Certificate Manager closes the loaded configuration.

Next step

Install the certificate configuration on the device or individual application certificates.

See also

Installing the certificate configuration or individual certificates (Page 173)

Exporting a certificate configuration (Page 170)

Making certificates available (Page 160)

5.9.3 Installing the certificate configuration or individual certificates

You can install the entire certificate configuration of a PC or individual application certificates.

The certificate configuration of a PC consists of its application certificates as well as the root certificate and the CRL file.

Requirement

- The WinCC Certificate Manager is opened on the PC where you want to install the certificates.
- The certificate configuration of the PC was imported onto the device with the Certificate Manager.

Installing

1. Select the "CA configuration" tab.
2. Select one of the following options:
   To install the entire certificate configuration:
   - Right-click the node of the local machine.
     The local machine has the following icon: 🖴
   - Select "Install all certificates".

   To install an individual application certificate:
   - Under the local machine node, right-click the certificate.
   - Select "Install".

Result

The entire certificate configuration of the device or the individual application certificate is installed.

Note

If the trust relationship between the device and its communication partners has already been established, the device can successfully communicate with its communication partners.

During installation, the following happens in detail:

- Application certificates are installed in the certificate store defined for the respective application.
- The root certificate is classified as trustworthy in the certificate stores.

- The CRL file is installed in the certificate store.
- If a WebUX | WebNavigator certificate is installed and the web page of the web server was already set up, the certificate is automatically bound to the web page by the installation. The certificate replaces any certificate selected during the installation. To enforce the use of the new certificate, the web page is then restarted. Any connected WebUX clients will be disconnected and will have to log in again.
  The connection cannot succeed if a WebUX | WebNavigator certificate is installed and the web page has not yet been set up. The certificate is not displayed in the "Installed certificates" tab. Certificate Manager logs this with an entry in the "Output" area.

Note

The OPC UA server certificate only takes effect after a restart of the Runtime.

See also

Establishing a trust relationship (Page 175)

Uninstalling application certificates (Page 174)

Importing a certificate configuration (Page 171)

Making certificates available (Page 160)

5.9.4 Uninstalling application certificates

You have the option of uninstalling the WinCC application certificates installed on a PC.

Requirement

- WinCC Certificate Manager is open on the PC.
- At least one application certificate has been installed on the PC with the Certificate Manager.

Procedure

1. Select the "Installed certificates" tab.
2. Right-click the certificate you would like to open.
3. Select "Delete".

See also

Installing the certificate configuration or individual certificates (Page 173)

5.10 Establishing a trust relationship

Requirement

- The certificate configuration of the PC has been installed on the device.
- For OPC UA communication: The certificate configuration of the OPC UA communication peers is completed.

Establishing the trust relationship with the web server

More information: WinCC Certificate Manager > Installing a root certificate at the WebUX/WebNavigator client (Page 179)

Establishing the trust relationship between the PC and its OPC UA communication peer

If the PC and its OPC UA communication peer have the same certificate authority, the devices trust each other automatically.

If the devices have different certificate authorities, the trust relationship must be established manually:

1. Export the WinCC root certificate and the CRL file on the certificate authority device to an external storage medium.
2. On the device of the OPC UA communication peer:
   - Connect the external storage medium.
   - Add the root certificate and the CRL file on the device to the trustworthy root certificate authorities.
     Proceed as described in the user help of the OPC UA communication peer.
3. Export the root certificate of the OPC UA communication peer and its CRL file to the external storage medium. Proceed as described in the user help of the communication peer.
4. Connect the external storage medium to the PC.
5. Copy the root certificate and the CRL file of the OPC UA communication peer on the PC to the following folder:
   - If PC is an OPC UA client:
     <Installation path>opc\UAClient\PKI\issuers
   - If PC is an OPC UA server:
     Copy the files to the storage location that was defined by the configuration file of the server. Copy the files to the folder with the trustworthy root certificate authorities.

At the next connection attempt, the PC and its communication partner accept each other's application certificates.

Note

If a device has multiple communication peers that have the same certificate authority, the trust relationship with the certificate authority only has to be established once.

The device then trusts all communication peers of this certificate authority.

See also

Exporting root certificate and CRL file (Page 177)

Creating the trust relationship (Page 152)

Security concept of OPC UA (Page 748)

Making certificates available (Page 160)

5.11 Exporting root certificate and CRL file

Introduction

WinCC Certificate Manager allows you to export and distribute the root certificate and CRL file as a public certificate separately from the certificate configuration. This is necessary to establish the trust relationship between a PC and its external communication partners, or to update an expired CRL file.

You have the following options:

- Exporting root certificate and CRL file
- Export the CRL file only

Requirement

A certificate authority has been created on the certificate authority device in Certificate Manager.

Exporting root certificate and CRL file

1. On the certificate authority device, open Certificate Manager.
2. In the "CA configuration" tab, click the root certificate on the right.
3. Select "Export > CA certificate ...".
4. Select a file format.
5. Confirm your entries.
6. Select a target folder.
7. Confirm your entries.

The root certificate and its CRL file are exported to the target folder, each to a separate file.

Export CRL file only

1. On the certificate authority device, open Certificate Manager.
2. Select the "CA configuration" tab.
3. Under the root certificate, right-click the Certificate Revocation list.
4. Select "Export".
5. Select a file format.
6. Confirm your entries.
7. Select a target folder.
8. Confirm your entries.

The CRL file is exported to the target folder.

Next step

Import the files to the external communication partners. Follow the steps described in the user help of the devices.

See also

Creating a certificate authority and root certificate (Page 162)

Making certificates available (Page 160)

5.12 Installing a root certificate at the WebUX/WebNavigator client

Introduction

To establish a trustworthy connection to a web server, WebUX clients and WebNavigator clients must trust the root certificate of the WebUX WebNavigator certificate. To do this, the root certificate must be installed at the web clients in the folder with the Trusted Certificate Authority folder.

Creating the trust relationship automatically

The web client trusts the root certificate of the web server automatically in the following cases:

- The web client is a WebNavigator client or a web client which uses Microsoft Edge or Chrome.
- The device of the web client is a PC with WinCC installed.
- The PC of the web server and the PC of the web client have the same certificate authority. The following application certificates were created on the certificate authority device in WinCC Certificate Manager:
  - For the PC of the web server: The WebUX | WebNavigator certificate
  - For the PC of the web client: At least one application certificate
- The following certificates were installed on the PCs with Certificate Manager:
  - On the PC of the web server: The WebUX | WebNavigator certificate
  - On the PC of the web client: At least one application certificate

Creating the trust relationship manually

This section describes how to install the root certificate if the web client does not yet trust the root certificate.

The procedure depends on which web client you are using.

Note

If a web client communicates with multiple web servers that have the same certificate authority, you only have to install the root certificate on the web client once.

Requirement

- The root certificate of the WebUX | WebNavigator certificate was exported on the certificate authority device, e. g. on an external storage medium.
- The WebUX client or WebNavigator client has access to the storage location.

Procedure for WinCCViewerRT and Internet Explorer (WebNavigator clients)

WebNavigator clients use the Microsoft Windows system certificate storage.

To install the root certificate of the web server in the folder with the trustworthy root certificate authorities, proceed as follows:

1. On the WebNavigator client device, double-click the root certificate file.
   The root certificate appears with the Windows standard dialog "Certificate".
2. Select "Install certificate...".
   The Certificate Import wizard opens.
3. Select "Local computer" as the storage location and "Trusted Root Certificate Authorities" as the certificate store.
4. Start the import.

Alternative procedure

1. You can also start the Windows System certificate store directly.
2. Click the "Trusted Root Certificate Authorities" folder and select "All tasks > Import...".
   The Certificate Import wizard opens.
3. Proceed as described above from step 3.

You can also install the root certificate when establishing the connection for the first time from WinCCViewerRT with the web server:

1. If WinCCViewerRT does not trust the web server, a security warning is issued. Click "View certificate".
2. Click "Install certificate...".
3. Proceed as described above from step 3.

Procedure for Edge and Chrome as WebUX client

Edge and Chrome use the Microsoft Windows system certificate store.

To install the root certificate of the web server in the folder with the trustworthy root certificate authorities, proceed as described above for WebNavigator clients.

Procedure for browsers with their own certificate store

If the browser of the WebNavigator client has its own certificate store and the root certificate in the certificate store is not yet in the folder with the trustworthy root certificate authorities, install it manually.

Follow the steps described in the user help of the browser.

For Firefox, for example, follow these steps:

1. In Firefox, under "Settings > Privacy & Security" under "Certificates", click "Show certificates".
2. In the "Certificate management" window, select the "Certification authorities" tab.
3. Click "Import" and select the root certificate file.
4. In the window that opens, select the option "This certificate can identify websites" and confirm your selection.

See also

Exporting root certificate and CRL file (Page 177)

Making certificates available (Page 160)

5.13 Exporting application certificates as public certificates

Requirement

An application certificate was added to a PC in WinCC Certificate Manager.

Exporting certificate to the certificate authority device

1. On the certificate authority device, open Certificate Manager.
2. Select the "CA configuration" tab.
3. Right-click on the application certificate under the device.
4. Select "Export certificate ...".
5. Select a file format.
6. Confirm your entries.
7. Select a target folder.
8. Confirm your entries.

Exporting the certificate to the PC

Additional requirements

- The application certificate was installed to the PC with the WinCC Certificate Manager.

Procedure

1. Open the Certificate Manager on the PC.
2. Select the "Installed certificates" tab.
3. Right-click on the application certificate.
4. Select "Export certificate ...".
5. Select a file format.
6. Confirm your entries.
7. Select a target folder.
8. Confirm your entries.

Result

The public key of the application certificate is exported. Distribute it to the external communication partners of the device.

See also

Add or delete application certificates (Page 168)

Installing the certificate configuration or individual certificates (Page 173)

5.14 Recreating certificates

WinCC Certificate Manager allows you to recreate existing certificates. This can become necessary in the following cases:

- Expiry of an application certificate:
  Recreate the application certificate.
- Expiry of the root certificate:
  Recreate the entire configuration of the certificate authority.
- Expiry of the CRL file
  Update the CRL file.
- Change in the IP address of a PC because the IP address was specified when adding the device in Certificate Manager.
  Add the device and its application certificates again.

See also

Recreating application certificates (Page 183)

Recreating the entire configuration (Page 184)

Updating a CRL file (Page 185)

Subsequent change of IP address (Page 187)

Making certificates available (Page 160)

5.14.1 Recreating application certificates

You recreate application certificates in the following cases:

- The lifetime of a certificate has expired.
- Entries for a valid certificate are to be edited, for example to correct entries.

Procedure

1. Open WinCC Certificate Manager on the certificate authority device.
2. Select the "CA configuration" tab.
3. Right-click on the application certificate of the desired device and select "Recreate".
   The "Recreate certificate" dialog opens. The entries of the old certificate are downloaded into the dialog.
4. Change the desired properties.
5. Click "Create".

Result

The certificate is recreated. Export the certificate configuration of the device and install the certificate on the device.

See also

Export, import and installation for PC (Page 170)

Making certificates available (Page 160)

5.14.2 Recreating the entire configuration

When the root certificate expires, you must recreate the entire configuration of the certificate authority. WinCC Certificate Manager supports you in this.

Requirement

A certificate authority has been created and configured in Certificate Manager on the certificate authority device.

Procedure

1. On the certificate authority device, open Certificate Manager.
2. Select the "CA configuration" tab.
   You will see the configuration of the certificate authority.
3. Right-click the root certificate and select "Recreate all".
4. The "Recreate certificate authority" dialog opens.
   The properties of the previous certificate authority are taken over as default. Change them if necessary.
5. Enter the same password as when you created the certificate authority and confirm it.
6. Click "Create".

Result

The entire configuration of the certificate authority is recreated:

- Private key
- Root certificate
- CRL file
- All devices and their application certificates

Next steps

- Export the certificate configuration of the PCs. Import and install it on the PCs.
- Distribute the root certificate and CRL file to the external communication peers.

See also

Export, import and installation for PC (Page 170)

Making certificates available (Page 160)

5.14.3 Updating a CRL file

When the root certificate is created in WinCC Certificate Manager, the CRL file receives a validity period of 24 months.

At the end of this lifetime, you must update the CRL file.

Requirement

A certificate authority has been generated on the certificate authority device in Certificate Manager.

Procedure

1. On the certificate authority device, open Certificate Manager.
2. Select the "CA configuration" tab.
3. Under the root certificate, right-click the "Certificate Revocation List" node.
4. Select "Update".

Result

A new CRL file with a lifetime of 24 months is created.

Next step

Export the file and distribute it.

See also

Exporting root certificate and CRL file (Page 177)

Creating a certificate authority and root certificate (Page 162)

Making certificates available (Page 160)

5.15 Backing up the certificate authority

Procedure

To back up all data of the certificate authority, proceed as follows:

1. Open WinCC Certificate Manager on the certificate authority device.
2. Select the "CA configuration" tab.
3. Right-click the root certificate and select "Export > Full backup".
4. Enter and repeat a password in the "Export" dialog to protect the backup file.
5. Click "Export".
6. Click on "Save" and select the storage location and the file name.

Result

The entire configuration of the certificate authority is written to a backup file.

Loading the backup

1. Open Certificate Manager.
2. In the "CA configuration" tab, double-click the "Open configuration ..." entry.
3. Select the backup file and confirm with "Open".
4. Enter the password set when creating the backup and confirm with "Open".

See also

Password requirements (Page 153)

Making certificates available (Page 160)

5.16 Subsequent change of IP address

Introduction

When you add a PC in WinCC Certificate Manager, you specify its device name or FQDN and/or the IP address. This information is written to the application certificates of the device.

If you entered the IP address and it is subsequently changed, the new address and the information in the certificates no longer match. Follow the procedure described below.

If you did not enter the IP address in Certificate Manager, no additional steps are necessary.

Procedure

1. Open Certificate Manager on the certificate authority device.
2. Select the "CA configuration" tab.
3. Delete the PC whose IP address has been changed.
4. Add the PC again with the new IP address.
5. Add the desired application certificates to the PC.
6. Export, distribute, and install the certificate configuration.

Change of IP address of the certificate authority device

If the PC with the changed IP address is also the certificate authority device, proceed as follows:

1. Delete the PC from Certificate Manager.
2. Add the PC again.
3. Recreate the entire configuration of the certificate authority manually. Distribute and install it.

See also

Deleting devices (Page 167)

Adding devices (Page 164)

Add or delete application certificates (Page 168)

Export, import and installation for PC (Page 170)

Creating a certificate authority and root certificate (Page 162)

Making certificates available (Page 160)

Process communication

6

6.1 Communication Basics

Introduction

Communication is defined as the exchange of data between two communication partners.

Communication

Communication partners can be any component of a network that is in a position to communication with others and to exchange data. In the WinCC, these can be central and communication modules in the automation system (AS) as well as communication processors in the PC.

The transferred data between communication partners can serve many different purposes. In the case of WinCC, these may be:

- Controlling a process
- Calling data from a process
- Indicating unexpected states in the process
- Process data archiving

6.2 Basic Rules for Configuring Connections

Acquisition cycle and update time

The acquisition cycles for the tags defined in the configuration software are major factors for the achievable update times.

The update time is the sum of the acquisition cycle, the transmission time and the processing time.

To achieve optimum update times, remember the following points during configuration:

- Optimize the maximum and minimum size of the data areas.
- Define data areas that belong together as belonging together. If you set up one large area instead of multiple small areas, it improves the update time.
- Acquisition cycles that are too small decrease performance. Set the acquisition cycle according to the rate of change of the process values. Take the temperature of an oven for example, it changes much more slowly than the speed of an electrical drive.
- Put the tags of an alarm or a screen in one data area without gaps.
- Changes in the controller can only be detected reliably if these are available for at least one acquisition cycle.
- Set the transmission rate to the highest possible value for error-free transmission.

Images

The refresh rate of screens is determined by the type and volume of data to be visualized.

In the interest of short update times, ensure that you only configure short acquisition times for objects that require fast updates.

Curves

When using bit-triggered curves, if the group bit is set in the "Curve transfer area", all curves for which the bit is set in this area are updated on the WinCC station. It resets the bits in the next cycle.

Only after all bits have been reset in the WinCC station may the group bit be set again in the PLC program.

6.3 WinCC process communication

6.3.1 WinCC process communication

Introduction

You can access process tags (external tags) in an automation system from WinCC. Before you configure the process connection in WinCC, however, you should use a checklist to check whether the following prerequisites have been met:

- The automation system must be equipped with a communication interface supported by a communication driver in WinCC.
- This interface must be configured in the automation system in such a way that the control program can access the interface with the communication calls. The configuration parameters for the communication hardware must be known.
- The addresses of the tags that WinCC should access must be known. Note that the addresses depend on the automation system used.
- The respective communication hardware (communication processor, standard I/O port COMx, ...) must be installed in the WinCC system. In order to install this hardware, the supplied operating system driver (hardware driver) must also have been installed previously. The settings for the hardware and software of the communication processor must be known.
- Depending on the communication processor used in the WinCC system, more settings may have to be made. When using industrial Ethernet or PROFIBUS, for example, a local database must be created. These connection parameters also have to be known.

For operation in runtime, a physical connection must also exist between WinCC and the AS so that you can access the external tags.

S7DOS configuration

If you are using S7DOS, you require the IPv4 protocol as of version "S7DOS V9".

Therefore, leave the IPv4 protocol activated in the Ethernet properties for the network adapter or the SIMATIC Ethernet CPs.

In this way, you ensure that the module detection of S7DOS works for the TCP, RFC1006 and ISO protocols.

6.3.2 Principle of WinCC communication

Introduction

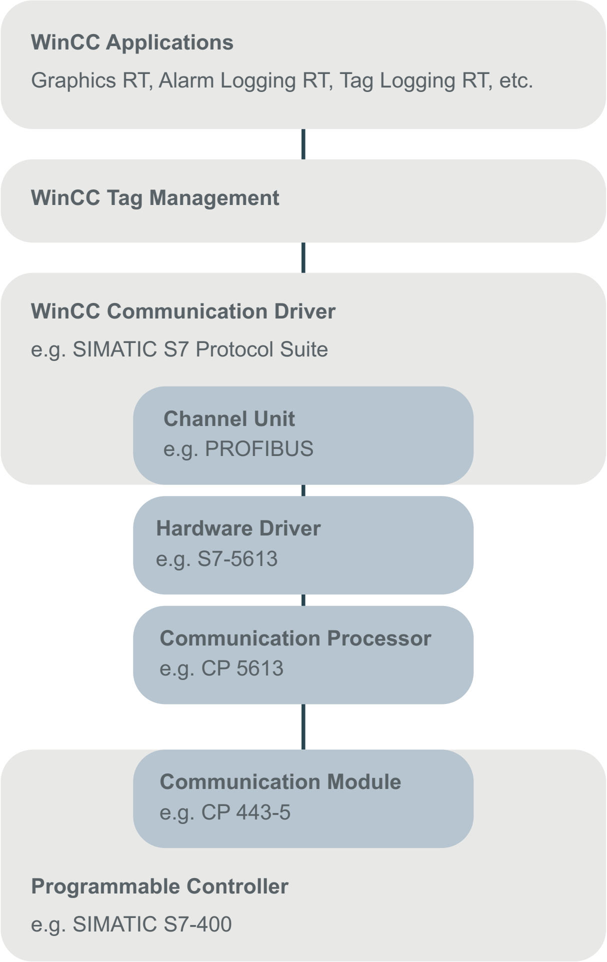
WinCC manages its tags centrally using so-called Tag Management. All data and tags created in a project and stored in the project database are acquired and managed in WinCC Runtime.

All applications, such as Graphics Runtime, Alarm Logging Runtime or Tag Logging Runtime (Global Script), must request the data in the form of WinCC tags from Tag Management.

Communication between WinCC and automation system (AS)

In industrial communication with WinCC, communication means that information is exchanged using tags and process values.

For acquisition of process values, the WinCC communication driver sends request frames to the AS. This in turn sends the requested process values back to WinCC in corresponding response frames.



A physical connection between WinCC and the AS must exist to begin with.

The properties of this connection, such as transmission medium and communication network, define the conditions for communication and are needed for configuring the communication in WinCC.

Communication driver

A communication driver is a software component that establishes a connection between an AS and Tag Management of WinCC, which enables the supply of WinCC tags with process values. In WinCC, there are a number of communication drivers for connecting different ASs using various bus systems.

Each communication driver can be integrated only once in a WinCC project.

In WinCC, a communication driver is also called a "channel" and has the file extension "*.chn". All of the communication drivers installed on the computer are located in subdirectory "\bin" of the WinCC installation directory.

A communication driver has different channel units for different communication networks.

Channel unit

Each channel unit forms the interface to exactly one underlying hardware driver and therefore to exactly one communications processor in the PC. Each channel unit used must therefore be assigned to the associated communications processor.

For some channel units, an additional configuration is performed in the so-called system parameters.

For channel units that work on the transport layer (Layer 4) of the OSI model, the transport parameters are also defined.

Connection (logical)

If WinCC and the AS are correctly connected physically, a communication driver and an associated channel unit are needed in WinCC in order to establish or configure a (logical) connection to the AS.

The data exchange will take place via this connection in Runtime. You can use system tags to establish or terminate this connection and to query the connection status.

In WinCC, a connection is a configured, logical assignment of two communication partners for executing a certain communication service.

Every connection has two end points that also contain necessary information for addressing the communication partner and other attributes for establishing the connection.

A connection is configured under a channel unit with its specific connection parameters. Several connections can also be created under a channel unit, depending on the communication driver.

See also

Configuring tags for the connection status in Runtime (Page 193)

6.3.3 Configuring tags for the connection status in Runtime

When WinCC Runtime is activated, the connection to the configured controllers is established.

To selectively deactivate or activate individual connections in Runtime, use a system tag of the "ConnectionStates" tag group.

Another system tag enables querying of the current connection status.

Channels that are supported

The system tags for the connection status are available for the communication drivers that are supported as of WinCC V7.5.

The "System Info" channel, which is only used to evaluate system information, is an exception.

Diagnostics: Performance of a connection

Use the WinCC performance tags to evaluate the time behavior of a connection. Additional information:

- "Check connection with performance tags (Page 611)"

Tag group "ConnectionStates"

In order to specify or determine the connection status of a channel, the following system tags are created for each connection:

- @<Connectionname>@ForceConnectionStateEx
  Use the tag to establish or terminate the connection in Runtime.
- @<Connectionname>@ConnectionStateEx
  Use the tag to determine the status of the connection in runtime.

The tags have the tag type "Unsigned 32-bit value" (DWORD).

If you change the name of the connection, the two system tags are also renamed.

Tag values

Tag	Use	Value	Explanation
@<...>@ForceConnectionStateEx	Determine the connection state	1	Establishment of connection Start value = 1: When Runtime is activated, the connection is established.
		0	Termination of connection Start value = 0: When Runtime is activated, the connection remains deactivated. The tags of the connection are not archived.
@<...>@ConnectionStateEx	Determine the current connection status	1	The connection is ready to use.
		0	The connection is interrupted or terminated.

Requirement

- The needed connections are created in Tag Management.

Procedure

1. Select the desired connection in the navigation area in Tag Management.
2. In the shortcut menu of the connection, select the entry "Create tags for activation/ deactivation".
   A new tag group "ConnectionStates" is created in the navigation tree under "Internal tags". This group contains the two created tags.
3. Configure a separate address for each tag on the control system.
   To this purpose use an unused or fictitious address. This address is only required for the tag transfer.

Using tags

Querying the connection status

To determine the status of the connection, read the value of the tag "@<...>@ConnectionStateEx".

Terminating a connection

To deactivate a connection, set the value "0" in the "@<...>@ForceConnectionStateEx" tag.

The archiving of the associated process tags is stopped.

Establishing a connection

To reactivate an interrupted connection, set the value "1" in the "@<...>@ForceConnectionStateEx" tag.

The process tags of the corresponding communication channel are archived again.

Visualize the connection status

For example, you can visualize the connection status in the process picture with SVG objects.

As of V2.0, the SVG library "IndustryGraphicLibrary" contains the following objects in the folder "SIMATIC > SystemDiagnostic":

- SysDiag_DiagnosticsIndicator
- SysDiag_SignalLamp
- SysDiag_SignalTower

See also

How to Use the "Status - Logical Connections" Function to Check a Channel (Page 614)

Using the Tag Status to Monitor Connection Status (Page 675)

Check connection with performance tags (Page 611)

STEP 7 "Secure Communication" in the SIMATIC S7-1200, S7-1500 Channel. (Page 508)

6.3.4 External tags

6.3.4.1 External tags

Introduction

In order to obtain access to certain data of an AS, tags are needed in WinCC.

These tags, which depend on the connection to an AS, are called external tags.

In contrast, tags that have no process connection are called internal tags.

Data type and type conversion

When configuring an external tag, you must specify the tag name as well as a data type and, for some data types, you must also specify a type conversion:

The data type determines the data format in WinCC.

The type conversion is used to specify the conversion from AS data format to WinCC data format. The type conversion applies to both transmission directions:

- In the AS: e.g. for certain functions (such as timer values / BCD displays) or due to the information to be addressed (e.g. byte or word address in data block or I/O area).
- In WinCC: e.g. for analog value processing or calculations.

In practice, the AS data format is usually preset. The following possibilities then exist for the choice of WinCC format:

- The WinCC data format can match the AS format.
  This is done by selecting a type conversion that uses the same formats on both sides and takes the sign into account, depending on the WinCC data type, e.g. "WordToSignedWord".
  If this cannot be achieved with the selected data type, it must be changed in WinCC.
- The WinCC format is based on the value processing in WinCC.

You must observe the following when choosing the data type and, if necessary, the type conversion:

- Sign:
  Should it be taken into account in the conversion?
  Can negative tag values also occur during operation, for example, in the case of closed-loop control errors as a percentage?
- Value range:
  Are tag values that occur during operation within the value range of both formats or can a possible overflow of the value be expected in WinCC or the AS?
  If an overflow occurs, a value cannot be represented on the other side or it can lead to problems during subsequent processing.
- Different type conversions with the same value range:
  It is possible that several type conversions of a data type have the same value range, such as "ByteToUnsignedDword" and "ByteToUnsignedWord" with value range [0...127].
  In this case, always check the format of the AS data and whether this format wastes resources unnecessarily because of over-dimensioning (e.g. DWord instead of Word).

If the value range required in the AS is not covered with the selected type conversion, you must change the data type in WinCC.

Note

Communication disruption due to incorrect configuration

If a process tag is configured incorrectly, for example, due to an addressing error, communication with the automation system can be disrupted.

WinCC data types and type conversion

The table shows which WinCC data types support a type conversion.

Data type	Type conversion
Binary tag	No
Unsigned 8-bit value	Yes
Signed 8-bit value	Yes
Unsigned 16-bit value	Yes
Signed 16-bit value	Yes
Unsigned 32-bit value	Yes
Signed 32-bit value	Yes
Floating-point number 32-bit IEEE 754	Yes
Floating-point number 64-bit IEEE 754	Yes
Text tag 8-bit character set	No
Text tag 16-bit character set	No
Raw data type	No

Note

Type conversion must be interpretable

For a type conversion, ensure that the data sent by the AS can be interpreted by WinCC within the selected type conversion.

If the data cannot be interpreted by WinCC, an error is entered in the file "WinCC_sys_0x.log" in directory "..\Siemens\WinCC\Diagnose".

Linear scaling of numerical tag types

A linear scaling can be performed for numerical data types.

The value range of a variable in the process can be linearly mapped onto a defined value range of a WinCC tag.

For example, the process may require a setpoint to be set in the unit bar, while this value is to be entered by the user in WinCC in mbar. With linear scaling, the value range in the process [0...1] can be converted to value range [0...1000] of the WinCC tag.

Length information for text tags

Length information is required for tags of data types "Text tag 8-bit character set" and "Text tag 16-bit character set".

A text tag that is to later accommodate 10 characters must have a length of 10 in the case of data type "Text tag 8-bit character set" and a length of 20 in the case of data type "Text tag 16-bit character set".

Addressing in the automation system

WinCC tags are assigned to a data area in the AS.

These must be addressed in the AS in a certain way. The addressing method depends on the type of communication partner.

Prefixes and suffixes for tag names

After downloading the tags from the AS, you can define prefixes and suffixes for the tag names for the instance of the connection.

The prefix or suffix is added automatically for all tags of the connection once you have imported the process tags using the tag selection dialog.

Changing the prefix or suffix does not affect tags already imported.

6.3.4.2 How to Create a New Connection

Introduction

External tags can only be created on the basis of a connection to an AS. If the required connection does not exist, it must be created first.

Requirements

- The required communication processor and the respective hardware driver are installed.
- The desired communication driver is also installed, e.g. "SIMATIC S7 Protocol Suite".

Procedure

1. Select "Tag Management" in the navigation bar in the Configuration Studio.
2. Select the required channel unit in the navigation area, e.g. "PROFIBUS".
3. Select the entry "New Connection..." in the pop-up menu of the channel unit.
4. Give the connection a unique name in the data area.
5. Define the required parameters for this connection in the "Properties" window. More information can be found under Help / Documentation for the relevant channel.

6.3.4.3 An external tag is configured as follows

Introduction

The procedures for creating a tag is similar for almost all data types.

For some data types however, special settings are required (steps 5 - 7).

Requirements

- The required communication processor and the hardware driver are installed.
- The desired communication driver is installed, e.g. "SIMATIC S7 Protocol Suite".
- A connection is already created based on a channel unit (e.g. "PROFIBUS").

Procedure

1. In the tree view of the navigation area, select the connection for which a tag is to be created.
2. Enter a tag name which is unique in the WinCC project, e.g. WinCCTag_01", in the first free cell of the "Name" column.
3. Define the data type for the tag in the "Data type" field, e.g. "Floating-point number 64-bit IEEE 754".
4. In the AS, specify the address area of the tag in the "AS Length" field.
   With channels that do not support bit-/byte-access with binary or 8 bit tags, first the dialog "Bit/Byte Tag" and then the dialog "Tag properties" will also be shown.
   You can find more information under "Principle of the BinWrite-Mechanism".
   Close the dialog "Bit/Byte Tag" or "Tag properties" with the "OK" button.
5. With numerical tags, WinCC suggests a format adaptation in the "Format adaptation" field. Select another format adaptation if necessary. The display is in sequence "X to Y", whereby X = WinCC format and Y = AS format, e.g. "DoubleToDouble".
6. Activate the check box "Linear scaling" to scale a numerical tag linearly. Enter the high and low limits for the "Process value range" (in AS) and "Tag value range" (in WinCC).
7. The "Length" field is activated for a text tag. Enter the length of the text tag in characters here.
8. Close all dialogs using the "OK" button.

6.3.4.4 Format adaptation sorted by WinCC data type

Introduction

When configuring external tags, another format adaptation must be done for all numeric data types.

The data type determines the data format on the WinCC side. The format adaptation also defines the conversion from WinCC format to the AS format. The definition applies for both transfer directions.

Choose the required WinCC data type in the following selection box. You are then provided with a list of the respective possible format adaptations and value ranges in the table below.

WinCC Data Type

Table 6-1 Signed 8-bit value

Format adaptation "Signed 8-bit value"	Value range
CharToUnsignedByte	0...127
CharToUnsignedWord	0...127
CharToUnsignedDword	0...127
CharToSignedByte	-128...+127 (no conversion)
CharToSignedWord	-128...+127
CharToSignedDword	-128...+127
CharToMSBByte	-127...+127
CharToMSBWord	-128...+127
CharToMSBDword	-128...+127
CharToBCDByte	0...99
CharToBCDWord	0...127
CharToBCDDword	0...127
CharToSignedBCDByte	-9...+9
CharToSignedBCDWord	-128...+127
CharToSignedBCDDword	-128...+127
CharToExtSignedBCDByte	-79...+79
CharToExtSignedBCDWord	-128...+127
CharToExtSignedBCDDword	-128...+127
CharToAikenByte	0...99
CharToAikenWord	0...127
CharToAikenDword	0...127
CharToSignedAikenByte	-9...+9
CharToSignedAikenWord	-128...+127
CharToSignedAikenDword	-128...+127
CharToExcessByte	0...99
CharToExcessWord	0...127
CharToExcessDword	0...127
CharToSignedExcessByte	-9...+9
CharToSignedExcessWord	-128...+127
CharToSignedExcessDword	-128...+127

Table 6-2 Unsigned 8-bit value

Format adaptation "Unsigned 8-bit value"	Value range
ByteToUnsignedByte	0...255 (no conversion)
ByteToUnsignedWord	0...255
ByteToUnsignedDword	0...255
ByteToSignedByte	0...127
ByteToSignedWord	0...255
ByteToSignedDword	0...255
ByteToBCDByte	0...99
ByteToBCDWord	0...255
ByteToBCDDword	0...255
ByteToAikenByte	0...99
ByteToAikenWord	0...255
ByteToAikenDword	0...255
ByteToExcessByte	0...99
ByteToExcessWord	0...255
ByteToExcessDword	0...255

Table 6-3 Signed 16-bit value

Format adaptation "Signed 16-bit value"	Value range
ShortToUnsignedByte	0...255
ShortToUnsignedWord	0...32767
ShortToUnsignedDword	0...32767
ShortToSignedByte	-128...+127
ShortToSignedWord	-32768...+32767 (no conversion)
ShortToSignedDword	-32768...+32767
ShortToMSBByte	-127...+127
ShortToMSBWord	-32767...+32767
ShortToMSBDword	-32768...+32767
ShortToBCDByte	0...99
ShortToBCDWord	0...9999
ShortToBCDDword	0...32767
ShortToSignedBCDByte	-9...+9
ShortToSignedBCDWord	-999...+999
ShortToSignedBCDDword	-32768...+32767
ShortToExtSignedBCDByte	-79...+79
ShortToExtSignedBCDWord	-7999...+7999
ShortToExtSignedBCDDword	-32768...+32767
ShortToAikenByte	0...99
ShortToAikenWord	0...9999

Format adaptation "Signed 16-bit value"	Value range
ShortToAikenDword	0...32767
ShortToSignedAikenByte	-9...+9
ShortToSignedAikenWord	-999...+999
ShortToSignedAikenDword	-32768...+32767
ShortToExcessByte	0...99
ShortToExcessWord	0...9999
ShortToExcessDword	0...32767
ShortToSignedExcessByte	-9...+9
ShortToSignedExcessWord	-999...+999
ShortToSignedExcessDword	-32768...+32767

Table 6-4 Unsigned 16-bit value

Format adaptation "Unsigned 16 bit value"	Value range
WordToUnsignedWord	0...65535 (no conversion)
WordToUnsignedByte	0...255
WordToUnsignedDword	0...65535
WordToSignedByte	0...127
WordToSignedWord	0...32767
WordToSignedDword	0...65535
WordToBCDByte	0...99
WordToBCDWord	0...9999
WordToBCDDword	0...65535
WordToAikenByte	0...99
WordToAikenWord	0...9999
WordToAikenDword	0...65535
WordToExcessByte	0...99
WordToExcessWord	0...9999
WordToExcessDword	0...65535
WordToSimaticBCDCounter	0...999
WordToSimaticCounter	0...999

Table 6-5 Signed 32-bit value

Format adaptation "Signed 32 bit value"	Value range
LongToSignedDword	-2147483648...+2147483647 (no conversion)
LongToUnsignedByte	0...255
LongToUnsignedWord	0...65535
LongToUnsignedDword	0...2147483647
LongToSignedByte	-128...+127

Format adaptation "Signed 32 bit value"	Value range
LongToSignedWord	-32768...+32767
LongToSignedQword	-2147483648...+2147483647
LongToMSBByte	-127...+127
LongToMSBWord	-32767...+32767
LongToMSBDword	-2147483647...+2147483647
LongToMSBQword	-2147483648...+2147483647
LongToBCDByte	0...99
LongToBCDWord	0...9999
LongToBCDDword	0...99999999
LongToSignedBCDByte	-9...+9
LongToSignedBCDWord	-999...+999
LongToSignedBCDDword	-9999999...+9999999
LongToSignedBCDQword	-2147483648...+2147483647
LongToExtSignedBCDByte	-79..+79
LongToExtSignedBCDWord	-7999...+7999
LongToExtSignedBCDDword	-79999999...+79999999
LongToExtSignedBCDQword	-79999999...+79999999
LongToAikenByte	0...99
LongToAikenWord	0...9999
LongToAikenDword	0...99999999
LongToSignedAikenByte	-9...+9
LongToSignedAikenWord	-999...+999
LongToSignedAikenDword	-9999999...+9999999
LongToSignedAikenQword	-999999999...+999999999
LongToExcessByte	0...99
LongToExcessWord	0...9999
LongToExcessDword	0...99999999
LongToSignedExcessByte	-9...+9
LongToSignedExcessWord	-999...+999
LongToSignedExcessDword	-9999999...+9999999
LongToSignedExcessQword	-999999999...+999999999
LongToSimaticBCDTimer	10...9990000
LongToSimaticTimer	10...9990000
LongToSimaticLTime	00:00:00.000...596.31.23.647

Table 6-6 Unsigned 32-bit value

Format adaptation "Unsigned 32 bit value"	Value range
DwordToUnsignedDword	0...4294967295 (no conversion)
DwordToUnsignedByte	0...255
DwordToUnsignedWord	0...65535

Format adaptation "Unsigned 32 bit value"	Value range
DwordToUnsignedQword	0...4294967295
DwordToSignedByte	0...127
DwordToSignedWord	0...32767
DwordToSignedDword	0...2147483647
DwordToBCDByte	0...99
DwordToBCDWord	0...9999
DwordToBCDDword	0...99999999
DwordToBCDQword	0...999999999
DwordToAikenByte	0...99
DwordToAikenWord	0...9999
DwordToAikenDword	0...99999999
DwordToAikenQword	0...999999999
DwordToExcessByte	0...99
DwordToExcessWord	0...9999
DwordToExcessDword	0...99999999
DwordToExcessQword	0...999999999
DwordToSimaticBCDTimer	10...9990000
DwordToSimaticTimer	10...9990000
DwordToSimaticLTimeOfDay	00:00:00.000...23.59.59.999

Table 6-7 Floating-point number 32-bit IEEE 754

Format adaptation "Floating-point number 32-bit IEEE 754"	Value range
FloatToFloat	+-3.402823e+38 (no conversion)
FloatToUnsignedByte	0...255
FloatToUnsignedWord	0...65535
FloatToUnsignedDword	0...4.294967e+09
FloatToSignedByte	-128...+127
FloatToSignedWord	-32768...+32767
FloatToSignedDword	-2.147483e+09...+2.147483e+09
FloatToDouble	+-3.402823e+38
FloatToMSBByte	-127...+127
FloatToMSBWord	-32767...+32767
FloatToMSBDword	-2.147483e+09...+2.147483e+09
FloatToBCDByte	0...99
FloatToBCDWord	0...9999
FloatToBCDDword	0...9.999999e+07
FloatToSignedBCDByte	-9...+9
FloatToSignedBCDWord	-999...+999
FloatToSignedBCDDword	-9999999...+9999999
FloatToExtSignedBCDByte	-79...+79

Format adaptation "Floating-point number 32-bit IEEE 754"	Value range
FloatToExtSignedBCDWord	-7999...+7999
FloatToExtSignedBCDDword	-7.999999e+07...+7.999999e+07
FloatToAikenByte	0...99
FloatToAikenWord	0...9999
FloatToAikenDword	0...9,999999e+07
FloatToSignedAikenByte	-9...+9
FloatToSignedAikenWord	-999...+999
FloatToSignedAikenDword	-9999999...+9999999
FloatToExcessByte	0...99
FloatToExcessWord	0...9999
FloatToExcessDword	0...9.999999e+07
FloatToSignedExcessByte	-9...+9
FloatToSignedExcessWord	-999...+999
FloatToSignedExcessDword	-9999999...+9999999
FloatToSimaticBCDTimer	10...9990000
FloatToS5Float	+-1.701411e+38
FloatToSimaticTimer	10...9990000

Table 6-8 Floating-point number 64-bit IEEE 754

Format adaptation "Floating-point number 64-bit IEEE 754"	Value range
DoubleToDouble	+-1.79769313486231e+308 (no conversion)
DoubleToUnsignedByte	0...255
DoubleToUnsignedWord	0...65535
DoubleToUnsignedDword	0...4294967295
DoubleToUnsignedQword	0...18446744073709551616
DoubleToSignedByte	-128...+127
DoubleToSignedWord	-32768...+32767
DoubleToSignedDword	-2147483648...+2147483647
DoubleToSignedQword	-9223372036854775808...+9223372036854775808
DoubleToFloat	+-3.402823e+38
DoubleToMSBByte	-127...+127
DoubleToMSBWord	-32767...+32767
DoubleToMSBDword	-2147483647...+2147483647
DoubleToMSBQword	-9223372036854775808...+9223372036854775808
DoubleToBCDByte	0...99
DoubleToBCDWord	0...9999
DoubleToBCDDword	0...99999999
DoubleToBCDQword	0...999999999999999
DoubleToSignedBCDByte	-9...+9
DoubleToSignedBCDWord	-999...+999

Format adaptation "Floating-point number 64-bit IEEE 754"	Value range
DoubleToSignedBCDDword	-9999999...+9999999
DoubleToSignedBCDQword	-999999999999999...+999999999999999
DoubleToExtSignedBCDByte	-79...+79
DoubleToExtSignedBCDWord	-7999...+7999
DoubleToExtSignedBCDDword	-79999999...+79999999
DoubleToExtSignedBCDQword	-799999999999999...+799999999999999
DoubleToAikenByte	0...99
DoubleToAikenWord	0...9999
DoubleToAikenDword	0...99999999
DoubleToAikenQword	0...999999999999999
DoubleToSignedAikenByte	-9...+9
DoubleToSignedAikenWord	-999...+999
DoubleToSignedAikenDword	-9999999...+9999999
DoubleToSignedAikenQword	-999999999999999...+999999999999999
DoubleToExcessByte	0...99
DoubleToExcessWord	0...9999
DoubleToExcessDword	0...99999999
DoubleToExcessQword	0...9999999999999999
DoubleToSignedExcessByte	-9...+9
DoubleToSignedExcessWord	-999...+999
DoubleToSignedExcessDword	-9999999...+9999999
DoubleToSignedExcessQword	-999999999999999...+999999999999999
DoubleToSimaticBCDTimer	10...9990000
DoubleToS5Float	+-1.701411e+38
DoubleToSimaticTimer	10...9990000

6.3.4.5 Format adaptation sorted by AS data type

Introduction

When configuring external tags, another format adaptation must be done for all numeric data types.

The data type determines the data format on the WinCC side. The format adaptation also defines the conversion from WinCC format to the AS format. The definition applies for both transfer directions.

Choose the required AS data type in the following selection box. You are then provided with a list of the respective possible format adaptations and respective value ranges in the table below.

AS data type

Type conversion and value range:

Table 6-9 AikenByte

Type conversion "AikenByte"	Value range
ByteToAikenByte	0...99
AikenByte	0...99
DoubleToAikenByte	0...99
DwordToAikenByte	0...99
FloatToAikenByte	0...99
LongToAikenByte	0...99
ShortToAikenByte	0...99
WordToAikenByte	0...99

Table 6-10 AikenWord

Type conversion "AikenWord"	Value range
ByteToAikenWord	0...255
CharToAikenWord	0...127
DoubleToAikenWord	0...9999
DwordToAikenWord	0...9999
FloatToAikenWord	0...9999
LongToAikenWord	0...9999
ShortToAikenWord	0...9999
WordToAikenWord	0...9999

Table 6-11 AikenDWord

Type conversion "AikenDWord"	Value range
ByteToAikenDword	0...255
CharToAikenDword	0...127
DoubleToAikenDword	0...99999999
DwordToAikenDword	0...99999999
FloatToAikenDword	0...9,999999e+07
LongToAikenDword	0...99999999
ShortToAikenDword	0...32767
WordToAikenDword	0...65535

Table 6-12 AikenQWord

Type conversion "AikenQWord"	Value range
DoubleToAikenQword	0...999999999999999
DwordToAikenQword	0...999999999

Table 6-13 BCDByte

Type conversion "BCDByte"	Value range
ByteToBCDByte	0...99
CharToBCDByte	0...99
DoubleToBCDByte	0...99
DwordToBCDByte	0...99
FloatToBCDByte	0...99
LongToBCDByte	0...99
ShortToBCDByte	0...99
WordToBCDByte	0...99

Table 6-14 BCDWord

Type conversion "BCDWord"	Value range
ByteToBCDWord	0...255
CharToBCDWord	0...127
DoubleToBCDWord	0...9999
DwordToBCDWord	0...9999
FloatToBCDWord	0...9999
LongToBCDWord	0...9999
ShortToBCDWord	0...9999
WordToBCDWord	0...9999

Table 6-15 BCDDWord

Type conversion "BCDDWord"	Value range
ByteToBCDDword	0...255
CharToBCDDword	0...127
DoubleToBCDDword	0...99999999
DwordToBCDDword	0...99999999
FloatToBCDDword	0...9.999999e+07
LongToBCDDword	0...99999999
ShortToBCDDword	0...32767
WordToBCDDword	0...65535

Table 6-16 BCDQWord

Type conversion "BCDQWord"	Value range
DoubleToBCDQword	0...999999999999999
DwordToBCDQword	0...999999999

Table 6-17 Double

Type conversion "Double"	Value range
DoubleToDouble	+-1.79769313486231e+308 (no conversion)
FloatToDouble	+-3.402823e+38

Table 6-18 ExcessByte

Type conversion "ExcessByte"	Value range
ByteToExcessByte	0...99
CharToExcessByte	0...99
DoubleToExcessByte	0...99
DwordToExcessByte	0...99
FloatToExcessByte	0...99
LongToExcessByte	0...99
ShortToExcessByte	0...99
WordToExcessByte	0...99

Table 6-19 ExcessWord

Type conversion "ExcessWord"	Value range
ByteToExcessWord	0...255
CharToExcessWord	0...127
DoubleToExcessWord	0...9999
DwordToExcessWord	0...9999
FloatToExcessWord	0...9999
LongToExcessWord	0...9999
ShortToExcessWord	0...9999
WordToExcessWord	0...9999

Table 6-20 ExcessDWord

Type conversion "ExcessDWord"	Value range
ByteToExcessDword	0...255
CharToExcessDword	0...127
DoubleToExcessDword	0...99999999
DwordToExcessDword	0...99999999
FloatToExcessDword	0...9.999999e+07
LongToExcessDword	0...99999999
ShortToExcessDword	0...32767
WordToExcessDword	0...65535

Table 6-21 ExcessQWord

Type conversion "ExcessQWord"	Value range
DoubleToExcessQword	0...9999999999999999
DwordToExcessQword	0...999999999

Table 6-22 ExtSignedBCDByte

Type conversion "ExtSignedBCDByte"	Value range
CharToExtSignedBCDByte	-79...+79
DoubleToExtSignedBCDByte	-79...+79
FloatToExtSignedBCDByte	-79...+79
LongToExtSignedBCDByte	-79..+79
ShortToExtSignedBCDByte	-79...+79

Table 6-23 ExtSignedBCDWord

Type conversion "ExtSignedBCDWord"	Value range
CharToExtSignedBCDWord	-128...+127
DoubleToExtSignedBCDWord	-7999...+7999
FloatToExtSignedBCDWord	-7999...+7999
LongToExtSignedBCDWord	-7999...+7999
ShortToExtSignedBCDWord	-7999...+7999

Table 6-24 ExtSignedBCDDWord

Type conversion "ExtSignedBCDDWord"	Value range
CharToExtSignedBCDDword	-128...+127
DoubleToExtSignedBCDDword	-79999999...+79999999
FloatToExtSignedBCDDword	-7.999999e+07...+7.999999e+07
LongToExtSignedBCDDword	-79999999...+79999999
ShortToExtSignedBCDDword	-32768...+32767

Table 6-25 ExtSignedBCDQWord

Type conversion "ExtSignedBCDQWord"	Value range
DoubleToExtSignedBCDQword	-799999999999999...+799999999999999
LongToExtSignedBCDQword	-79999999...+79999999

Table 6-26 Float

Type conversion "Float"	Value range
DoubleToFloat	+-3.402823e+38
FloatToFloat	+-3.402823e+38 (no conversion)

Table 6-27 MSBByte

Type conversion "MSBByte"	Value range
CharToMSBByte	-127...+127
DoubleToMSBByte	-127...+127
FloatToMSBByte	-127...+127
LongToMSBByte	-127...+127
ShortToMSBByte	-127...+127

Table 6-28 MSBWord

Type conversion "MSBWord"	Value range
CharToMSBWord	-128...+127
DoubleToMSBWord	-32767...+32767
FloatToMSBWord	-32767...+32767
LongToMSBWord	-32767...+32767
ShortToMSBWord	-32767...+32767

Table 6-29 MSBDWord

Type conversion "MSBDWord"	Value range
CharToMSBDword	-128...+127
DoubleToMSBDword	-2147483647...+2147483647
FloatToMSBDword	-2.147483e+09...+2.147483e+09
LongToMSBDword	-2147483647...+2147483647
ShortToMSBDword	-32768...+32767

Table 6-30 MSBQWord

Type conversion "MSBQWord"	Value range
DoubleToMSBQword	-9223372036854775808...+9223372036854775808
LongToMSBQword	-2147483648...+2147483647

Table 6-31 S5Float

Type conversion "S5Float"	Value range
DoubleToS5Float	+-1.701411e+38
FloatToS5Float	+-1.701411e+38

Table 6-32 SignedByte

Type conversion "SignedByte"	Value range
ByteToSignedByte	0...127
CharToSignedByte	-128...+127 (no conversion)
DoubleToSignedByte	-128...+127
DwordToSignedByte	0...127
FloatToSignedByte	-128...+127
LongToSignedByte	-128...+127
ShortToSignedByte	-128...+127
WordToSignedByte	0...127

Table 6-33 SignedWord

Type conversion "SignedWord"	Value range
ByteToSignedWord	0...255
CharToSignedWord	-128...+127
DoubleToSignedWord	-32768...+32767
DwordToSignedWord	0...32767
FloatToSignedWord	-32768...+32767
LongToSignedWord	-32768...+32767
ShortToSignedWord	-32768...+32767 (no conversion)
WordToSignedWord	0...32767

Table 6-34 SignedDWord

Type conversion "SignedDWord"	Value range
ByteToSignedDword	0...255
CharToSignedDword	-128...+127
DoubleToSignedDword	-2147483648...+2147483647
DwordToSignedDword	0...2147483647
FloatToSignedDword	-2.147483e+09...+2.147483e+09
LongToSignedDword	-2147483648...+2147483647 (no conversion)

Type conversion "SignedDWord"	Value range
ShortToSignedDword	-32768...+32767
WordToSignedDword	0...65535

Table 6-35 SignedQWord

Type conversion "SignedQWord"	Value range
DoubleToSignedQword	-9223372036854775808...+9223372036854775808
LongToSignedQword	-2147483648...+2147483647

Table 6-36 SignedAikenByte

Type conversion "SignedAikenByte"	Value range
CharToSignedAikenByte	-9...+9
DoubleToSignedAikenByte	-9...+9
FloatToSignedAikenByte	-9...+9
LongToSignedAikenByte	-9...+9
ShortToSignedAikenByte	-9...+9

Table 6-37 SignedAikenWord

Type conversion "SignedAikenWord"	Value range
CharToSignedAikenWord	-128...+127
DoubleToSignedAikenWord	-999...+999
FloatToSignedAikenWord	-999...+999
LongToSignedAikenWord	-999...+999
ShortToSignedAikenWord	-999...+999

Table 6-38 SignedAikenDWord

Type conversion "SignedAikenDWord"	Value range
CharToSignedAikenDword	-128...+127
DoubleToSignedAikenDword	-9999999...+9999999
FloatToSignedAikenDword	-9999999...+9999999
LongToSignedAikenDword	-9999999...+9999999
ShortToSignedAikenDword	-32768...+32767

Table 6-39 SignedAikenQWord

Type conversion "SignedAikenQWord"	Value range
DoubleToSignedAikenQword	-999999999999999...+999999999999999
LongToSignedAikenQword	-999999999...+999999999

Table 6-40 SignedBCDByte

Type conversion "SignedBCDByte"	Value range
CharToSignedBCDByte	-9...+9
DoubleToSignedBCDByte	-9...+9
FloatToSignedBCDByte	-9...+9
LongToSignedBCDByte	-9...+9
ShortToSignedBCDByte	-9...+9

Table 6-41 SignedBCDWord

Type conversion "SignedBCDWord"	Value range
CharToSignedBCDWord	-128...+127
DoubleToSignedBCDWord	-999...+999
FloatToSignedBCDWord	-999...+999
LongToSignedBCDWord	-999...+999
ShortToSignedBCDWord	-999...+999

Table 6-42 SignedBCDDWord

Type conversion "SignedBCDDWord"	Value range
CharToSignedBCDDword	-128...+127
DoubleToSignedBCDDword	-9999999...+9999999
FloatToSignedBCDDword	-9999999...+9999999
LongToSignedBCDDword	-9999999...+9999999
ShortToSignedBCDDword	-32768...+32767

Table 6-43 SignedBCDQWord

Type conversion "SignedBCDQWord"	Value range
DoubleToSignedBCDQword	-999999999999999...+999999999999999
LongToSignedBCDQword	-2147483648...+2147483647

Table 6-44 SignedExcessByte

Type conversion "SignedExcessByte"	Value range
CharToSignedExcessByte	-9...+9
DoubleToSignedExcessByte	-9...+9
FloatToSignedExcessByte	-9...+9
LongToSignedExcessByte	-9...+9
ShortToSignedExcessByte	-9...+9

Table 6-45 SignedExcessWord

Type conversion "SignedExcessWord"	Value range
CharToSignedExcessWord	-128...+127
DoubleToSignedExcessWord	-999...+999
FloatToSignedExcessWord	-999...+999
LongToSignedExcessWord	-999...+999
ShortToSignedExcessWord	-999...+999

Table 6-46 SignedExcessDWord

Type conversion "SignedExcessDWord"	Value range
CharToSignedExcessDword	-128...+127
DoubleToSignedExcessDword	-9999999...+9999999
FloatToSignedExcessDword	-9999999...+9999999
LongToSignedExcessDword	-9999999...+9999999
ShortToSignedExcessDword	-32768...+32767

Table 6-47 SignedExcessQWord

Type conversion "SignedExcessQWord"	Value range
DoubleToSignedExcessQword	-999999999999999...+999999999999999
LongToSignedExcessQword	-999999999...+999999999

Table 6-48 SimaticCounter

Type conversion "SimaticCounter"	Value range
WordToSimaticCounter	0...999

Table 6-49 SimaticBCDCounter

Type conversion "SimaticBCDCounter"	Value range
WordToSimaticBCDCounter	0...999

Table 6-50 SimaticTimer

Type conversion "SimaticTimer"	Value range
DoubleToSimaticTimer	10...9990000
DwordToSimaticTimer	10...9990000
FloatToSimaticTimer	10...9990000
LongToSimaticTimer	10...9990000

Table 6-51 SimaticBCDTimer

Type conversion "SimaticBCDTimer"	Value range
DoubleToSimaticBCDTimer	10...9990000
DwordToSimaticBCDTimer	10...9990000
FloatToSimaticBCDTimer	10...9990000
LongToSimaticBCDTimer	10...9990000

Table 6-52 SimaticLTime

Type conversion "SimaticLTime"	Value range
LongToSimaticLTime	00:00:00.000...596.31.23.647

Table 6-53 SimaticLTimeOfDay

Type conversion "SimaticLTimeOfDay"	Value range
DwordToSimaticLTimeOfDay	00:00:00.000...23.59.59.999

Table 6-54 UnsignedByte

Type conversion "UnsignedByte"	Value range
ByteToUnsignedByte	0...255 (no conversion)
CharToUnsignedByte	0...127
DoubleToUnsignedByte	0...255
DwordToUnsignedByte	0...255
FloatToUnsignedByte	0...255
LongToUnsignedByte	0...255
ShortToUnsignedByte	0...255
WordToUnsignedByte	0...255

Table 6-55 UnsignedWord

Type conversion "UnsignedWord"	Value range
ByteToUnsignedWord	0...255
CharToUnsignedWord	0...127
DoubleToUnsignedWord	0...65535
DwordToUnsignedWord	0...65535
FloatToUnsignedWord	0...65535
LongToUnsignedWord	0...65535
ShortToUnsignedWord	0...32767
WordToUnsignedWord	0...65535 (no conversion)

Table 6-56 UnsignedDWord

Type conversion "UnsignedDWord"	Value range
ByteToUnsignedDword	0...255
CharToUnsignedDword	0...127
DoubleToUnsignedDword	0...4294967295
DwordToUnsignedDword	0...4294967295 (no conversion)
FloatToUnsignedDword	0...4.294967e+09
LongToUnsignedDword	0...2147483647
ShortToUnsignedDword	0...32767
WordToUnsignedDword	0...65535

Table 6-57 UnsignedQWord

Type conversion "UnsignedQWord"	Value range
DoubleToUnsignedQword	0...18446744073709551616
DwordToUnsignedQword	0...4294967295

6.3.4.6 Principle of the BinWrite-Mechanism

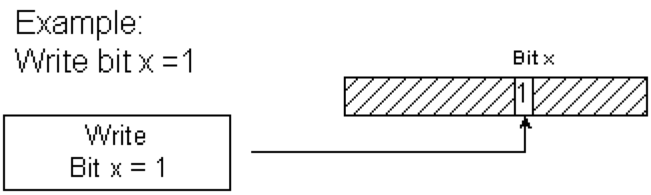
Introduction

In WinCC, not all communication drivers and their channel units support the direct bit-wise or byte-wise access (short: Bit-/Byte-access) to address ranges in a connected automation system. Instead, they use the BinWrite mechanism.

Bit-/Byte-access

With channel units of communication drivers with bit-/byte-access, the desired bit or byte can be read and written directly.

In the following figure, a bit x is allocated the value = 1 via direct bit-/byte-access.



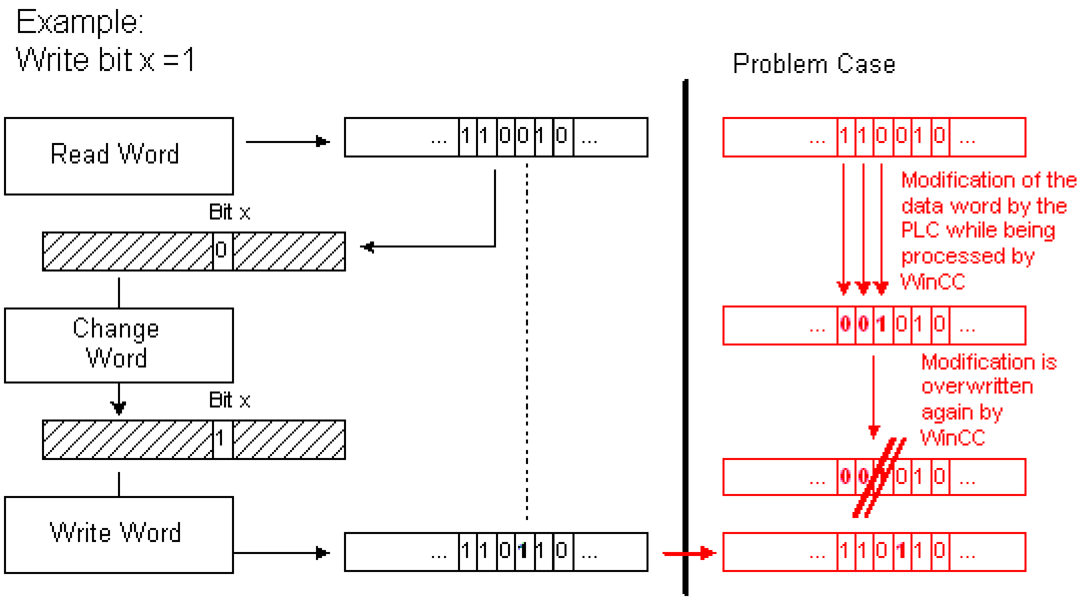
BinWrite Mechanism

The following communication drivers do not support bit-/byte-access and instead use the BinWrite mechanism for the respective channel units:

- Modbus Serial
- SIMATIC S5 Ethernet Layer 4
- SIMATIC S5 Programmers Port AS511
- SIMATIC S5 Serial 3964R
- SIMATIC TI Ethernet Layer 4
- SIMATIC TI Serial

To write a bit or byte, the channel unit first reads the entire data word with the BinWrite mechanism. The data to be addressed is then changed in the word that is read. Then, instead of just the changed bit or byte, the entire (!) word is written back.

In the following figure, a bit x is allocated the value = 1 via the BinWrite mechanism.



Note

If a data word changes in an AS at the same time as this data word was read via the BinWrite mechanism in the WinCC (see figure "Problem case"), then the change is lost in the AS, as soon as WinCC writes the data word back.

6.3.4.7 How to Configure a Tag with "BinWrite"

Introduction

If you want to configure a "Binary tag" for the channel unit of a communication driver, which does not support bit-/byte-access, you have to activate and configure the BinWrite mechanism using a dialog, which otherwise does not exist.

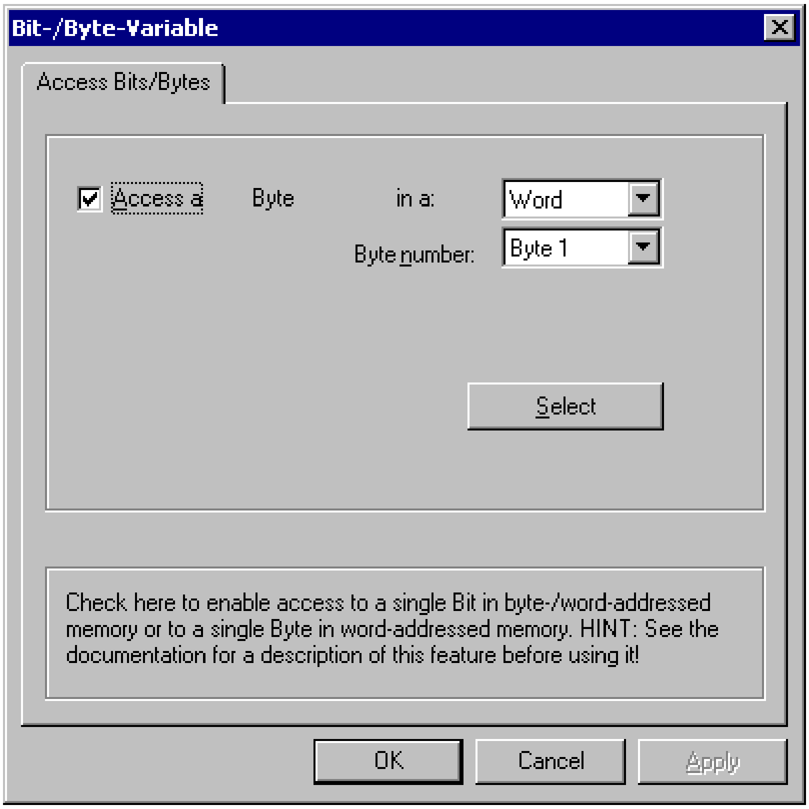
Requirements

- The required communication processor and the hardware driver are installed.
- The desired communication driver, which does not support bit-/byte-access however, is installed, e.g. "SIMATIC S5 Ethernet Layer 4".
- A connection has already been created based on its channel units.

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 one of the following data types:
   - "Binary tag"
   - "Unsigned 8-bit value"
   - "Signed 8-bit value"

5. Open the "Bit/Byte Tag" dialog.
   For this purpose, click in the "Address" field and then the ... button.
   The "Bit/Byte Tag" dialog opens. The address range in the AS is set with this button for channels with bit/byte access.



☑ Access a Byte in a: Word

Byte number: Byte 1

Check here to enable access to a single Bit in byte-/word-addressed memory or to a single Byte in word-addressed memory. HINT: See the documentation for a description of this feature before using it!

6. Activate the check box "Access to one bit" or "Access to one byte" and make the normal settings.
   The display in this dialog depends on the data type selected in step 2.
7. Click "OK" to close the dialog.

6.3.5 Port Addresses for Coupling via Ethernet

Port Addresses

⚠ CAUTION

Ethernet communication

When using Ethernet-based communication, the end users is responsible for the security of his/her own data network. If targeted attacks lead to an overload of the device for instance, the functionality is no longer guaranteed.

When connected via Ethernet, you may require information on the port addresses. This information is required for configuring a firewall or a router. The port addresses that WinCC applications use as defaults are shown in the table.

	Port address TCP/IP	**Port address UDP**
S7 Communication	102	
HTTP (communication; transfer)	80	
HTTPS (communication; transfer)	443	
WebServices (SOAP)	80 HTTP 443 HTTPS	
OPC-XML (CE as OPC Master)	80 HTTP 443 HTTPS	
SendEmail	25	
Transfer (via Ethernet; CE-Stub; PC Loader; PC)	2308 alternative 50523	
Logging (via Ethernet) CSV File	139, 445	137, 138
Modbus Ethernet	502	
Allen-Bradley Ethernet CIP	44818	
Allen-Bradley Ethernet CSP2	2222	

6.3.6 SIMATIC S7-PLCSIM Advanced

WinCC supports the simulation of a virtual controller with the SIMATIC S7-PLCSIM Advanced simulation software.

Additional information can be found in the PLCSIM product documentation.

- Industry Online Support: SIMATIC S7-PLCSIM Advanced (https://support.industry.siemens.com/cs/us/en/ps/24466/man)

See also

Industry Online Support: SIMATIC S7-PLCSIM Advanced (https://support.industry.siemens.com/cs/us/en/ps/24466/man)

Communication channels

7

7.1 Allen Bradley - Ethernet IP

7.1.1 WinCC Channel "Allen Bradley - Ethernet IP"

Introduction

The channel "Allen Bradley - Ethernet IP" is used for linking to Allen-Bradley automation systems. The communication is handled with 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

7.1.2 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

7.1.3 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)

7.1.4 Configuring the Channel

7.1.4.1 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 implementing the TCP/IP protocol, you must define the IP address of the AS for the logic connection. 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.

Connectable controllers

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

- Allen-Bradley ControlLogix 5500
- Allen-Bradley CompactLogix 5300

- PLC-5 with Ethernet Port
- SLC 500 with Ethernet Port, e.g. SLC 5/05
- MicroLogix

Released communication types

The following types of communication are system-tested and released 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

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

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

Introduction

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

- Allen Bradley E/IP ControlLogix
- 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

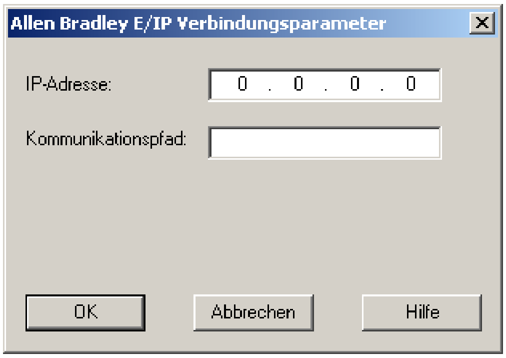
Requirements

- 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.



Allen Bradley E/IP Verbindungsparameter

IP-Adresse: 0 . 0 . 0 . 0

Kommunikationspfad:

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.

7.1.4.3 Examples: Communication path

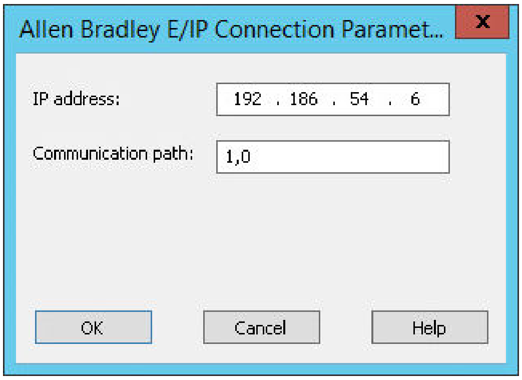
Example 1: Direct connection

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.



Allen Bradley E/IP Connection Paramet...

IP address: 192 . 186 . 54 . 6

Communication path: 1,0

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.:

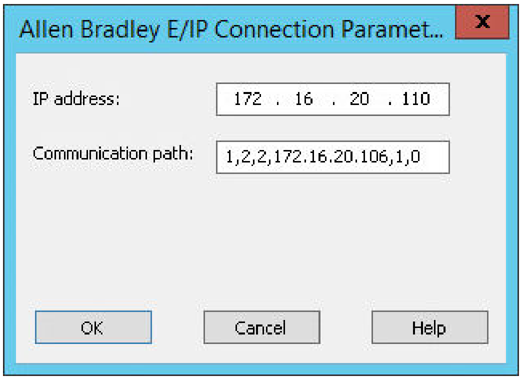
- WinCC - Control Logix (1) - Control Logix (2)
- WinCC - Control Logix (1) - SLC50x (1)

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


Allen Bradley E/IP Connection Paramet...

IP address: 172 . 16 . 20 . 110

Communication path: 1,2,2,172.16.20.106,1,0

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:

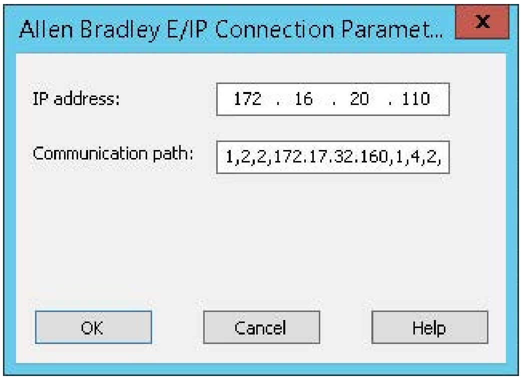
- WinCC - Control Logix (1) - Control Logix (2) - Control Logix (3)
- WinCC - Control Logix (1) - Control Logix (2) - SLC50x (1)

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


Allen Bradley E/IP Connection Paramet...

IP address: 172 . 16 . 20 . 110

Communication path: 1,2,2,172.17.32.160,1,4,2,

7.1.4.4 Configuring the tags

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

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.

Addressing syntax

Notation of addresses

The following tables define the possibilities for writing individual addressing.

Table 7-1 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.

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.

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]

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

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

Introduction

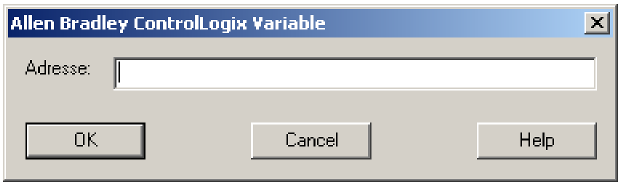
This section shows how you configure a tag for channel unit "Allen Bradley E/IP ControlLogix" in the automation system (AS) address range.

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.

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

Introduction

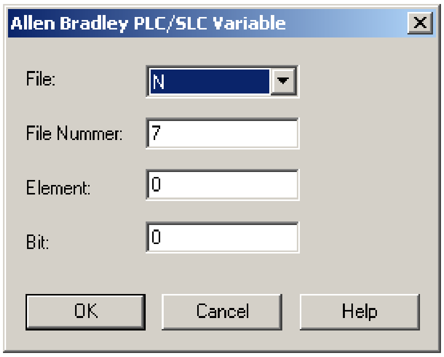
This section shows you how to configure a tag for bit by bit access for the address area in the automation system (AS).

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.



Allen Bradley PLC/SLC Variable

File: N

File Nummer: 7

Element: 0

Bit: 0

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.

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

Introduction

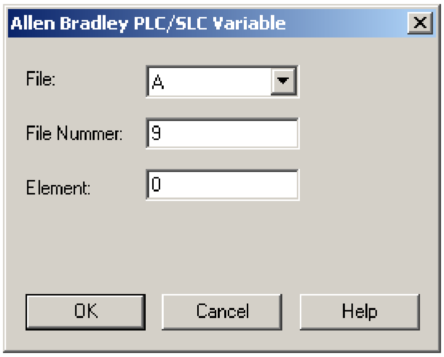
This section shows you how to configure a tag for byte by byte access for the address area in the automation system (AS).

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.



Allen Bradley PLC/SLC Variable

File: A

File Nummer: 9

Element: 0

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.

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

Introduction

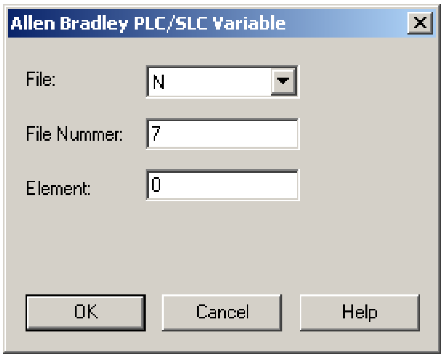
This section shows you how to configure a tag for word by word access to the address area in the automation system (AS).

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.



Allen Bradley PLC/SLC Variable

File: N

File Nummer: 7

Element: 0

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.

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

Introduction

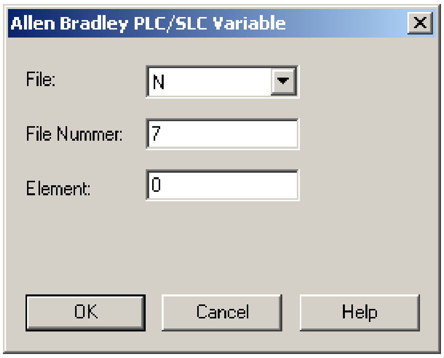
This section shows you how to configure a tag for word by word access to the address area in the automation system (AS).

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.



Allen Bradley PLC/SLC Variable

File: N

File Nummer: 7

Element: 0

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.

7.2 Mitsubishi Ethernet

7.2.1 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

7.2.2 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

7.2.3 Configuring the Channel

7.2.3.1 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 241)

How to configure a "Mitsubishi Q Series" channel unit connection (Page 242)

How to configure a "Mitsubishi iQ-R series" channel unit connection (Page 244)

Configuring the tags (Page 247)

7.2.3.2 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.

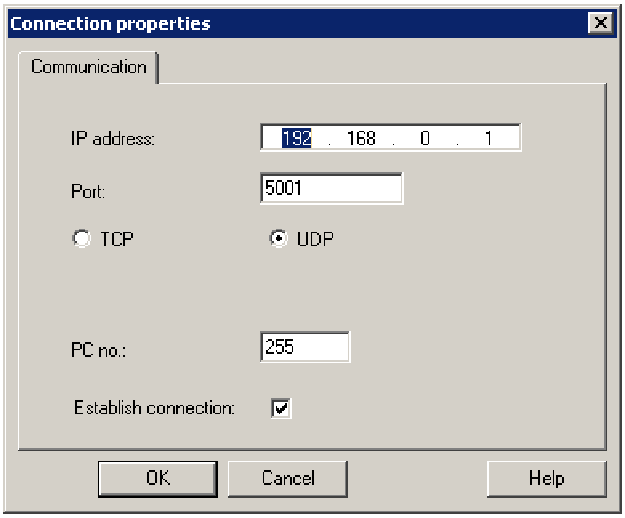
Requirements

- 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.



Connection properties

Communication

IP address: 192 . 168 . 0 . 1

Port: 5001

○ TCP ● UDP

PC no.: 255

Establish connection: ☑

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 240)

How to configure a tag (Page 249)

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

Introduction

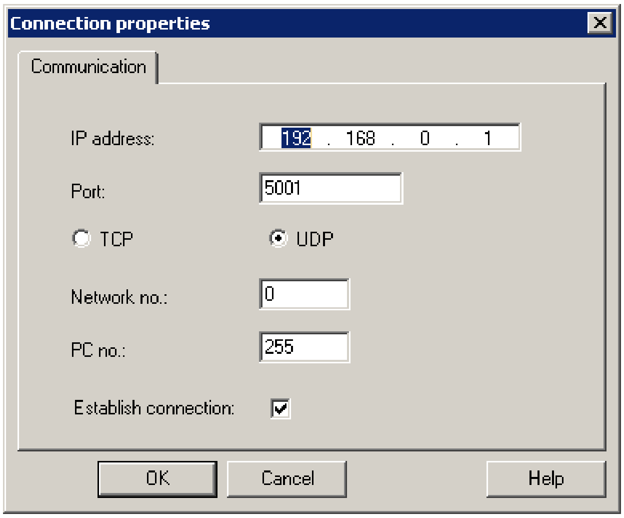
This section shows you how to configure the connection for the "Mitsubishi Q Series" channel unit.

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.



IP address: 192 . 168 . 0 . 1

Port: 5001

○ TCP ● UDP

Network no.: 0

PC no.: 255

Establish connection: ☑

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 240)

How to configure a tag (Page 249)

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

Introduction

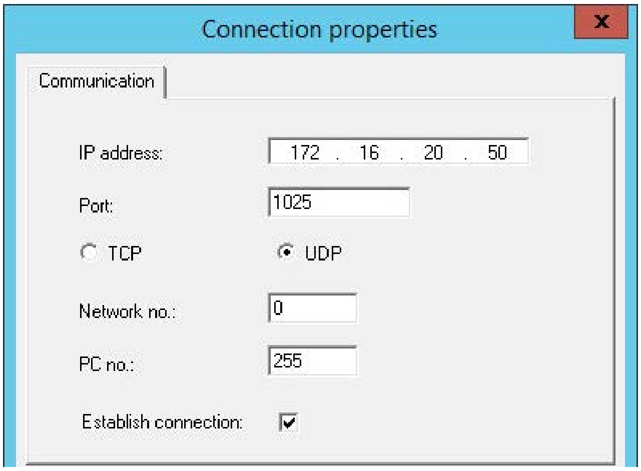
This section shows you how to configure a connection for the "Mitsubishi iQ-R series" channel unit.

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.



Connection properties

Communication

IP address: 172 . 16 . 20 . 50

Port: 1025

○ TCP ● UDP

Network no.: 0

PC no.: 255

Establish connection: ☑

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 240)

How to configure a tag (Page 249)

7.2.3.5 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.

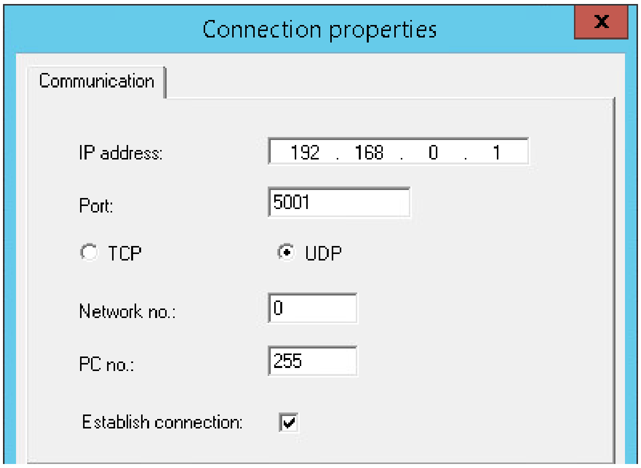
Requirements

- 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.



Connection properties

Communication

IP address: 192 . 168 . 0 . 1

Port: 5001

○ TCP ● UDP

Network no.: 0

PC no.: 255

Establish connection: ☑

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 240)

How to configure a tag (Page 249)

7.2.3.6 Configuring the tags

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				

Address type	Code	Display	Data type	AS that are supported
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
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 249)

How to configure a tag

Introduction

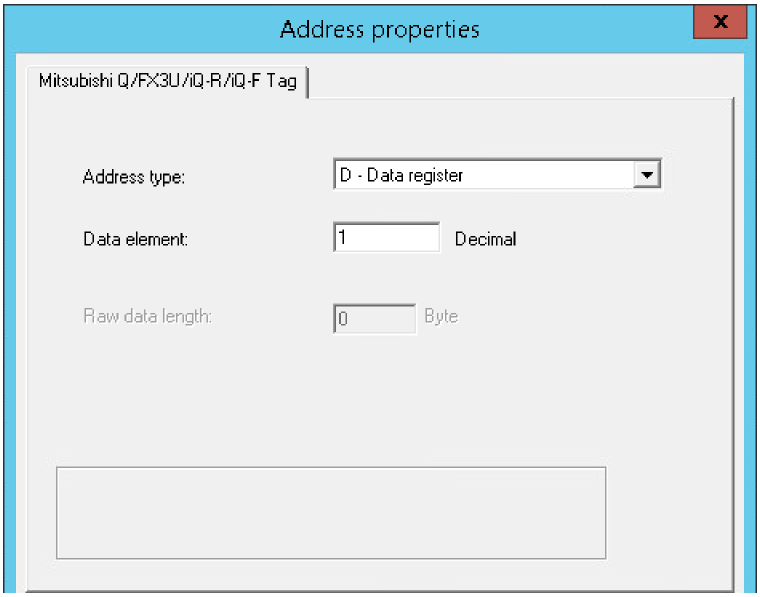
This section shows you how to configure a tag access for the address range in the automation system (AS).

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.



Address type: D - Data register

Data element: 1 Decimal

Raw data length: 0 Byte

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 241)

How to configure a "Mitsubishi Q Series" channel unit connection (Page 242)

How to configure a "Mitsubishi iQ-R series" channel unit connection (Page 244)

Configuring the tags (Page 247)

7.3 Modbus TCPIP

7.3.1 "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.

7.3.2 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

7.3.3 Configuring the Channel

7.3.3.1 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.

When implementing the TCP/IP protocol, you must define the IP address of the AS for the logic connection. 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 communication
- 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.

7.3.3.2 How to configure a connection

Introduction

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

1. Configuring a connection
2. Configuring tags

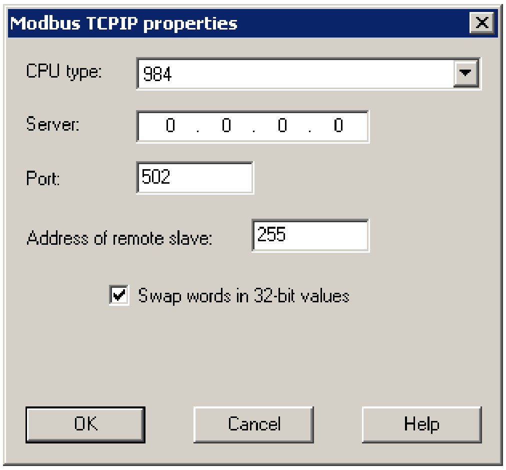
Requirements

- 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.



Modbus TCPIP properties

CPU type: 984

Server: 0 . 0 . 0 . 0

Port: 502

Address of remote slave: 255

☑ Swap words in 32-bit values

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".

7.3.3.3 Configuring the tags

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

Name	Area with CPU Premium/Micro	Area with CPU 984, Compact, Quantum, Momentum	data type
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 255)

How to Configure a Tag with Word by Word Access (Page 257)

How to Configure a Text Tag (Page 258)

How to Configure a Tag with Bit by Bit Access

Introduction

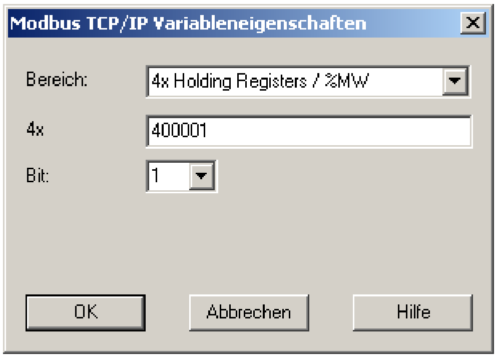
This section shows you how to configure a tag for bit by bit access for the address area in the automation system (AS).

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.



Modbus TCP/IP Variableneigenschaften

Bereich: 4x Holding Registers / %MW

4x: 400001

Bit: 1

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 258)

How to Configure a Tag with Word by Word Access

Introduction

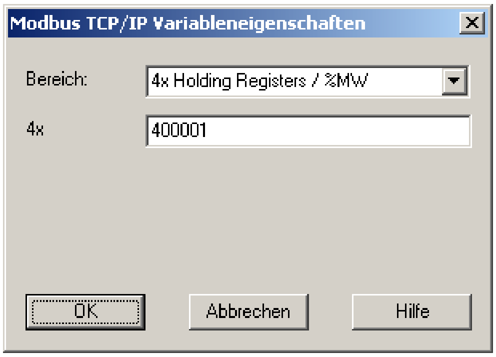
This section shows you how to configure a tag for word by word access to the address area in the automation system (AS).

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.



Modbus TCP/IP Variableneigenschaften

Bereich: 4x Holding Registers / %MW

4x 400001

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.

How to Configure a Text Tag

Introduction

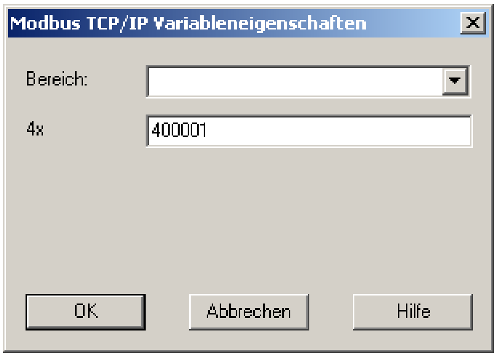
This section shows you how to configure a tag for word by word access to the address area in the automation system (AS).

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.



Modbus TCP/IP Variableneigenschaften

Bereich:

4x 400001

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.

7.4 Omron Ethernet-IP

7.4.1 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

7.4.2 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

7.4.3 Configuring the channel

7.4.3.1 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 both controller families. The configuration differs only in relation to the usable address types of the specific controller family.

Protocol

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

Configuring the logical connection for the "Omron Ethernet-IP" 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.

7.4.3.2 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 CS1 Series" channel unit.

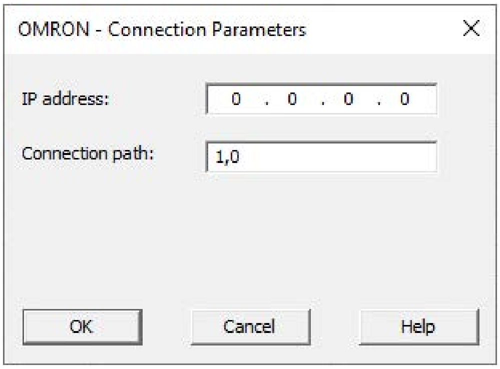
The configuration is carried out identically for the three available channel units.

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 CS1 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.



OMRON - Connection Parameters

IP address: 0 . 0 . 0 . 0

Connection path: 1,0

5. Enter the IP address of the controller.
6. Enter the connection path used for the connection.
7. Close the dialog with "OK".

7.4.3.3 Configuring the tags

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.

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 264)

How to configure a tag

Introduction

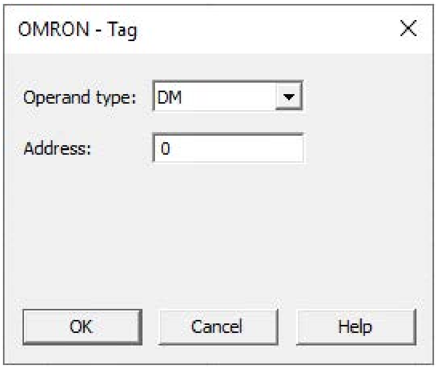
This section shows you how to configure a tag access for the address range in the automation system (AS).

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 Serie

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.



OMRON - Tag

Operand type: DM

Address: 0

6. Set the operand type.

7. Enter the corresponding address.
   The value depends on the configuration of the controller.
8. 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

Address ranges for Omron Ethernet/IP (Page 264)

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

7.5 OPC Channel

7.5.1 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:

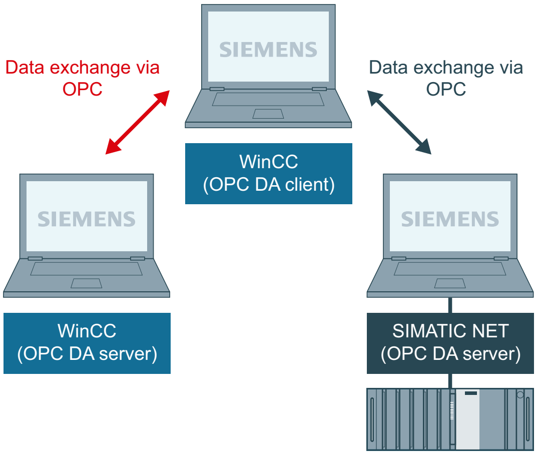
- OPC communication driver
- OPC Item Manager

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.

7.5.2 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

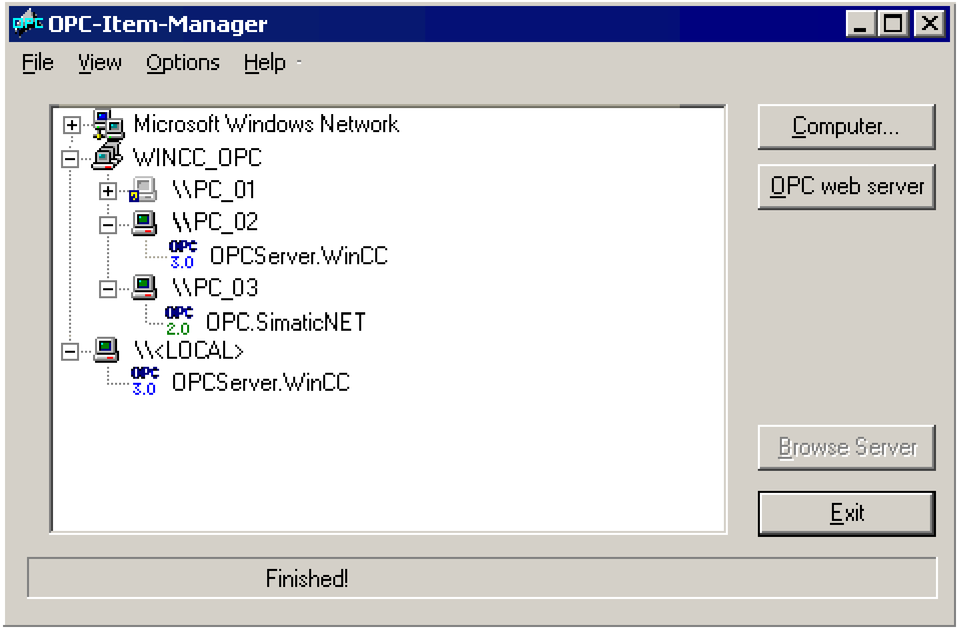
The OPC Item Manager assumes the following tasks:

- 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 ".



Finished!

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.
OPC 3.0	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.

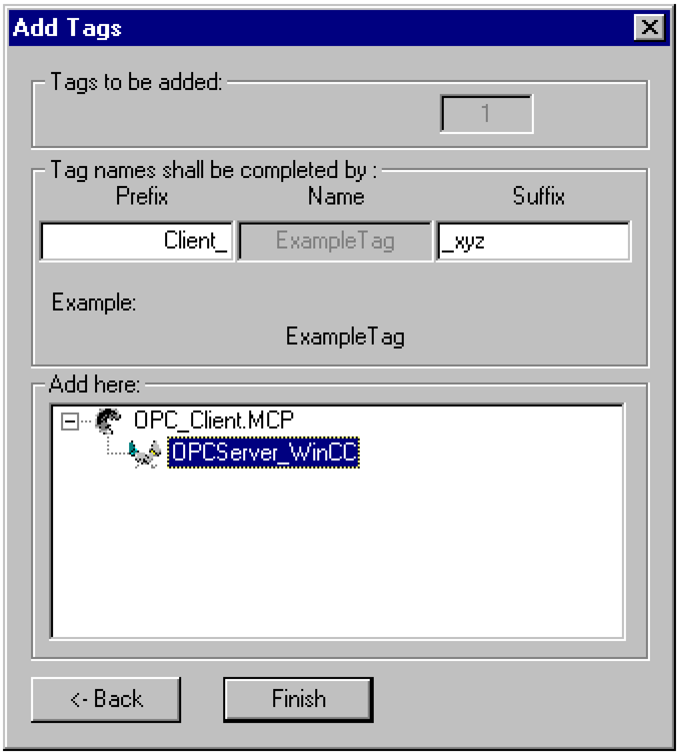
Creating a connection

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



Add Tags

Tags to be added: 1

Tag names shall be completed by :

Prefix	Name	Suffix
Client_	ExampleTag	_xyz

Example:

ExampleTag

Add here:

- OPC_Client.MCP
  - OPCServer_WinCC

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 276)

7.5.3 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 283)

7.5.4 WinCC OPC DA Client

7.5.4.1 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 283)

Accessing a WinCC Tag without the OPC Item Manager (Page 281)

Configuring Access with the OPC Item Manager (Page 277)

Overview of the Supported WinCC Data Types (Page 273)

OPC Item Manager (Page 270)

Diagnosis of Channels and Tags (Page 609)

OPC specifications and compatibility (Page 683)

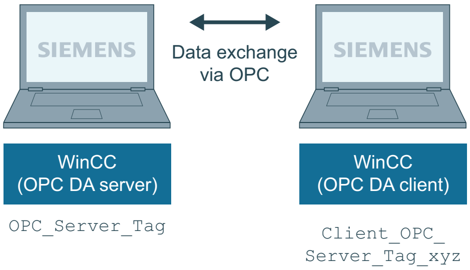
Functionality of the WinCC OPC DA Server (Page 687)

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

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 277)

Configuring the OPC Channel on the WinCC OPC DA Client (Page 276)

Example of WinCC to WinCC Connection (Page 690)

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 277)

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.

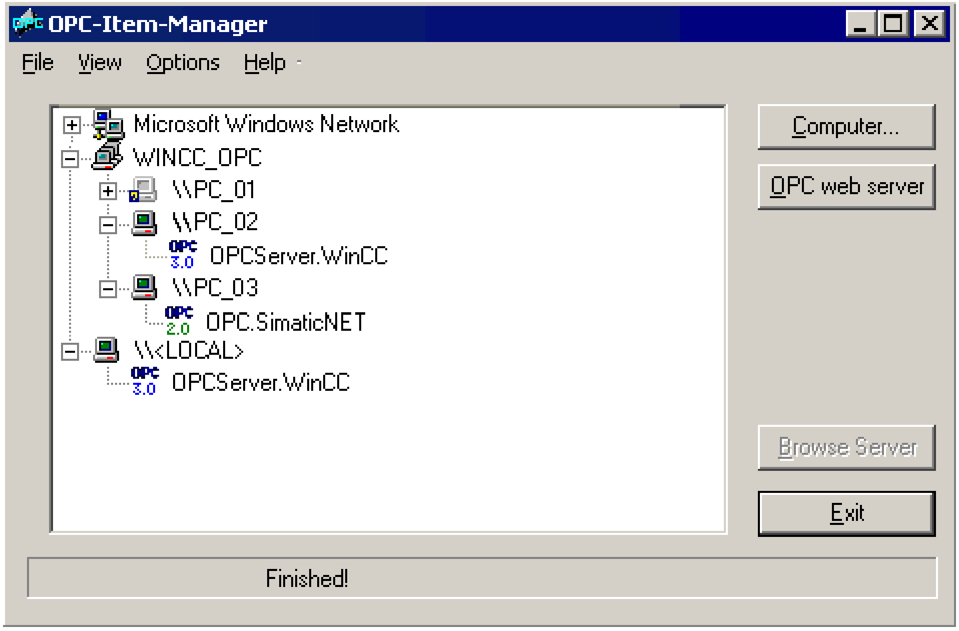
Note

"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.

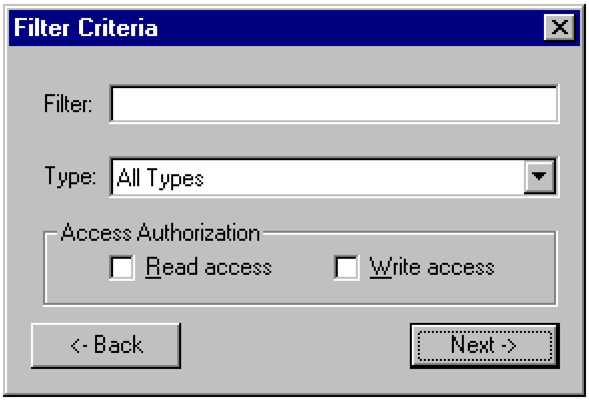
Procedure

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.

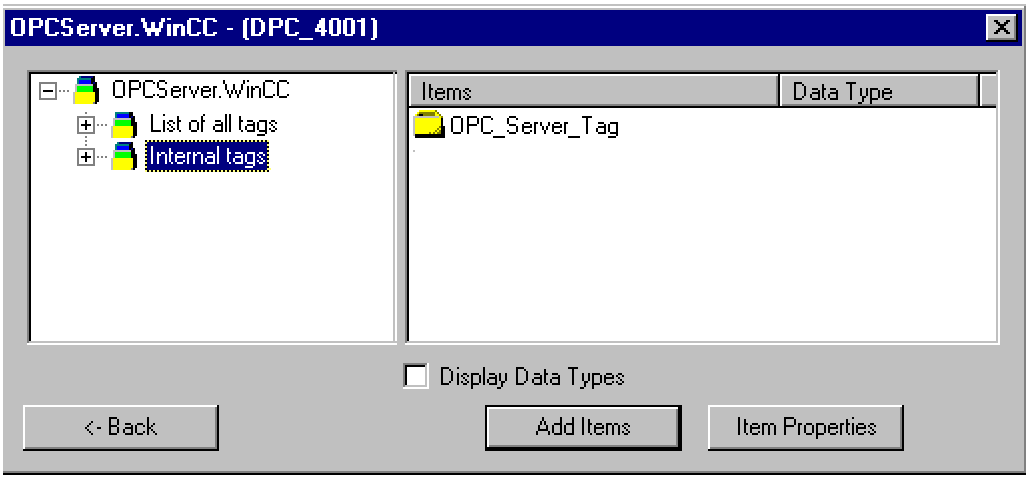


2. Choose the name of the computer to be used as the WinCC OPC DA server from the selection dialog.
   Select "OPCServer.WinCC" from the list displayed.

3. Click the "Browse Server" button.
   The "Filter criteria" dialog is opened.



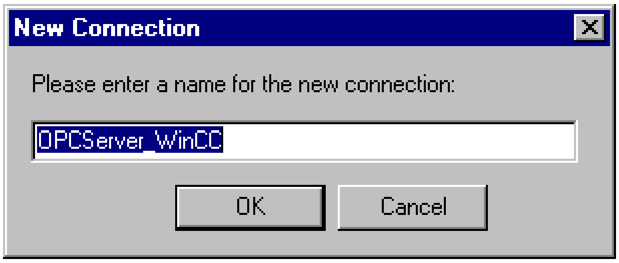
4. Click "Next".
   The "OPCServer.WinCC ..." dialog is opened.



OPCServer.WinCC - (DPC_4001)

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.

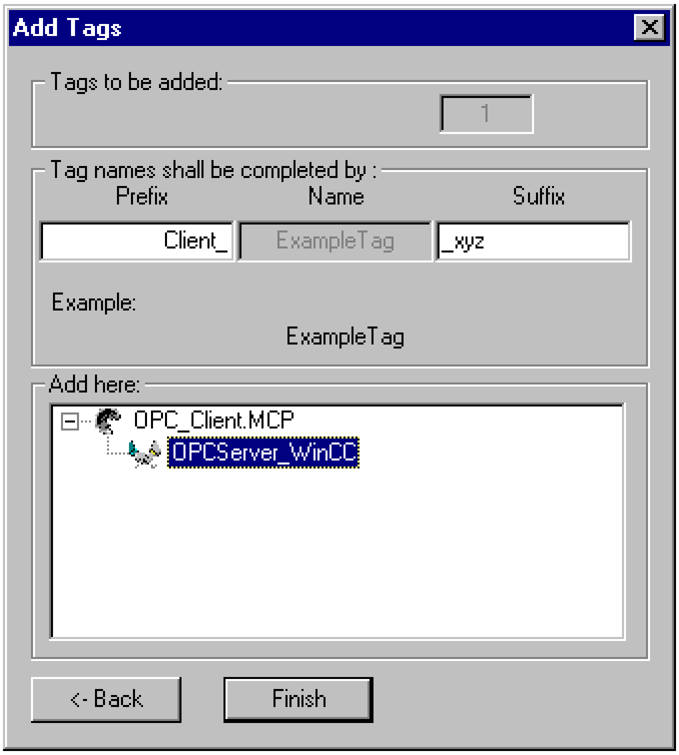


New Connection

Please enter a name for the new connection:

OPCServer_WinCC

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



Add Tags

Tags to be added: 1

Tag names shall be completed by :

Prefix	Name	Suffix
Client_	ExampleTag	_xyz

Example:

ExampleTag

Add here:

OPC_Client.MCP
OPCServer_WinCC

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.

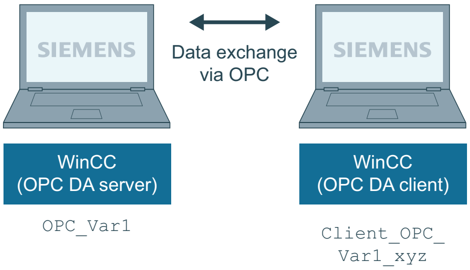
See also

Configuring the OPC Channel on the WinCC OPC DA Client (Page 276)

7.5.4.3 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.

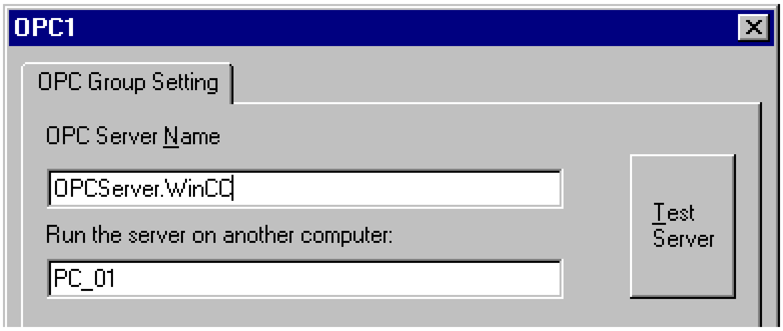
Note

"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.

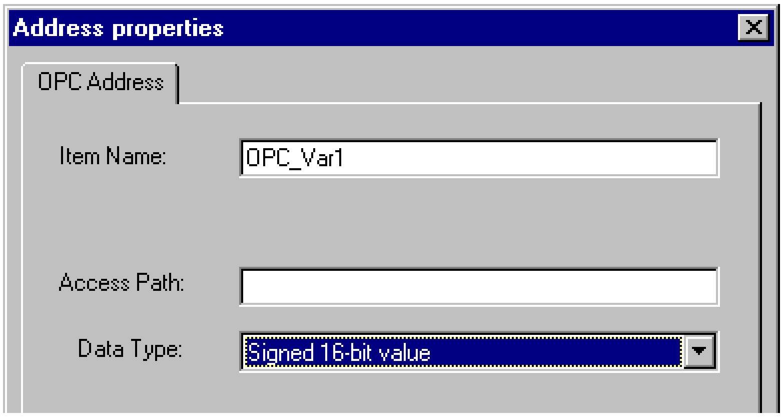


OPC Server Name: OPCServer.WinCC

Run the server on another computer: PC_01

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.



Item Name: OPC_Var1

Access Path:

Data Type: Signed 16-bit value

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.

7.5.4.4 Using Structures on a WinCC OPC DA Client

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 284)

Configuring Structures and Structure Tags on the WinCC OPC DA Server (Page 283)

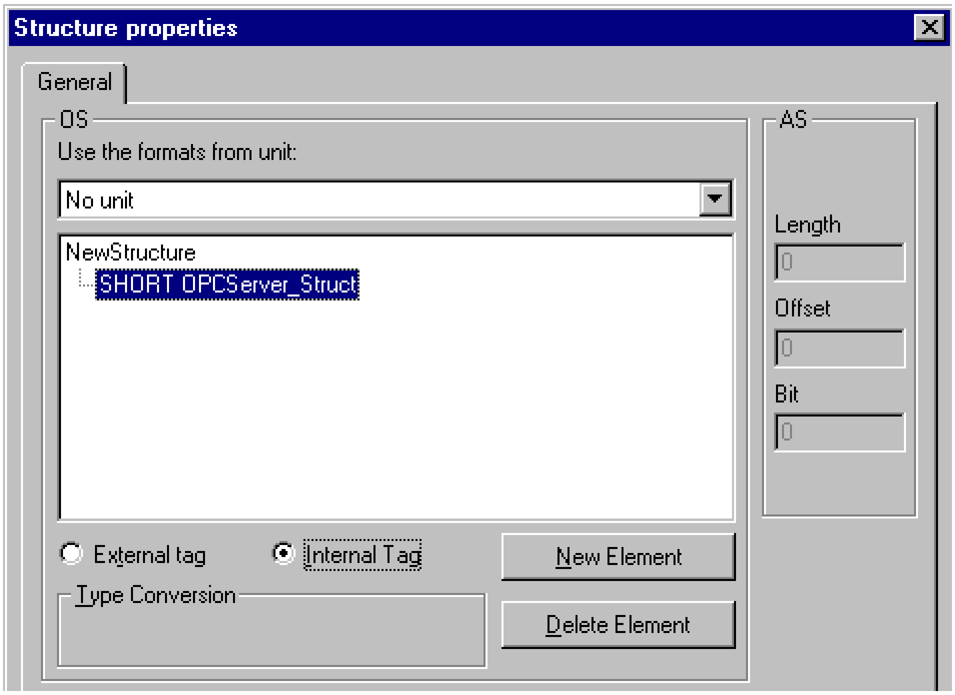
Configuring Structures and Structure Tags on the WinCC OPC DA Server

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.

   Structure properties

   General

   OS

   Use the formats from unit: No unit

   NewStructure
   SHORT OPCServer_Struct

   ○ External tag ● Internal Tag

   Type Conversion

   AS

   Length: 0

   Offset: 0

   Bit: 0

   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 284)

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 276)

Configuring Structures and Structure Tags on the WinCC OPC DA Server (Page 283)

7.5.4.5 Error Handling in the Event of Disturbed OPC DA Communication

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 290)

WinCC is used as the OPC DA client, and the connection is established. (Page 289)

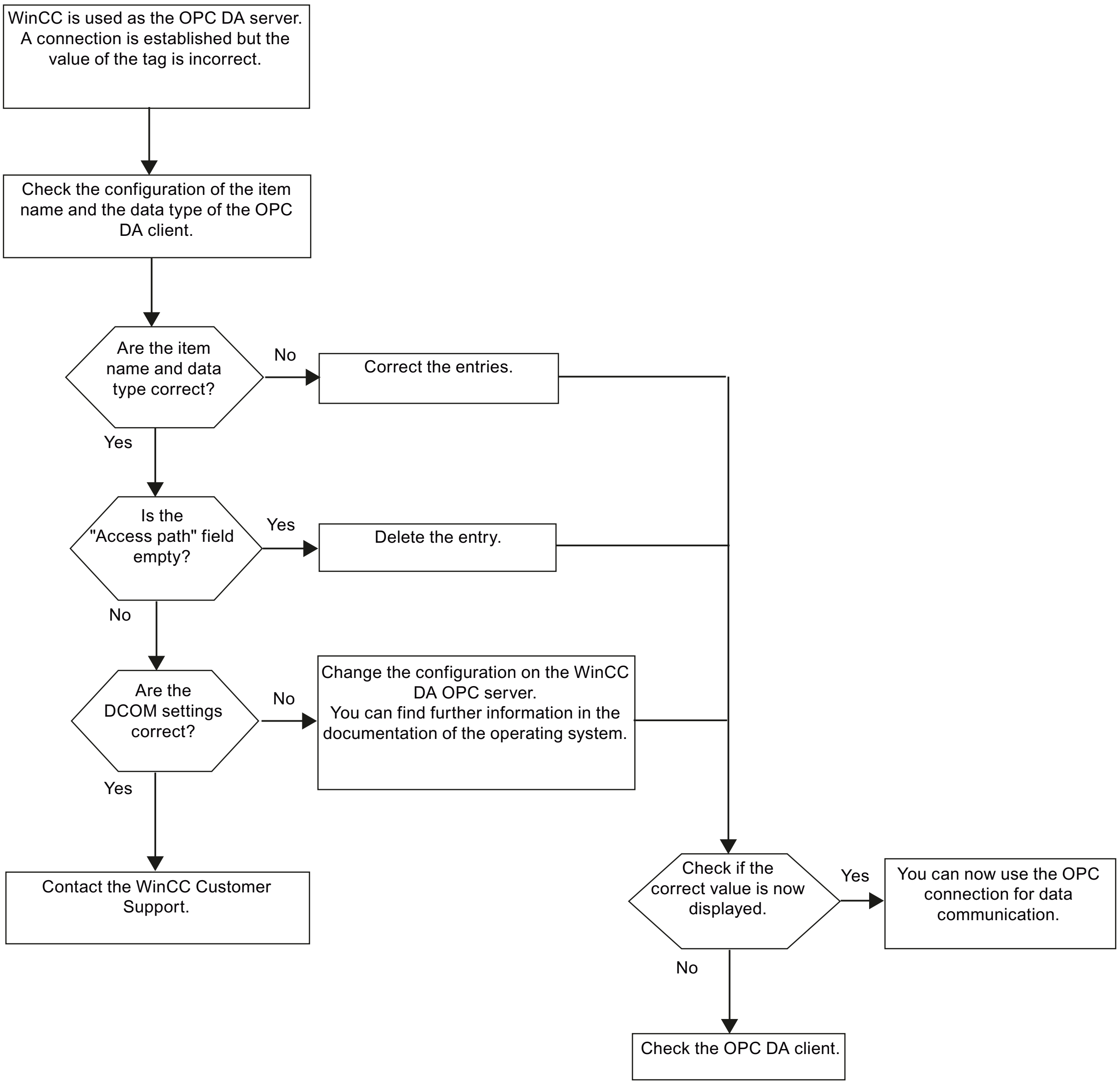
WinCC is used as the OPC DA server, and the connection is not established. (Page 288)

WinCC is used as the OPC DA server, and the connection is established successfully. (Page 287)

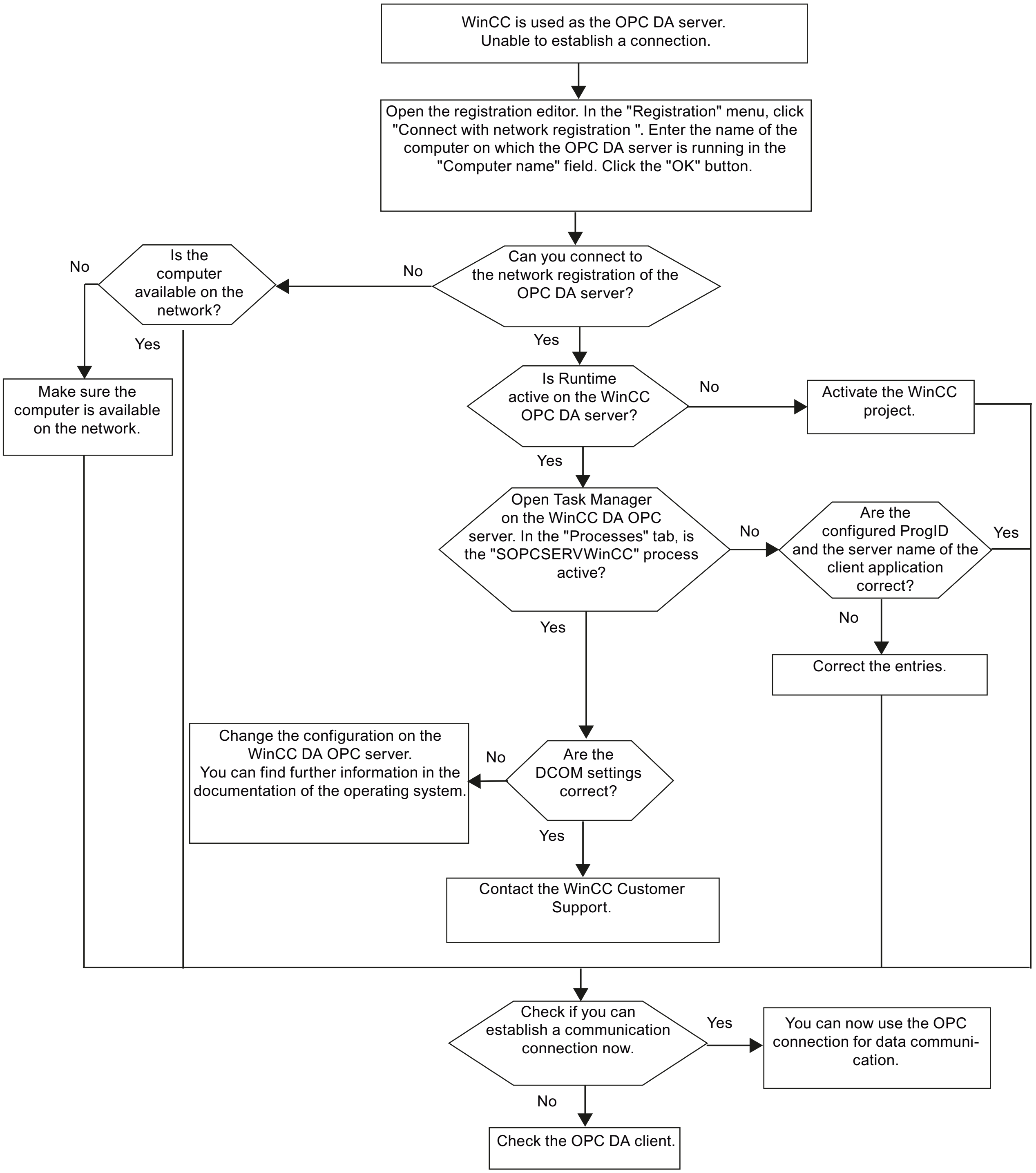
Possibilities for Diagnosing the "OPC" Channel (Page 658)

WinCC as OPC DA Server

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

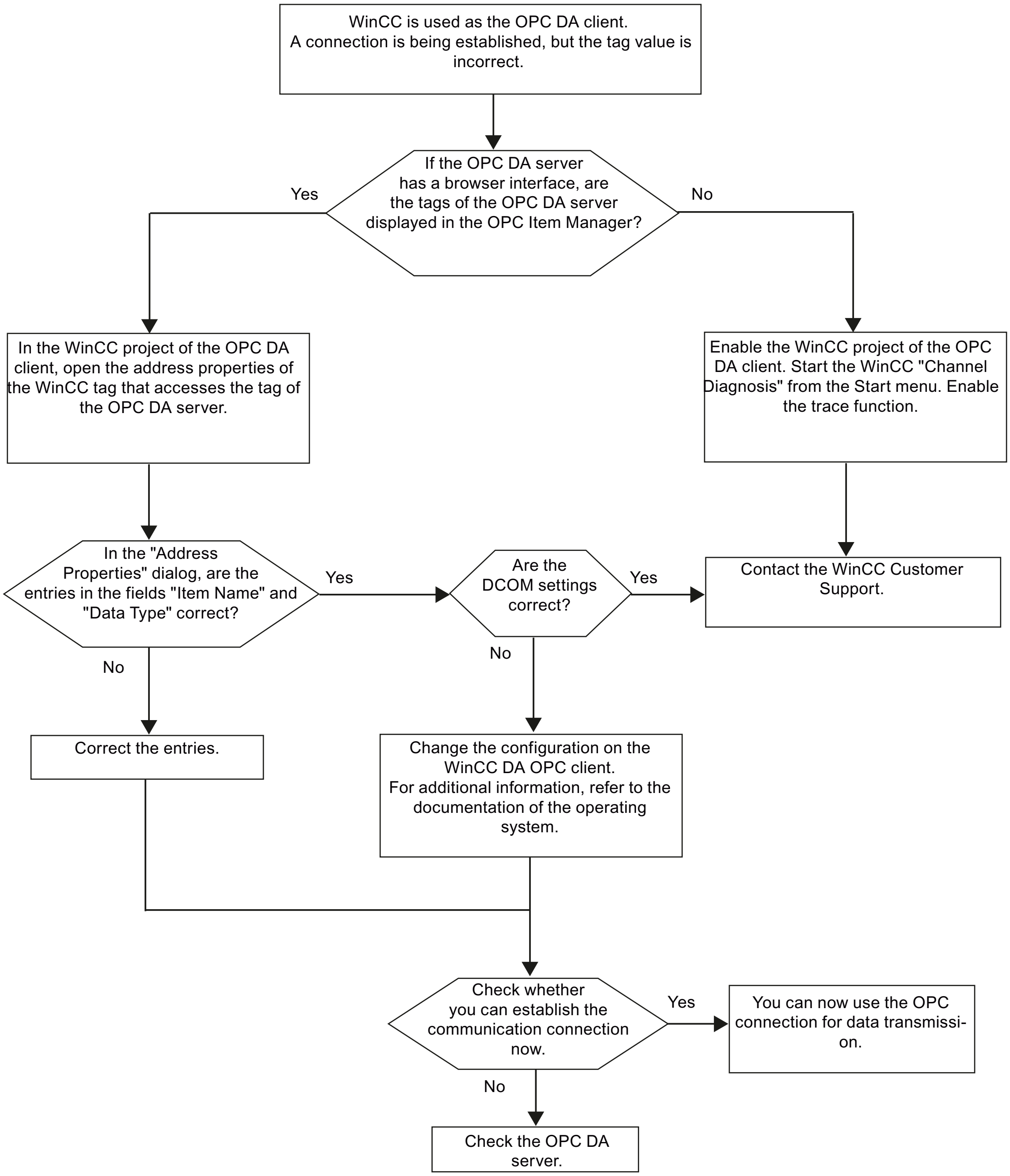

WinCC is used as the OPC DA server. A connection is established but the value of the tag is incorrect.
Check the configuration of the item name and the data type of the OPC DA client.
Are the item name and data type correct?
No
Correct the entries.
Yes
Is the "Access path" field empty?
Yes
Delete the entry.
No
Are the DCOM settings correct?
No
Change the configuration on the WinCC DA OPC server. You can find further information in the documentation of the operating system.
Yes
Contact the WinCC Customer Support.
Check if the correct value is now displayed.
Yes
You can now use the OPC connection for data communication.
No
Check the OPC DA client.

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

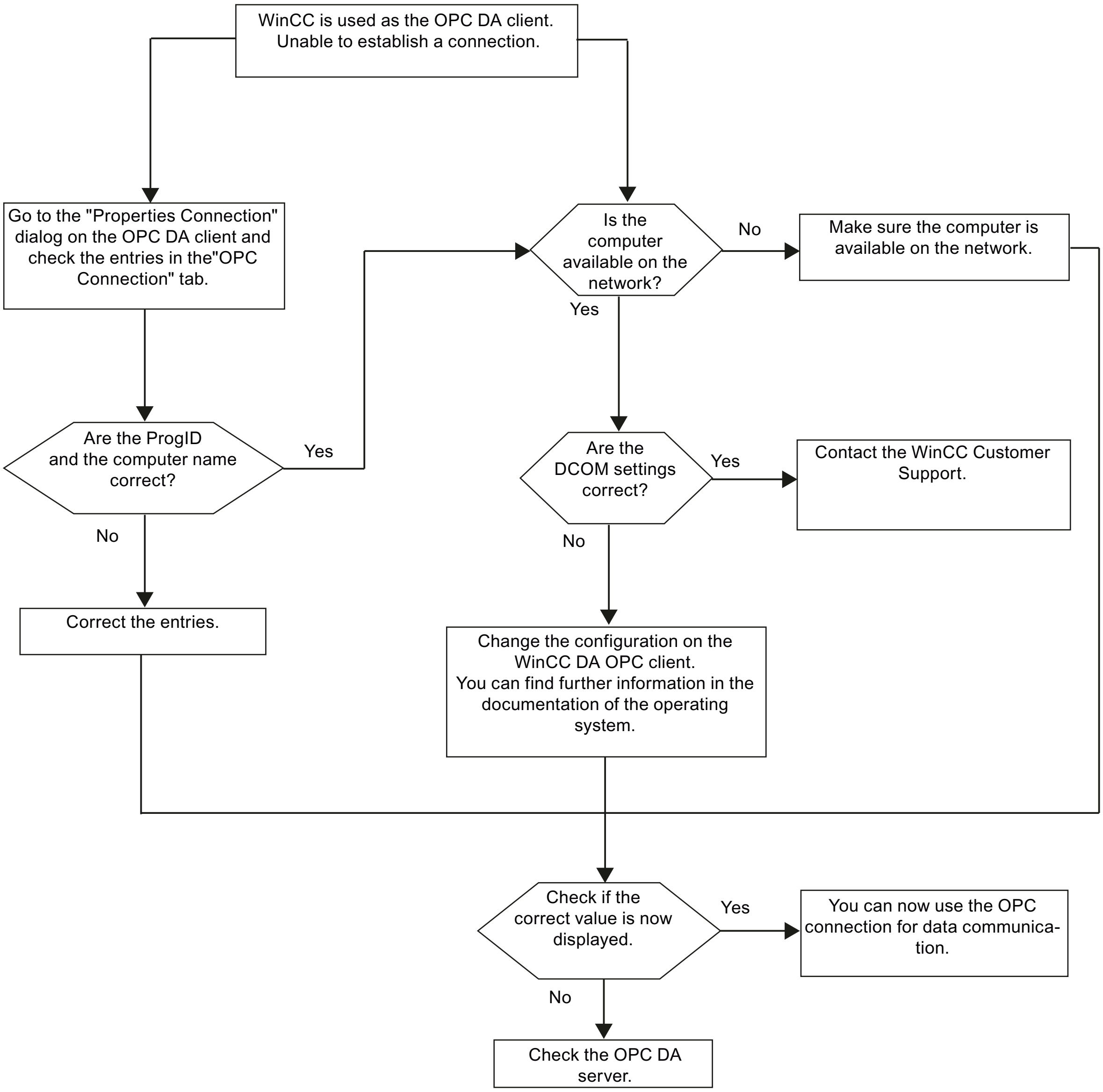

WinCC is used as the OPC DA server.
Unable to establish a connection.
Open the registration editor. In the "Registration" menu, click "Connect with network registration ". Enter the name of the computer on which the OPC DA server is running in the "Computer name" field. Click the "OK" button.
Can you connect to the network registration of the OPC DA server?
No
Is the computer available on the network?
No
Yes
Make sure the computer is available on the network.
Yes
Is Runtime active on the WinCC OPC DA server?
No
Activate the WinCC project.
Yes
Open Task Manager on the WinCC DA OPC server. In the "Processes" tab, is the "SOPCSERVWinCC" process active?
No
Are the configured ProgID and the server name of the client application correct?
Yes
No
Correct the entries.
Yes
Change the configuration on the WinCC DA OPC server.
You can find further information in the documentation of the operating system.
No
Are the DCOM settings correct?
Yes
Contact the WinCC Customer Support.
Check if you can establish a communication connection now.
Yes
You can now use the OPC connection for data communi-cation.
No
Check the OPC DA client.

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.
A connection is being established, but the tag value is incorrect.
If the OPC DA server has a browser interface, are the tags of the OPC DA server displayed in the OPC Item Manager?
Yes
No
In the WinCC project of the OPC DA client, open the address properties of the WinCC tag that accesses the tag of the OPC DA server.
Enable the WinCC project of the OPC DA client. Start the WinCC "Channel Diagnosis" from the Start menu. Enable the trace function.
In the "Address Properties" dialog, are the entries in the fields "Item Name" and "Data Type" correct?
Yes
Are the DCOM settings correct?
Yes
Contact the WinCC Customer Support.
No
No
Correct the entries.
Change the configuration on the WinCC DA OPC client.
For additional information, refer to the documentation of the operating system.
Check whether you can establish the communication connection now.
Yes
You can now use the OPC connection for data transmissi-on.
No
Check the OPC DA server.

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


WinCC is used as the OPC DA client. Unable to establish a connection.
Go to the "Properties Connection" dialog on the OPC DA client and check the entries in the"OPC Connection" tab.
Are the ProgID and the computer name correct?
Yes
No
Correct the entries.
Is the computer available on the network?
No
Make sure the computer is available on the network.
Yes
Are the DCOM settings correct?
Yes
Contact the WinCC Customer Support.
No
Change the configuration on the WinCC DA OPC client. You can find further information in the documentation of the operating system.
Check if the correct value is now displayed.
Yes
You can now use the OPC connection for data communica-tion.
No
Check the OPC DA server.

7.6 OPC UA WinCC Channel

7.6.1 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.

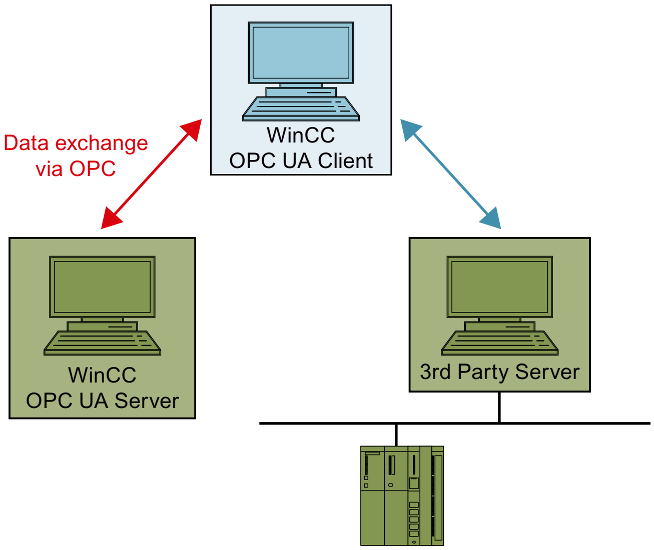
OPC Unified Architecture (OPC UA) provides, for example, additional mechanisms such as authentication and encryption to secure communication between involved partners.

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.



7.6.2 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

7.6.3 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 292)

7.6.4 Configuring the OPC UA channel

7.6.4.1 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 295)
- Creating a connection to the OPC UA server (Page 296)
- Setting up authentication via certificates. (Page 303)
- Configuring the OPC UA tags (Page 314)
- Using OPC UA alarms in WinCC (Page 326)

7.6.4.2 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.

7.6.4.3 Creating a connection to the OPC UA server

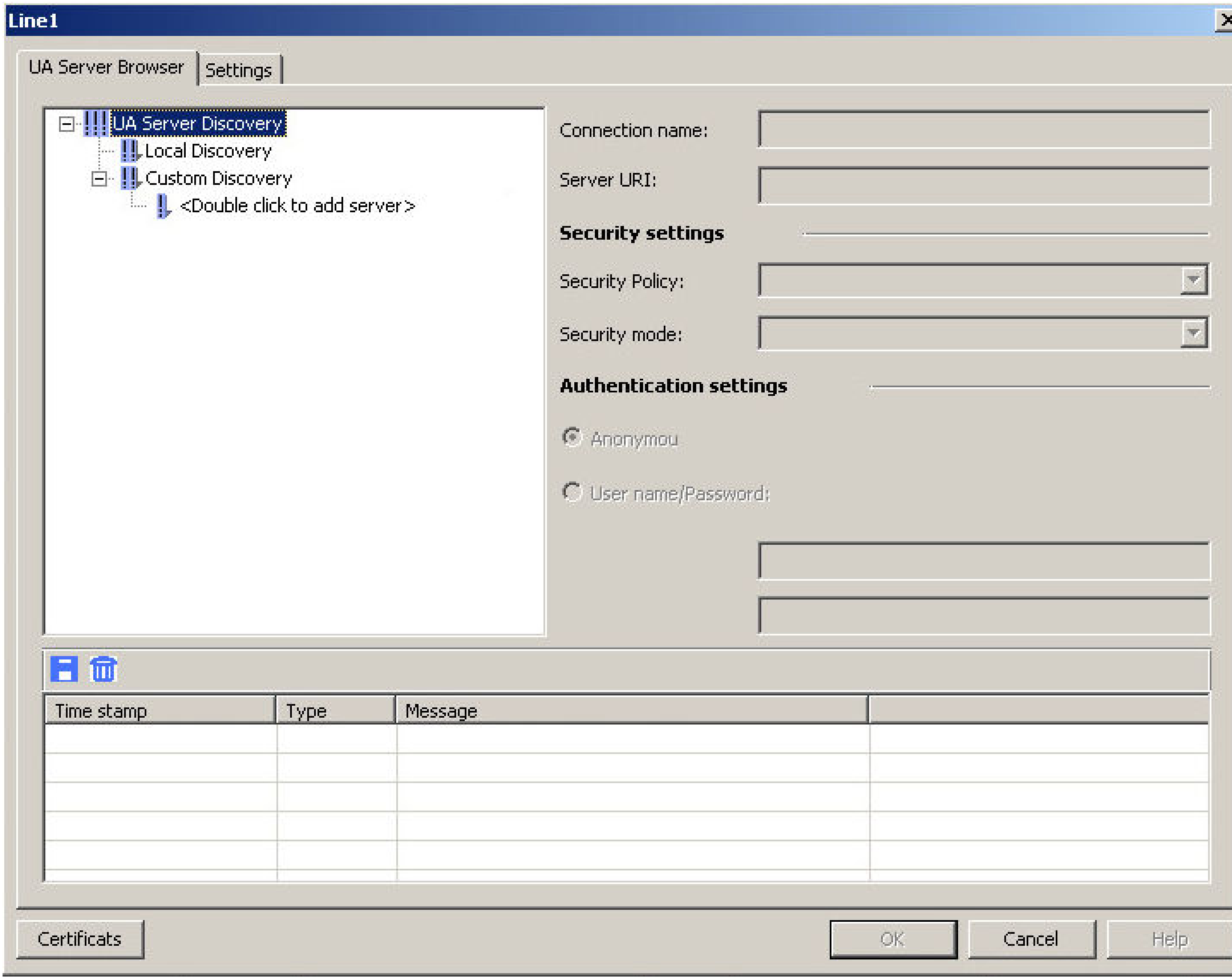
To connect to an OPC UA server, you need information about the server and the security settings.

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.

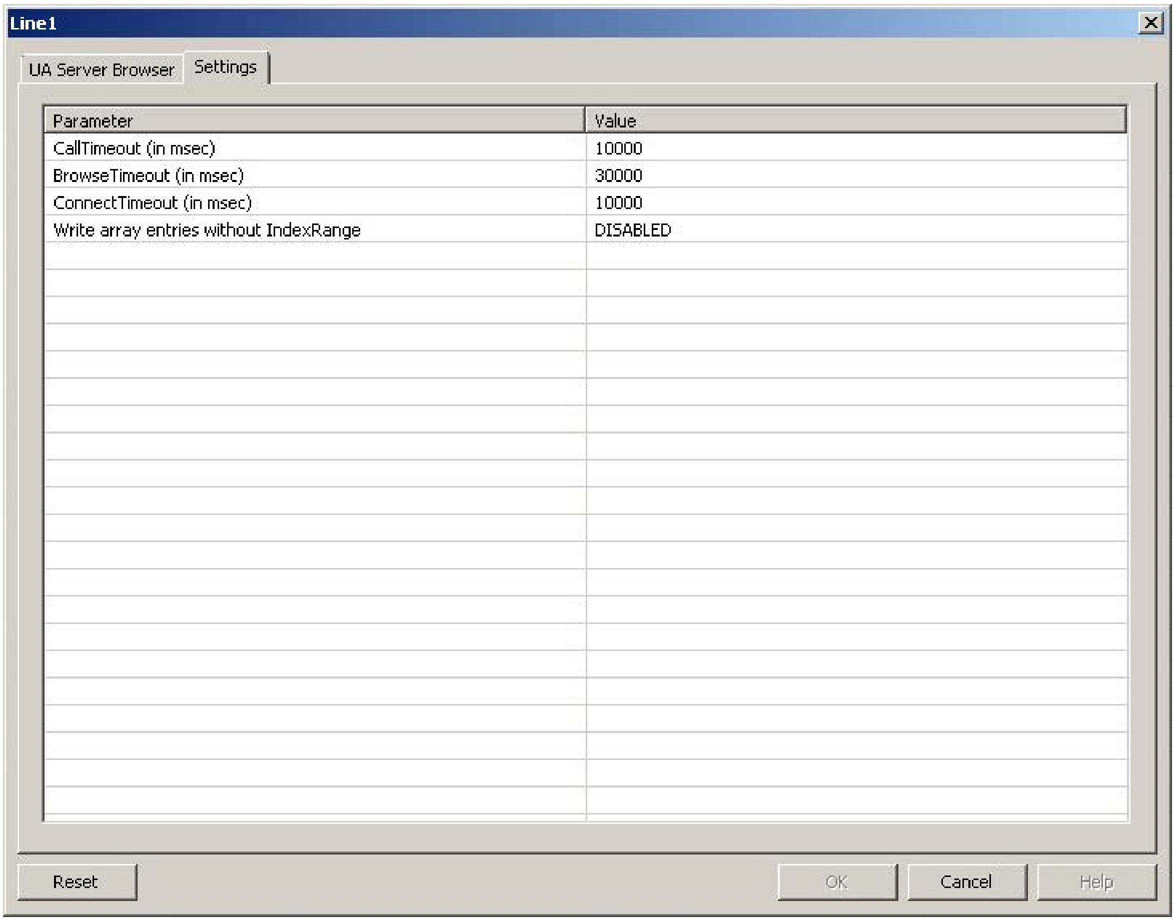


Field / setting	Contents
UA Server Discovery	The UA Discovery Server provides a list of available OPC UA servers.
Local Discovery	Local Discovery Lists all OPC UA Servers on the local computer, which are registered for the Local Discovery Server (LDS). Requirement: The LDS is installed on the local computer.

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.

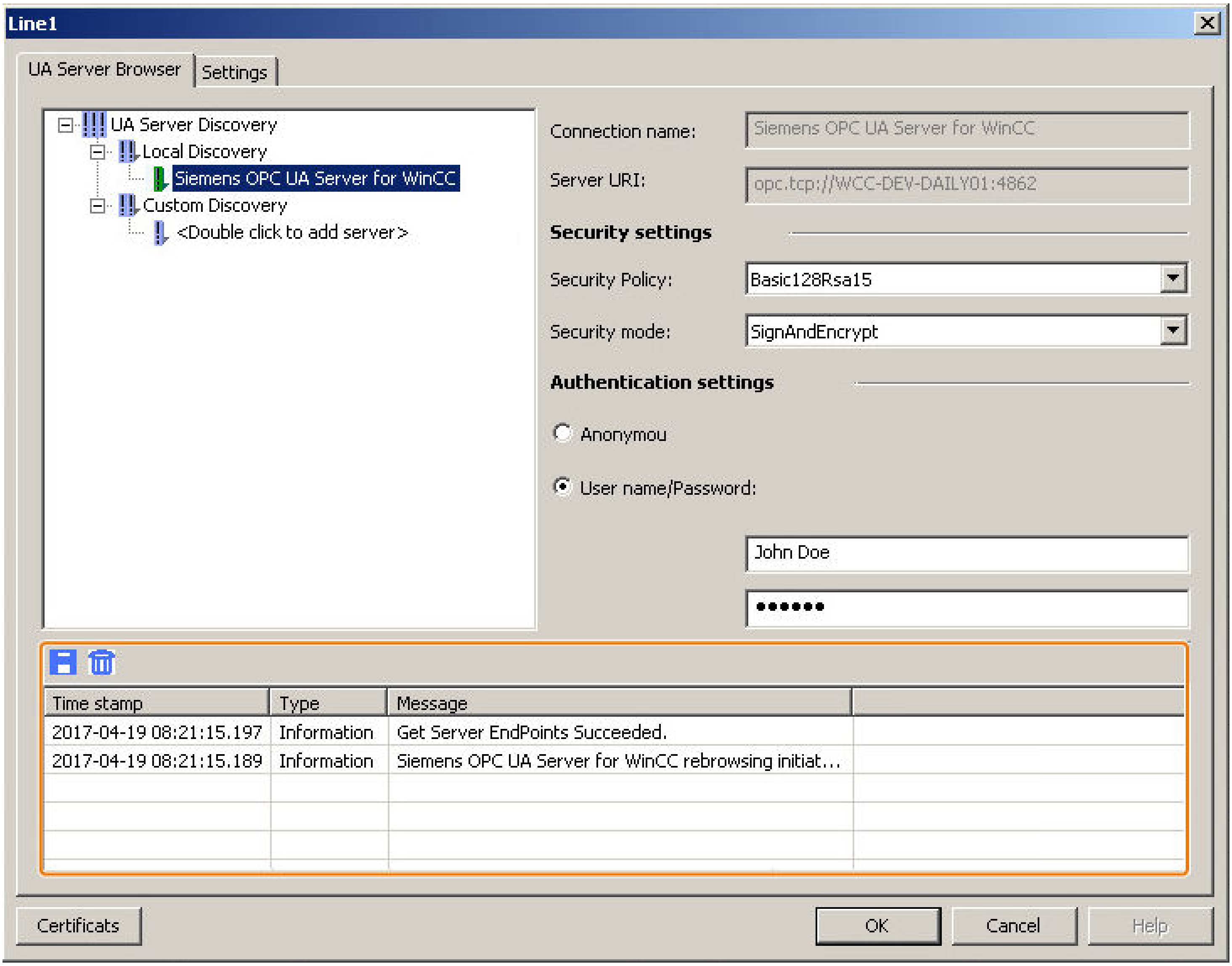


Parameter	Value
CallTimeout (in msec)	10000
BrowseTimeout (in msec)	30000
ConnectTimeout (in msec)	10000
Write array entries without IndexRange	DISABLED

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 Index-Range	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.



Connection name: Siemens OPC UA Server for WinCC

Server URI: opc.tcp://WCC-DEV-DAILY01:4862

Security settings

Security Policy: Basic128Rsa15

Security mode: SignAndEncrypt

Authentication settings

○ Anonymou

● User name/Password:

John Doe

●●●●●●

Time stamp	Type	Message
2017-04-19 08:21:15.197	Information	Get Server EndPoints Succeeded.
2017-04-19 08:21:15.189	Information	Siemens OPC UA Server for WinCC rebrowsing initiat...

Deleting messages

You delete all entries with the following button: 🗑

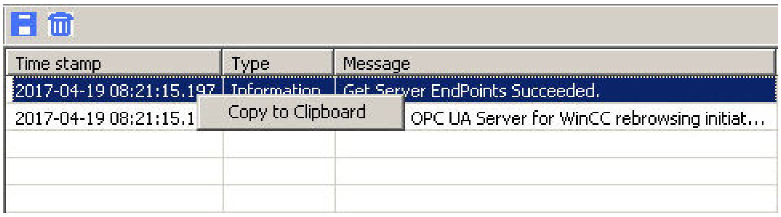
NOTICE
Delete: "Undo" not possible
Deleted entries cannot be restored.

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.



Time stamp	Type	Message
2017-04-19 08:21:15.197	Information	Get Server EndPoints Succeeded.
2017-04-19 08:21:15.1		OPC UA Server for WinCC rebrowsing initiat...

See also

How to configure a connection to the OPC UA server (Page 311)

Creating a connection to the OPC UA server via a NAT router (Page 301)

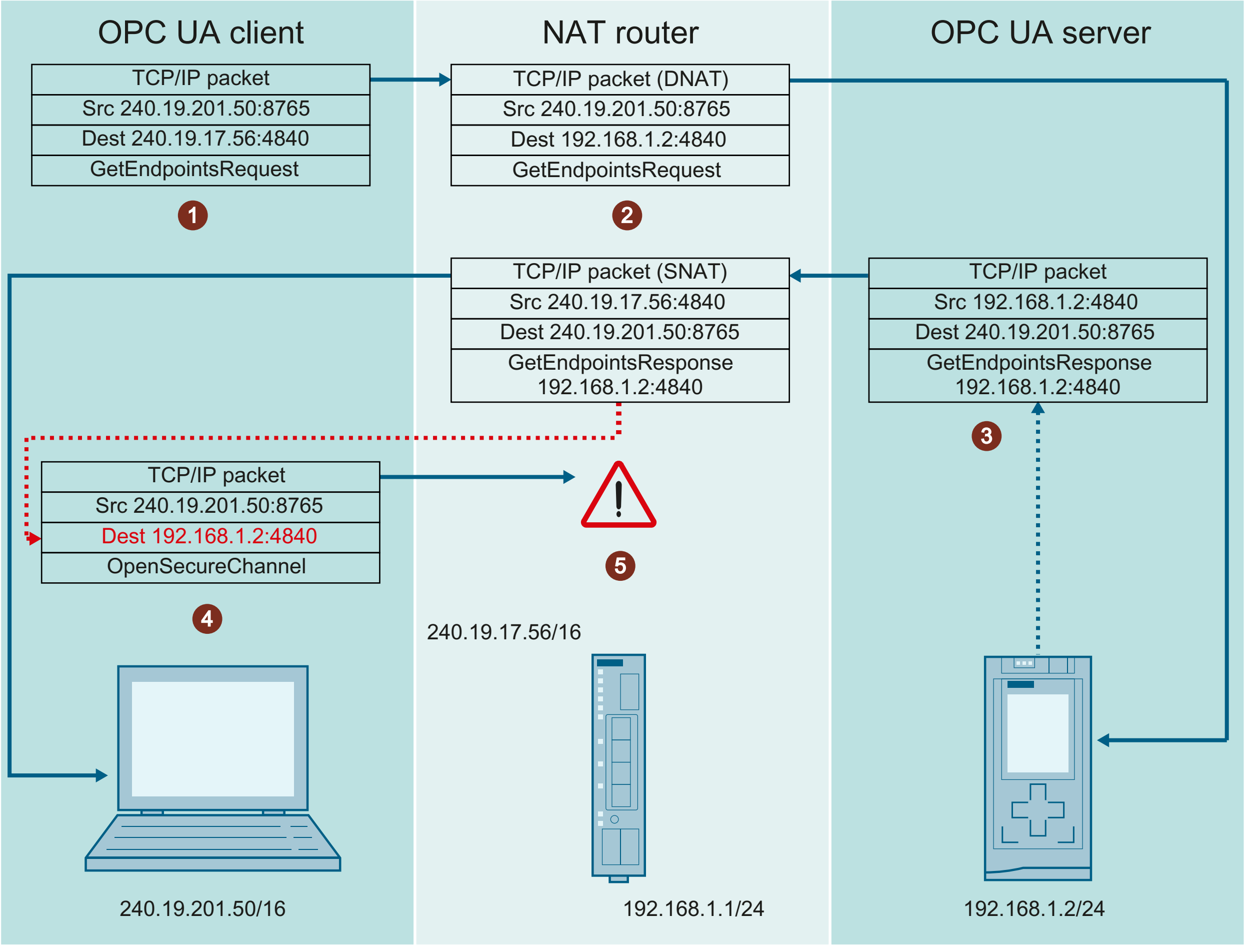
7.6.4.4 Creating a connection to the OPC UA server via a NAT router

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 303)".

See also

Creating a connection to the OPC UA server (Page 296)

7.6.4.5 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 (Page 748)
- Communication > Communication Diagnostics > Diagnostics Channel "OPC" (Page 658)

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:

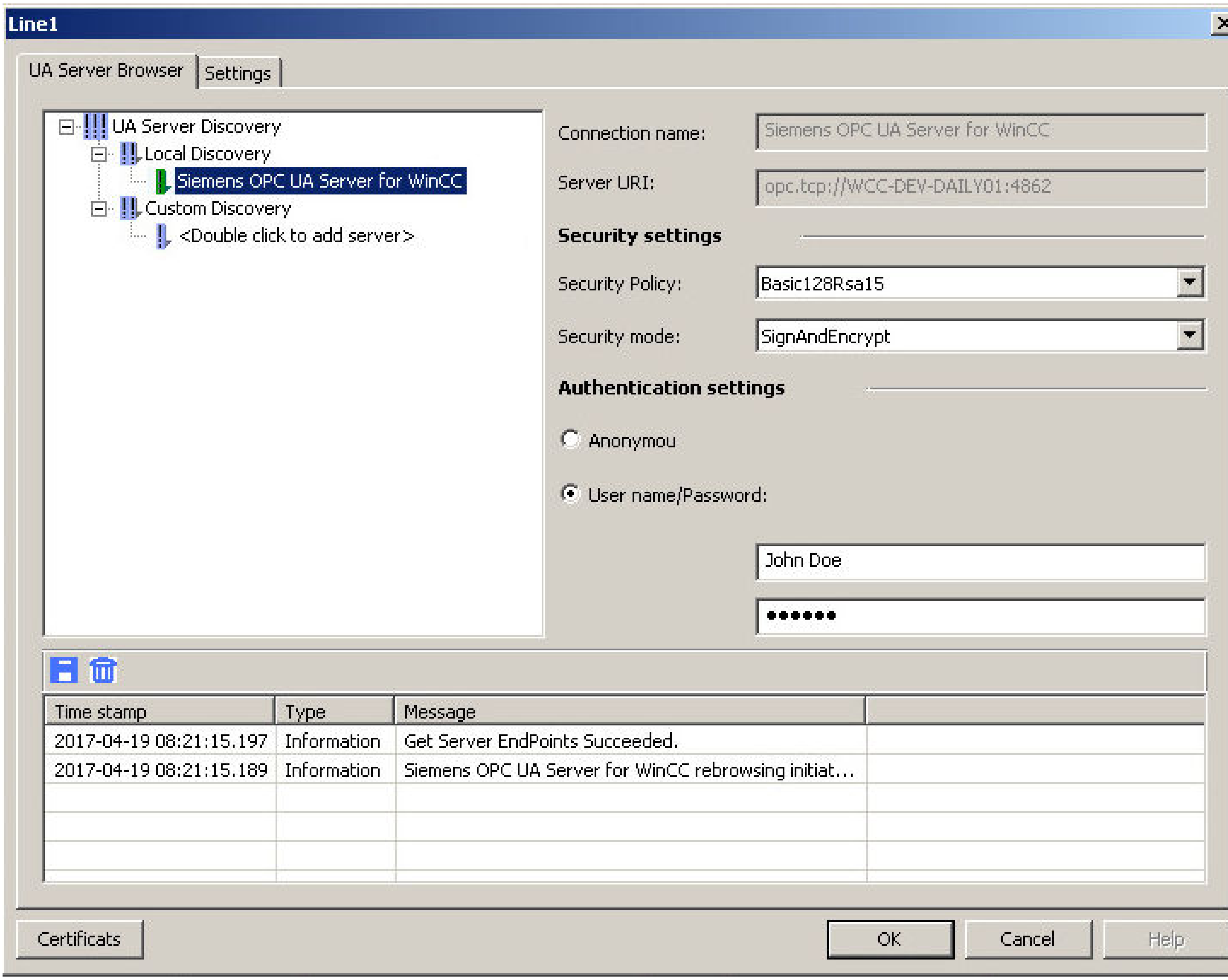
- Discovery of UA Server failed - The Certificate is not trusted.

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.



Connection name: Siemens OPC UA Server for WinCC

Server URI: opc.tcp://WCC-DEV-DAILY01:4862

Security settings

Security Policy: Basic128Rsa15

Security mode: SignAndEncrypt

Authentication settings

○ Anonymou

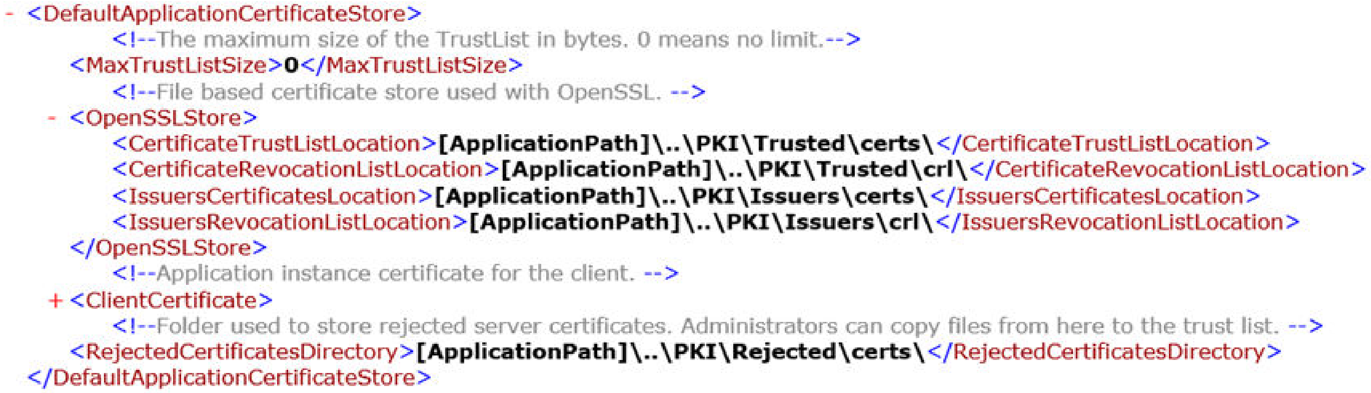
● User name/Password:

John Doe

••••••

Time stamp	Type	Message	
2017-04-19 08:21:15.197	Information	Get Server EndPoints Succeeded.	
2017-04-19 08:21:15.189	Information	Siemens OPC UA Server for WinCC rebrowsing initiat...	

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.



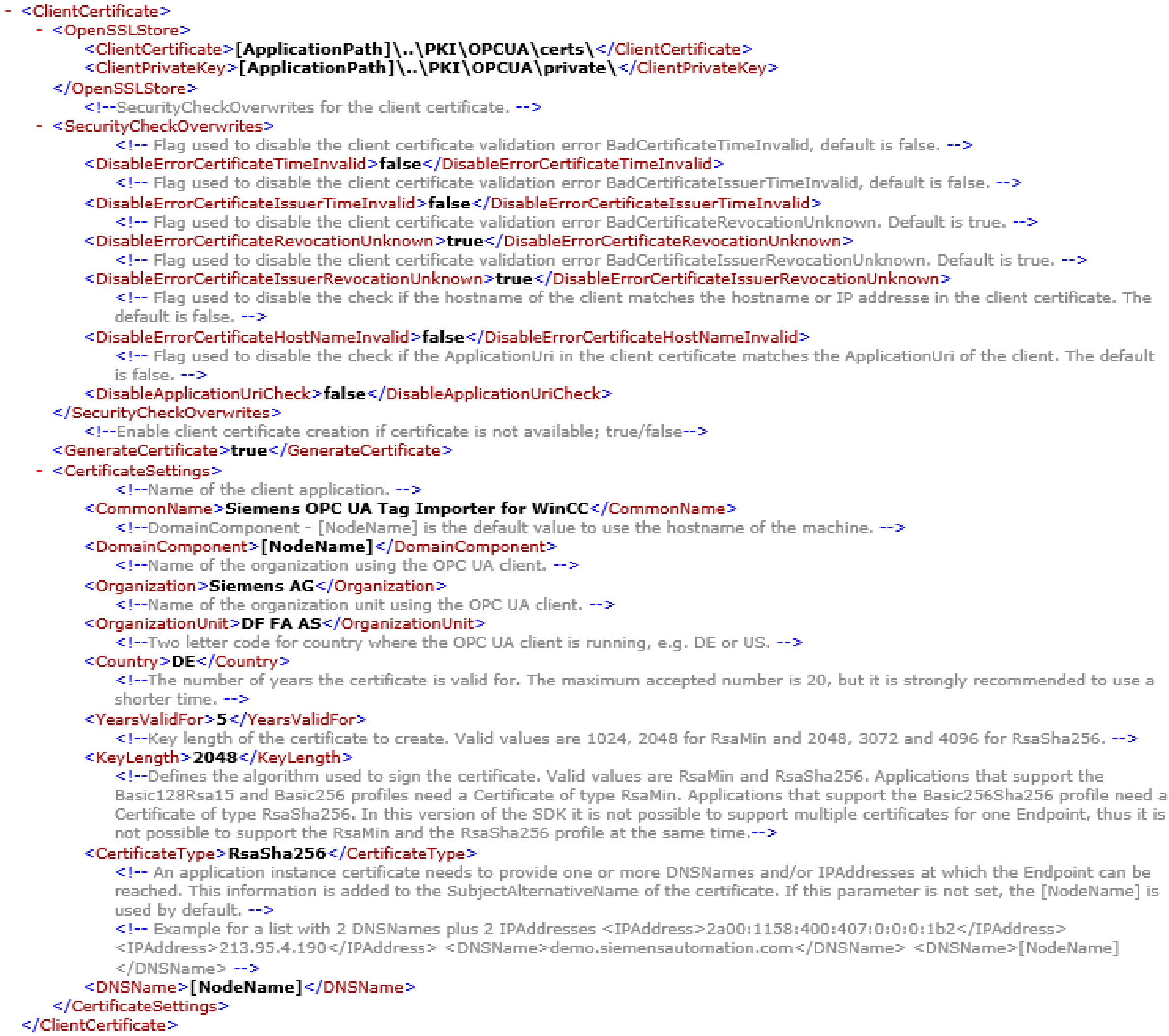
```
- <DefaultApplicationCertificateStore>
        <!--The maximum size of the TrustList in bytes. 0 means no limit.-->
    <MaxTrustListSize>0</MaxTrustListSize>
        <!--File based certificate store used with OpenSSL. -->
  - <OpenSSLStore>
        <CertificateTrustListLocation>[ApplicationPath]\..\PKI\Trusted\certs\</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 DomainComponent OrganizationUnit Organization Country	Descriptive elements The parameters can be changed and have no effect on the function of the applications.
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.

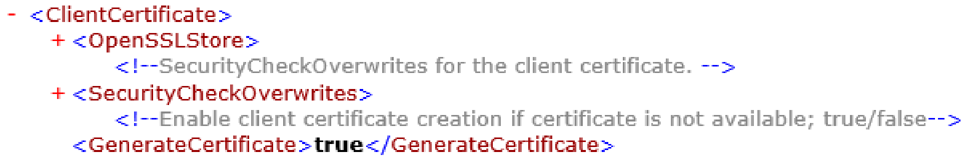
1. Create a backup.
2. Delete the existing certificates and the associated private keys in the corresponding folders.
3. In the configuration files, update the certificate parameters and save the XML files.
4. Open the DOS window "cmd.exe" in Microsoft Windows with administrator rights.
5. To create 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.

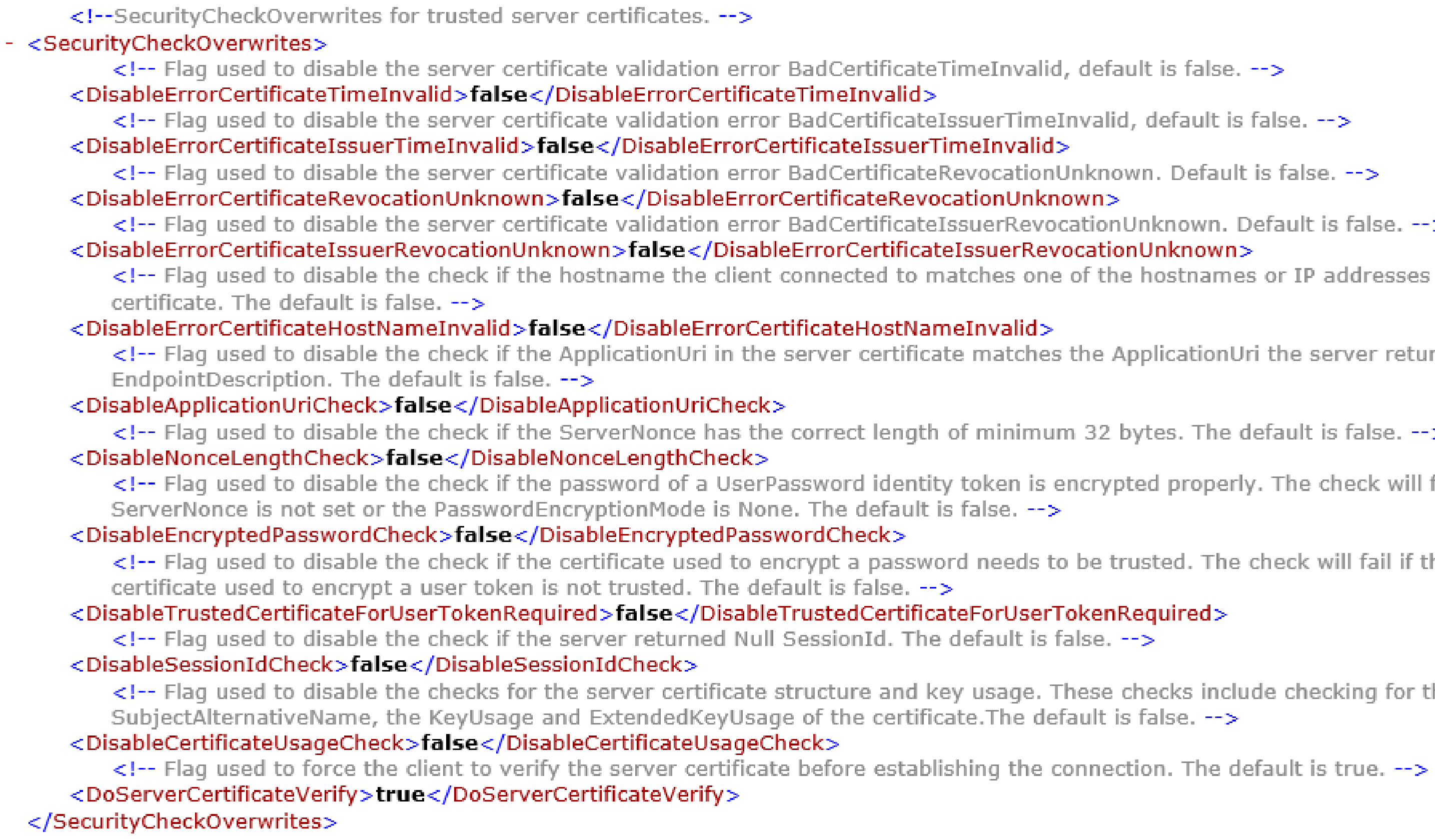
To that end, adjust "SecurityCheckOverwrites" in the section "UaClient" to match.

"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
      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 retur
      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 f
      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 th
      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 tl
      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

Security concept of OPC UA (Page 748)

How to configure a connection to the OPC UA server (Page 311)

Creating a connection to the OPC UA server via a NAT router (Page 301)

Diagnosis of the "OPC" Channel (Page 658)

7.6.4.6 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

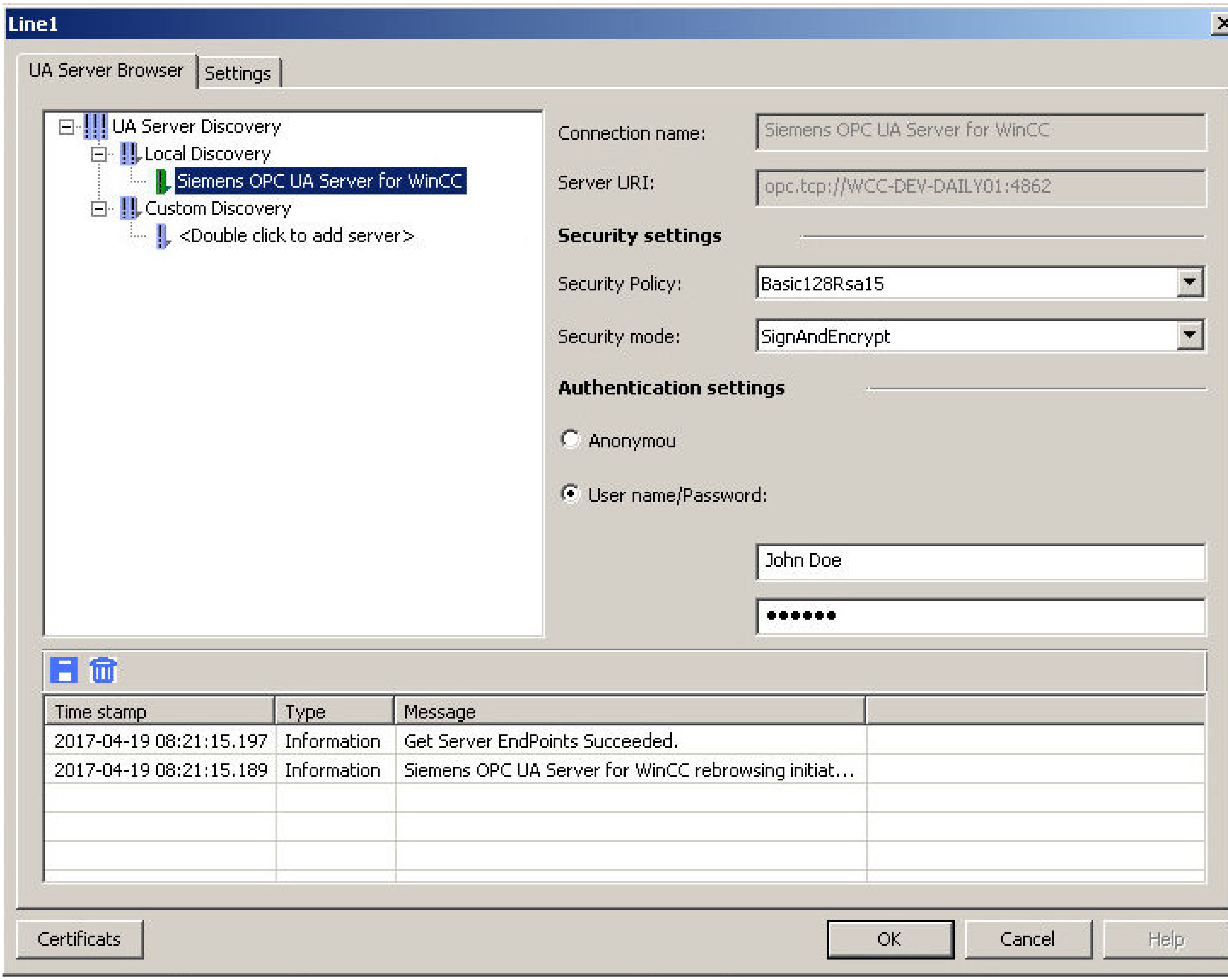
1. Open WinCC Tag Management in the WinCC Configuration Studio.
2. Create a new connection using the shortcut menu of "OPC UA Connections".
3. Name the connection.
   Keep the following restrictions in mind when naming the connection:
   – The first character must be a letter.
   – Only letters contained 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.



Connection name: Siemens OPC UA Server for WinCC

Server URI: opc.tcp://WCC-DEV-DAILY01:4862

Security settings

Security Policy: Basic128Rsa15

Security mode: SignAndEncrypt

Authentication settings

○ Anonymou

● User name/Password:

John Doe

●●●●●●

Time stamp	Type	Message	
2017-04-19 08:21:15.197	Information	Get Server EndPoints Succeeded.	
2017-04-19 08:21:15.189	Information	Siemens OPC UA Server for WinCC rebrowsing initiat...	

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 317)

OPC UA tags (Page 314)

Setting up authentication via certificates. (Page 303)

7.6.4.7 Configuring the OPC UA tags

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 318)".

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

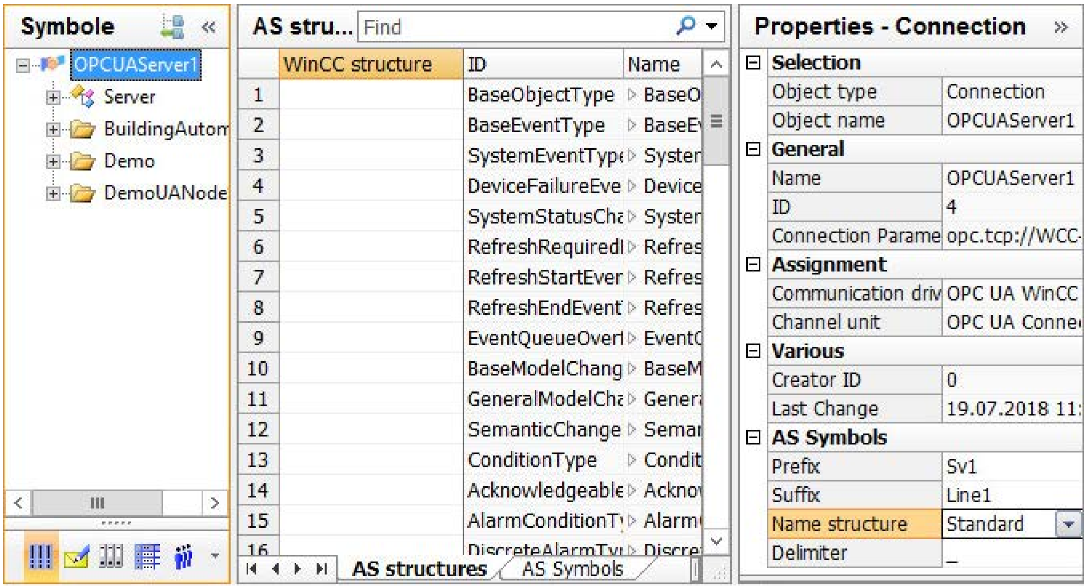
To display the settings in the "AS symbols" property group, click the connection name in the "Symbols" view.

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.



Symbole

- OPCUAServer1
  - Server
  - BuildingAutom
  - Demo
  - DemoUANode

AS stru... Find

	WinCC structure	ID	Name
1		BaseObjectType	BaseO
2		BaseEventType	BaseE
3		SystemEventType	Syster
4		DeviceFailureEve	Device
5		SystemStatusCha	Syster
6		RefreshRequiredI	Refres
7		RefreshStartEver	Refres
8		RefreshEndEvent	Refres
9		EventQueueOverl	EventC
10		BaseModelChang	BaseM
11		GeneralModelCha	Genera
12		SemanticChange	Semar
13		ConditionType	Condit
14		Acknowledgeable	Acknov
15		AlarmConditionTy	Alarm
16		DiscreteAlarmTy	Discre

AS structures | AS Symbols

Properties - Connection

Selection	
Object type	Connection
Object name	OPCUAServer1
General	
Name	OPCUAServer1
ID	4
Connection Parame	opc.tcp://WCC-
Assignment	
Communication driv	OPC UA WinCC
Channel unit	OPC UA Conne
Various	
Creator ID	0
Last Change	19.07.2018 11:
AS Symbols	
Prefix	Sv1
Suffix	Line1
Name structure	Standard
Delimiter	_

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 317)

To import an OPC UA node as a WinCC tag (Page 318)

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.

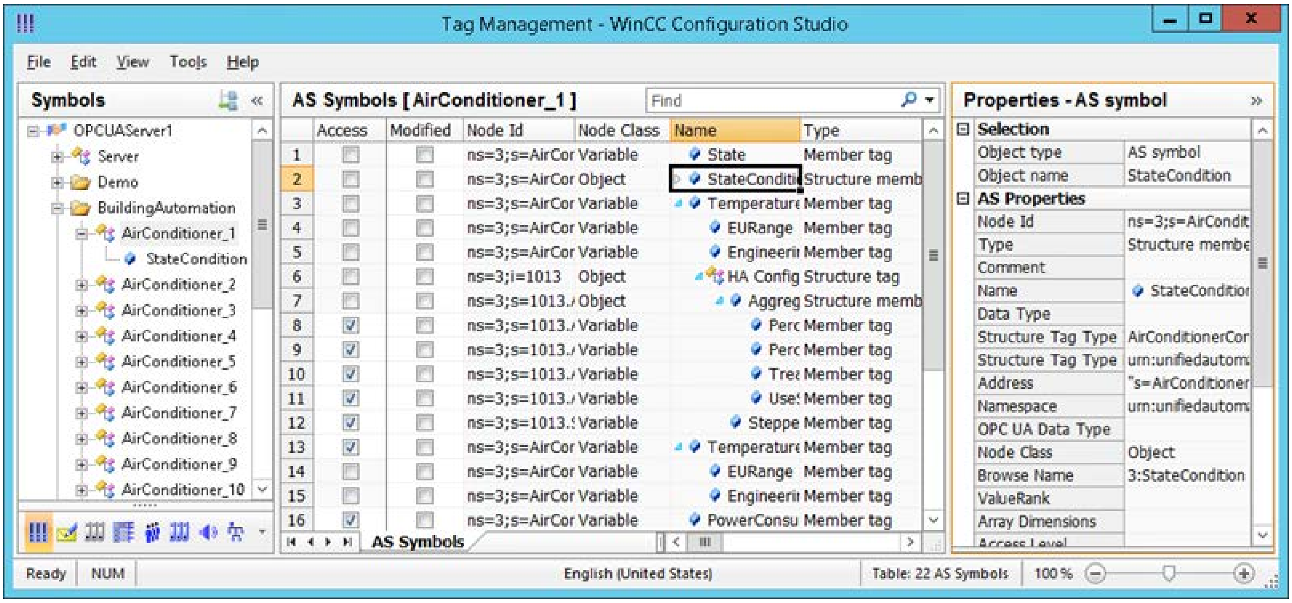
Complex data structures are displayed in the data area.

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 Symbols [AirConditioner_1]

	Access	Modified	Node Id	Node Class	Name	Type
1	□	□	ns=3;s=AirCor	Variable	State	Member tag
2	□	□	ns=3;s=AirCor	Object	▷ StateConditi	Structure memb
3	□	□	ns=3;s=AirCor	Variable	Temperature	Member tag
4	□	□	ns=3;s=AirCor	Variable	EURange	Member tag
5	□	□	ns=3;s=AirCor	Variable	Engineerii	Member tag
6	□	□	ns=3;i=1013	Object	HA Config	Structure tag
7	□	□	ns=3;s=1013./	Object	Aggreg	Structure memb
8	☑	□	ns=3;s=1013./	Variable	Perc	Member tag
9	☑	□	ns=3;s=1013./	Variable	Perc	Member tag
10	☑	□	ns=3;s=1013./	Variable	Trea	Member tag
11	☑	□	ns=3;s=1013./	Variable	Use:	Member tag
12	☑	□	ns=3;s=1013.:	Variable	Steppe	Member tag
13	☑	□	ns=3;s=AirCor	Variable	Temperature	Member tag
14	□	□	ns=3;s=AirCor	Variable	EURange	Member tag
15	□	□	ns=3;s=AirCor	Variable	Engineerii	Member tag
16	☑	□	ns=3;s=AirCor	Variable	PowerConsu	Member tag

Properties - AS symbol

Selection	
Object type	AS symbol
Object name	StateCondition
AS Properties	
Node Id	ns=3;s=AirCondit
Type	Structure membe
Comment	
Name	StateConditior
Data Type	
Structure Tag Type	AirConditionerCor
Structure Tag Type	urn:unifiedautom
Address	"s=AirConditioner
Namespace	urn:unifiedautom
OPC UA Data Type	
Node Class	Object
Browse Name	3:StateCondition
ValueRank	
Array Dimensions	
Access Level	

"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 322)

To import an OPC UA node as a WinCC tag (Page 318)

How to configure a connection to the OPC UA server (Page 311)

OPC UA tags (Page 314)

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 314)".

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 322)

The "Symbols" view (Page 317)

7.6.4.8 Using OPC UA types in WinCC

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

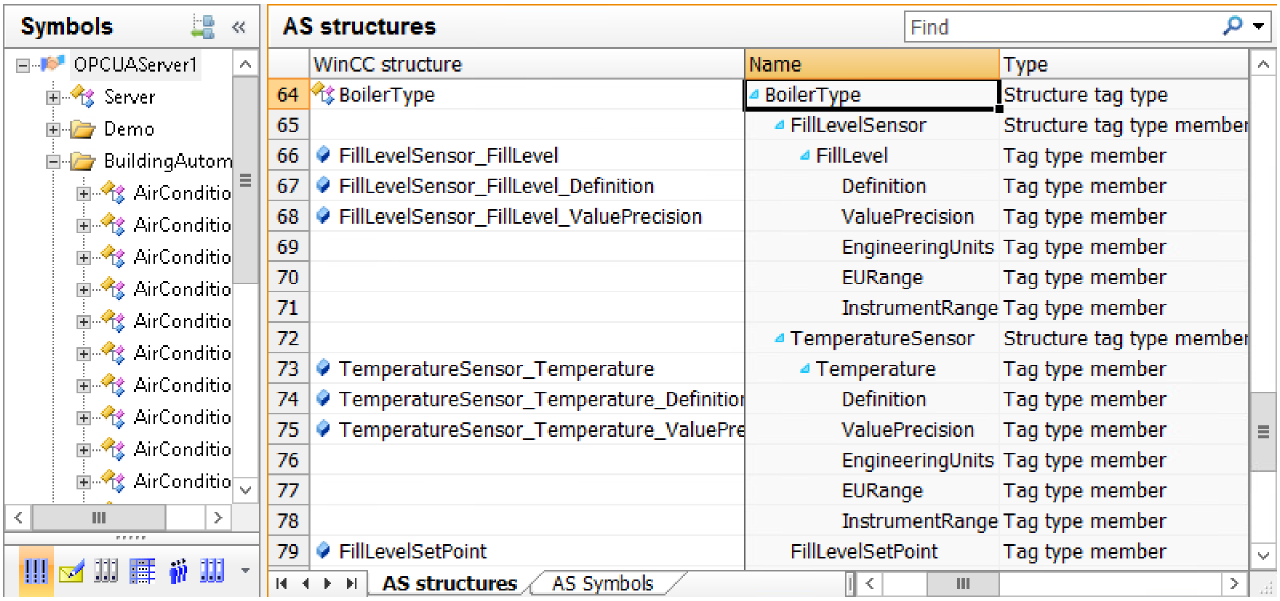
Configuration step 1: Assigning object types

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.



	WinCC structure	Name	Type
64	BoilerType	BoilerType	Structure tag type
65		FillLevelSensor	Structure tag type member
66	FillLevelSensor_FillLevel	FillLevel	Tag type member
67	FillLevelSensor_FillLevel_Definition	Definition	Tag type member
68	FillLevelSensor_FillLevel_ValuePrecision	ValuePrecision	Tag type member
69		EngineeringUnits	Tag type member
70		EURange	Tag type member
71		InstrumentRange	Tag type member
72		TemperatureSensor	Structure tag type member
73	TemperatureSensor_Temperature	Temperature	Tag type member
74	TemperatureSensor_Temperature_Definitior	Definition	Tag type member
75	TemperatureSensor_Temperature_ValuePre	ValuePrecision	Tag type member
76		EngineeringUnits	Tag type member
77		EURange	Tag type member
78		InstrumentRange	Tag type member
79	FillLevelSetPoint	FillLevelSetPoint	Tag type member

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.

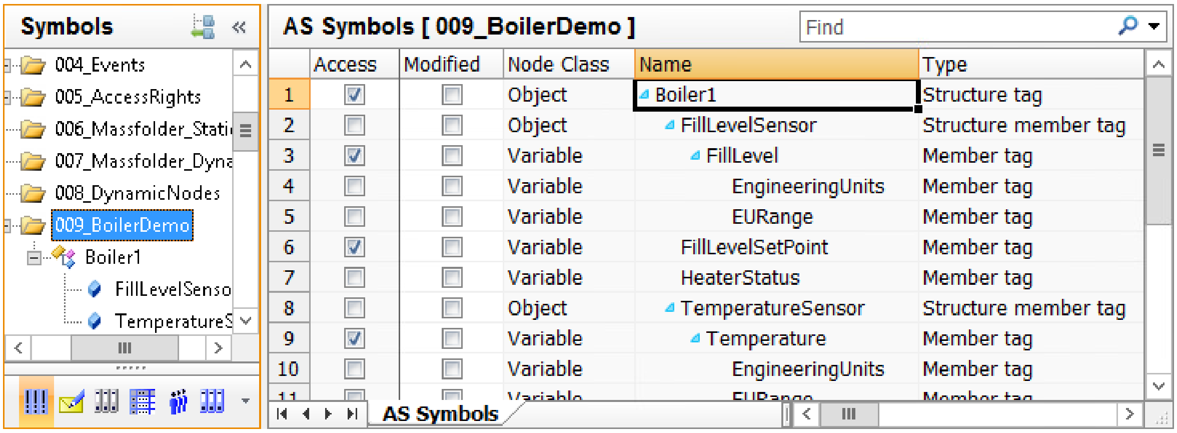
If the structure type elements and properties or tags have the same name and data type, they are assigned automatically.

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.



	Access	Modified	Node Class	Name	Type
1	☑	□	Object	Boiler1	Structure tag
2	□	□	Object	FillLevelSensor	Structure member tag
3	☑	□	Variable	FillLevel	Member tag
4	□	□	Variable	EngineeringUnits	Member tag
5	□	□	Variable	EURange	Member tag
6	☑	□	Variable	FillLevelSetPoint	Member tag
7	□	□	Variable	HeaterStatus	Member tag
8	□	□	Object	TemperatureSensor	Structure member tag
9	☑	□	Variable	Temperature	Member tag
10	□	□	Variable	EngineeringUnits	Member tag
11	□	□	Variable	EURange	Member tag

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.

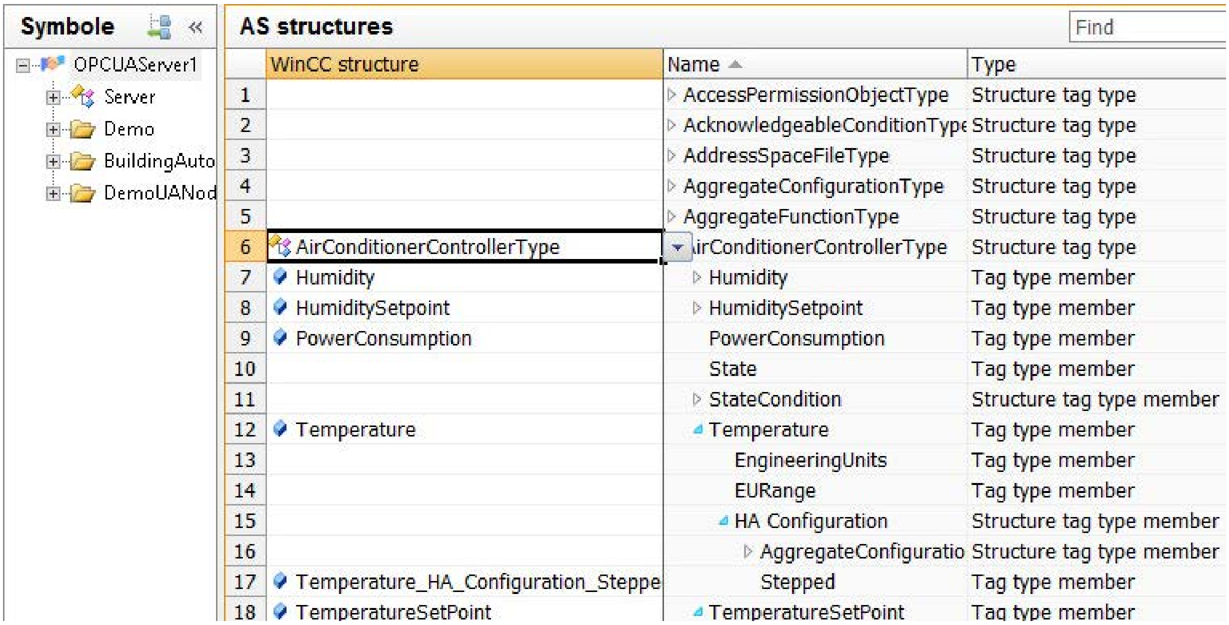
Requirements

- 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.

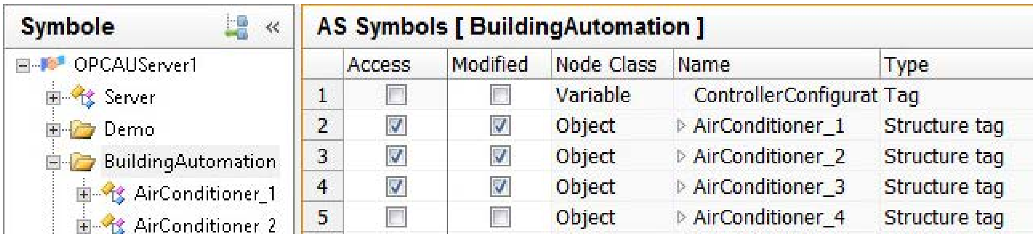
3. Select the "Create structure" entry in the shortcut menu of the row.
   - A structure type with the name of the OPC UA object type is created in WinCC Tag Management.
   - One structure type element each is created for all properties and tags of the OPC UA object type that can be mapped.
   - The 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.



	WinCC structure	Name	Type
1		AccessPermissionObjectType	Structure tag type
2		AcknowledgeableConditionType	Structure tag type
3		AddressSpaceFileType	Structure tag type
4		AggregateConfigurationType	Structure tag type
5		AggregateFunctionType	Structure tag type
6	AirConditionerControllerType	ırConditionerControllerType	Structure tag type
7	Humidity	Humidity	Tag type member
8	HumiditySetpoint	HumiditySetpoint	Tag type member
9	PowerConsumption	PowerConsumption	Tag type member
10		State	Tag type member
11		StateCondition	Structure tag type member
12	Temperature	Temperature	Tag type member
13		EngineeringUnits	Tag type member
14		EURange	Tag type member
15		HA Configuration	Structure tag type member
16		AggregateConfiguratio	Structure tag type member
17	Temperature_HA_Configuration_Steppe	Stepped	Tag type member
18	TemperatureSetPoint	TemperatureSetPoint	Tag type member

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.



AS Symbols [BuildingAutomation]

	Access	Modified	Node Class	Name	Type
1	☐	☐	Variable	ControllerConfigurat	Tag
2	☑	☑	Object	▷ AirConditioner_1	Structure tag
3	☑	☑	Object	▷ AirConditioner_2	Structure tag
4	☑	☑	Object	▷ AirConditioner_3	Structure tag
5	☐	☐	Object	▷ AirConditioner_4	Structure tag

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 318)

The "Symbols" view (Page 317)

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.

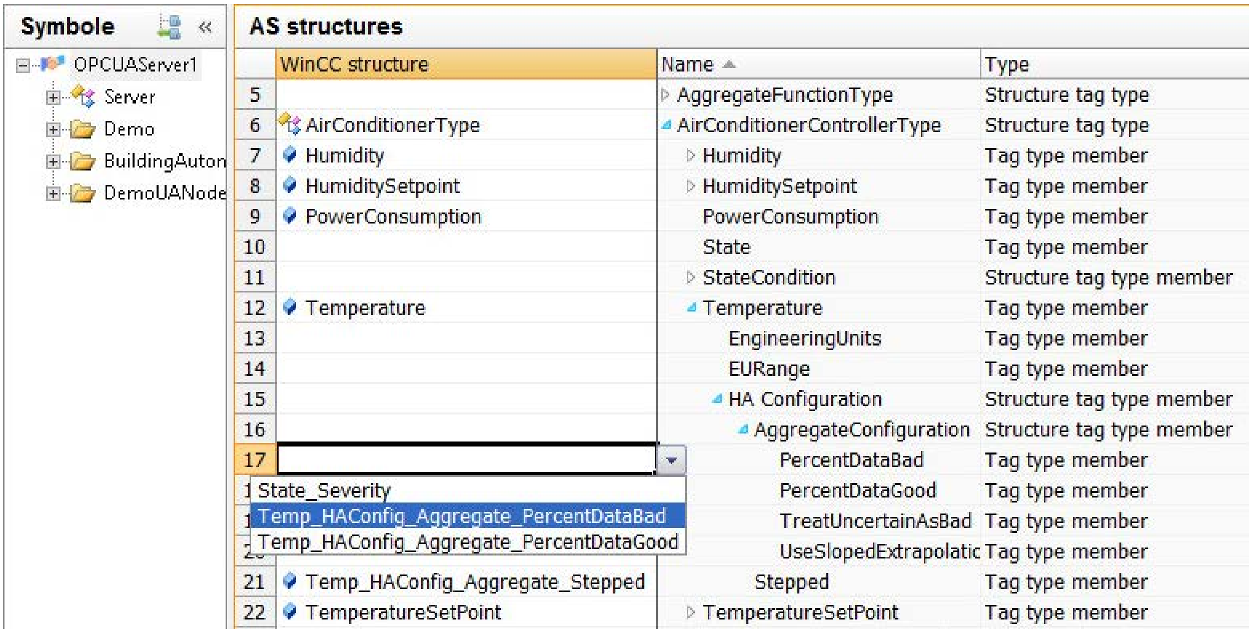
Requirements

- 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.



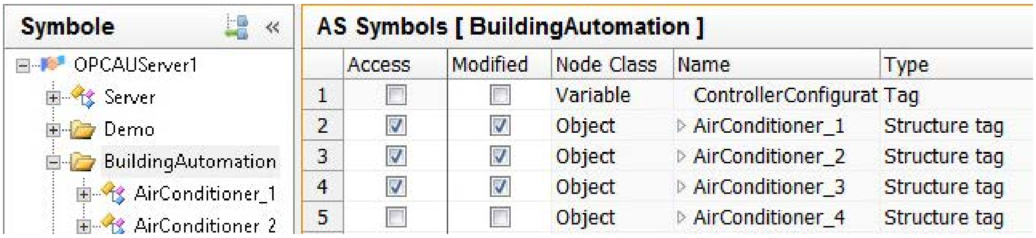
Symbole

- OPCUAServer1
  - Server
  - Demo
  - BuildingAuton
  - DemoUANode

AS structures

	WinCC structure	Name	Type
5		AggregateFunctionType	Structure tag type
6	AirConditionerType	AirConditionerControllerType	Structure tag type
7	Humidity	Humidity	Tag type member
8	HumiditySetpoint	HumiditySetpoint	Tag type member
9	PowerConsumption	PowerConsumption	Tag type member
10		State	Tag type member
11		StateCondition	Structure tag type member
12	Temperature	Temperature	Tag type member
13		EngineeringUnits	Tag type member
14		EURange	Tag type member
15		HA Configuration	Structure tag type member
16		AggregateConfiguration	Structure tag type member
17		PercentDataBad	Tag type member
1	State_Severity	PercentDataGood	Tag type member
1	Temp_HAConfig_Aggregate_PercentDataBad	TreatUncertainAsBad	Tag type member
2	Temp_HAConfig_Aggregate_PercentDataGood	UseSlopedExtrapolatic	Tag type member
21	Temp_HAConfig_Aggregate_Stepped	Stepped	Tag type member
22	TemperatureSetPoint	TemperatureSetPoint	Tag type member

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.



Symbole

- OPCAUServer1
  - Server
  - Demo
  - BuildingAutomation
    - AirConditioner_1
    - AirConditioner_2

AS Symbols [BuildingAutomation]

	Access	Modified	Node Class	Name	Type
1	☐	☐	Variable	ControllerConfigurat	Tag
2	☑	☑	Object	AirConditioner_1	Structure tag
3	☑	☑	Object	AirConditioner_2	Structure tag
4	☑	☑	Object	AirConditioner_3	Structure tag
5	☐	☐	Object	AirConditioner_4	Structure tag

   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.

7.6.4.9 Using OPC UA alarms in WinCC

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.

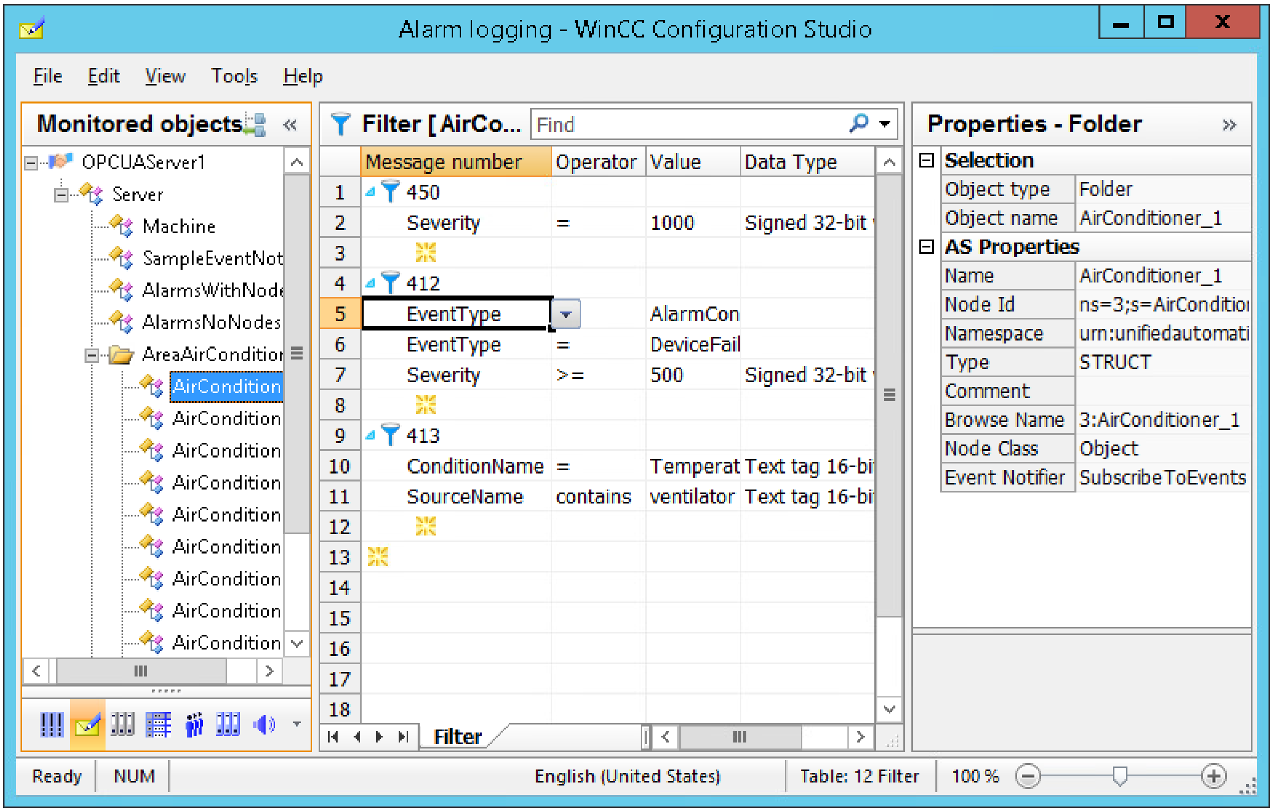
Display of the symbols

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.



	Message number	Operator	Value	Data Type
1	450			
2	Severity	=	1000	Signed 32-bit
3				
4	412			
5	EventType		AlarmCon	
6	EventType	=	DeviceFail	
7	Severity	>=	500	Signed 32-bit
8				
9	413			
10	ConditionName	=	Temperat	Text tag 16-bi
11	SourceName	contains	ventilator	Text tag 16-bi
12				
13				
14				
15				
16				
17				
18				

Selection	
Object type	Folder
Object name	AirConditioner_1
AS Properties	
Name	AirConditioner_1
Node Id	ns=3;s=AirConditio
Namespace	urn:unifiedautomati
Type	STRUCT
Comment	
Browse Name	3:AirConditioner_1
Node Class	Object
Event Notifier	SubscribeToEvents

Ready | NUM | English (United States) | Table: 12 Filter | 100 %

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 333)

Filters for the OPC UA alarms (Page 329)

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.

A WinCC message should be configured in such a way that it maps the properties of the alarms or events, e.g. acknowledgment theory and message 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 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 Capitalization must be taken into account. Operator "contains": • Contains the entered text. Placeholders are not used.

Filter criterion	Operators		Description
Severity	=	Is 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)
	!=	Is not equal to	
	>	Is greater than	
	<	Is less than	
	>=	Is greater than or equal to	
	<=	Is less than or equal to	
	be- tween	Range from, to	

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.

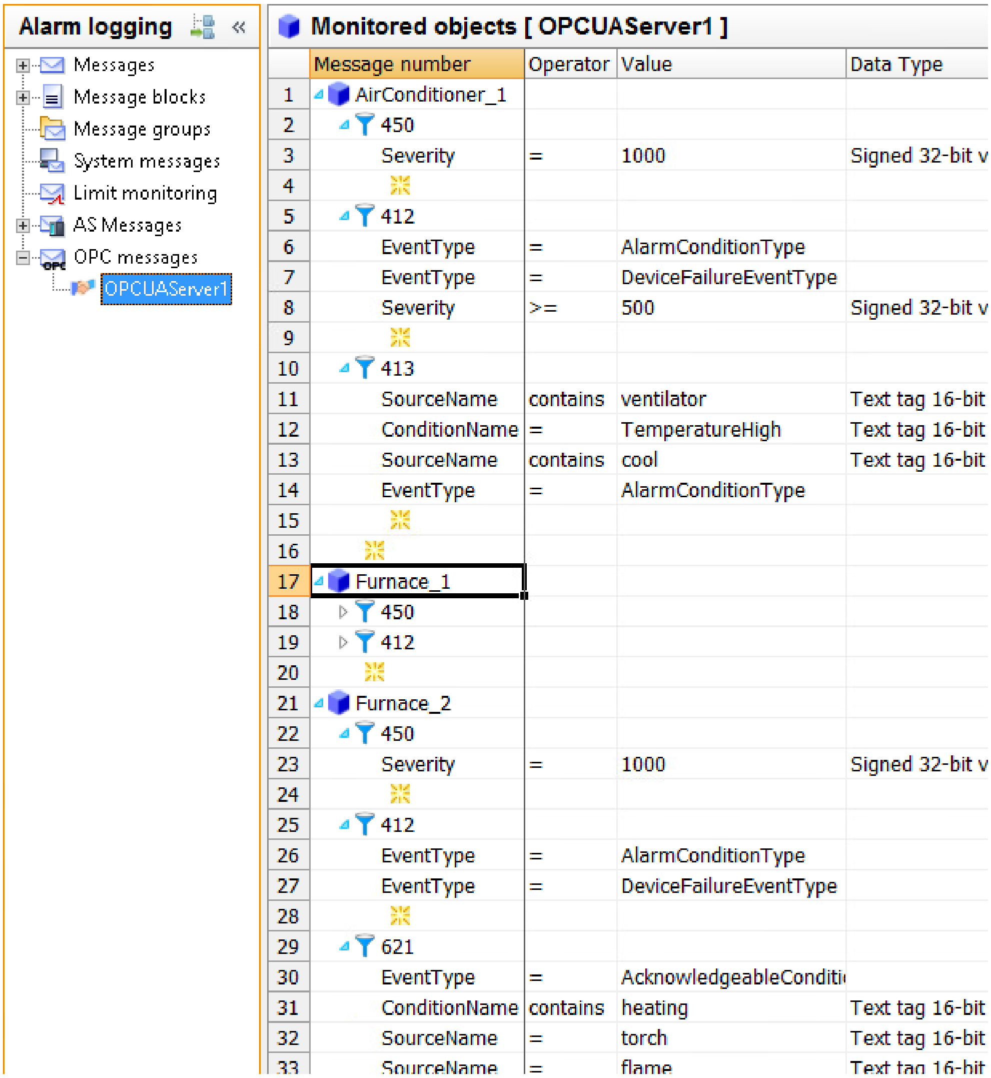
Example: "Alarm Logging" view

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:



Monitored objects [OPCUAServer1]

	Message number	Operator	Value	Data Type
1	AirConditioner_1			
2	450			
3	Severity	=	1000	Signed 32-bit v
4	✳			
5	412			
6	EventType	=	AlarmConditionType	
7	EventType	=	DeviceFailureEventType	
8	Severity	>=	500	Signed 32-bit v
9	✳			
10	413			
11	SourceName	contains	ventilator	Text tag 16-bit
12	ConditionName	=	TemperatureHigh	Text tag 16-bit
13	SourceName	contains	cool	Text tag 16-bit
14	EventType	=	AlarmConditionType	
15	✳			
16	✳			
17	Furnace_1			
18	450			
19	412			
20	✳			
21	Furnace_2			
22	450			
23	Severity	=	1000	Signed 32-bit v
24	✳			
25	412			
26	EventType	=	AlarmConditionType	
27	EventType	=	DeviceFailureEventType	
28	✳			
29	621			
30	EventType	=	AcknowledgeableConditi	
31	ConditionName	contains	heating	Text tag 16-bit
32	SourceName	=	torch	Text tag 16-bit
33	SourceName	=	flame	Text tag 16-bit

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 326)

How to import Event Notifiers as WinCC messages (Page 333)

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.

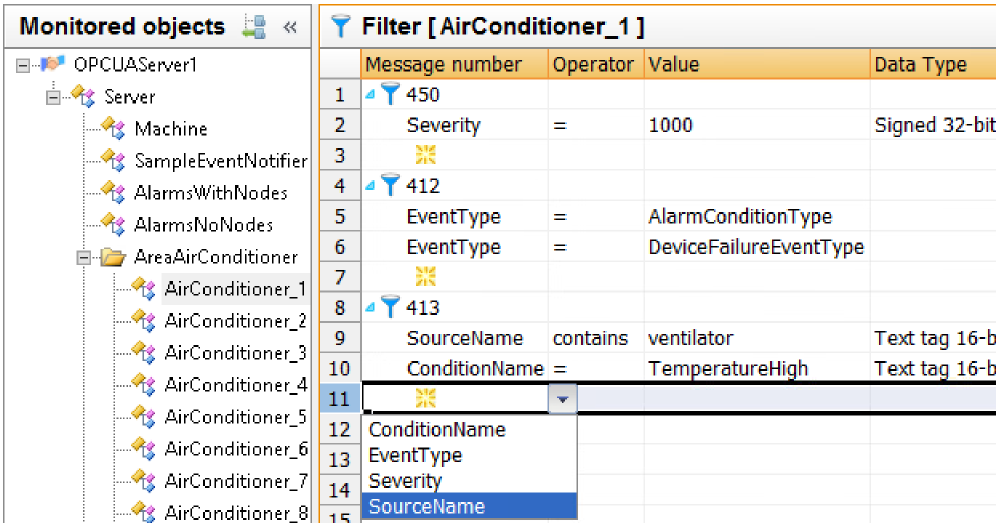
Linking of WinCC messages with Event Notifiers of a local WinCC OPC UA server can lead to a continuous loop of the Alarm Logging in the case of unfavorable configuration.

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.



Monitored objects

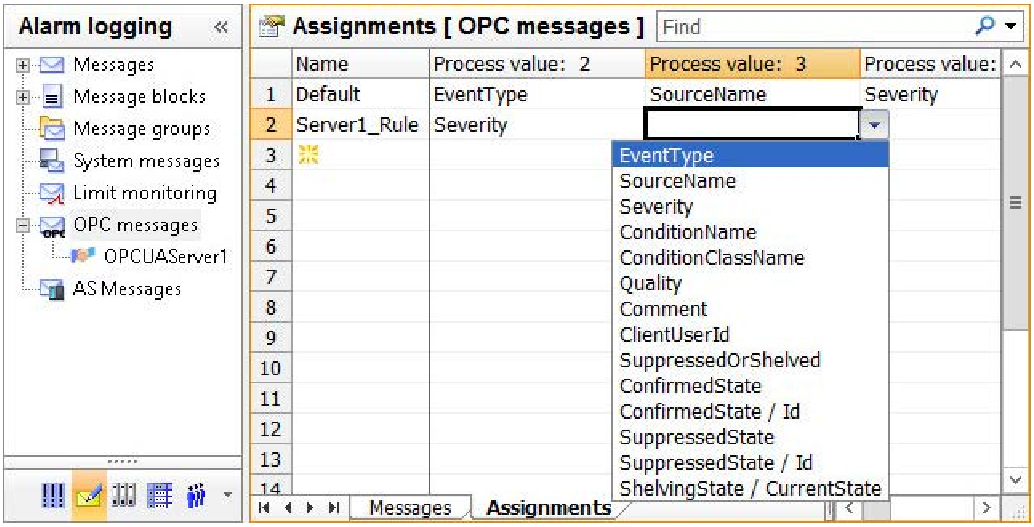
- OPCUAServer1
  - Server
    - Machine
    - SampleEventNotifier
    - AlarmsWithNodes
    - AlarmsNoNodes
    - AreaAirConditioner
      - AirConditioner_1
      - AirConditioner_2
      - AirConditioner_3
      - AirConditioner_4
      - AirConditioner_5
      - AirConditioner_6
      - AirConditioner_7
      - AirConditioner_8

Filter [AirConditioner_1]

	Message number	Operator	Value	Data Type
1	450			
2	Severity	=	1000	Signed 32-bit
3				
4	412			
5	EventType	=	AlarmConditionType	
6	EventType	=	DeviceFailureEventType	
7				
8	413			
9	SourceName	contains	ventilator	Text tag 16-b
10	ConditionName	=	TemperatureHigh	Text tag 16-b
11				
12				
13				
14				
15				

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.

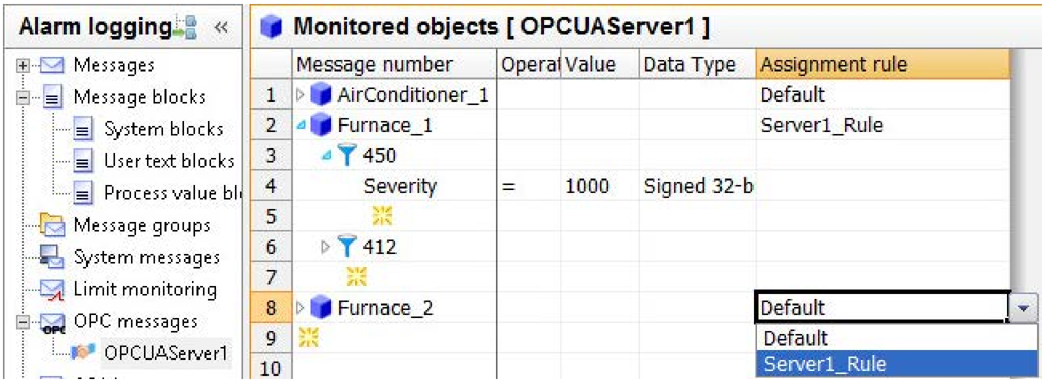


	Name	Process value: 2	Process value: 3	Process value:
1	Default	EventType	SourceName	Severity
2	Server1_Rule	Severity		
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				

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.



	Message number	Operat	Value	Data Type	Assignment rule
1	AirConditioner_1				Default
2	Furnace_1				Server1_Rule
3	450				
4	Severity	=	1000	Signed 32-b	
5					
6	412				
7					
8	Furnace_2				Default
9					
10					

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 326)

Filters for the OPC UA alarms (Page 329)

7.6.5 OPC UA Arrays in the OPC UA WinCC Channel

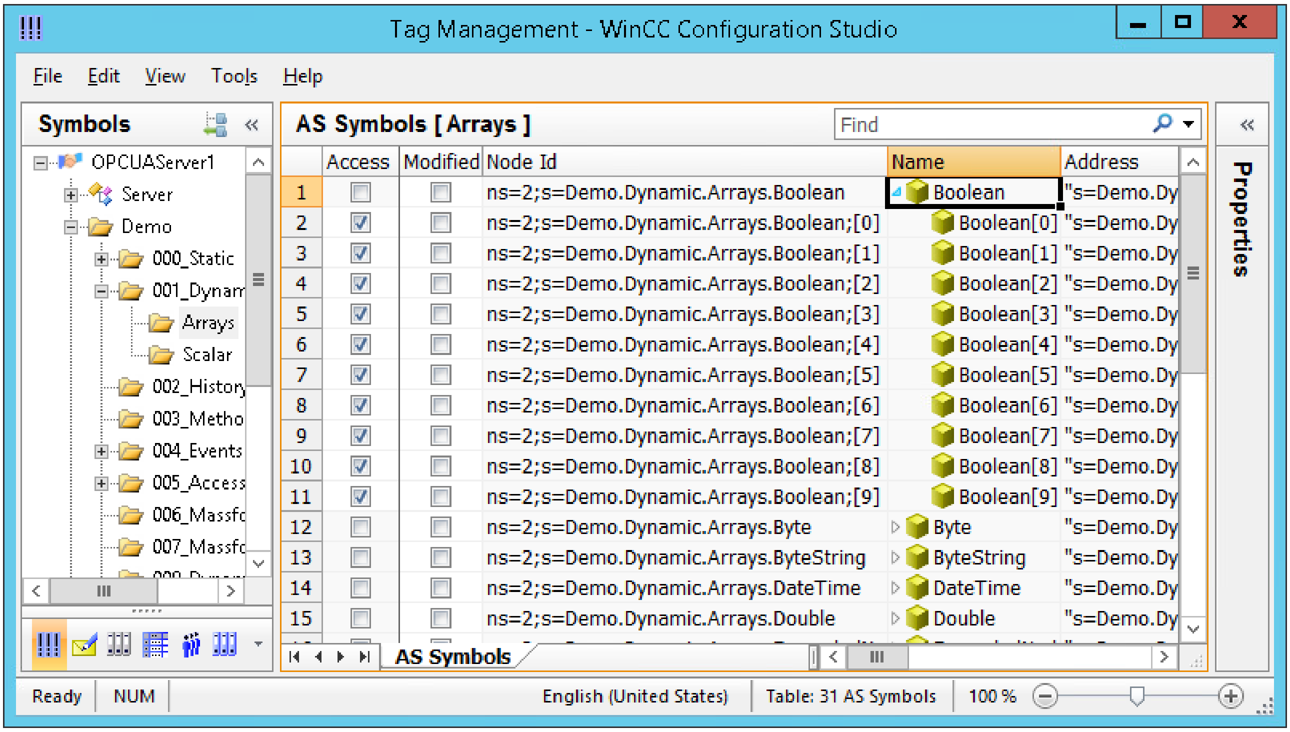
7.6.5.1 Arrays in WinCC

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.



	Access	Modified	Node Id	Name	Address
1	☐	☐	ns=2;s=Demo.Dynamic.Arrays.Boolean	Boolean	"s=Demo.Dy
2	☑	☐	ns=2;s=Demo.Dynamic.Arrays.Boolean;[0]	Boolean[0]	"s=Demo.Dy
3	☑	☐	ns=2;s=Demo.Dynamic.Arrays.Boolean;[1]	Boolean[1]	"s=Demo.Dy
4	☑	☐	ns=2;s=Demo.Dynamic.Arrays.Boolean;[2]	Boolean[2]	"s=Demo.Dy
5	☑	☐	ns=2;s=Demo.Dynamic.Arrays.Boolean;[3]	Boolean[3]	"s=Demo.Dy
6	☑	☐	ns=2;s=Demo.Dynamic.Arrays.Boolean;[4]	Boolean[4]	"s=Demo.Dy
7	☑	☐	ns=2;s=Demo.Dynamic.Arrays.Boolean;[5]	Boolean[5]	"s=Demo.Dy
8	☑	☐	ns=2;s=Demo.Dynamic.Arrays.Boolean;[6]	Boolean[6]	"s=Demo.Dy
9	☑	☐	ns=2;s=Demo.Dynamic.Arrays.Boolean;[7]	Boolean[7]	"s=Demo.Dy
10	☑	☐	ns=2;s=Demo.Dynamic.Arrays.Boolean;[8]	Boolean[8]	"s=Demo.Dy
11	☑	☐	ns=2;s=Demo.Dynamic.Arrays.Boolean;[9]	Boolean[9]	"s=Demo.Dy
12	☐	☐	ns=2;s=Demo.Dynamic.Arrays.Byte	Byte	"s=Demo.Dy
13	☐	☐	ns=2;s=Demo.Dynamic.Arrays.ByteString	ByteString	"s=Demo.Dy
14	☐	☐	ns=2;s=Demo.Dynamic.Arrays.DateTime	DateTime	"s=Demo.Dy
15	☐	☐	ns=2;s=Demo.Dynamic.Arrays.Double	Double	"s=Demo.Dy

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

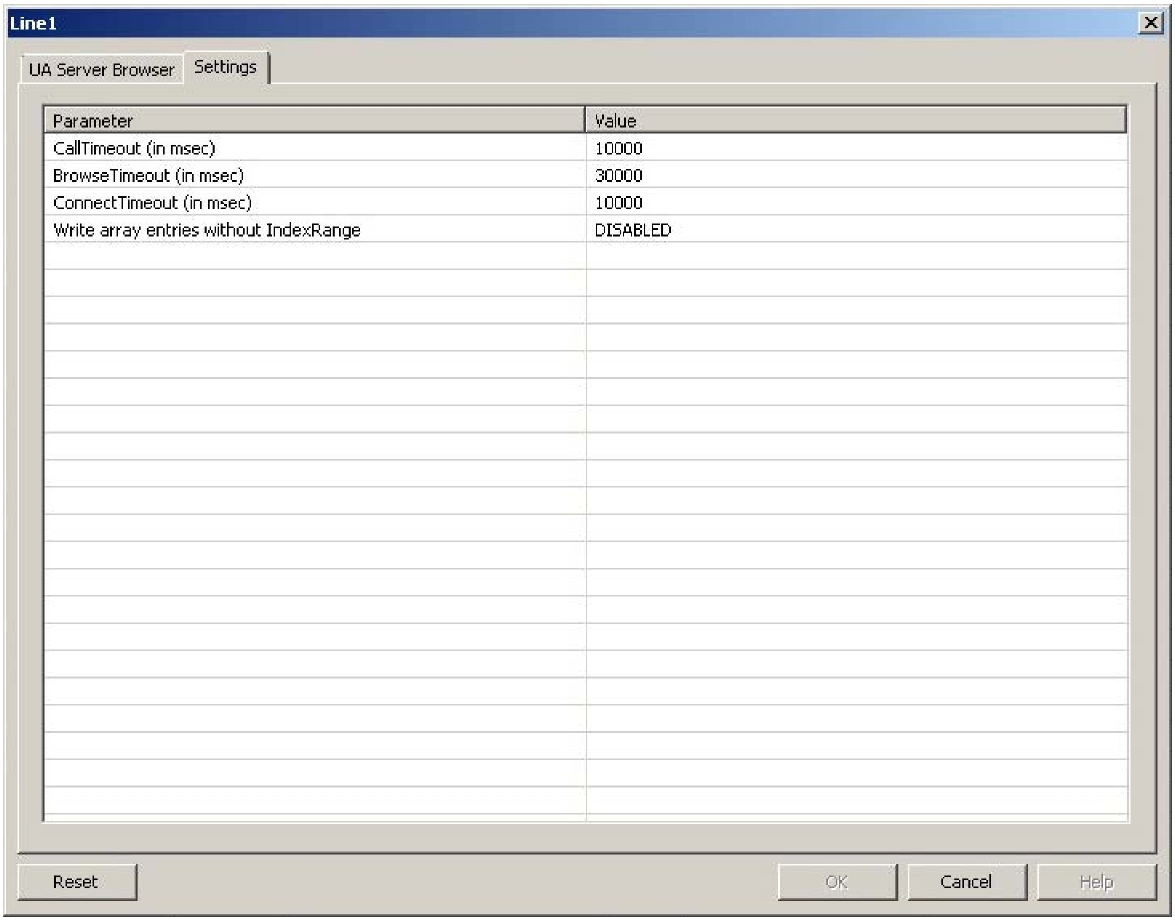
When reading and writing an OPC UA Array the OPC UA Client can access the entire Array or only a section of it by the IndexRange function specified in the OPC-UA norm being used.

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.



Parameter	Value
CallTimeout (in msec)	10000
BrowseTimeout (in msec)	30000
ConnectTimeout (in msec)	10000
Write array entries without IndexRange	DISABLED

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 339).

7.6.5.2 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.

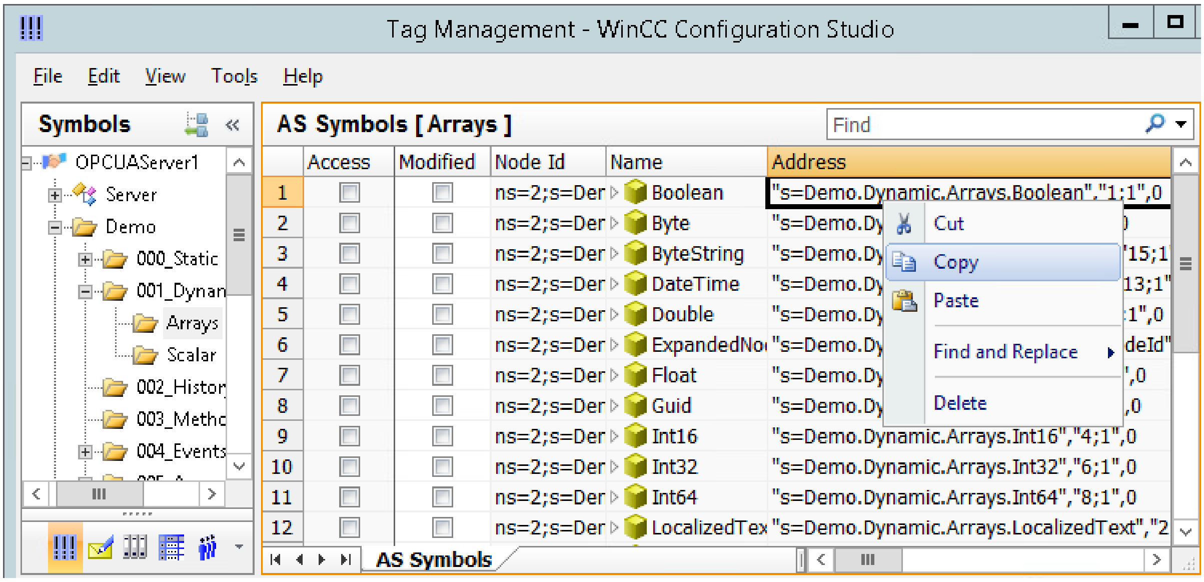
   Note

   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.

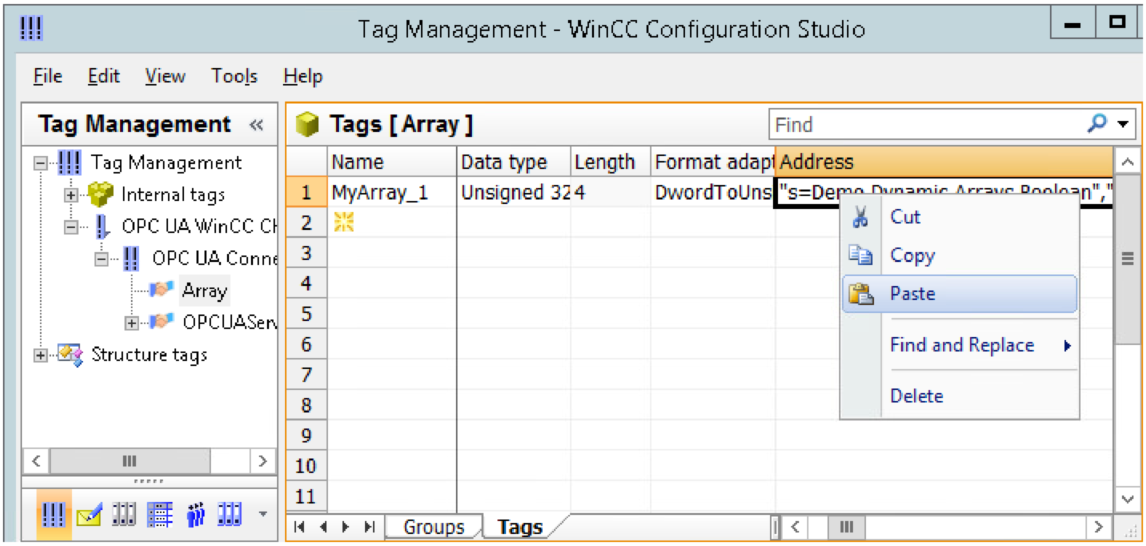
2. Rename the newly created tag, if applicable.

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



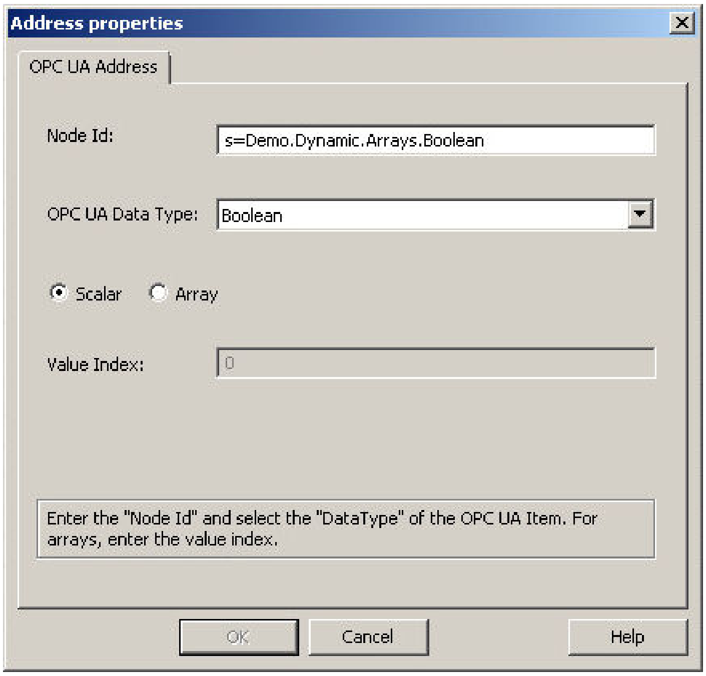
	Access	Modified	Node Id	Name	Address
1	☐	☐	ns=2;s=Der	Boolean	"s=Demo.Dynamic.Arrays.Boolean","1;1",0
2	☐	☐	ns=2;s=Der	Byte	"s=Demo.Dy)"
3	☐	☐	ns=2;s=Der	ByteString	"s=Demo.Dy '15;1
4	☐	☐	ns=2;s=Der	DateTime	"s=Demo.Dy 13;1"
5	☐	☐	ns=2;s=Der	Double	"s=Demo.Dy 1",0
6	☐	☐	ns=2;s=Der	ExpandedNo	"s=Demo.Dy deId"
7	☐	☐	ns=2;s=Der	Float	"s=Demo.Dy ',0
8	☐	☐	ns=2;s=Der	Guid	"s=Demo.Dy ,0
9	☐	☐	ns=2;s=Der	Int16	"s=Demo.Dynamic.Arrays.Int16","4;1",0
10	☐	☐	ns=2;s=Der	Int32	"s=Demo.Dynamic.Arrays.Int32","6;1",0
11	☐	☐	ns=2;s=Der	Int64	"s=Demo.Dynamic.Arrays.Int64","8;1",0
12	☐	☐	ns=2;s=Der	LocalizedTex	"s=Demo.Dynamic.Arrays.LocalizedText","2

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



	Name	Data type	Length	Format adapt	Address
1	MyArray_1	Unsigned 32	4	DwordToUns	"s=Demo.Dynamic.Arrays.Boolean","
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

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



Address properties

OPC UA Address

Node Id: s=Demo.Dynamic.Arrays.Boolean

OPC UA Data Type: Boolean

● Scalar ○ Array

Value Index: 0

Enter the "Node Id" and select the "DataType" of the OPC UA Item. For arrays, enter the value index.

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

7.6.6 OPC UA methods in the OPC UA WinCC Channel

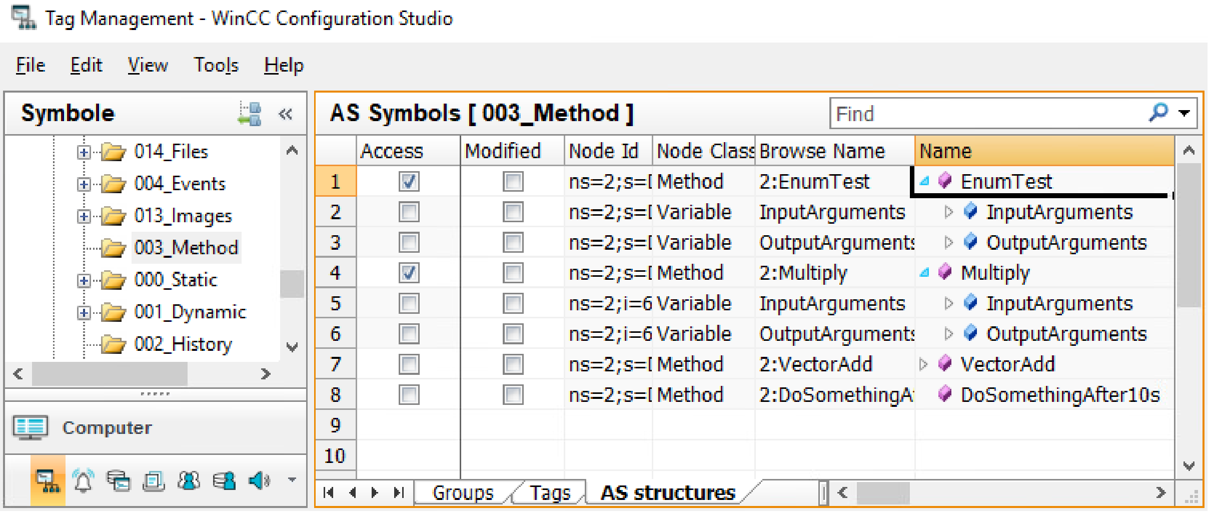
7.6.6.1 OPC UA methods in WinCC

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.



	Access	Modified	Node Id	Node Class	Browse Name	Name
1	☑	□	ns=2;s=[	Method	2:EnumTest	EnumTest
2	□	□	ns=2;s=[	Variable	InputArguments	InputArguments
3	□	□	ns=2;s=[	Variable	OutputArgument	OutputArguments
4	☑	□	ns=2;s=[	Method	2:Multiply	Multiply
5	□	□	ns=2;i=6	Variable	InputArguments	InputArguments
6	□	□	ns=2;i=6	Variable	OutputArgument	OutputArguments
7	□	□	ns=2;s=[	Method	2:VectorAdd	VectorAdd
8	□	□	ns=2;s=[	Method	2:DoSomethingA	DoSomethingAfter10s
9						
10						

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.

7.6.6.2 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.

7.6.7 Error handling

7.6.7.1 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.

7.7 PROFIBUS DP

7.7.1 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.

7.7.2 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.

7.7.3 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.

7.7.4 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

Field	Meaning
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.

7.8 S5 Ethernet Layer 4

7.8.1 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).

7.8.2 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

7.8.3 Configuring the Channel

7.8.3.1 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

7.8.3.2 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.

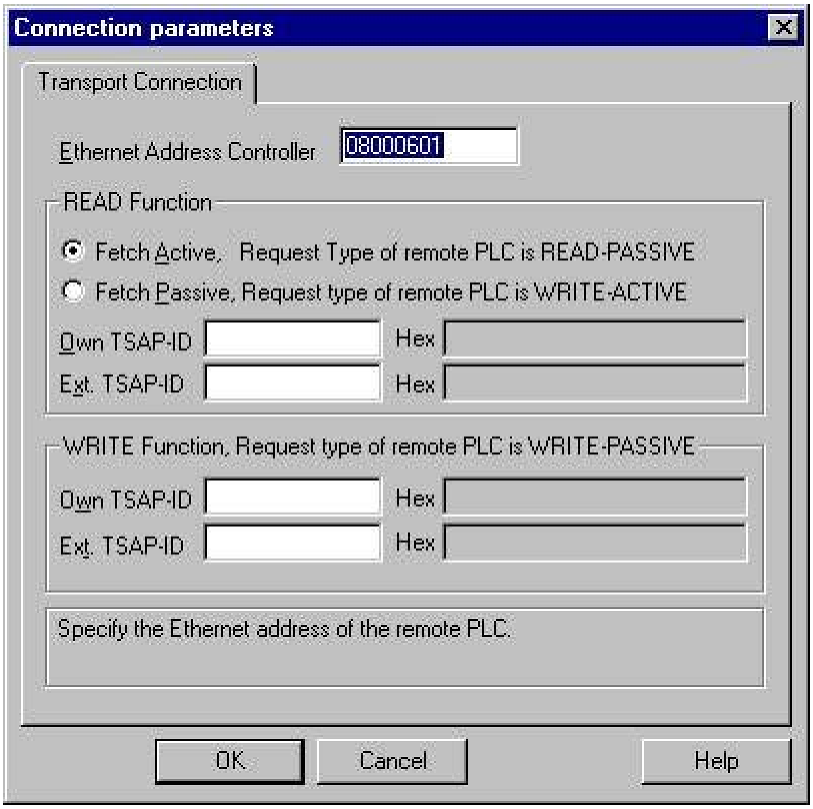
A FETCH Passive connections is only assigned the "OK" status if at least one telegram has been sent from AS to WinCC.

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.



Connection parameters

Transport Connection

Ethernet Address Controller: 08000601

READ Function

● Fetch Active, Request Type of remote PLC is READ-PASSIVE

○ Fetch Passive, Request type of remote PLC is WRITE-ACTIVE

Own TSAP-ID: Hex:

Ext. TSAP-ID: Hex:

WRITE Function, Request type of remote PLC is WRITE-PASSIVE

Own TSAP-ID: Hex:

Ext. TSAP-ID: Hex:

Specify the Ethernet address of the remote PLC.

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.

7.8.3.3 Configuring the tags

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

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.

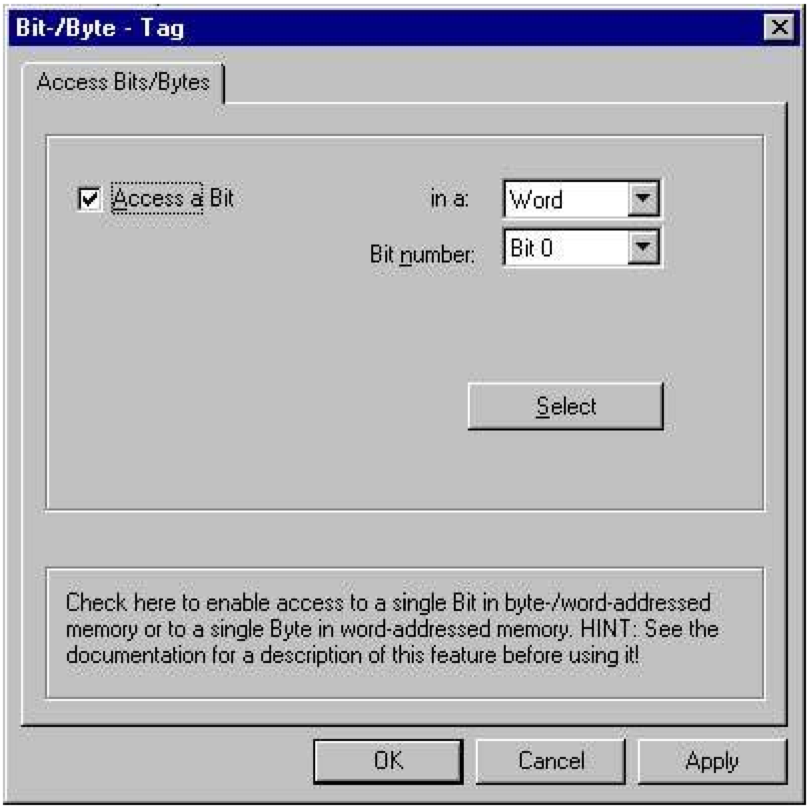
Configuring the address of a tag is done independent of the tag type:

- 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.

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.



☑ Access a Bit

in a: Word

Bit number: Bit 0

Check here to enable access to a single Bit in byte-/word-addressed memory or to a single Byte in word-addressed memory. HINT: See the documentation for a description of this feature before using it!

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.

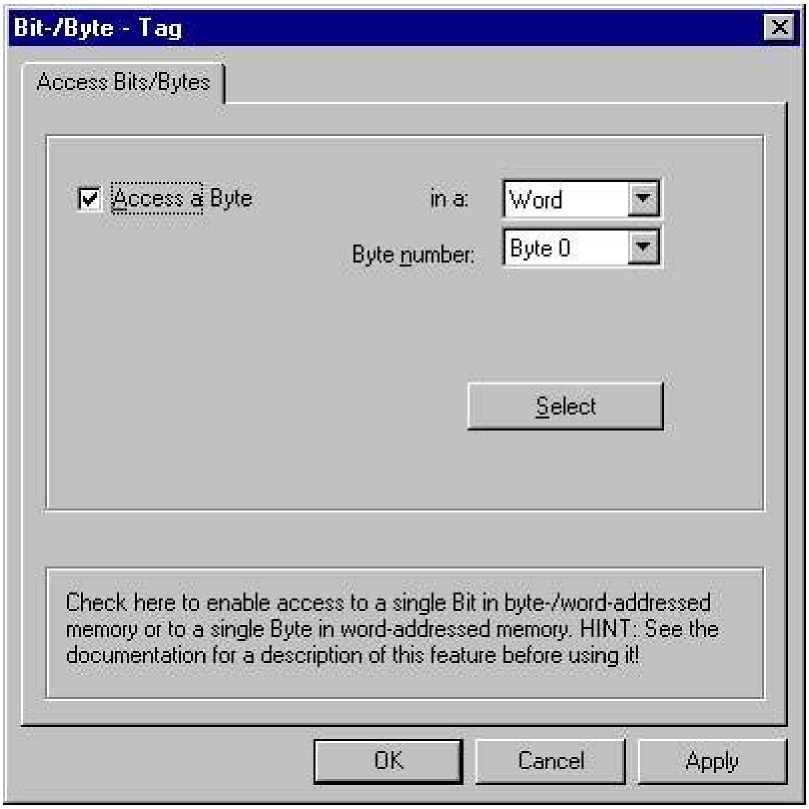
Activating the check box "Access a bit" affects the display of the fields of the "Address properties" dialog.

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.

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.



Bit-/Byte - Tag

Access Bits/Bytes

☑ Access a Byte

in a: Word

Byte number: Byte 0

Check here to enable access to a single Bit in byte-/word-addressed memory or to a single Byte in word-addressed memory. HINT: See the documentation for a description of this feature before using it!

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.

How to Configure a Tag with Word by Word Access

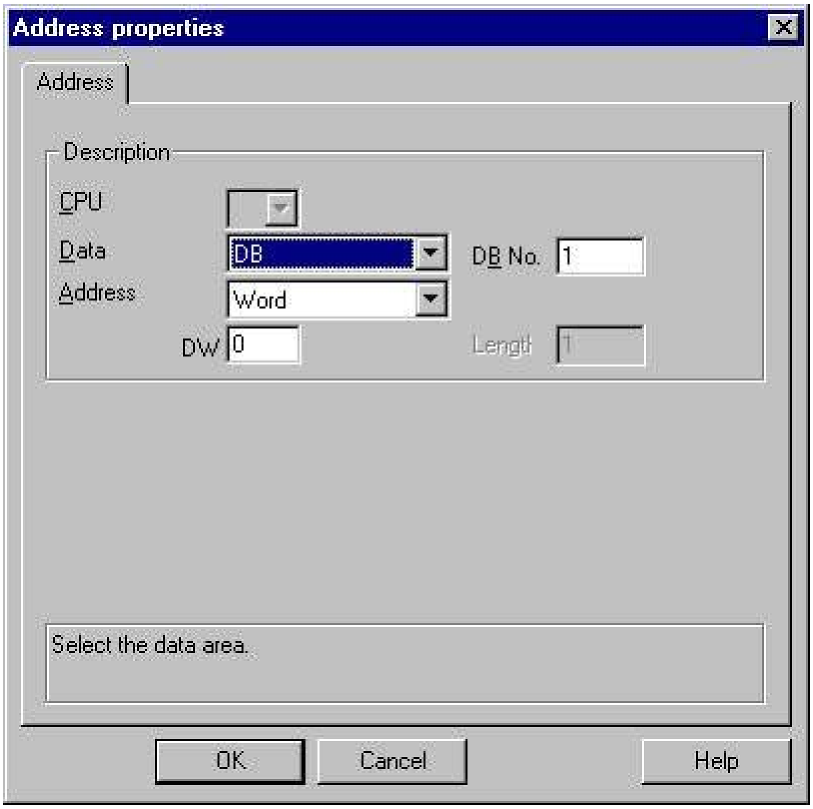
Introduction

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.



CPU:
Data: DB
DB No. 1
Address: Word
DW 0
Length 1

Select the data area.

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.

How to configure a raw data tag

Introduction

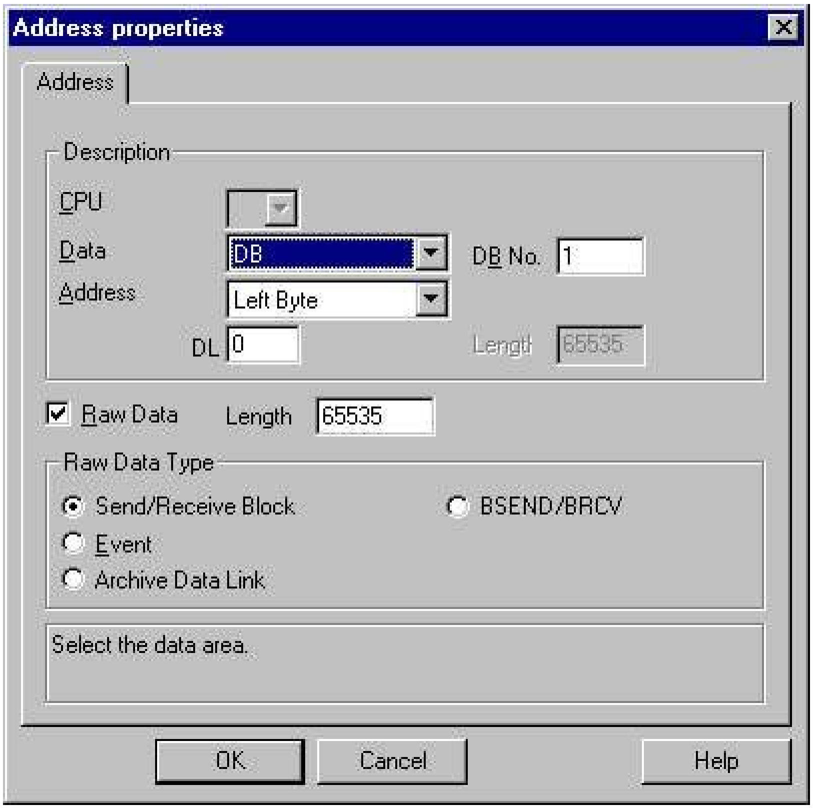
The following is a description of how the address of a raw data tag is defined.

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.



Address properties

Address

Description

CPU

Data: DB   DB No.: 1

Address: Left Byte

DL: 0   Length: 65535

☑ Raw Data   Length: 65535

Raw Data Type

● Send/Receive Block   ○ BSEND/BRCV

○ Event

○ Archive Data Link

Select the data area.

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.

7.8.3.4 System parameters

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.

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.

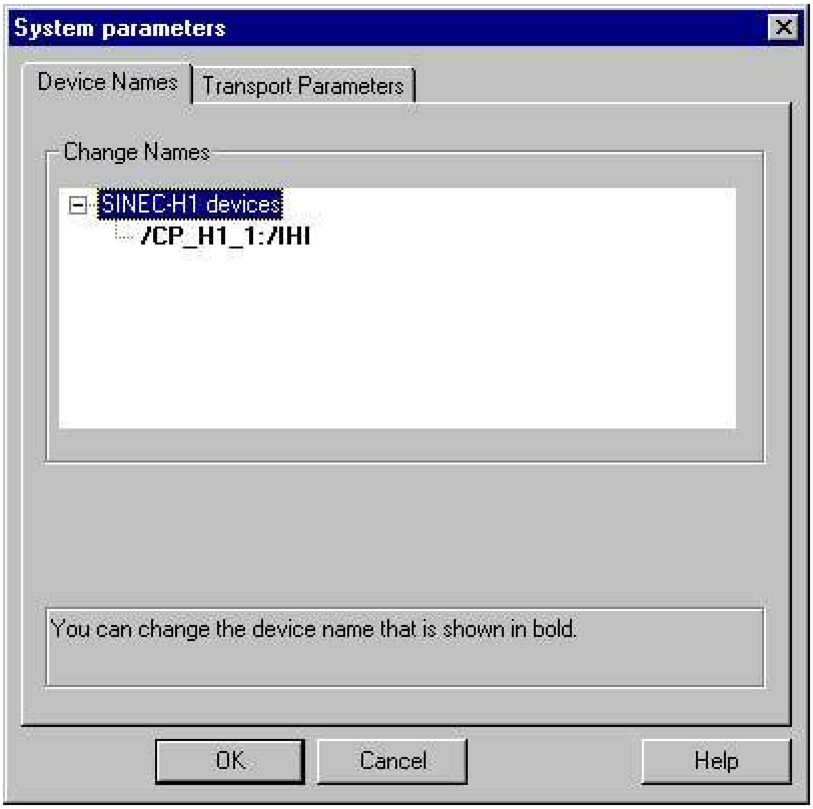
The system parameters are almost identical for all protocols used.

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.



You can change the device name that is shown in bold.

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.

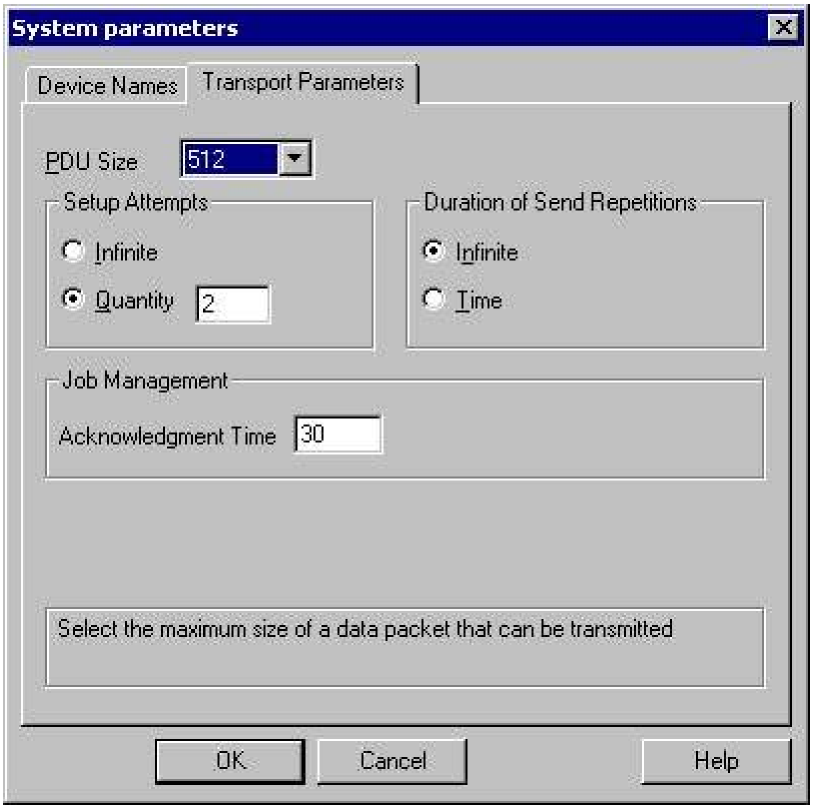
Note

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.

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.



System parameters

Device Names | Transport Parameters

PDU Size: 512

Setup Attempts
○ Infinite
● Quantity: 2

Duration of Send Repetitions
● Infinite
○ Time

Job Management
Acknowledgment Time: 30

Select the maximum size of a data packet that can be transmitted

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).

7.8.4 Appendix

7.8.4.1 Appendix

Introduction

Added information on the channel "SIMATIC S5 Ethernet Layer 4" is provided in the appendix.

7.8.4.2 Internal error codes and constants

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

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.

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

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

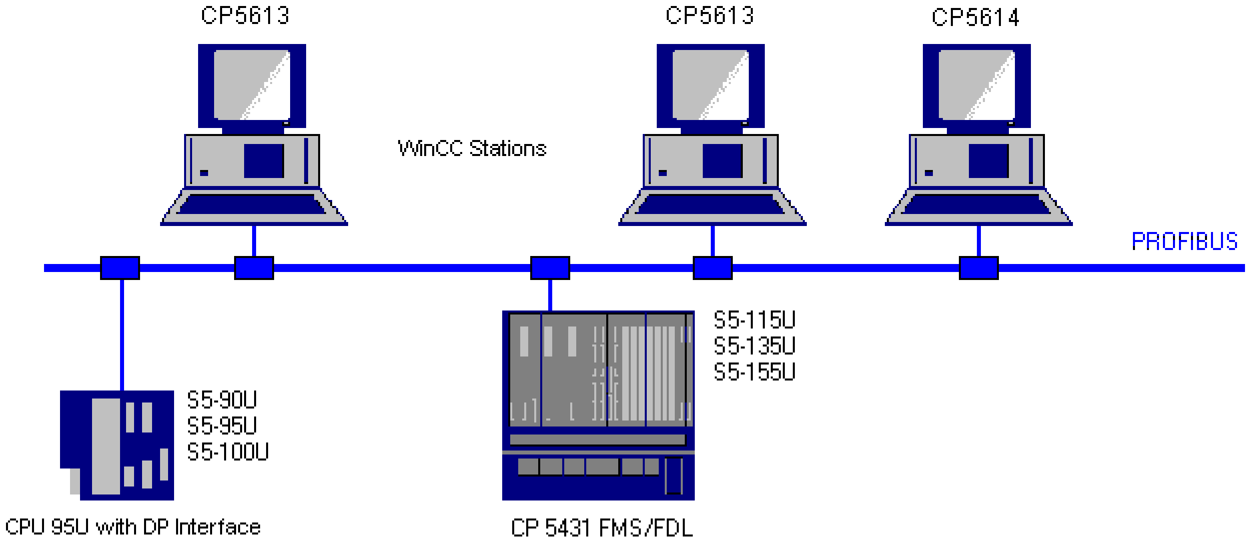
7.9 S5 PROFIBUS FDL

7.9.1 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.

7.9.2 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.

7.9.3 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.

7.9.4 Configuring the Channel

7.9.4.1 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.

7.9.4.2 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.

7.9.4.3 How to configure a connection

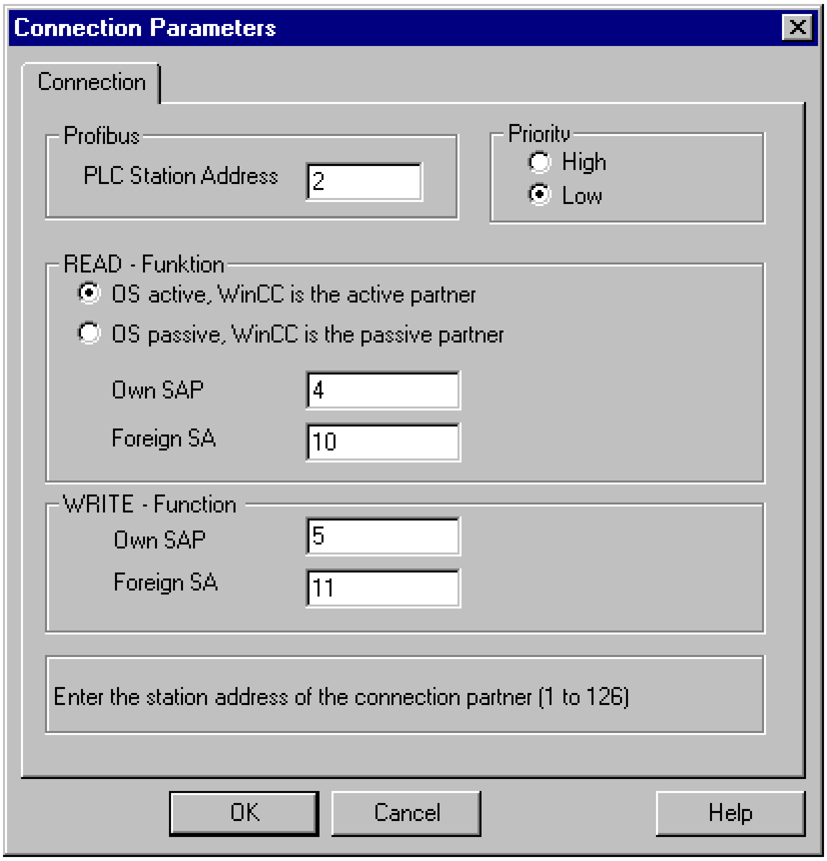
Requirements

- 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.



Connection Parameters

Connection

Profibus

PLC Station Address: 2

Priority

○ High

● Low

READ - Funktion

● OS active, WinCC is the active partner

○ OS passive, WinCC is the passive partner

Own SAP: 4

Foreign SA: 10

WRITE - Function

Own SAP: 5

Foreign SA: 11

Enter the station address of the connection partner (1 to 126)

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.

7.9.4.4 Configuring the tags

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.

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.

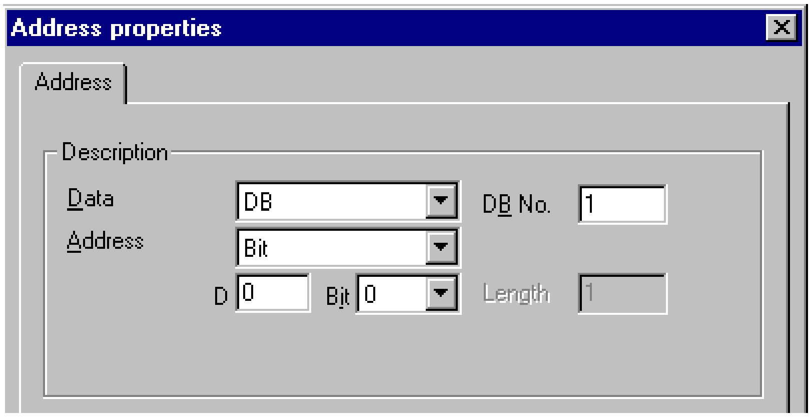
Requirements

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.



Data: DB
DB No.: 1
Address: Bit
D: 0
Bit: 0
Length: 1

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.

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

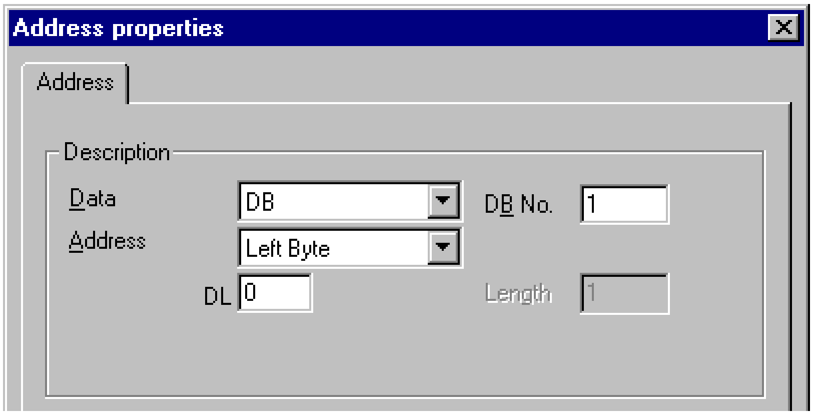
The byte-wise access to a tag is only read access.

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.



Address properties

Address

Description

Data: DB

DB No.: 1

Address: Left Byte

DL: 0

Length: 1

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.

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

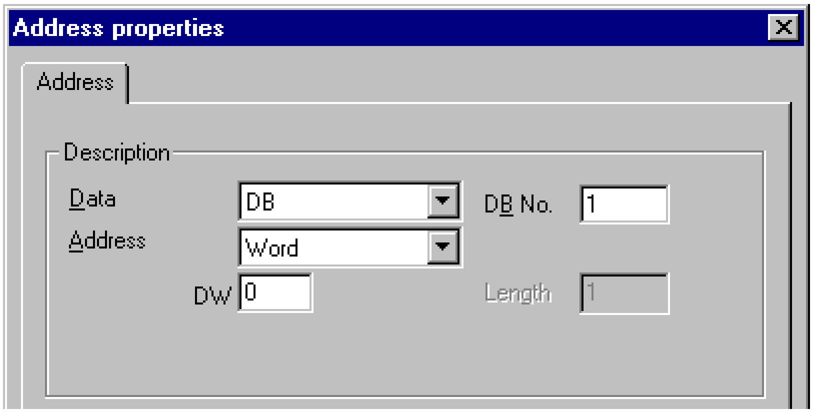
The word by word access to a tag is read and/or write access.

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.



Data: DB
DB No.: 1
Address: Word
DW: 0
Length: 1

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.

7.9.4.5 System parameters

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.

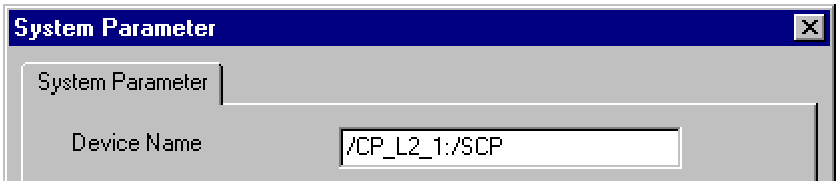
How to Change the Device Name

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.



Device Name: /CP_L2_1:/SCP

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.

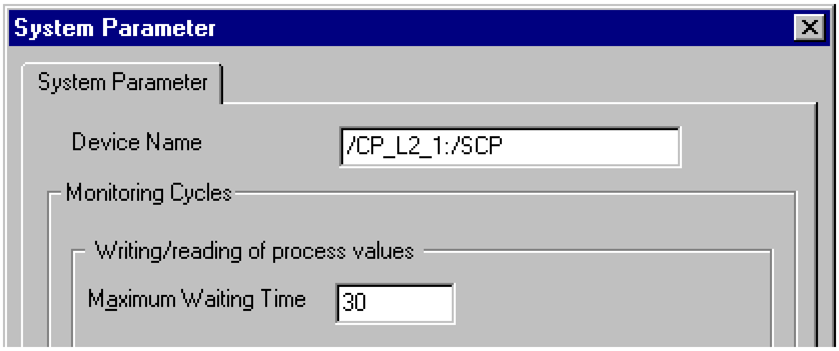
How to change the write/read monitoring time of process values

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.



Device Name: /CP_L2_1:/SCP

Maximum Waiting Time: 30

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.

7.9.5 Special Functions

7.9.5.1 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.

7.9.5.2 Raw data tags of the "SIMATIC S5 Profibus FDL" channel

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.

How to configure raw data tags

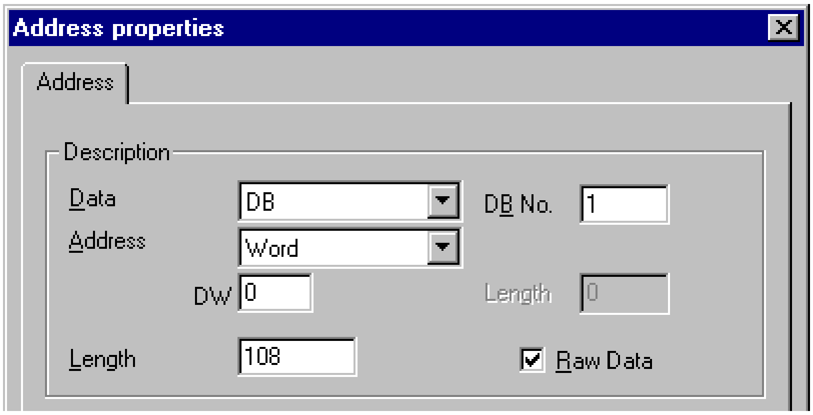
Requirements

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.



Address properties

Address

Description

Data: DB

DB No.: 1

Address: Word

DW: 0

Length: 0

Length: 108

☑ Raw Data

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.

7.9.5.3 Configuring the communication types

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.

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

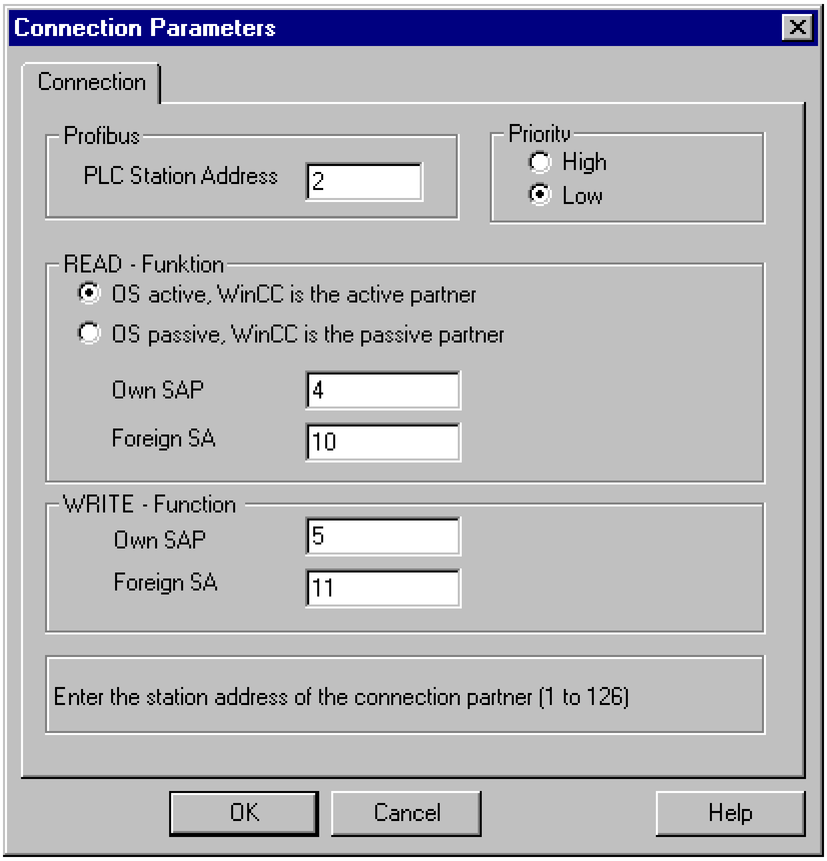
If more than one connection is configured, note that an SAP can only be assigned one time.

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.



PLC Station Address: 2

○ High
● Low

● OS active, WinCC is the active partner
○ OS passive, WinCC is the passive partner

Own SAP: 4
Foreign SA: 10

Own SAP: 5
Foreign SA: 11

Enter the station address of the connection partner (1 to 126)

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.

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

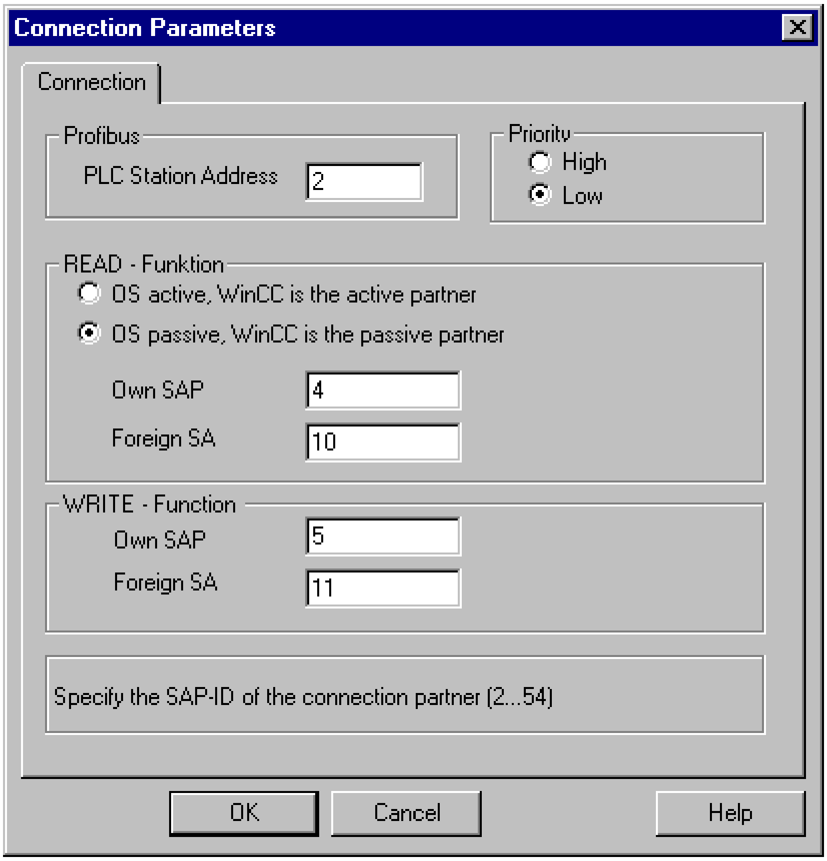
If more than one connection is configured, note that an SAP can only be assigned one time.

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.



Connection Parameters

Connection

Profibus

PLC Station Address: 2

Priority

○ High
● Low

READ - Funktion

○ OS active, WinCC is the active partner
● OS passive, WinCC is the passive partner

Own SAP: 4

Foreign SA: 10

WRITE - Function

Own SAP: 5

Foreign SA: 11

Specify the SAP-ID of the connection partner (2...54)

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.

7.9.6 Example of configuring the "SIMATIC S5 Profibus FDL" channel

7.9.6.1 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.

7.9.6.2 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.

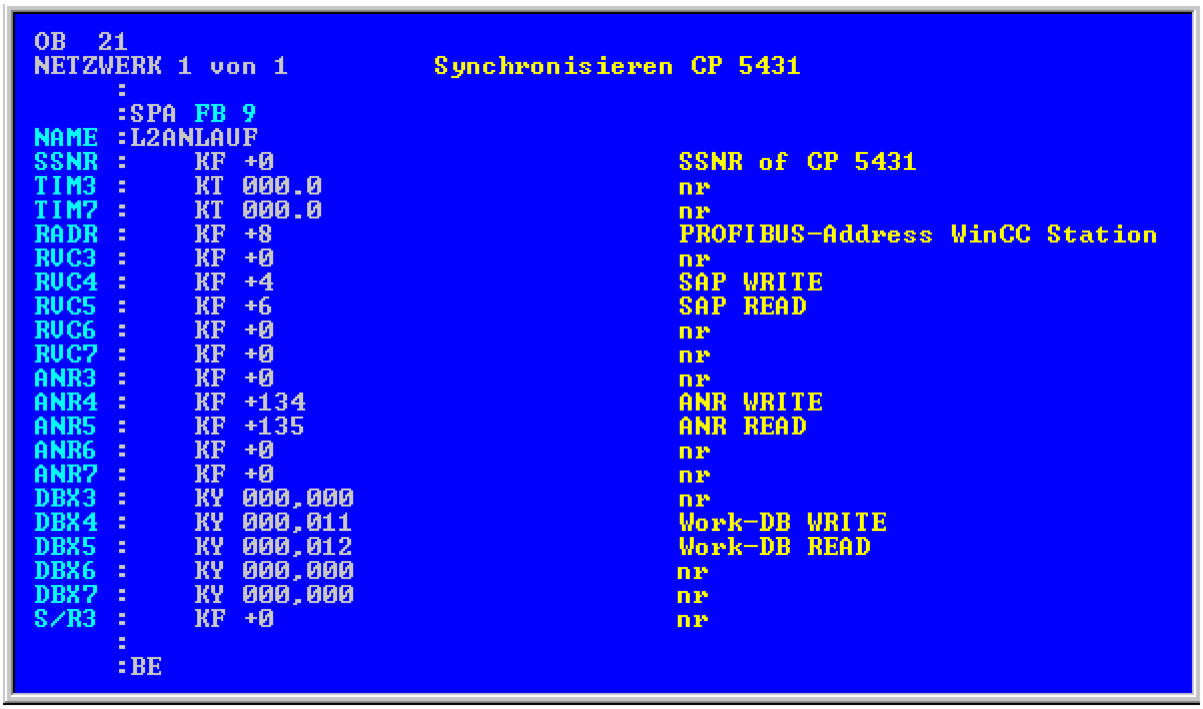
Feedback occurs after a completed transmission.

Requirements

- The data handling blocks SYNCHRON, CONTROL, SEND and RECEIVE must be available in the AS.

Procedure

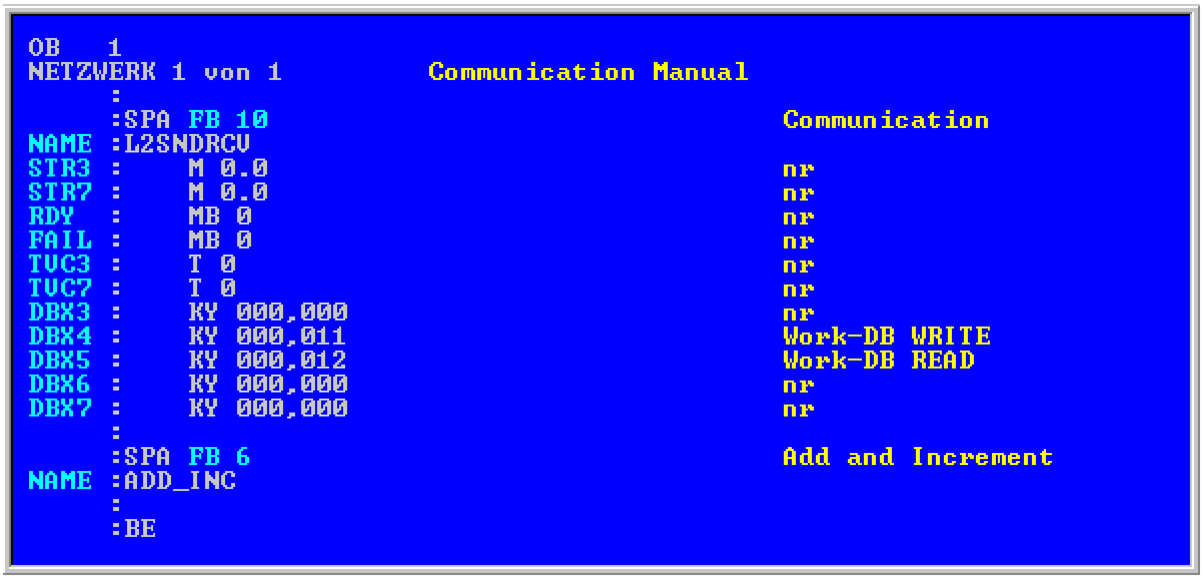
1. A startup block (OB 20, 21, 22) is created in the STEP 5 software using menu item "Editor" → "STEP5 Block" → "in the program file".
   The name of the program block is "L2ANLAUF" in the example.
2. The following parameters must be preassigned:
   – Interface number (SSNR) of the CP (e.g. of CP5431)
   – 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.

7.9.6.3 How to configure an I/O Field

Introduction

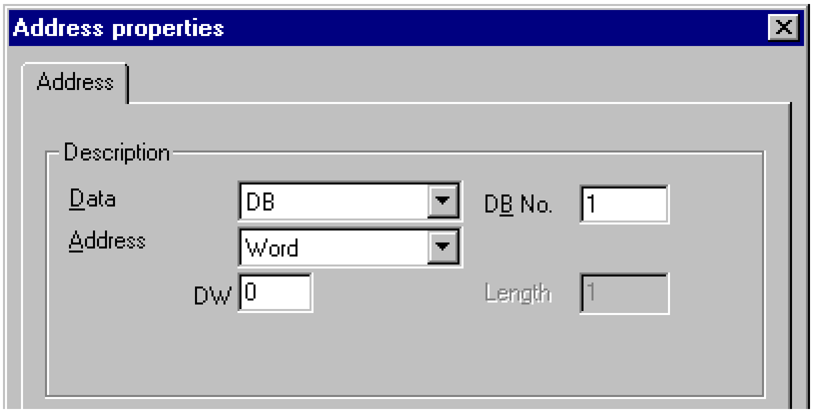
You will configure an I/O Field in this section.

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.



Address properties

Address

Description

Data: DB

DB No.: 1

Address: Word

DW: 0

Length: 1

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.

7.10 S5 Programmers Port AS511

7.10.1 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.

7.10.2 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

7.10.3 Configuring the Channel

7.10.3.1 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

7.10.3.2 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.

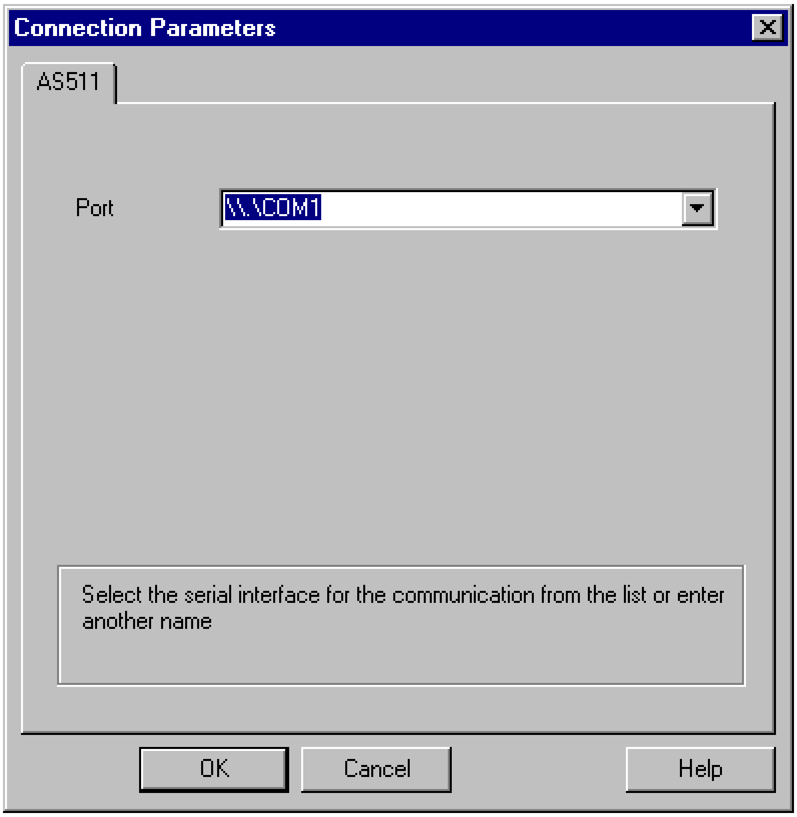
The following procedure can be used to assign one of the PC's serial ports to the AS511-NT drivers.

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.



Port: \\.\COM1

Select the serial interface for the communication from the list or enter another name

3. In the "Port" field, select the port to be used for the serial link.

7.10.3.3 Configuring the tags

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.

How to configure the address of a tag

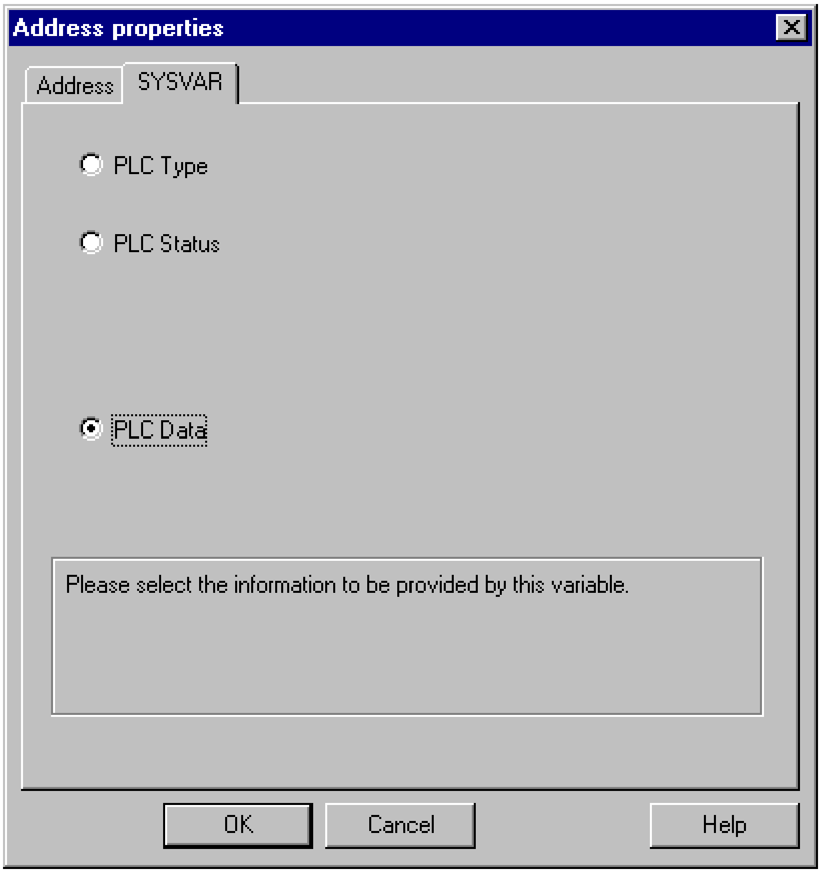
Introduction

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.



○ PLC Type

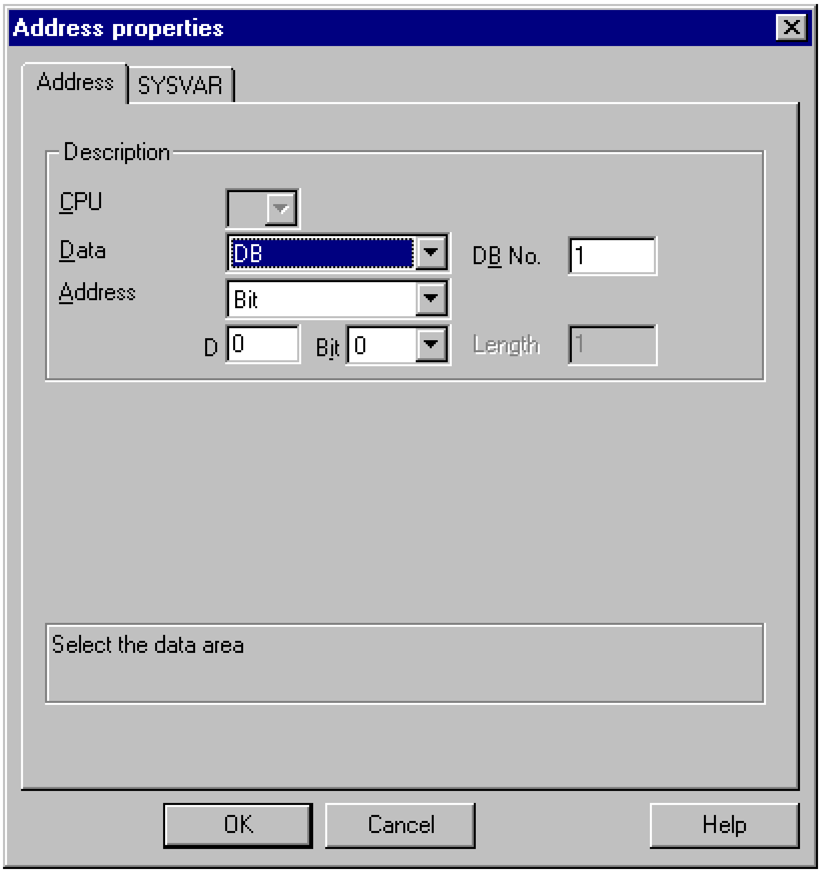
○ PLC Status

● PLC Data

Please select the information to be provided by this variable.

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.



Address properties

Description

CPU:

Data: DB

DB No.: 1

Address: Bit

D: 0

Bit: 0

Length: 1

Select the data area

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.

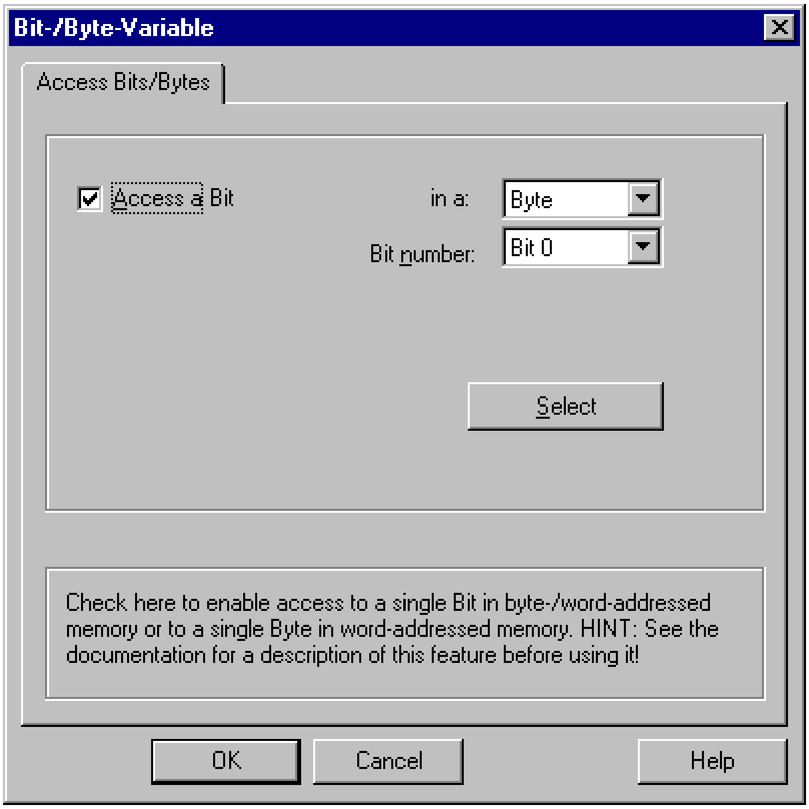
Times cannot be written.

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.



☑ Access a Bit

in a: Byte

Bit number: Bit 0

Check here to enable access to a single Bit in byte-/word-addressed memory or to a single Byte in word-addressed memory. HINT: See the documentation for a description of this feature before using it!

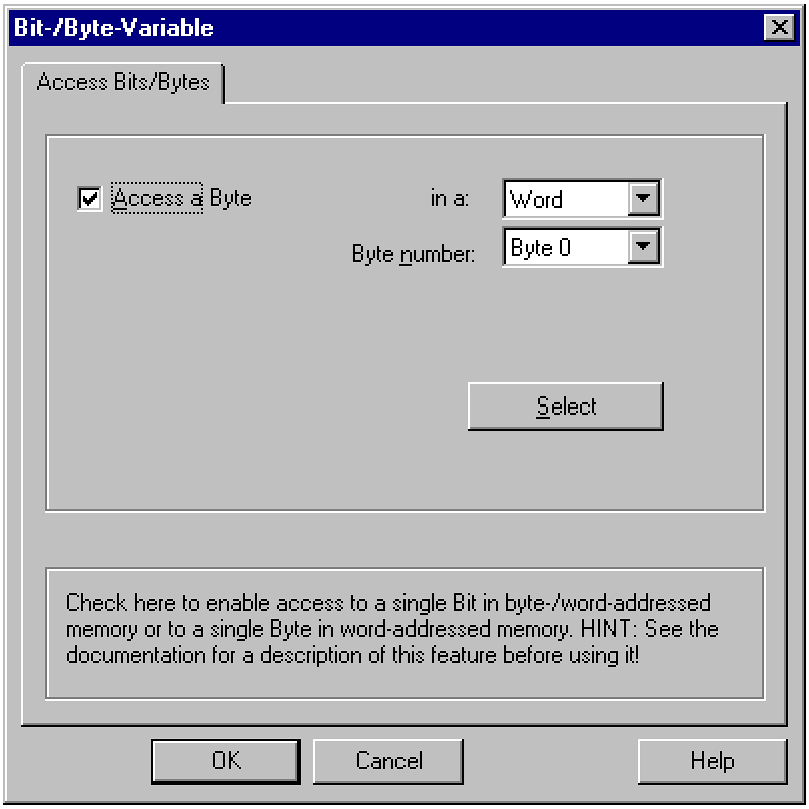
6. Click the "Select" button. The "Address properties" dialog is opened.
7. Select the addressing type of the PLC memory in the selection field.
8. Select the number of bit to be changed in the selection field.

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.



Bit-/Byte-Variable

Access Bits/Bytes

☑ Access a Byte

in a: Word

Byte number: Byte 0

Check here to enable access to a single Bit in byte-/word-addressed memory or to a single Byte in word-addressed memory. HINT: See the documentation for a description of this feature before using it!

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.

7.11 S5 Serial 3964R

7.11.1 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.

7.11.2 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

7.11.3 Configuring the Channel

7.11.3.1 Configuring the "SIMATIC S5 Serial 3964R" channel

Introduction

The following steps are required for configuring the channel "SIMATIC S5 Serial 3964R".

7.11.3.2 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.

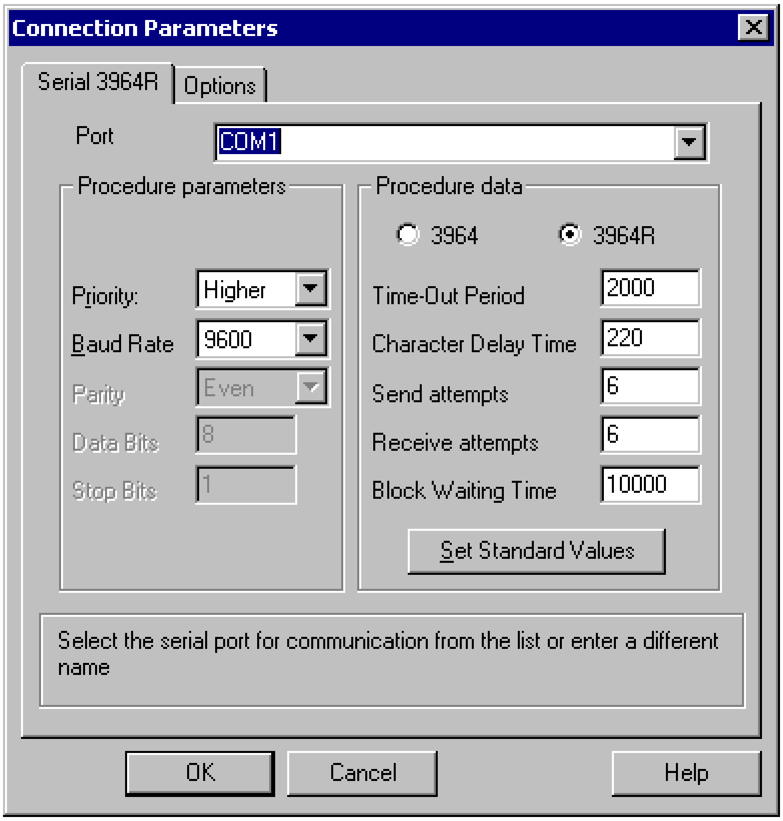
This serial link supports transmission rates of up to 19200 baud.

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.



Connection Parameters

Serial 3964R | Options

Port: COM1

Procedure parameters

Priority: Higher

Baud Rate: 9600

Parity: Even

Data Bits: 8

Stop Bits: 1

Procedure data

○ 3964 ● 3964R

Time-Out Period: 2000

Character Delay Time: 220

Send attempts: 6

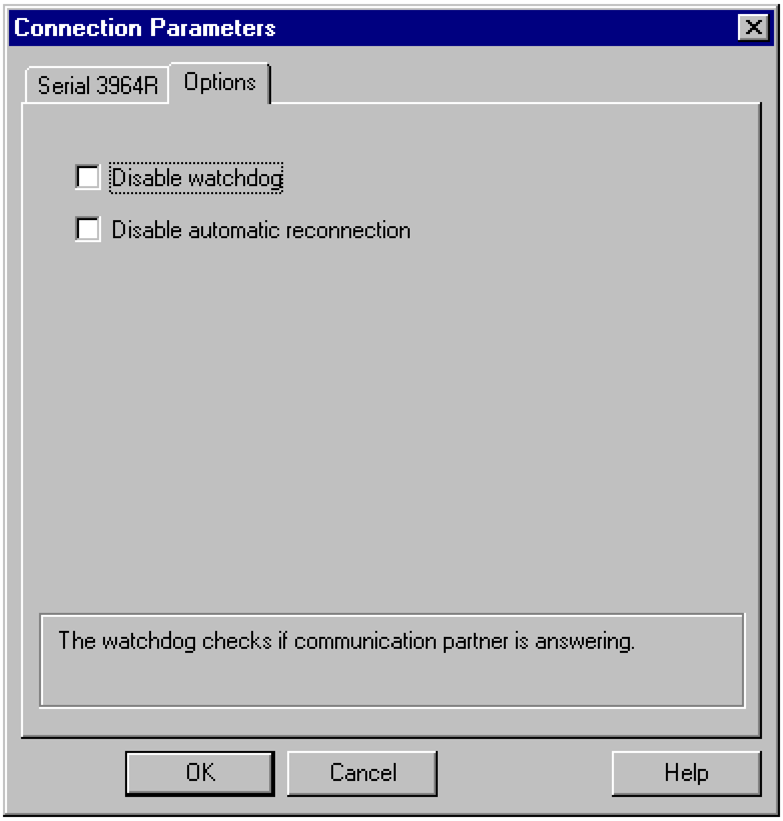
Receive attempts: 6

Block Waiting Time: 10000

Select the serial port for communication from the list or enter a different name

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.



□ Disable watchdog

□ Disable automatic reconnection

The watchdog checks if communication partner is answering.

7. You can disable cyclic life beat monitoring and disable the automatic reconnection on the "Options" tab.

7.11.3.3 Configuring the tags

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.

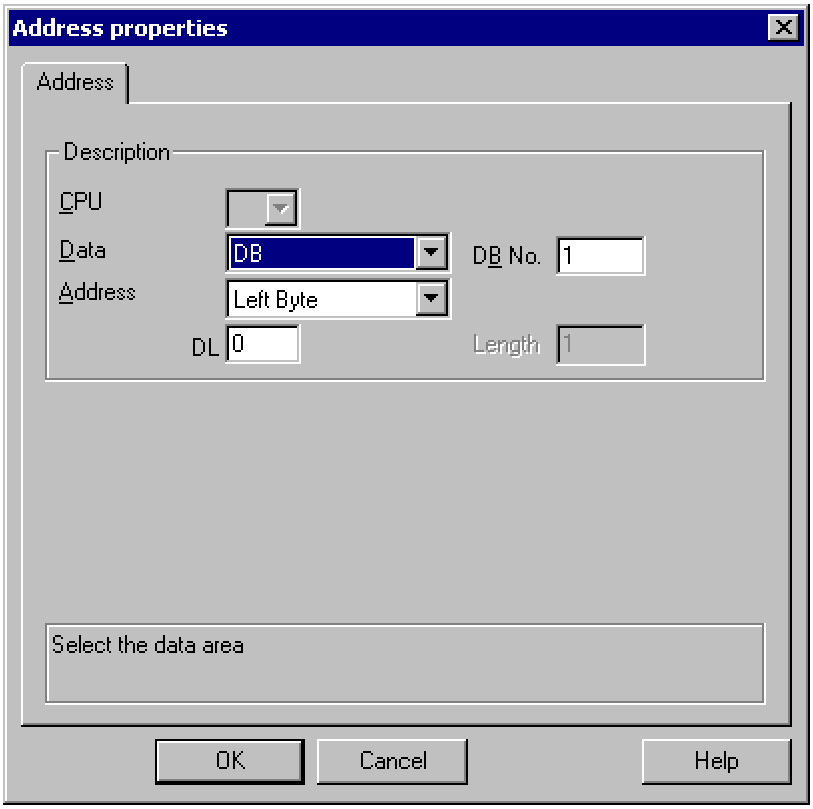
How to configure the address of the tag

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.



Address properties

Description

CPU:

Data: DB

DB No.: 1

Address: Left Byte

DL: 0

Length: 1

Select the data area

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.

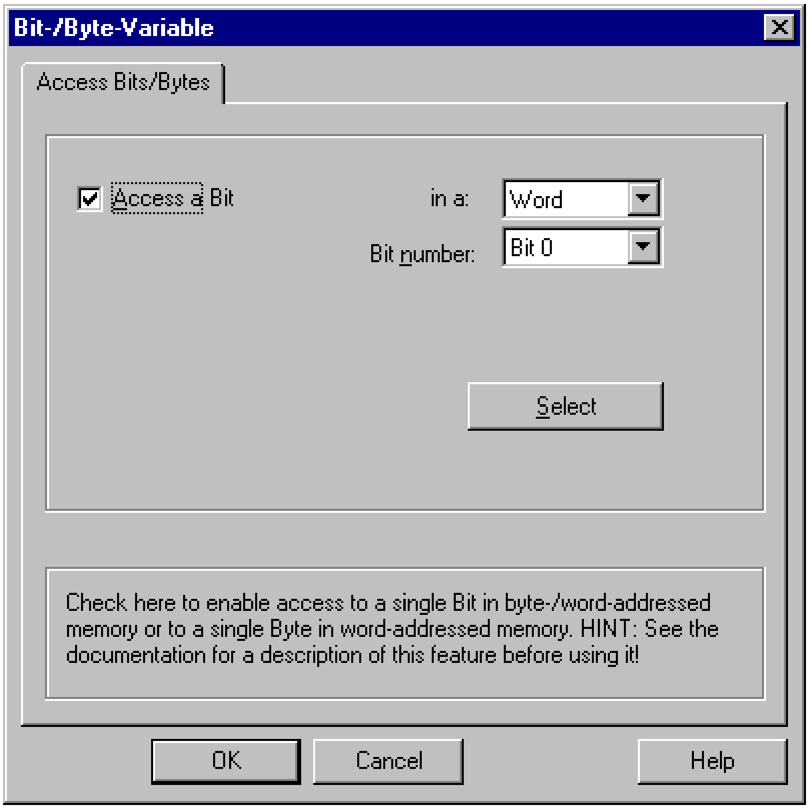
Depending on the type of tag, you can access the automation system's memory bit-wise or byte-wise.

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.



Check here to enable access to a single Bit in byte-/word-addressed memory or to a single Byte in word-addressed memory. HINT: See the documentation for a description of this feature before using it!

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.

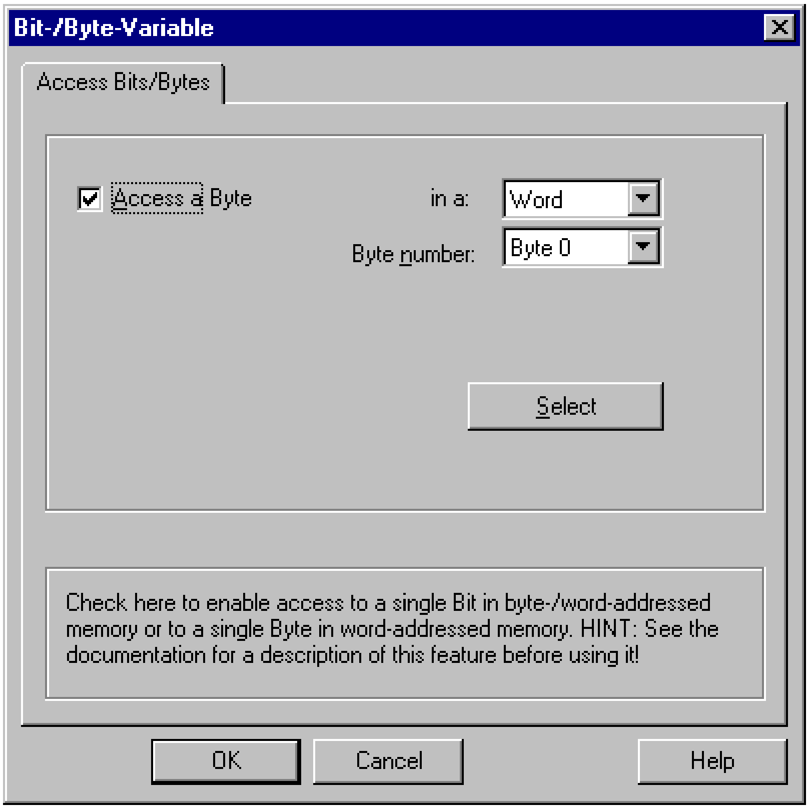
Note

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).

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.



Check here to enable access to a single Bit in byte-/word-addressed memory or to a single Byte in word-addressed memory. HINT: See the documentation for a description of this feature before using it!

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).

7.12 SIMATIC S7 Protocol Suite

7.12.1 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 Channel "SIMATIC S7 Protocol Suite" is used to link SIMATIC S7-300 and SIMATIC S7-400 automation systems.

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

- Industrial Ethernet and Industrial Ethernet (II): 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), via an MPI communications processor or a communication 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.

For more information regarding the diagnosis of channels and tags, refer to "Communication Diagnostics".

Detailed procedures

Additional information and detailed examples of channel configuration can be found 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 additional information on diagnosing channels and tags in "Communication Diagnostics (Page 609)".

See also

Software Redundancy - Connection-specific internal tags (Page 494)

"SIMATIC S7 Protocol Suite" Channel - Configuration (Page 415)

Overview of the supported data types (Page 414)

Channel unit selection (Page 411)

Diagnosis of Channels and Tags (Page 609)

http://support.automation.siemens.com/WW/view/en/21320307 (http://support.automation.siemens.com/WW/view/en/21320307)

7.12.2 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 predefined parameters and limitations on node count and transmission rate. 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 communication module. The table shows the recommended components.

System	CPU or communication module (recommended)
S7-300	CPU 31x CP 342-5 CP 343-5
S7-400	CPU 41x CP 443-5 Ext. CP 443-5 Basic

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 communication module. The table shows the recommended components.

System	CPU or communication module
S7-300	CPU 31x CP 342-5 CP 343-5
S7-400	CPU 41x CP 443-5 Ext. CP 443-5 Basic

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 communication module. The table shows the recommended components.

System	Communication module for Industrial Ethernet	Communication module for TCP/IP protocol
S7-300	CP 343-1	CP 343-1 TCP
S7-400	CP 443-1	CP 443-1 TCP CP 443-1 IT

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

7.12.3 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

Data Types	Type conversion
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.

7.12.4 Configuring the Channel

7.12.4.1 "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 447)

Configuring the tags (Page 442)

Channel units of the "SIMATIC S7 Protocol Suite" channel (Page 424)

Diagnosis of Channels and Tags (Page 609)

7.12.4.2 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 417)

7.12.4.3 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:

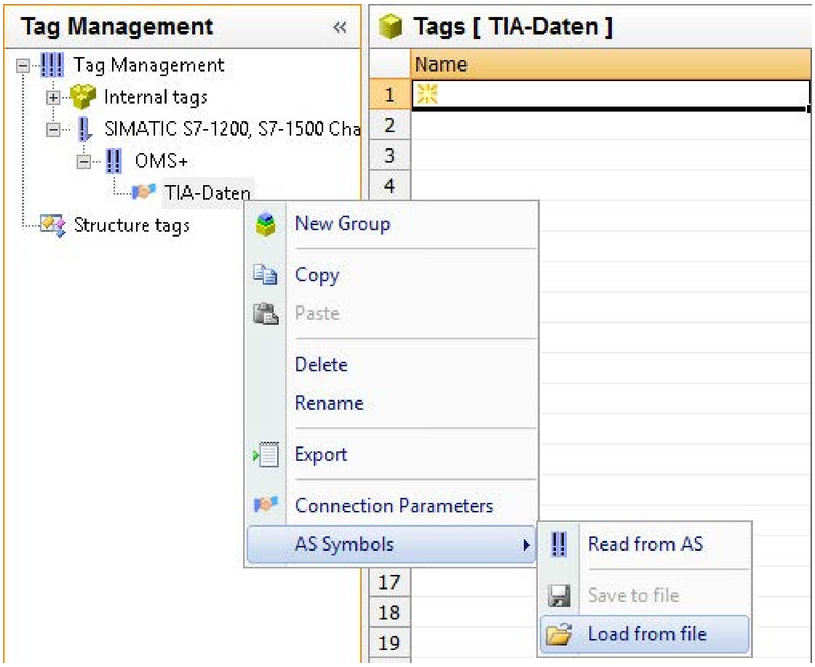
- Industry Online Support: Download "SIMATIC SCADA Export for TIA Portal" (ID 109748955) (https://support.industry.siemens.com/cs/ww/en/view/109748955)
- Industry Online Support: "SIMATIC SCADA Export" documentation (ID 101908495) (https://support.industry.siemens.com/cs/ww/en/view/101908495)

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.

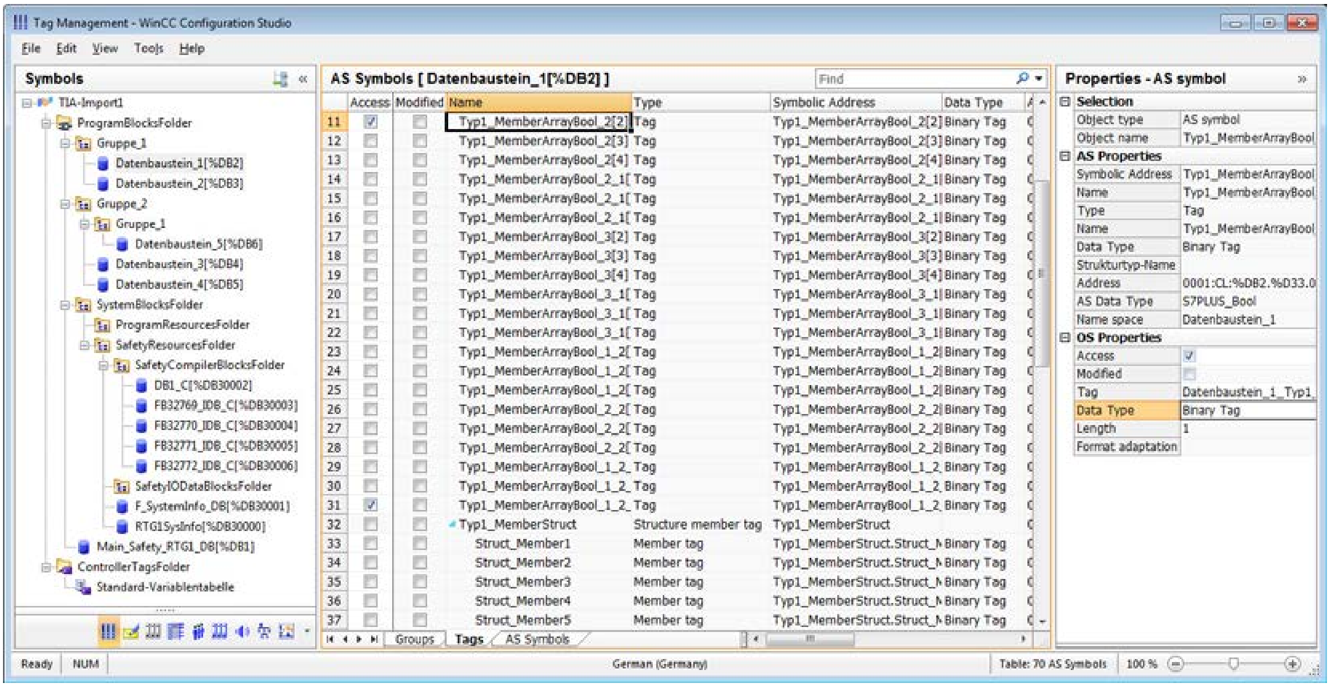
Result

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.



Tag Management - WinCC Configuration Studio

Symbols

- TIA-Import1
  - ProgramBlocksFolder
    - Gruppe_1
      - Datenbaustein_1[%DB2]
      - Datenbaustein_2[%DB3]
    - Gruppe_2
      - Gruppe_1
        - Datenbaustein_5[%DB6]
      - Datenbaustein_3[%DB4]
      - Datenbaustein_4[%DB5]
    - SystemBlocksFolder
      - ProgramResourcesFolder
      - SafetyResourcesFolder
        - SafetyCompilerBlocksFolder
          - DB1_C[%DB30002]
          - FB32769_IDB_C[%DB30003]
          - FB32770_IDB_C[%DB30004]
          - FB32771_IDB_C[%DB30005]
          - FB32772_IDB_C[%DB30006]
        - SafetyIODataBlocksFolder
        - F_SystemInfo_DB[%DB30001]
        - RTG1SysInfo[%DB30000]
    - Main_Safety_RTG1_DB[%DB1]
  - ControllerTagsFolder
    - Standard-Variablentabelle

AS Symbols [Datenbaustein_1[%DB2]]

	Access	Modified	Name	Type	Symbolic Address	Data Type	A
11	☑	□	Typ1_MemberArrayBool_2[2]	Tag	Typ1_MemberArrayBool_2[2]	Binary Tag	C
12	□	□	Typ1_MemberArrayBool_2[3]	Tag	Typ1_MemberArrayBool_2[3]	Binary Tag	C
13	□	□	Typ1_MemberArrayBool_2[4]	Tag	Typ1_MemberArrayBool_2[4]	Binary Tag	C
14	□	□	Typ1_MemberArrayBool_2_1[	Tag	Typ1_MemberArrayBool_2_1|	Binary Tag	C
15	□	□	Typ1_MemberArrayBool_2_1[	Tag	Typ1_MemberArrayBool_2_1|	Binary Tag	C
16	□	□	Typ1_MemberArrayBool_2_1[	Tag	Typ1_MemberArrayBool_2_1|	Binary Tag	C
17	□	□	Typ1_MemberArrayBool_3[2]	Tag	Typ1_MemberArrayBool_3[2]	Binary Tag	C
18	□	□	Typ1_MemberArrayBool_3[3]	Tag	Typ1_MemberArrayBool_3[3]	Binary Tag	C
19	□	□	Typ1_MemberArrayBool_3[4]	Tag	Typ1_MemberArrayBool_3[4]	Binary Tag	C
20	□	□	Typ1_MemberArrayBool_3_1[	Tag	Typ1_MemberArrayBool_3_1|	Binary Tag	C
21	□	□	Typ1_MemberArrayBool_3_1[	Tag	Typ1_MemberArrayBool_3_1|	Binary Tag	C
22	□	□	Typ1_MemberArrayBool_3_1[	Tag	Typ1_MemberArrayBool_3_1|	Binary Tag	C
23	□	□	Typ1_MemberArrayBool_1_2[	Tag	Typ1_MemberArrayBool_1_2|	Binary Tag	C
24	□	□	Typ1_MemberArrayBool_1_2[	Tag	Typ1_MemberArrayBool_1_2|	Binary Tag	C
25	□	□	Typ1_MemberArrayBool_1_2[	Tag	Typ1_MemberArrayBool_1_2|	Binary Tag	C
26	□	□	Typ1_MemberArrayBool_2_2[	Tag	Typ1_MemberArrayBool_2_2|	Binary Tag	C
27	□	□	Typ1_MemberArrayBool_2_2[	Tag	Typ1_MemberArrayBool_2_2|	Binary Tag	C
28	□	□	Typ1_MemberArrayBool_2_2[	Tag	Typ1_MemberArrayBool_2_2|	Binary Tag	C
29	□	□	Typ1_MemberArrayBool_1_2_	Tag	Typ1_MemberArrayBool_1_2	Binary Tag	C
30	□	□	Typ1_MemberArrayBool_1_2_	Tag	Typ1_MemberArrayBool_1_2	Binary Tag	C
31	☑	□	Typ1_MemberArrayBool_1_2_	Tag	Typ1_MemberArrayBool_1_2	Binary Tag	C
32	□	□	Typ1_MemberStruct	Structure member tag	Typ1_MemberStruct		C
33	□	□	Struct_Member1	Member tag	Typ1_MemberStruct.Struct_M	Binary Tag	C
34	□	□	Struct_Member2	Member tag	Typ1_MemberStruct.Struct_M	Binary Tag	C
35	□	□	Struct_Member3	Member tag	Typ1_MemberStruct.Struct_M	Binary Tag	C
36	□	□	Struct_Member4	Member tag	Typ1_MemberStruct.Struct_M	Binary Tag	C
37	□	□	Struct_Member5	Member tag	Typ1_MemberStruct.Struct_M	Binary Tag	C

Properties - AS symbol

Selection	
Object type	AS symbol
Object name	Typ1_MemberArrayBool
AS Properties	
Symbolic Address	Typ1_MemberArrayBool
Name	Typ1_MemberArrayBool
Type	Tag
Name	Typ1_MemberArrayBool
Data Type	Binary Tag
Strukturtyp-Name	
Address	0001:CL:%DB2.%D33.0
AS Data Type	S7PLUS_Bool
Name space	Datenbaustein_1
OS Properties	
Access	☑
Modified	□
Tag	Datenbaustein_1_Typ1_
Data Type	Binary Tag
Length	1
Format adaptation	

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.

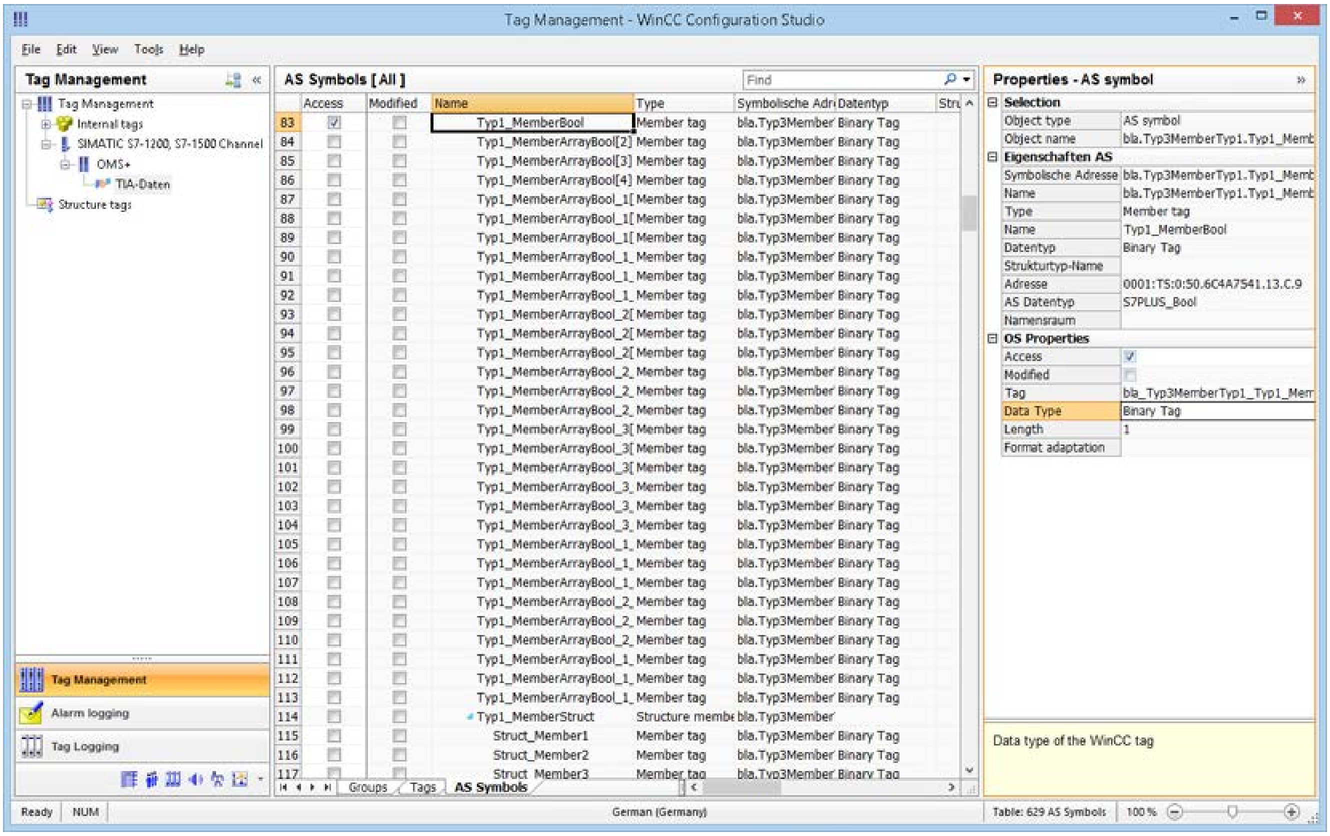
By selecting the check boxes in the "Access" column, you create a WinCC tag from the found AS tags.

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.



	Access	Modified	Name	Type	Symbolische Adr	Datentyp	Stru
83	☑	□	Typ1_MemberBool	Member tag	bla.Typ3Member	Binary Tag	
84	□	□	Typ1_MemberArrayBool[2]	Member tag	bla.Typ3Member	Binary Tag	
85	□	□	Typ1_MemberArrayBool[3]	Member tag	bla.Typ3Member	Binary Tag	
86	□	□	Typ1_MemberArrayBool[4]	Member tag	bla.Typ3Member	Binary Tag	
87	□	□	Typ1_MemberArrayBool_1[	Member tag	bla.Typ3Member	Binary Tag	
88	□	□	Typ1_MemberArrayBool_1[	Member tag	bla.Typ3Member	Binary Tag	
89	□	□	Typ1_MemberArrayBool_1[	Member tag	bla.Typ3Member	Binary Tag	
90	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
91	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
92	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
93	□	□	Typ1_MemberArrayBool_2[	Member tag	bla.Typ3Member	Binary Tag	
94	□	□	Typ1_MemberArrayBool_2[	Member tag	bla.Typ3Member	Binary Tag	
95	□	□	Typ1_MemberArrayBool_2[	Member tag	bla.Typ3Member	Binary Tag	
96	□	□	Typ1_MemberArrayBool_2_	Member tag	bla.Typ3Member	Binary Tag	
97	□	□	Typ1_MemberArrayBool_2_	Member tag	bla.Typ3Member	Binary Tag	
98	□	□	Typ1_MemberArrayBool_2_	Member tag	bla.Typ3Member	Binary Tag	
99	□	□	Typ1_MemberArrayBool_3[	Member tag	bla.Typ3Member	Binary Tag	
100	□	□	Typ1_MemberArrayBool_3[	Member tag	bla.Typ3Member	Binary Tag	
101	□	□	Typ1_MemberArrayBool_3[	Member tag	bla.Typ3Member	Binary Tag	
102	□	□	Typ1_MemberArrayBool_3_	Member tag	bla.Typ3Member	Binary Tag	
103	□	□	Typ1_MemberArrayBool_3_	Member tag	bla.Typ3Member	Binary Tag	
104	□	□	Typ1_MemberArrayBool_3_	Member tag	bla.Typ3Member	Binary Tag	
105	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
106	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
107	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
108	□	□	Typ1_MemberArrayBool_2_	Member tag	bla.Typ3Member	Binary Tag	
109	□	□	Typ1_MemberArrayBool_2_	Member tag	bla.Typ3Member	Binary Tag	
110	□	□	Typ1_MemberArrayBool_2_	Member tag	bla.Typ3Member	Binary Tag	
111	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
112	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
113	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
114	□	□	Typ1_MemberStruct	Structure membe	bla.Typ3Member		
115	□	□	Struct_Member1	Member tag	bla.Typ3Member	Binary Tag	
116	□	□	Struct_Member2	Member tag	bla.Typ3Member	Binary Tag	
117	□	□	Struct_Member3	Member tag	bla.Typ3Member	Binary Tag	

Selection	
Object type	AS symbol
Object name	bla.Typ3MemberTyp1.Typ1_Memb
Eigenschaften AS	
Symbolische Adresse	bla.Typ3MemberTyp1.Typ1_Memb
Name	bla.Typ3MemberTyp1.Typ1_Memb
Type	Member tag
Name	Typ1_MemberBool
Datentyp	Binary Tag
Strukturtyp-Name	
Adresse	0001:TS:0:50.6C4A7541.13.C.9
AS Datentyp	S7PLUS_Bool
Namensraum	
OS Properties	
Access	☑
Modified	□
Tag	bla_Typ3MemberTyp1_Typ1_Mem
Data Type	Binary Tag
Length	1
Format adaptation	

See also

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 415)

How to export AS project data (Page 423)

How to configure AS structures (Page 420)

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)

7.12.4.4 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".

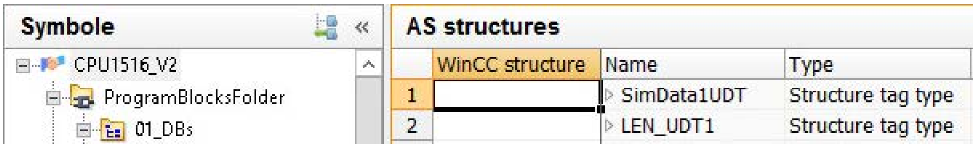
You change the name of the WinCC structure type and the structure type elements in the Tag Management. The assignment of the AS structure is automatically adjusted.

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.

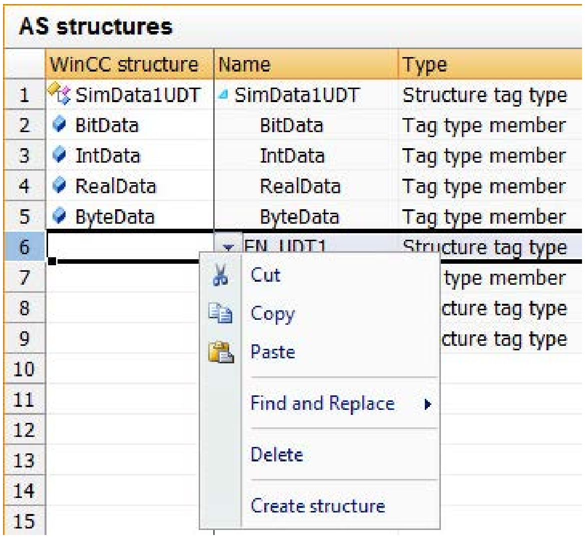


AS structures

	WinCC structure	Name	Type
1		SimData1UDT	Structure tag type
2		LEN_UDT1	Structure tag type

2. Click "AS structures".
   To display the elements of a structure, click the arrow in front of the structure name.

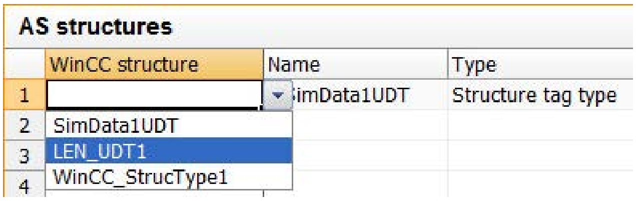
3. Select the entire row of a structure and select the "Create structure" entry in the shortcut menu.



AS structures

	WinCC structure	Name	Type
1	SimData1UDT	SimData1UDT	Structure tag type
2	BitData	BitData	Tag type member
3	IntData	IntData	Tag type member
4	RealData	RealData	Tag type member
5	ByteData	ByteData	Tag type member
6		EN_UDT1	Structure tag type
7			type member
8			cture tag type
9			cture tag type
10			
11			
12			
13			
14			
15			

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".



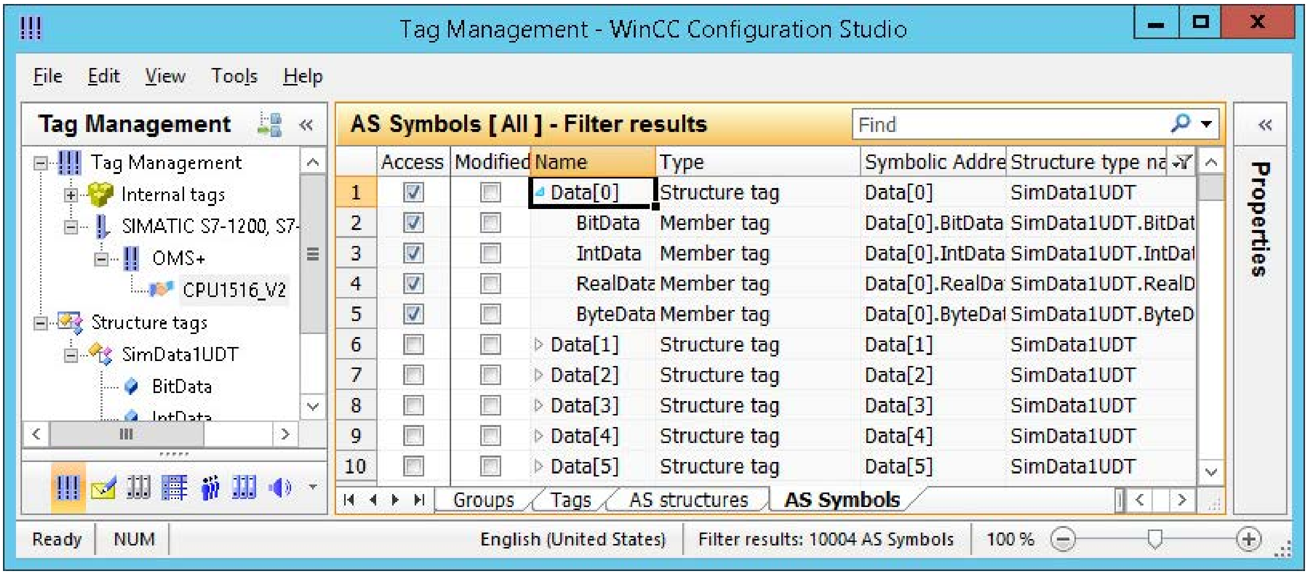
AS structures

	WinCC structure	Name	Type
1		imData1UDT	Structure tag type
2			
3			
4			

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.



	Access	Modified	Name	Type	Symbolic Addre	Structure type na
1	☑	□	Data[0]	Structure tag	Data[0]	SimData1UDT
2	☑	□	BitData	Member tag	Data[0].BitData	SimData1UDT.BitDat
3	☑	□	IntData	Member tag	Data[0].IntData	SimData1UDT.IntDat
4	☑	□	RealDat	Member tag	Data[0].RealDa	SimData1UDT.RealD
5	☑	□	ByteData	Member tag	Data[0].ByteDat	SimData1UDT.ByteD
6	□	□	Data[1]	Structure tag	Data[1]	SimData1UDT
7	□	□	Data[2]	Structure tag	Data[2]	SimData1UDT
8	□	□	Data[3]	Structure tag	Data[3]	SimData1UDT
9	□	□	Data[4]	Structure tag	Data[4]	SimData1UDT
10	□	□	Data[5]	Structure tag	Data[5]	SimData1UDT

Ready | NUM | English (United States) | Filter results: 10004 AS Symbols | 100 %

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 417)

How to export AS project data (Page 423)

7.12.4.5 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 417)

How to configure AS structures (Page 420)

7.12.4.6 Channel units

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 439)

"Soft PLC" channel unit (Page 438)

"Slot PLC" channel unit (Page 436)

Channel Units "PROFIBUS (I + II)" (Page 433)

"Named Connections" channel unit (Page 430)

"MPI" channel unit (Page 427)

Channel Units "Industrial Ethernet" + "Industrial Ethernet (II)" (Page 425)

"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 participants.

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 442)

How to configure a "Industrial Ethernet" channel unit connection (Page 426)

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 communication 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.

Further information regarding the diagnostics of the channel, connection and tags can be found 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 communications link will not be established.

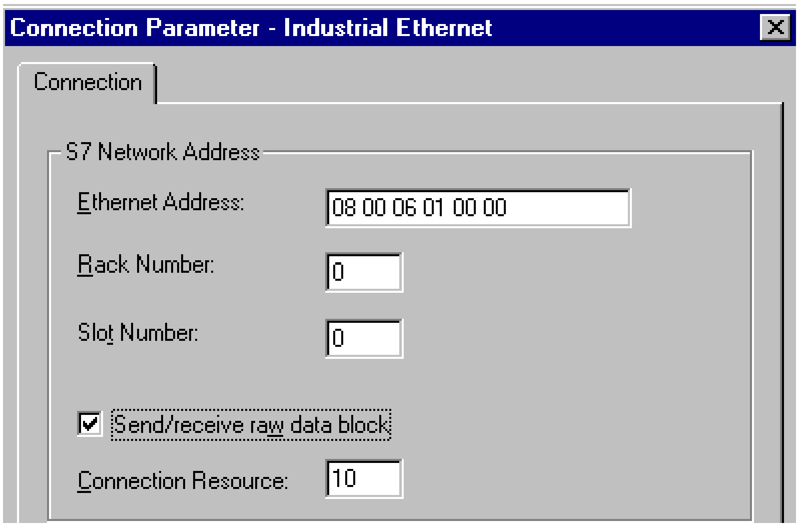
Requirements

- 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.



Ethernet Address: 08 00 06 01 00 00

Rack Number: 0

Slot Number: 0

☑ Send/receive raw data block

Connection Resource: 10

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 corresponding field "Slot Number".
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 442)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 415)

Diagnosis of Channels and Tags (Page 609)

"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 communication module.

Unit-typical terminology

MPI

MPI means Multi Point Interface and is a communication connection in which multiple nodes 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 communication 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 442)

How to configure a "MPI" channel unit connection (Page 428)

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 PLCs either use the internal MPI interface or a communication 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. Alternately, you can also use a communication module.

Further information regarding the diagnostics of the channel, connection and tags can be found 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 incorrect rack or slot number is entered, the communication connection will not be established.

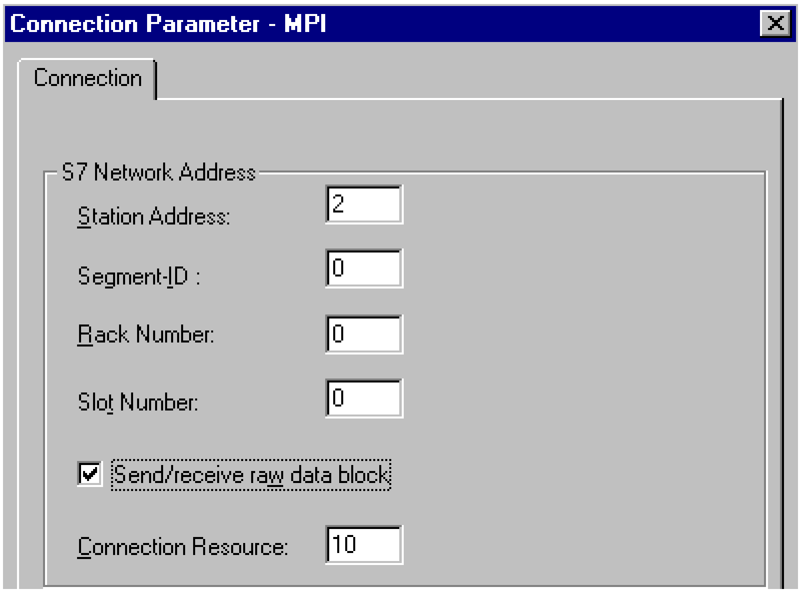
While using a S7-300, for the link via the internal MPI interface of the CPU, the rack/ slot number = 0 must be given.

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.



Connection Parameter - MPI

Connection

S7 Network Address

Station Address: 2

Segment-ID : 0

Rack Number: 0

Slot Number: 0

☑ Send/receive raw data block

Connection Resource: 10

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 442)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 415)

"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 442)

How to configure a "Named Connections" channel unit connection (Page 431)

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:

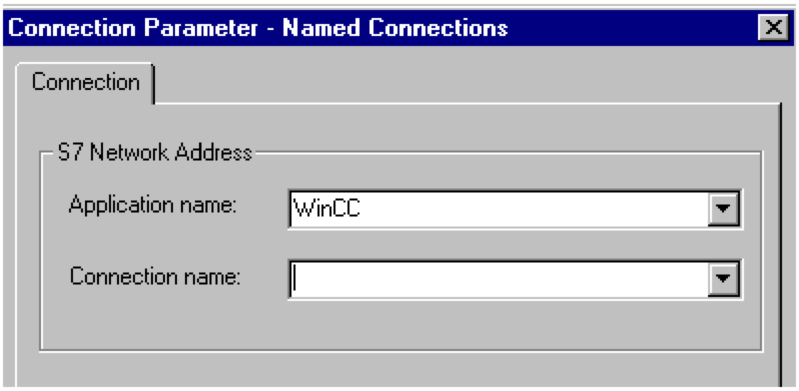
- No XDB file is available for the symbolic connection name. In this case transfer the configuration directly to the "Components configurator".
- The project is to be transferred to another computer.

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.



Application name: WinCC
Connection name:

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 442)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 415)

"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 the S7 automation systems, a communication 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 nodes. 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 442)

How to configure a "PROFIBUS" channel unit connection (Page 433)

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.

Further information regarding the diagnostics of the channel, connection and tags can be found 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 communications link will not be established.

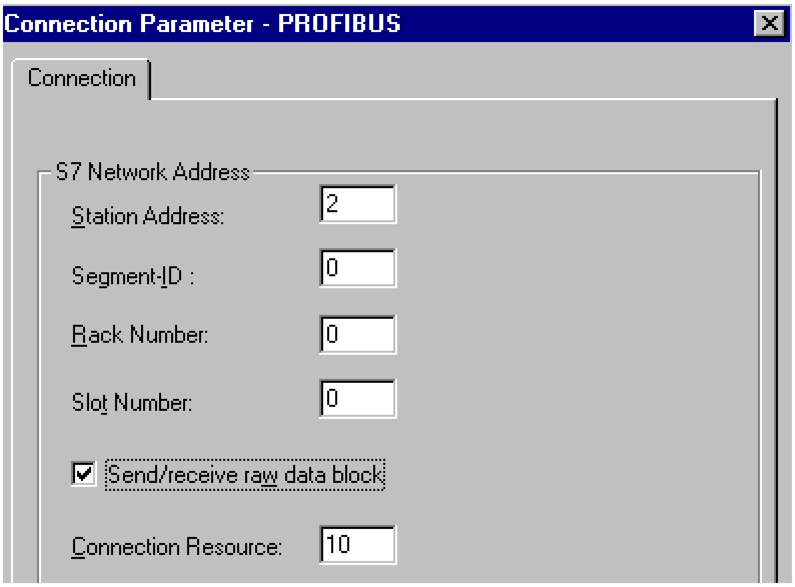
Requirements

- 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.



Connection Parameter - PROFIBUS

Connection

S7 Network Address

Station Address: 2

Segment-ID : 0

Rack Number: 0

Slot Number: 0

☑ Send/receive raw data block

Connection Resource: 10

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 442)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 415)

"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 442)

How to Configure a "Slot PLC" Channel Unit Connection (Page 436)

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".

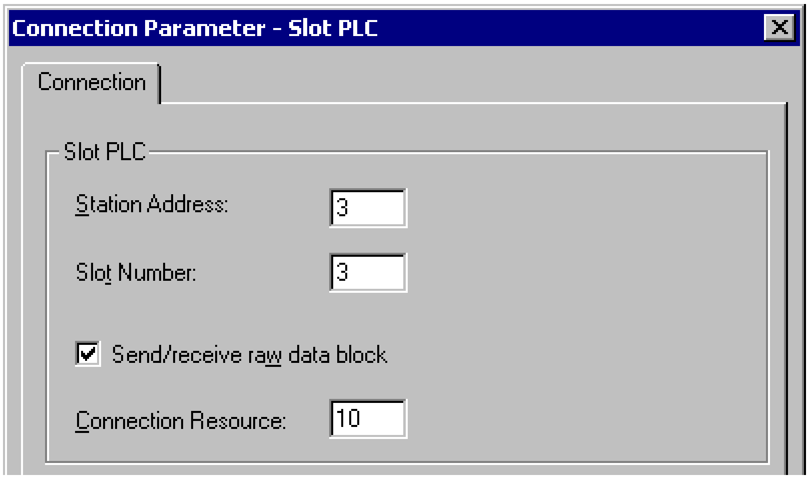
Requirements

- 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.



Station Address: 3

Slot Number: 3

☑ Send/receive raw data block

Connection Resource: 10

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 442)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 415)

"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 442)

How to configure a connection on the "Soft PLC" channel unit (Page 438)

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".

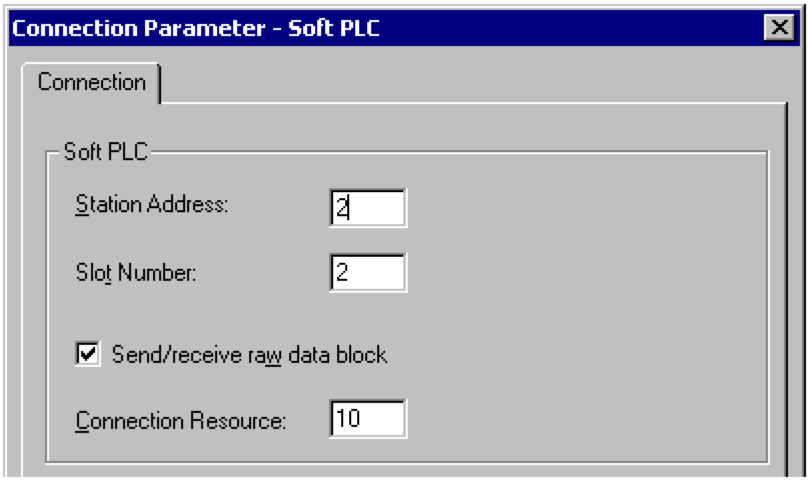
Requirements

- 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.



Connection Parameter - Soft PLC

Connection

Soft PLC

Station Address: 2

Slot Number: 2

☑ Send/receive raw data block

Connection Resource: 10

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 442)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 415)

"TCP/IP" channel unit

"TCP/IP" channel unit

Principle of operation

The channel unit "TCP/IP" is used to connect WinCC to the automation systems SIMATIC S7-300 and S7-400 via an Industrial Ethernet with protocol "ISO-on-TCP Transport".

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 automation system S7-300, 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 participants.

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 442)

How to configure a "TCP/IP" channel unit connection (Page 440)

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.

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.

Further information regarding the diagnostics of the channel, connection and tags can be found 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.

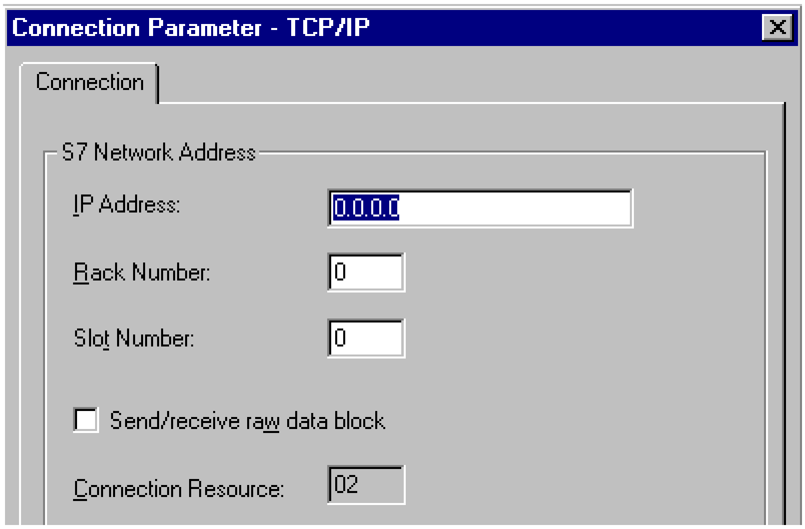
If the wrong Rack or Slot Number is entered, the communications link will not be established.

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.



Connection Parameter - TCP/IP

Connection

S7 Network Address

IP Address: 0.0.0.0

Rack Number: 0

Slot Number: 0

☐ Send/receive raw data block

Connection Resource: 02

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 corresponding field "Slot Number".

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 442)

How to configure the "SIMATIC S7 Protocol Suite" channel (Page 415)

7.12.4.7 Configuring the tags

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 446)

How to Configure a Tag with Word by Word Access (Page 445)

How to Configure a Tag with Byte by Byte Access (Page 443)

How to Configure a Tag with Bit by Bit Access (Page 442)

How to Configure a Tag with Bit by Bit Access

Introduction

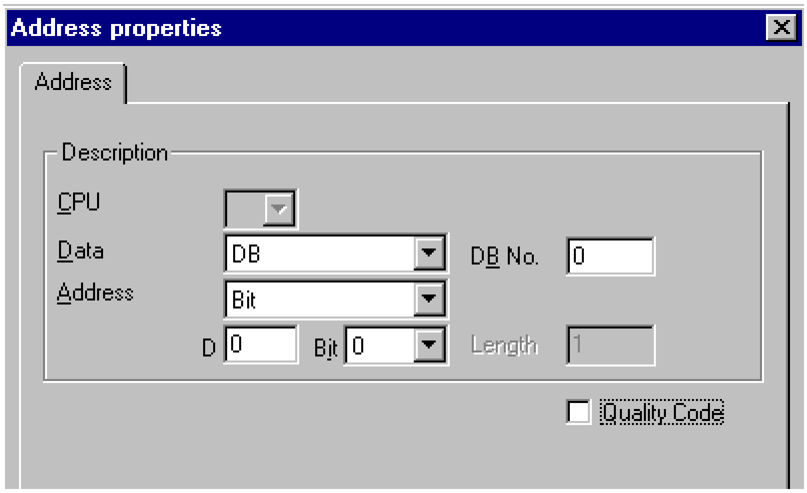
This section shows you how to configure a tag for bit by bit access for the address area in PLC.

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.



Address properties

Address

Description

CPU:

Data: DB

DB No.: 0

Address: Bit

D: 0

Bit: 0

Length: 1

□ Quality Code

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 415)

How to Configure a Tag with Byte by Byte Access

Introduction

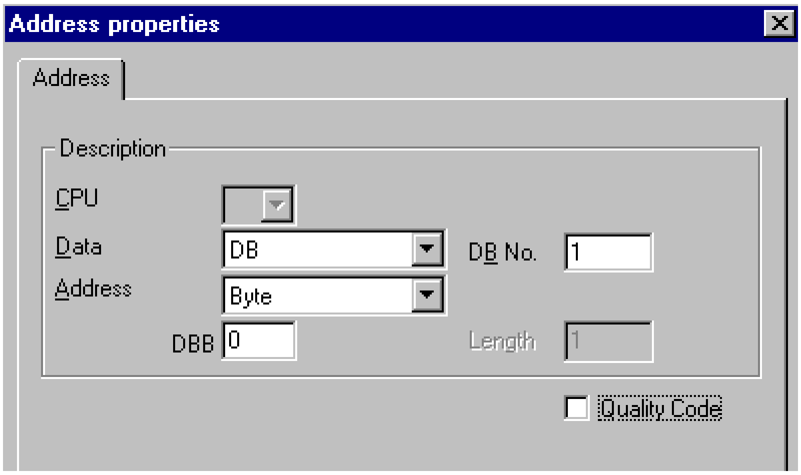
This section shows you how to configure a tag for byte by byte access for the address area in PLC.

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.



CPU
Data: DB
DB No.: 1
Address: Byte
DBB: 0
Length: 1
□ Quality Code

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 415)

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.

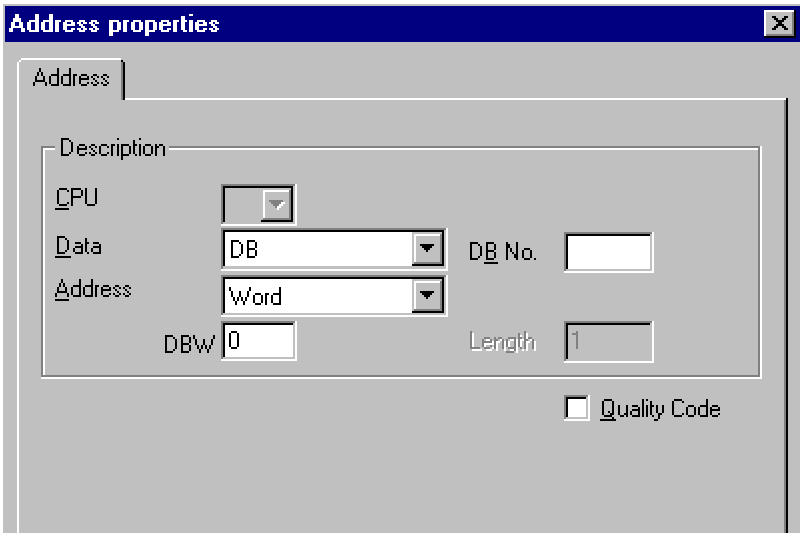
This procedure is also applicable for tags with length of 4 byte ("double word") and more.

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.



Address properties

Address

Description

CPU:

Data: DB

DB No.:

Address: Word

DBW: 0

Length: 1

☐ Quality Code

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 415)

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.

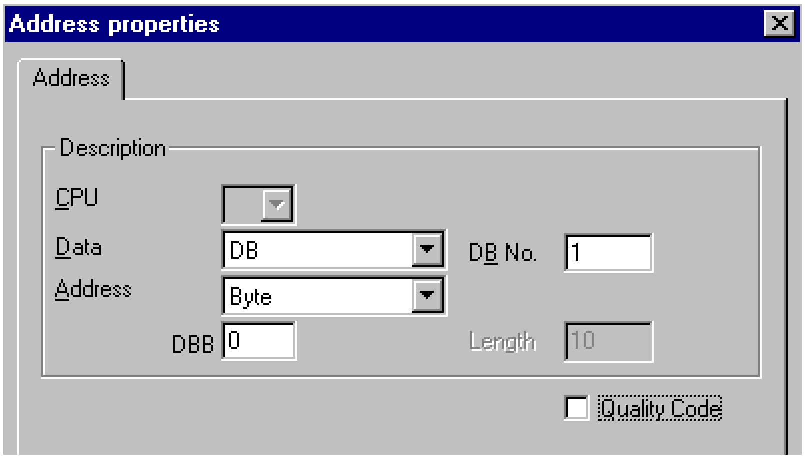
Requirements

- 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.



CPU:
Data: DB
DB No.: 1
Address: Byte
DBB: 0
Length: 10
□ Quality Code

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 415)

7.12.4.8 System parameters

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 451)

How to Configure the System Parameters (Page 449)

Cyclic read services in PLC (Page 448)

Cyclic read services in PLC

Introduction

In the system parameters of the "SIMATIC S7 Protocol Suite" channel, it is also 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 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 requested data is no longer required, e.g. in the case of a screen change, WinCC will delete the cyclic read service in the PLC.

In normal cases, use should be made of the cyclic read services in the PLC. 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.

Modification transfers can only be used when the cyclic read services are activated. The data will then only transferred from the AS when a value has changed and only once per AS cycle. The function must be supported by the PLC.

The use of the cyclic read services and modification 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 PLC. The PLC only sends the required data once. The formation of the cycle for the request is carried out by WinCC.

The 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-PLC. A maximum of four cyclic services are available for an S7-300 max. and a maximum of 32 for an S7-416 or 417. This number applies for all participants communicating with the PLC, i.e. if several WinCC systems are communicating with an S7-PLC, they will have to share the resources that are available. If the maximum number of resources 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 execute cycle formation.

Requesting external tags in scripts

The utilization of the cyclic read service has no influence on the initial update once a picture has been opened if the picture that has been selected does not contain any scripts that request external tags using the function "GetTagWord()". If scripts are executed with "GetTagWord()" when a picture is opened, the incorrect configuration of this script could result in new tag requests being sent to this channel repeatedly following a picture change. If external tags are required in a script, "Tag" should be entered as a trigger event.

How to Configure the System Parameters

Introduction

In this section, we will show you how to configure the system parameters of the Channel "SIMATIC S7 Protocol Suite".

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 channel unit "MPI" is used in all examples.

Any changes that are made to the parameter values will only take effect after WinCC has been restarted.

Note

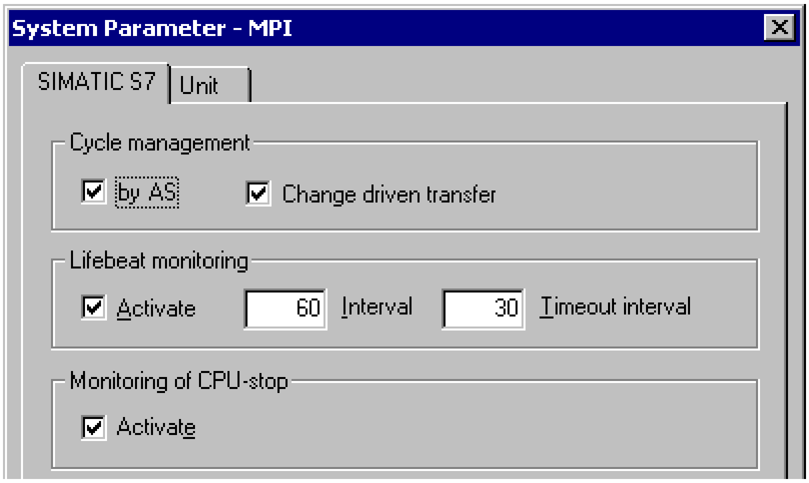
When copying the project to another computer, the settings on the Unit tab will be retained, the settings on the SIMATIC S7 tab, however, will not.

Requirements

- The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.

Procedure

1. Select the Channel "SIMATIC S7 Protocol Suite" in Tag Management. Open the dialog "System Parameters" using the pop-up menu of the desired channel unit.
2. Select the SIMATIC S7 tab. Place a tick in the check boxes "by AS" and "with modification transfer", if you want to activate cyclic reading of tags by the channel and the use of modification transfers. If available, the cyclic services in the PLC will be used here. Further information can be found under "PLC Cyclic Read Services".



System Parameter - MPI

Cycle management

☑ by AS ☑ Change driven transfer

Lifebeat monitoring

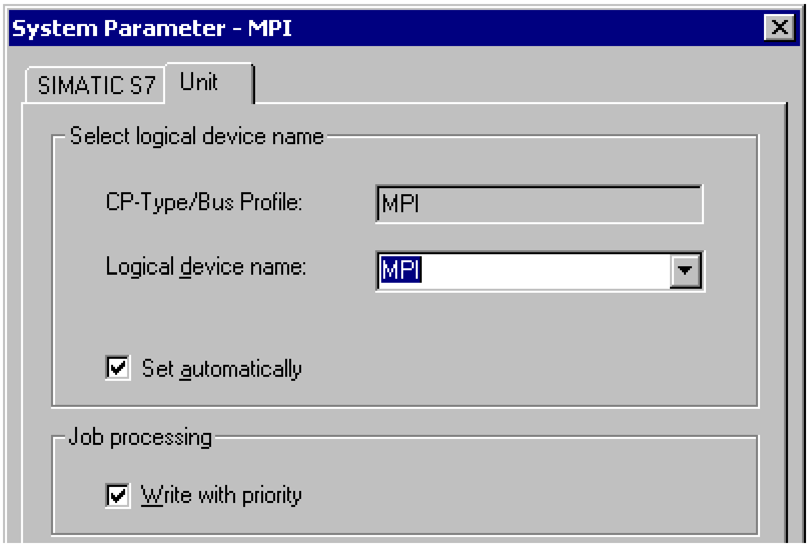
☑ Activate 60 Interval 30 Timeout interval

Monitoring of CPU-stop

☑ Activate

3. Activate the check box "Enable" in the "Lifebeat Monitoring" area if you wish to use this function. In the Interval field enter the time interval in seconds for transferring the lifebeat telegrams.
   In the Monitoring Time field enter the seconds value for monitoring the response to a lifebeat telegram.
4. If WinCC should indicate that communication is faulty when the S7-CPU is in the Stop status, activate the check box "Enable" in the "CPU Stop Monitoring" area.

5. Select the Unit tab. A name, which will depend on the communications processor installed, will be displayed in the field "Logical Device Name". You should only change this name if you have selected a different name when installing the communications processor. Further information can be found under "Changing Logical Device Names".



System Parameter - MPI

SIMATIC S7 | Unit

Select logical device name

CP-Type/Bus Profile: MPI

Logical device name: MPI

☑ Set automatically

Job processing

☑ Write with priority

6. If only one communications processor has been installed for this communication type, activate the check box "Set Automatically", if the device name should be set automatically when Runtime is started.
7. Activate the check box "Write with Priority", if the processing of write requests should take priority over the processing of read requests.
8. Close the dialog with the "OK" button.

See also

How to Change the Logical Device Name (Page 451)

Cyclic read services in PLC (Page 448)

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 presettings 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.

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.
   All possible names will be 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.
   Manual entries should only be made if the target station uses a communications card which is not installed on the configuring station.
5. Close the dialog by clicking the "OK" button.

Note

Logical devices 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.

7.12.5 Special functions

7.12.5.1 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 492)

Raw data tags of the Channel "SIMATIC S7 Protocol Suite" (Page 484)

Data exchange with the S7 function block AR_SEND (Page 453)

7.12.5.2 Data exchange with the S7 function block AR_SEND

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 466)

How to configure the AR _SEND variant for multiple archive tags (Page 482)

How to configure the AR _SEND variant for an archive tag (Page 480)

AR_SEND variant for multiple archive tags (optimized) (Page 478)

AR_SEND variant for an archive tag (Page 462)

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"

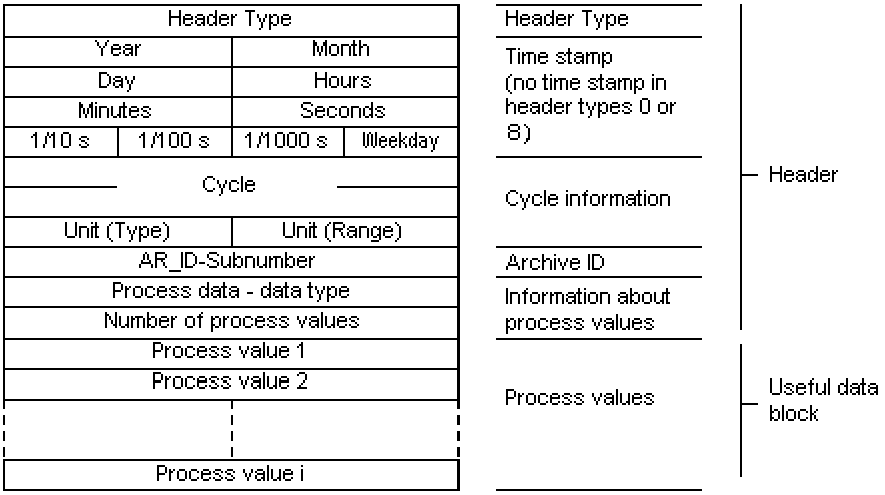
If the message system has been configured with the WinCC System Messages, this diagnostics entry will also trigger the OS process control message numbered 1003080. The text of the logbook entry is contained in this message's comment.

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 482)

How to configure the AR _SEND variant for an archive tag (Page 480)

AR_SEND variant for multiple archive tags (optimized) (Page 478)

AR_SEND variant for multiple archive tags (Page 466)

AR_SEND variant for an archive tag (Page 462)

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 (Timestampt 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 (Timestampt 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 Subnumber	9	1	Relevant	0	4	0	0	1; 2 3; 4; 5	3992 2660	1 Word Subnumber + 1 Word process value

See also

AR_SEND variant for multiple archive tags (optimized) (Page 478)

AR_SEND variant for multiple archive tags (Page 466)

AR_SEND variant for an archive tag (Page 462)

AR_SEND variant for an archive tag

Introduction

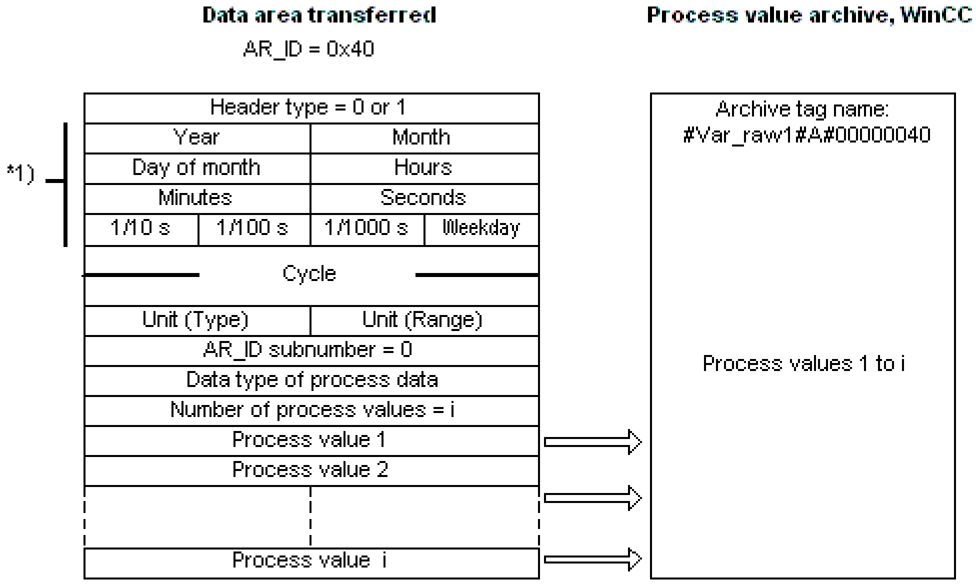
This variant can be used to supply an archive tag with process values. It can also be used with older versions of WinCC (prior to V5.0).

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 465)

Example 3 for data block structure: An archive tag; each process value has its own time stamp (Page 464)

Example 2 for data block structure: One archive tag; equally spaced time stamp (Page 464)

Example 1 for data block structure: An archive tag; each process value has a time stamp (Page 463)

Overview of the properties of the AR_SEND variants (Page 459)

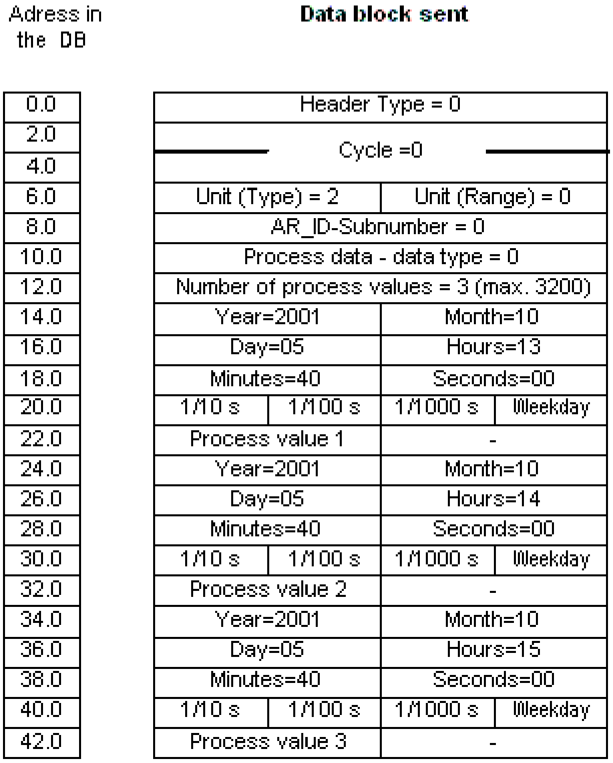
Example 1 for data block structure: An archive tag; each process value has a time stamp

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 454)

Example 2 for data block structure: One archive tag; equally spaced time stamp

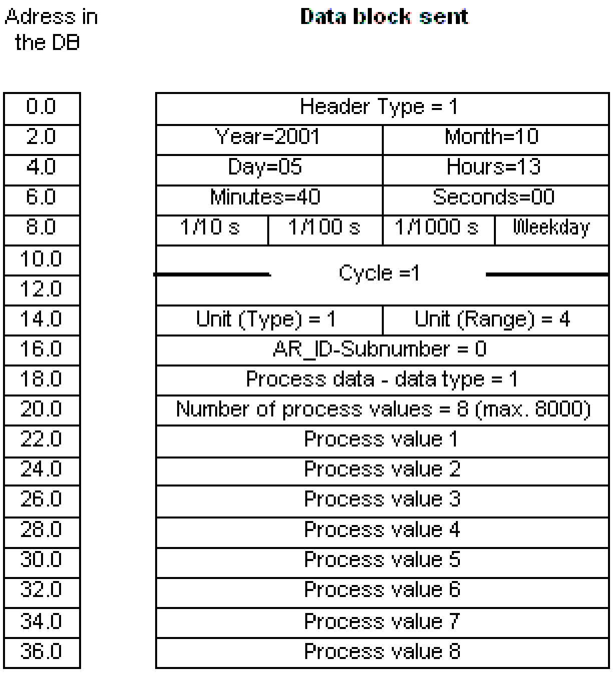
Introduction

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 454)

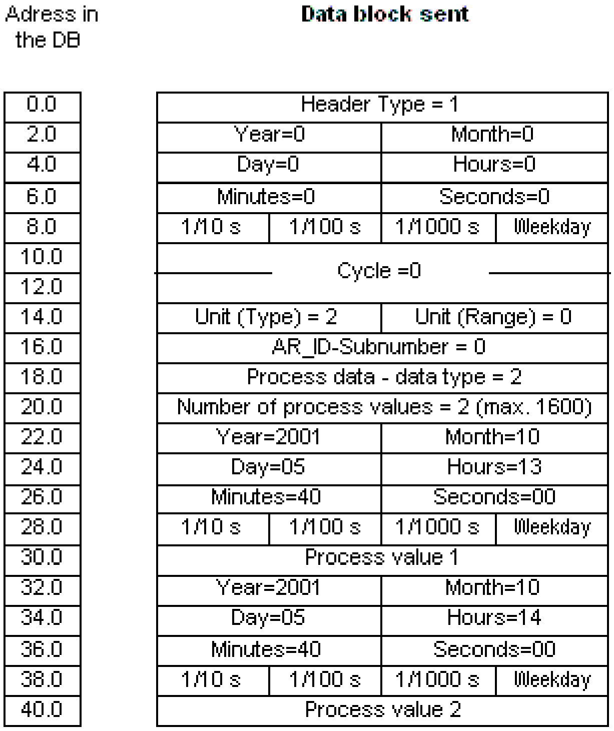
Example 3 for data block structure: An archive tag; each process value has its own time stamp

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 454)

Example 4 for data block structure: An archive tag; each process value with relative time stamp (time difference)

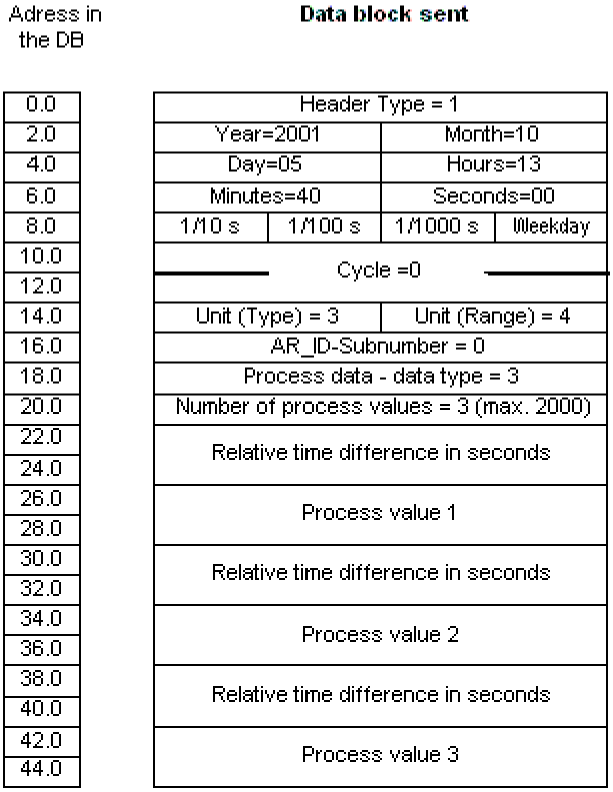
Introduction

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 454)

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.

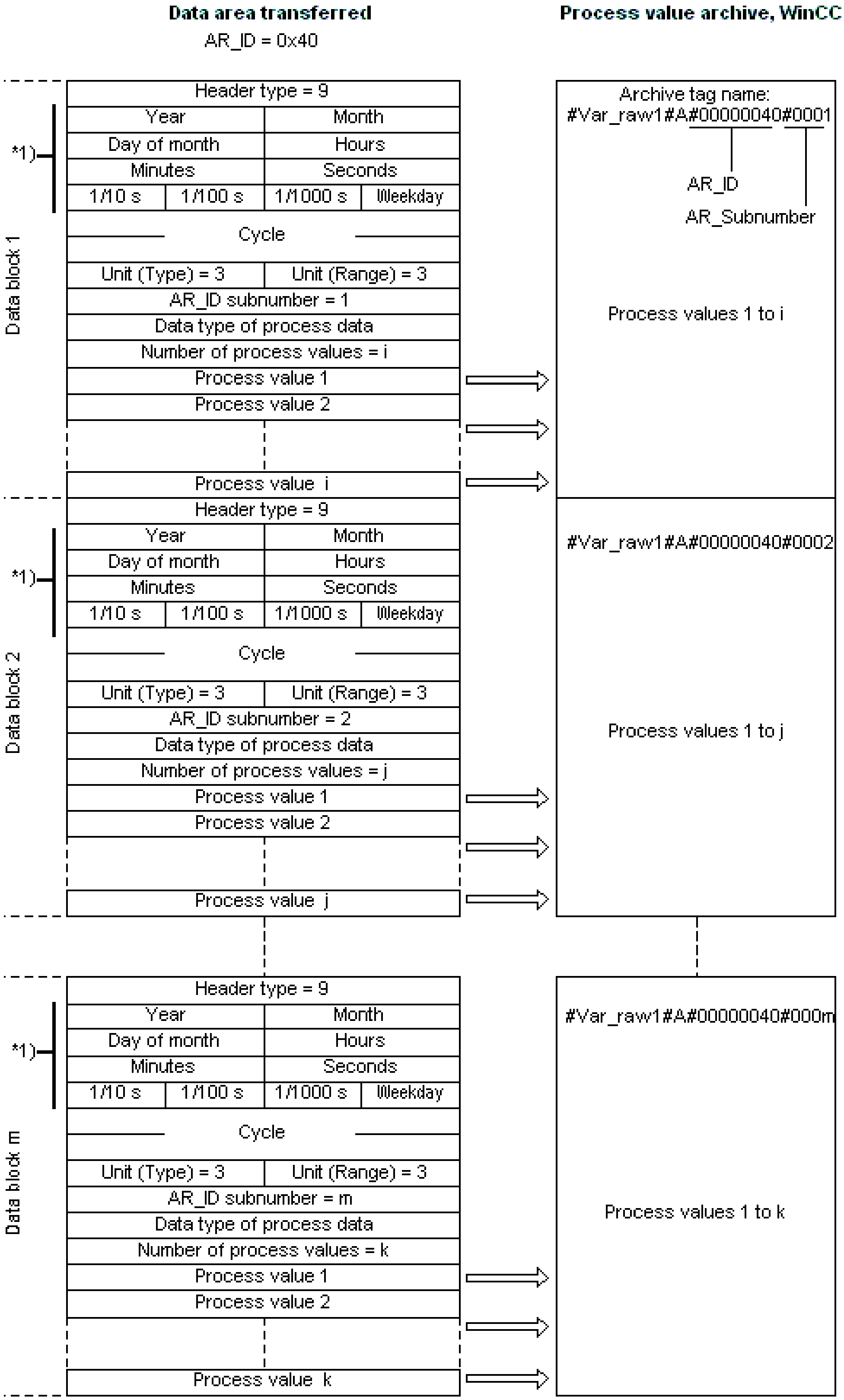
Note

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 476)

Example 7 for data block structure: Multiple archive tags; each process value has its own time stamp (Page 474)

Example 6 for data block structure: Multiple archive tags; equally spaced time stamp (Page 472)

Example 5 for data block structure: Multiple archive tags; each process value has its own time stamp (Page 470)

Overview of the properties of the AR_SEND variants (Page 459)

Example 5 for data block structure: Multiple archive tags; each process value has its own time stamp

Introduction

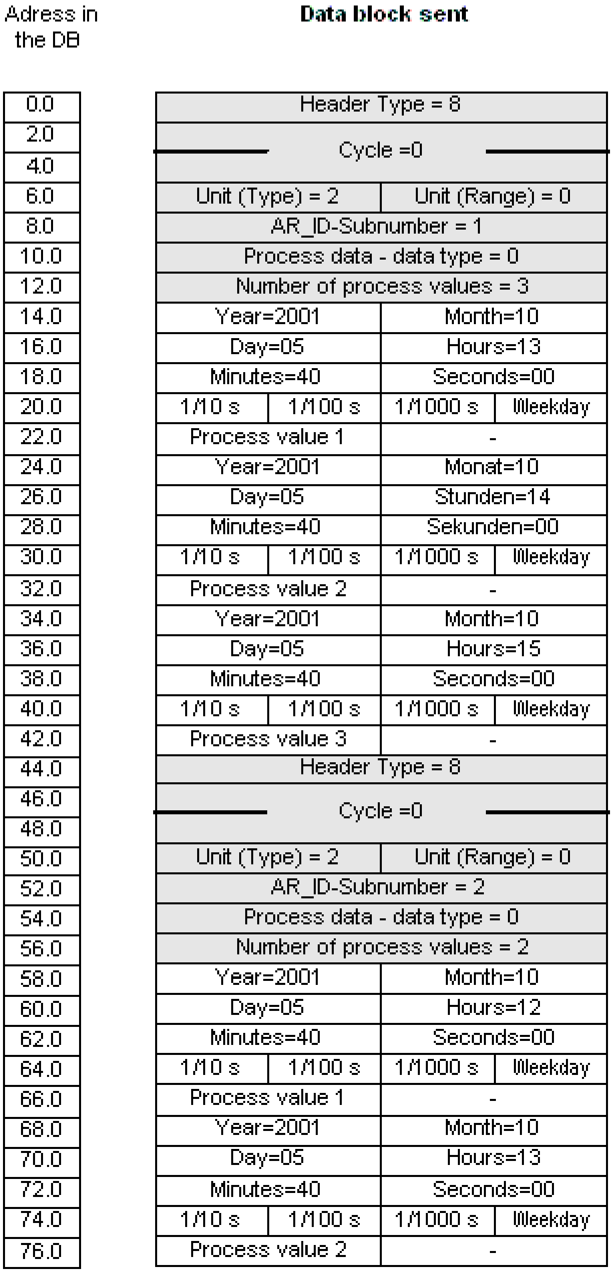
In this example, process values are transferred for multiple archive tags.

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 454)

Example 6 for data block structure: Multiple archive tags; equally spaced time stamp

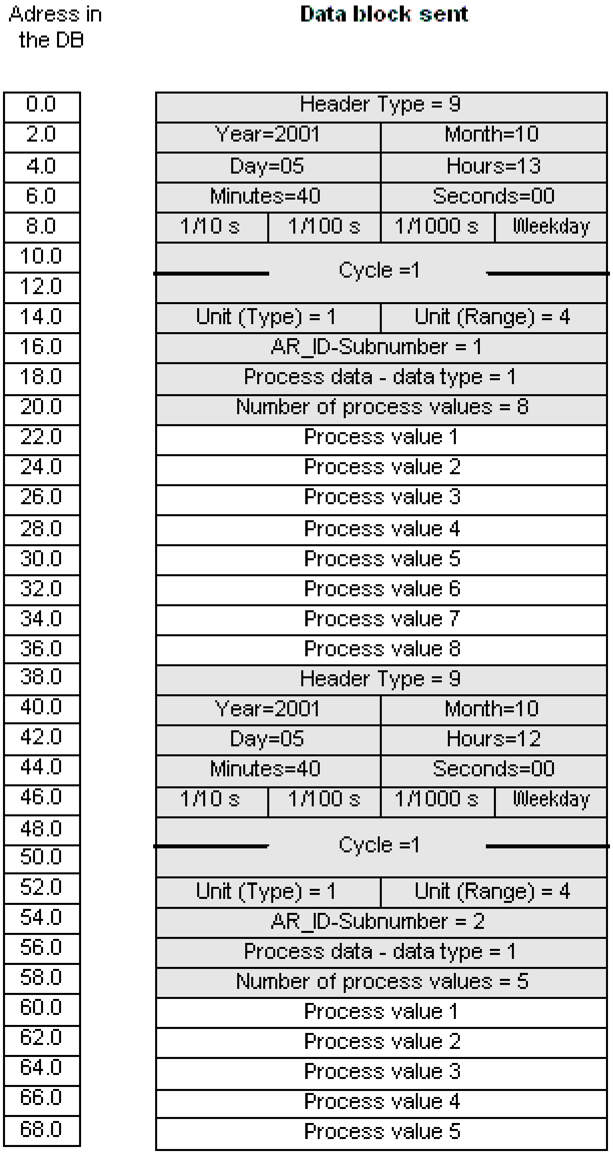
Introduction

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 454)

Example 7 for data block structure: Multiple archive tags; each process value has its own time stamp

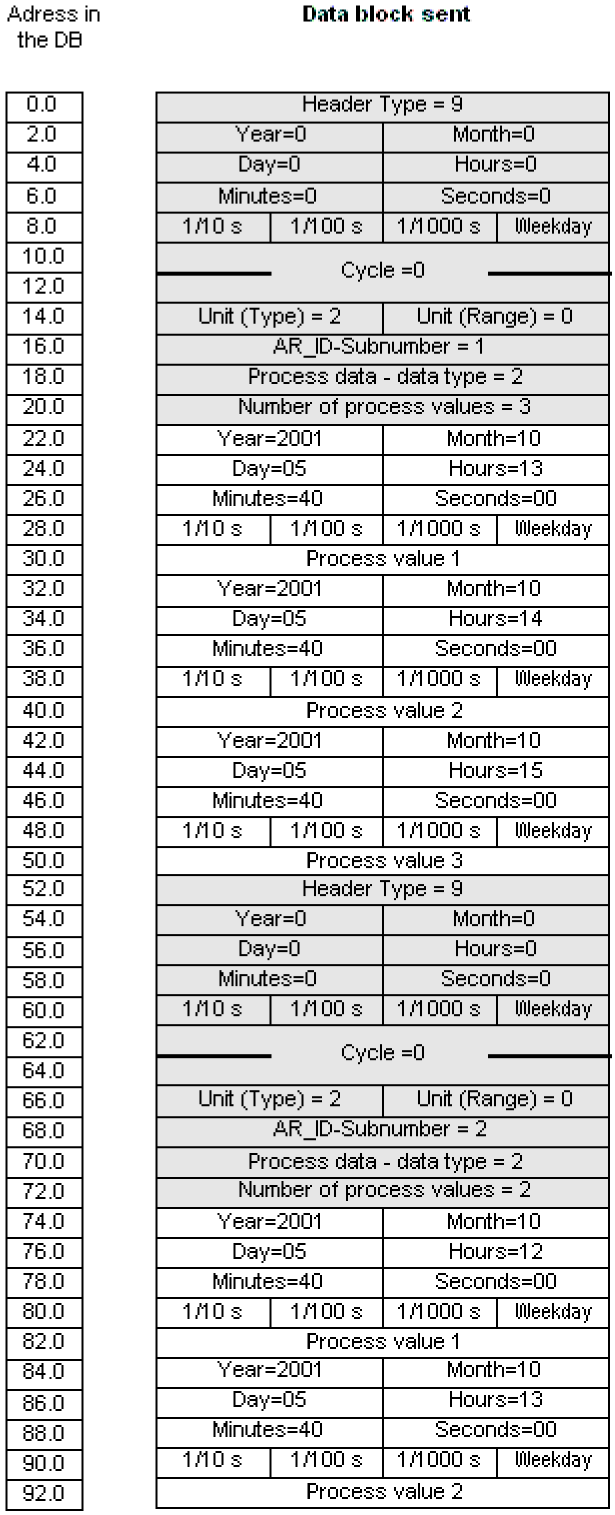
Introduction

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 454)

Example 8 for data block structure: Multiple archive tags; process values with relative time stamp (time difference)

Introduction

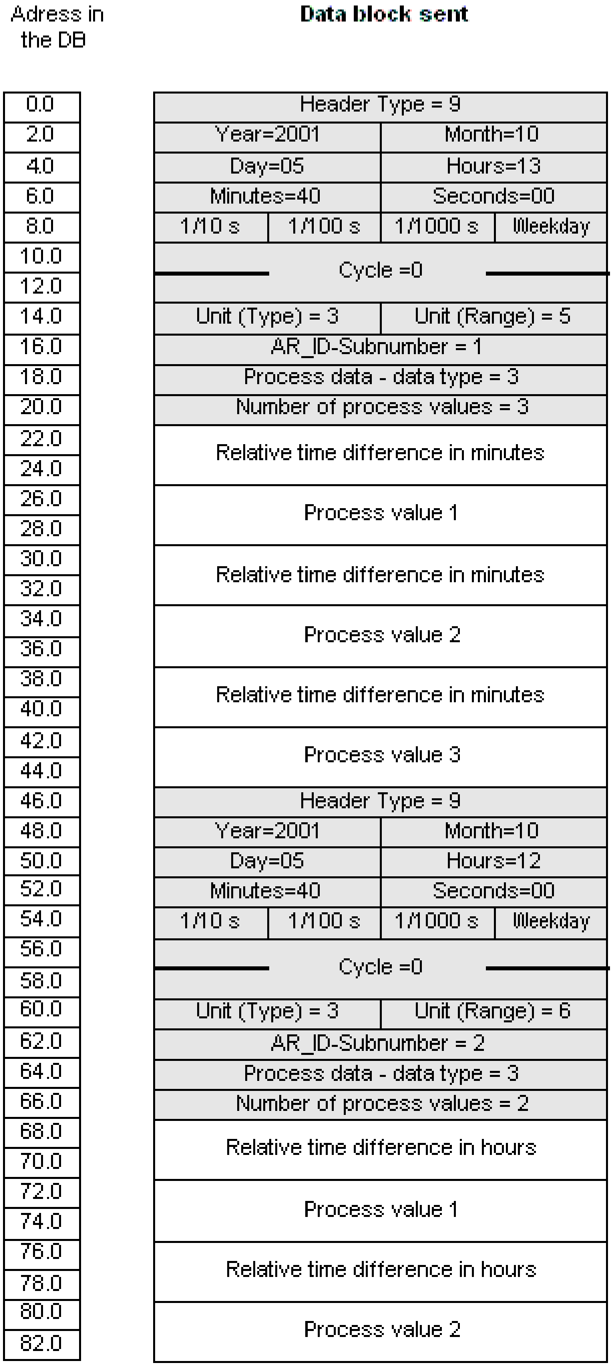
In this example, the process values are transferred for one archive tag with time stamp.

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 454)

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:

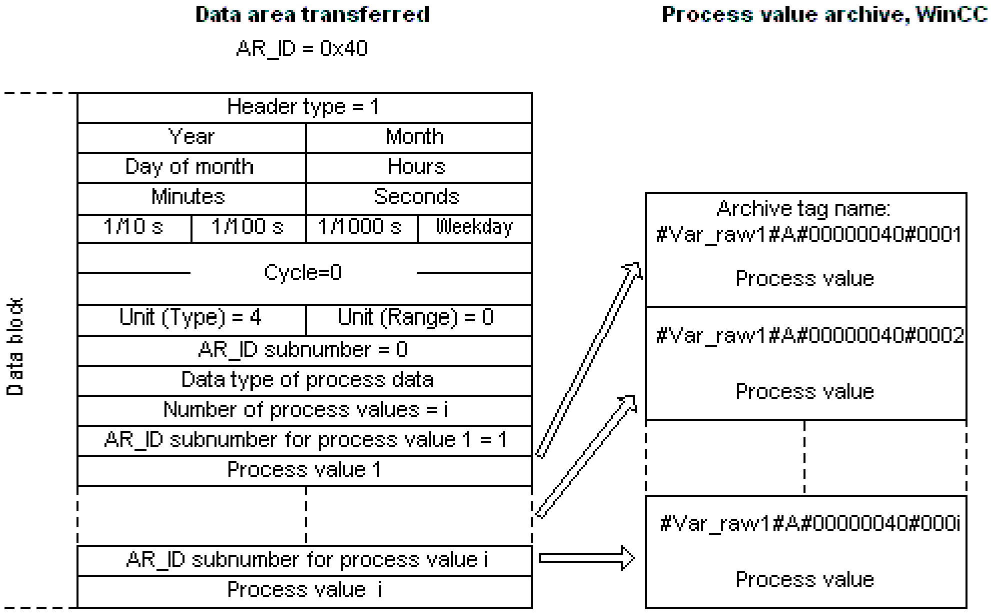
- The Header Type must be 1 (with time stamp and without AR_ID Subnumber).
- The AR_ID Subnumbers for the associated process values in the data block must be greater than zero. The AR_ID Subnumber in the header will not be evaluated.
- The "Unit (Type)" parameter must be 4, i.e. the process value has 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 459)

Example 9 for data block structure: multiple archive tags;optimized (Page 479)

Example 9 for data block structure: multiple archive tags;optimized

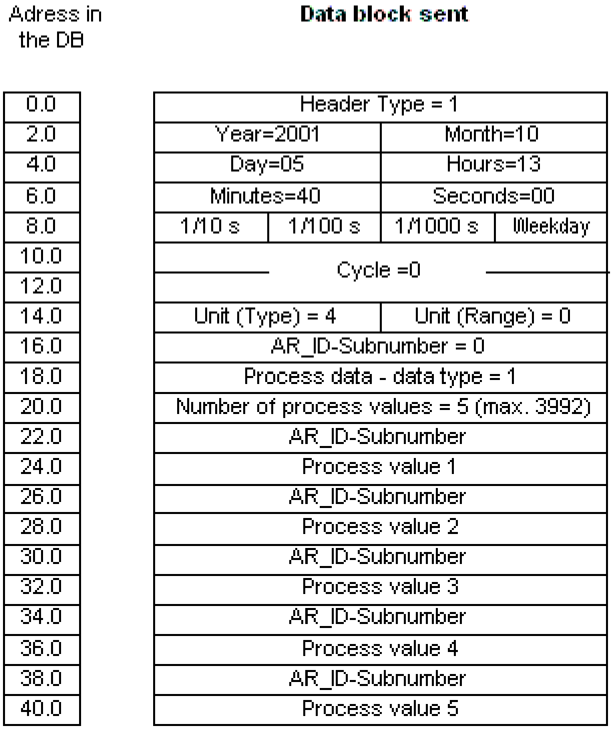
Introduction

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 454)

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. For the "One Archive Tag" variant, only the AR_ID is used. The AR_ID Subnumber is not used.

The AR_ID is used to establish the assignment of the data in the AS to the archive tags and is configured together with other parameters while configuring the data blocks and the SFB 37 "AR_SEND" function block in the AS.

In WinCC, this allocation is performed in the properties for the process controlled tag. This assignment is the only configuration that is necessary in WinCC and will be described in this section.

The other parameters in WinCC need not be configured, since they are evaluated automatically.

Note

To use this variant with just the AR_ID, the Header Type must be configure as 0 or 1. The AR_ID Subnumber must be set to zero.

Since they do not have an AR_ID Subnumber, all of the archive tags, which were configured before WinCC Version V5.0, can be used with this variant.

Since WinCC Version 5.1 Hotfix 4 it has been possible to specify an alias for the archive tag name with process-controlled tags or to use the internal names generated by the system. The name generated by the system contains the name of the assigned raw data tag instead of the raw data ID from Version V5.1 HF4. In projects migrated to a version from V5.1 HF4, the archive tag names can be used in their original format or can be converted. The names are converted by once opening and closing the properties dialog box of the process-controlled archive tags. An alias does not have to be assigned.

If in a project all external tags are remapped using the "AS-OS-Transfer" function, the archive tag name must therefore be converted once to the new structure! The new structure is then maintained.

Requirements

- The "AR_SEND" function block and the data block structure must first be configured in the AS and this configuration information must be available during the following procedure.
- 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".
- A process value archive must be configured in the "Tag Logging" editor.

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 "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.
4. Select "Raw Data Type" in the "Data type" field.
5. Open the dialog "Address properties".
   For this purpose, click in the "Address" field and then on the ... button.
6. Select the "Raw Data" check box. In the "Raw Data Type" area select the type "Archive Data Link". Click "OK" to close all open dialogs.

7. Open the "Tag Logging" editor. Select the "Process Value Archives" folder in the navigation area of the "Tag Logging" editor. In the table area, go to the "Process-controlled tags" tab and add the raw data tag "Var_raw_arsend".
8. Edit the properties in the "Properties" area.
9. In the "Conversion DLL" field, select the entry "nrms7pmc.nll".
10. Enter the AR_ID as a hexadecimal value in the "Block ID". The value is defined through the configuration in the AS.
    Do not enter anything for "Subnumber" because no subnumber is used in this AR_SEND variant.
11. "Tag Name" shows the internal archive tag name generated by the system. It contains the name of the assigned raw data tag and the AR_ID. In "Archive Tag Name", you can define an alias for this archive tag, if required. If no alias is entered, the internal archive tag name is used for management in the process value archive and for addressing the archive tag in WinCC.
12. Close Tag Logging.

How to configure the AR _SEND variant for multiple archive tags

Introduction

There are a number of variants for using the AR_SEND function to transfer data for multiple archive tags.

- Use the "Multiple Archive Tags" variant to supply multiple values to multiple archive tags at various times.
- Use the "Multiple Archive Tags - optimized" variant to supply one value each to the maximum number of archive tags at one time.

The AR_ID and AR_ID Subnumber are used in both of these variants.

AR_ID and AR_ID-Subnumber establish the assignment between the data in the AS and the archive tag. They are defined in the AS with other parameters when configuring the data blocks and the function module SFB 37 "AR_SEND".

In WinCC, this allocation is performed in the properties for the process controlled tag. This assignment is the only configuration that is necessary in WinCC and will be described in this section.

The other parameters in WinCC need not be configured, since they are evaluated automatically.

Note

To use the AR_ID Subnumber, the Header Type must be configured as 8 or 9.

Archive tags configured in WinCC Version V5.0 have no AR_ID Subnumber and can therefore only be used in the "One Tag" variant.

Since WinCC Version 5.1 Hotfix 4 it has been possible to specify an alias for the archive tag name with process-controlled tags or to use the internal names generated by the system. The name generated by the system contains the name of the assigned raw data tag instead of the raw data ID from Version V5.1 HF4. In projects migrated to a version from V5.1 HF4, the archive tag names can be used in their original format or can be converted. The names are converted by once opening and closing the properties dialog box of the process-controlled archive tags. An alias does not have to be assigned.

If in a project all external tags are remapped using the "AS-OS-Transfer" function, the archive tag name must therefore be converted once to the new structure! The new structure is then maintained.

Requirements

- The "AR_SEND" function block and the data block structure must first be configured in the AS and this configuration information must be available during the following procedure.
- 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".
- A process value archive must be configured in the "Tag Logging" editor.

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 "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.
4. Select "Raw Data Type" in the "Data type" field.
5. In the channel "SIMATIC S7 Protocol Suite", select the connection which should be used for the data transfer. Choose the "New Tag" option from the shortcut menu for the connection. The "Tag Properties" dialog opens.
6. Enter "Var_raw_arsend" as the name of the tag in the "Name" field. 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.

7. Open the dialog "Address properties".
   For this purpose, click in the "Address" field and then on the ... button.
8. Select the "Raw Data" check box. In the "Raw Data Type" area select the type "Archive Data Link". Click "OK" to close all open dialogs.
9. Open the "Tag Logging" editor. Select the "Process Value Archives" folder in the navigation area of the "Tag Logging" editor. In the table area, go to the "Process-controlled tags" tab and add the raw data tag "Var_raw_arsend".
10. Edit the properties in the "Properties" area.
11. In the "Conversion DLL" field, select the entry "nrms7pmc.nll". Enter the AR_ID as a hexadecimal value in the "Block ID". The value is defined through the configuration in AS. For "Subnumber", enter the AR_ID subnumber as hexadecimal value. The value is also specified by the configuration in AS.
12. "Tag Name" shows the internal archive tag name generated by the system. It contains the name of the assigned raw data tag and the AR_ID. In "Archive Tag Name", you can define an alias for this archive tag, if required. If no alias is entered, the internal archive tag name is used for management in the process value archive and for addressing the archive tag in WinCC.
13. Close Tag Logging.

7.12.5.3 Raw data tags of the Channel "SIMATIC S7 Protocol Suite"

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 488)

Raw data tag as byte array (Page 485)

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).

The raw data length is limited to one transferable data block and must be fully transferable uisng 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:

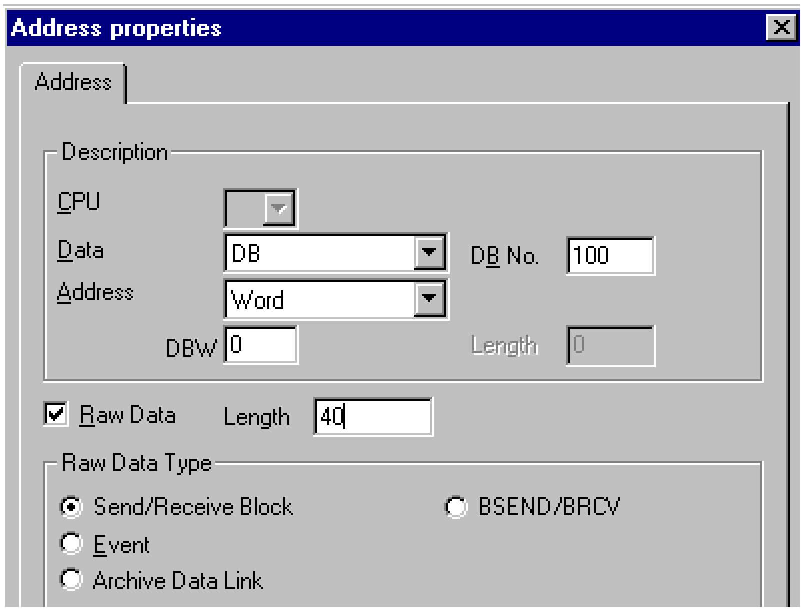
- S7-300: PDU length 240 Byte, max. data block length 208 Byte
- S7-400: PDU length 480 Byte, max. data block length 448 Byte

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:



Address properties

Address

Description

CPU

Data: DB

DB No.: 100

Address: Word

DBW: 0

Length: 0

☑ Raw Data Length: 40

Raw Data Type

● Send/Receive Block

○ BSEND/BRCV

○ Event

○ Archive Data Link

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 486)

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.

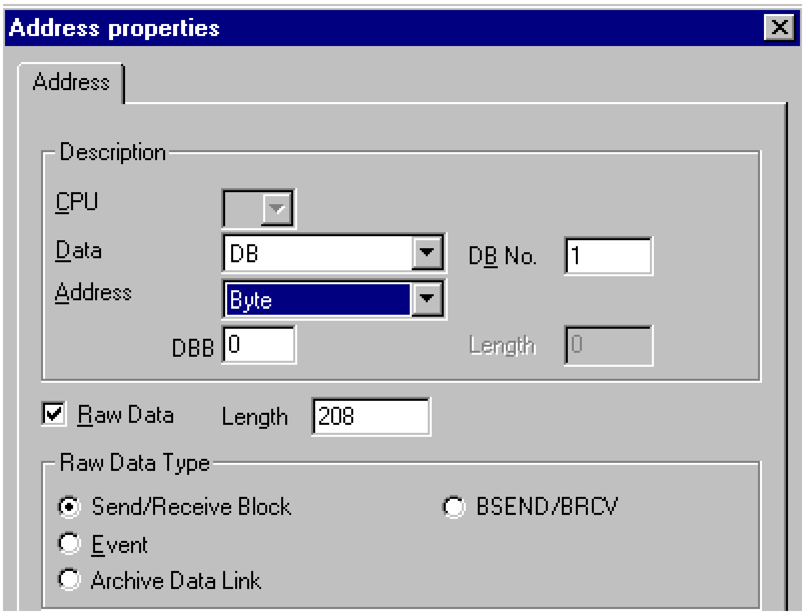
The configuration is identical for all channel units of the channel. The "MPI channel unit and its connection is used in the example.

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.
   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 "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 the data area is selected as "DB", enter the number of the data block in the enabled "DB No." field.
   The "Cpu" field is disabled for a connection of the "MPI" channel unit.



Address properties

Address

Description

CPU

Data: DB

DB No.: 1

Address: Byte

DBB: 0

Length: 0

☑ Raw Data

Length: 208

Raw Data Type

● Send/Receive Block

○ BSEND/BRCV

○ Event

○ Archive Data Link

9. Set up the addressing type in the "Addressing" field. The entries "Byte", "Word" or "Doubleword" 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. "DBB" for data area "DB" "Byte" for addressing type.
11. Click "OK" to close all open dialogs.

Note

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 Byte, max. data block length 208 Byte
- 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.

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 communication 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.

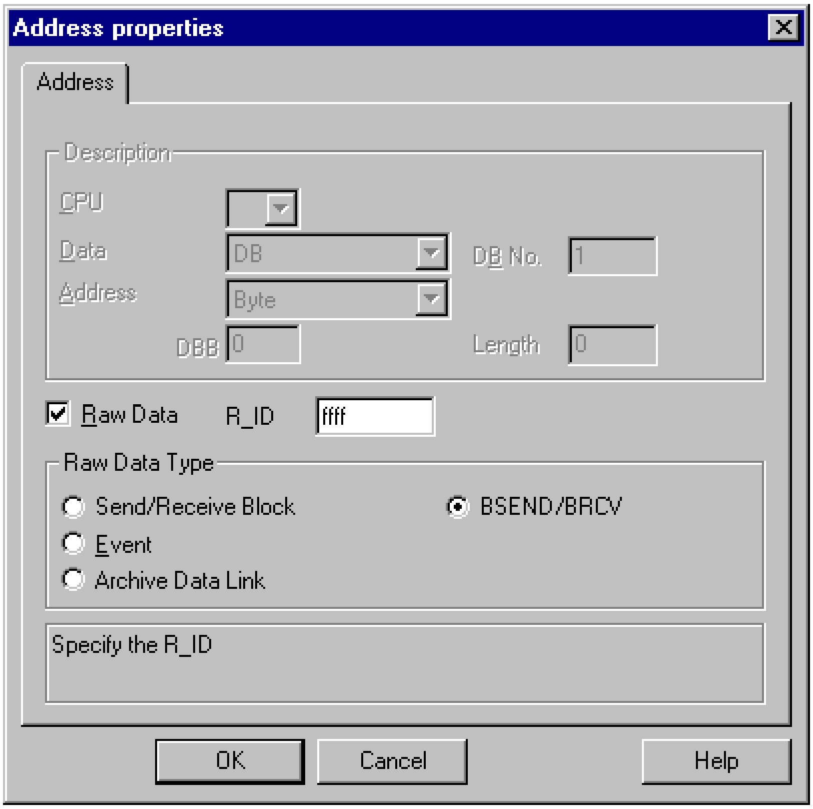
If very large data areas are to be transferred via the connection, then the data blocks are transferred in multiple PDUs.

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.


Address properties

Address

Description

CPU:

Data: DB

DB No.: 1

Address: Byte

DBB: 0

Length: 0

☑ Raw Data

R_ID: ffff

Raw Data Type

○ Send/Receive Block

○ Event

○ Archive Data Link

● BSEND/BRCV

Specify the R_ID

"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 491)

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.

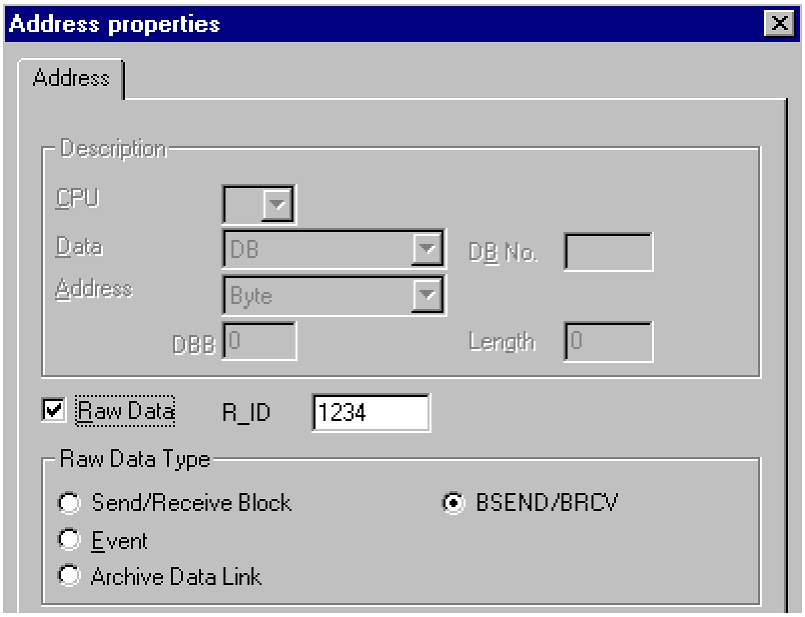
Requirements

- 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



CPU
Data: DB
DB No.
Address: Byte
DBB: 0
Length: 0

☑ Raw Data
R_ID: 1234

○ Send/Receive Block
● BSEND/BRCV
○ Event
○ Archive Data Link

8. Close both of the dialogs by clicking the "OK" buttons.

7.12.5.4 Software Redundancy

Software Redundancy

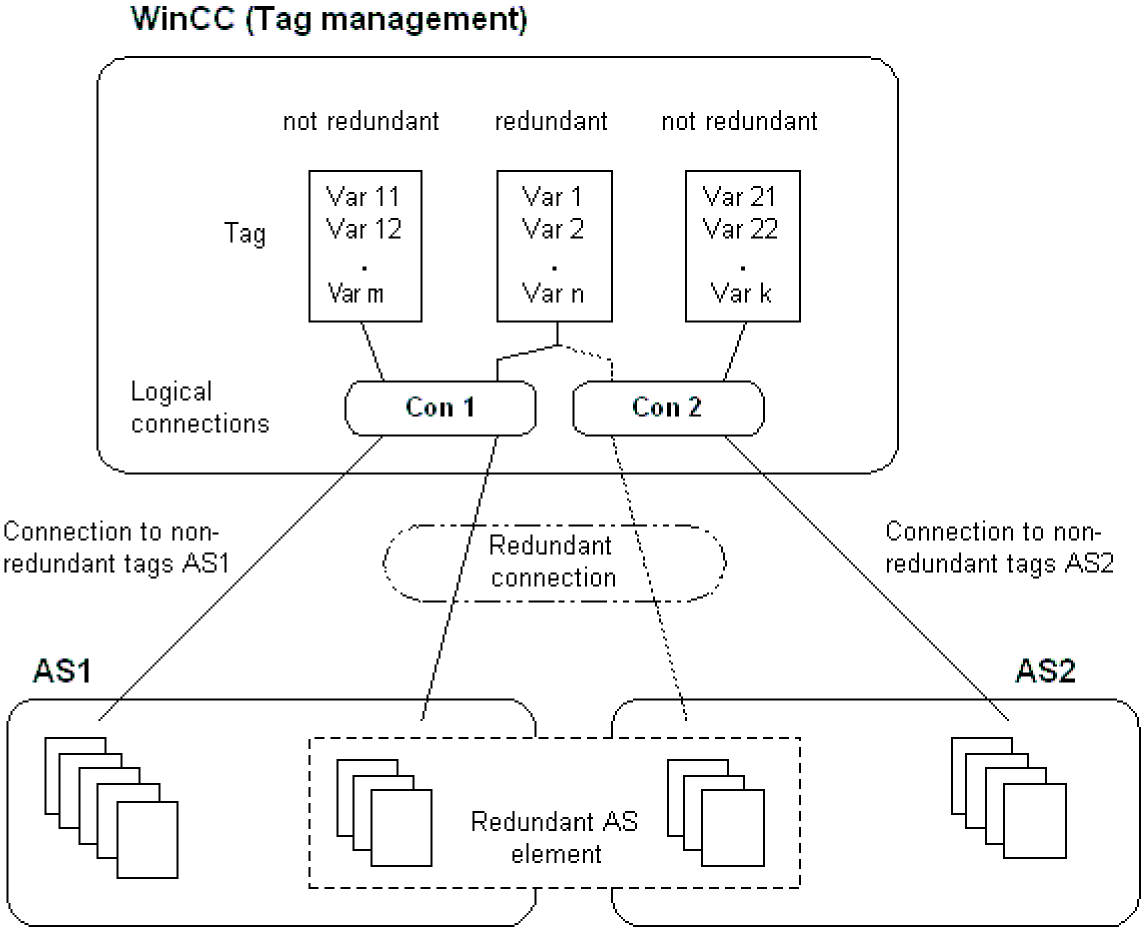
Introduction

The software redundancy offers a cost-effective option for monitoring the safety-related parts of the system that do not have time criticality 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.

Apart from the application program, the software redundancy package is installed on both the automation systems. This program package is not included in the scope of WinCC delivery. For matching data, you need a redundancy connection via MPI, PROFIBUS-DP or Industrial Ethernet between the two automation systems whereby even the existing communication links can be used.

WinCC

The redundancy connection can also be configured amongst the connection of the same channel unit. At the time of configuration, only one connection, the so-called main connection is configured. The reserve connection is only inserted via the Dynamic Wizard "Set up redundant connection".

The Wizard also inserts the connection-specific internal tags and a script. This controls the switching between connections during runtime and also 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 "@<connectionname>@ForceConnectionAddress".

For AS2, you need a second fixed configuration connection if data is to be used from the non-redundant part of AS2 in WinCC.

The use of software redundancy does not means that the hard configured connections can only be used for the redundancy. Each single connection can also be used without redundance.

See also

How to Clear a Software Redundancy in WinCC (Page 500)

How To Configure a Software Redundancy (Page 498)

Software Redundancy - Connection-specific internal tags (Page 494)

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.

How To Configure a Software Redundancy

Introduction

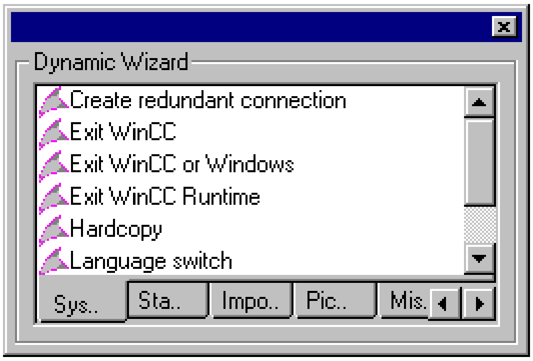
This section describes how to configure the software redundancy for connections of the Channel "SIMATIC S7 Protocol Suite" in WinCC. The PLC must also be configured to use this function, but this will not be described in this document.

Requirements

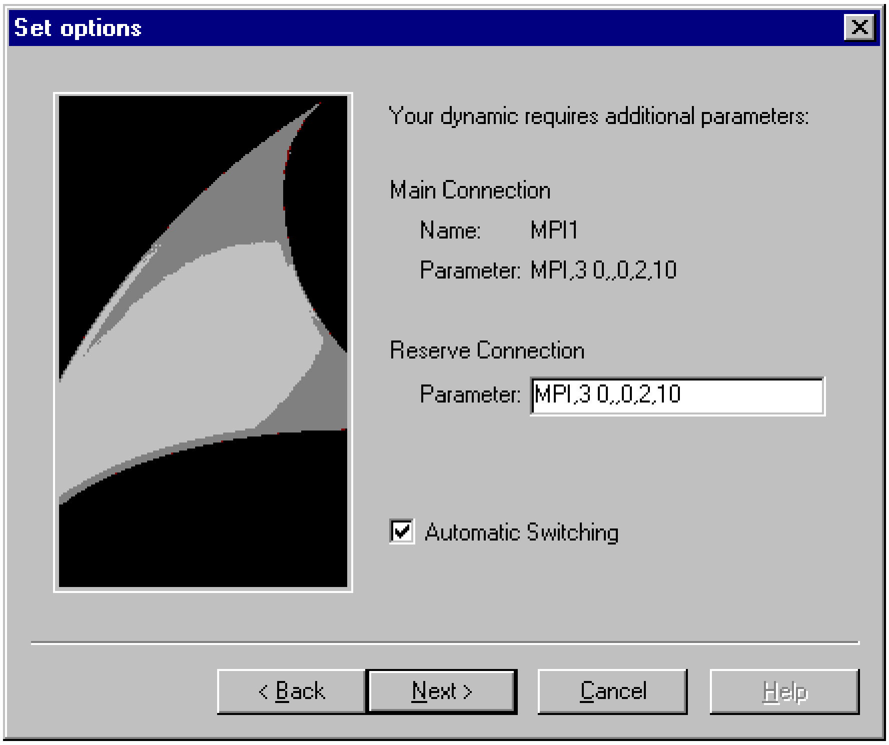
1. The Channel "SIMATIC S7 Protocol Suite" must be integrated into the project.
2. A connection must be created in one of this channel's channel-units to which a redundant connection should be configured.

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:



Set options

Your dynamic requires additional parameters:

Main Connection

Name: MPI1

Parameter: MPI,3 0,,0,2,10

Reserve Connection

Parameter: MPI,3 0,,0,2,10

☑ Automatic Switching

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 501)

How to Check the WinCC Startup Parameters (Page 500)

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.

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.

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.

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

7.13 SIMATIC S7-1200, S7-1500 Channel

7.13.1 "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 526)"

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 (Page 630)".

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

System diagnostics with SysDiagControl (Page 630)

Redundant System S7-1500R/H (Page 526)

STEP 7 "Secure Communication" in the SIMATIC S7-1200, S7-1500 Channel. (Page 508)

7.13.2 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

7.13.3 Configuring the channel

7.13.3.1 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 > Configuring tags for the connection status in Runtime (Page 193)".

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 531)".

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)"

See also

Configuring tags for the connection status in Runtime (Page 193)

Configuring raw data tags (Page 506)

Software redundancy for S7-1500R/H (Page 531)

STEP 7 "Secure Communication" in the SIMATIC S7-1200, S7-1500 Channel. (Page 508)

https://support.industry.siemens.com/cs/ww/en/view/109478253 (https://support.industry.siemens.com/cs/ww/en/view/109478253)

7.13.3.2 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.

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 the address and length of the data area (e.g. 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

Observe the maximum length of data blocks that can be sent by the communication driver:

- S7-1200 / S7-1500: Data block length max. 8000 bytes

Exchanging large data volumes

If you can use raw data to transfer large amounts of data from the controller to WinCC, 37873547 is written in the application example:

- Exchanging large amounts of data between S7-300/400/1500 and WinCC (https://support.industry.siemens.com/cs/ww/en/view/37873547)

See also

Configuration of the "SIMATIC S7-1200, S7-1500 Channel" channel (Page 503)

Exchanging large amounts of data between S7-300/400/1500 and WinCC (https://support.industry.siemens.com/cs/ww/en/view/37873547)

7.13.3.3 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:

- STEP 7 "Secure Communication" in the SIMATIC S7-1200, S7-1500 Channel. (Page 508)

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. Click "OK" to close the dialog.
10. 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

Restriction for S7-1500 software controller

Due to incompatibility of the S7-1507S with Simatic-Net, installation of Simatic-Net on the Soft PLC is currently not possible.

This means that, in the absence of Simatic-Net, a connection to external automation systems via additional channels is not possible.

Other channels such as OPC UA can be used.

See also

Configuring tags for the connection status in Runtime (Page 193)

STEP 7 "Secure Communication" in the SIMATIC S7-1200, S7-1500 Channel. (Page 508)

7.13.3.4 STEP 7 "Secure Communication" in the SIMATIC S7-1200, S7-1500 Channel.

STEP 7 "Secure Communication"

WinCC supports secure STEP 7 communication using the TLS protocol, which is available with TIA Portal as of V17.

STEP 7 components for which "Secure Communication" is configured use an asymmetric key procedure with public key (Public Key) and private key (Private Key). TLS (Transport Layer Security) is used as the encryption protocol.

To use the "Secure Communication" of TIA Portal V17 in the WinCC project, import the data records from a TIA Portal project with the corresponding settings.

Behavior in Runtime

Even with "Secure Communication" enabled, the following actions are possible during operation:

- Updating certificates
- Switching between the configured connections of the "SIMATIC S7-1200, S7-1500 Channel"

System messages

The following system messages document the status of the certificates:

- 1000306: General certificate error at a connection
- 1000307: Expired certificate
- 1000308: Untrusted certificate, manual trust is possible.
- 1000309: Untrusted certificate, manual trust is not possible
- 1000310: Revoked certificate

More information

- Industry Online Support: "WinCC V7 - Secure Communication" (ID 109798498) (https://support.industry.siemens.com/cs/ww/en/view/109798498)
- 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)
- Industry Online Support: Documentation on STEP 7 (TIA Portal V17) (https://support.industry.siemens.com/cs/products?search=%22secure%20communication%22&dtp=Manual&mfn=ps&pnid=24471&lc=en-US)
- TIA Portal information system (V17):
  "Editing devices and networks > Configuring devices and networks > Configure networks > Secure Communication"

Requirement

- In the TIA Portal project, the "Secure Communication" is configured for the S7 controller.
- The AS was compiled in the TIA Portal.

Procedure: Configuring a new WinCC project

1. Export the AS data from the TIA Portal project with the tool "SIEMENS SIMATIC SCADA Export":
   In the TIA Portal project, select the "Export to SIMATIC SCADA" entry in the shortcut menu of the PLC.
2. If necessary, create the desired connection in the "SIMATIC S7-1200, S7-1500 Channel" communication channel.
   Alternatively, select the connection that has already been created.
3. To import the exported AS data in the WinCC tag management, select "AS Symbol > Load from file" from the shortcut menu of the connection.
4. Select the desired data records to be loaded.
   The available controller data is loaded.
   The required certificates are also transferred in the process.
5. To import the required certificates, confirm the corresponding prompt with "Yes".
6. If the WinCC project was newly created, configure the imported data in the WinCC project:
   - Tag Management
     More information: "How to download AS symbols offline (Page 518)"
   - Alarm Logging
     More information: "Working with WinCC > Setting up a Message System > Working with AS Messages"

Procedure: Configuring an existing WinCC project

1. Export the AS data from the TIA Portal project with the tool "SIEMENS SIMATIC SCADA Export": In the TIA Portal project, select the "Export to SIMATIC SCADA" entry in the shortcut menu of the PLC.
2. Select the desired connection in the communication channel "SIMATIC S7-1200, S7-1500 Channel".
3. To import the exported AS data in the WinCC tag management, select "AS Symbol > Load from file" from the shortcut menu of the connection.
4. Select the desired data records to be loaded.
   The available controller data is loaded.
   The required certificates are also transferred in the process.
5. To import the required certificates, confirm the corresponding prompt with "Yes".
   To delete imported certificates again, select the "Delete certificate" entry in the shortcut menu of the connection.

Updating certificates

Expiry date of certificates

If the certificates used have expired, the secure connection remains.

The new certificates are used as soon as the connection is closed and re-established.

However, to increase the security of your installation, update certificates as soon as the expiration date is reached.

Updating a certificate in Runtime

You can manage the certificates without exiting Runtime. This allows you to renew or change the certificates on the controllers while the system continues to run.

To do this, import the current certificates from the TIA Portal project with the tool "SIEMENS SIMATIC SCADA Export".

The imported certificates will be applied the next time Runtime starts.

Note

Secure Communication and Runtime

A connection established via "Secure Communication" remains open until you stop WinCC Runtime or disconnect the connection from the controller.

Even in the "CPU Stop" state, the secure connection remains active until the connection is re-established.

This allows you to update certificates independently in the WinCC project and on the controller. Updating the controller and closing and restarting WinCC Runtime do not have to happen at the same time.

Changing connections ("Change Connection")

You can also change between the connections of the communication channel in WinCC Runtime when using "Secure Communication".

You need to change connections, for example, when you replace hardware or install hardware updates.

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=abc;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.

Requirements for changing connections

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:

The combination possibilities of the CPUs depend on the type of installed certificates on the CPUs.

Source CPU	CPU after connection change *. *	Comments
"Self-Signed End-Entity" certificate	Unknown "Self-Signed End-Entity" certificate	Manual acknowledgment required ("Manual Trust")
	Unknown root certificate (CA) "End-Entity" certificate	Import of certificate data from TIA Portal required
	Known root certificate (CA) "End-Entity" certificate	Combination occurs, for example, when the root certificate has already been imported into WinCC.
Root certificate (CA) and "End-Entity" certificate	Unknown "Self-Signed End-Entity" certificate	Manual acknowledgment required ("Manual Trust")
	Unknown root certificate (CA) "End-Entity" certificate	Import of certificate data from TIA Portal required
	Known root certificate (CA) "End-Entity" certificate	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.

Certificate management: "Manual Trust" and certificate revocation lists (CRL)

To manage the certificates, use the "Device Certificate Store" folder in the following path:

- <Installation path>\Siemens\Automation\device-certificate-store
  Example: "C:\ProgramData\Siemens\Automation\device-certificate-store"

In the "Device Certificate Store", you can store certificate revocation lists, as well as manually confirm or reject certificates as trusted ("Manual Trust").

If the target CPU uses unknown certificates after a connection change, these certificates are stored in the "untrusted" folder.

- 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.

A certificate revocation list contains certificates that have been revoked. The "*.DER" certificate revocation lists are located in the following folder:

- <Installation path>\Siemens\Automation\device-certificate-store\trusted\crl
  Example: "C:\ProgramData\Siemens\Automation\device-certificate-store\trusted\crl"

Note

Managing root certificates

To use an "end-entity" certificate combined with a root certificate, import the root certificate into WinCC. This makes the root certificate public and confirms it as trusted.

You cannot manage root certificates with the "Device Certificate Store".

More information about root certificates:

- Industry Online Support: STEP 7 (TIA Portal) - Documentation: Signatures and certificates (https://support.industry.siemens.com/cs/ww/en/view/109798671/143786688779)

See also

Configuration of the "SIMATIC S7-1200, S7-1500 Channel" channel (Page 503)

"SIMATIC S7-1200, S7-1500 Channel" channel (Page 502)

How to configure a connection (Page 507)

How to download AS symbols offline (Page 518)

Configuring tags for the connection status in Runtime (Page 193)

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)

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)

Industry Online Support: STEP 7 (TIA Portal) - Documentation: Signatures and certificates (https://support.industry.siemens.com/cs/ww/en/view/109798671/143786688779)

7.13.3.5 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.

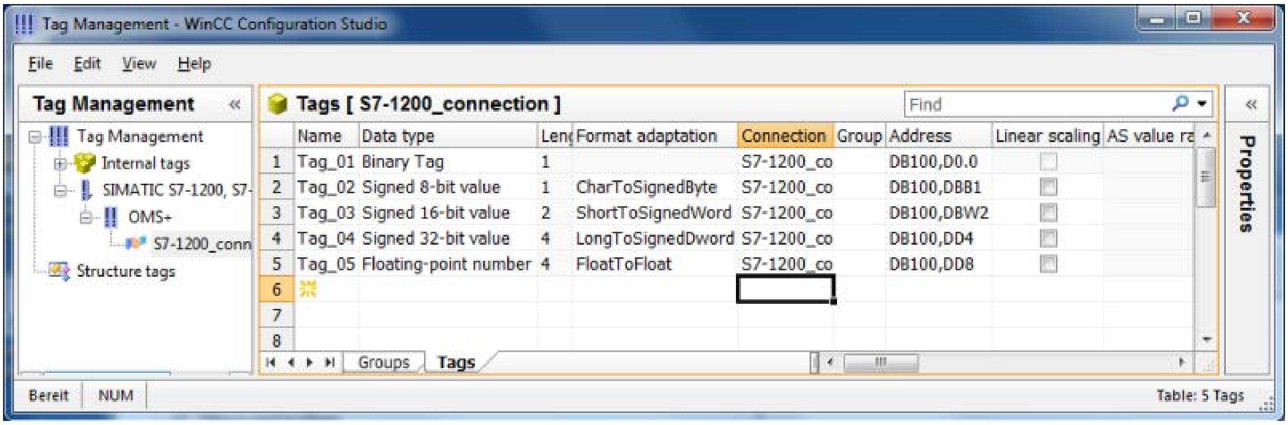
Procedure

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.



Tags [S7-1200_connection]

	Name	Data type	Len(	Format adaptation	Connection	Group	Address	Linear scaling	AS value ra
1	Tag_01	Binary Tag	1		S7-1200_co		DB100,D0.0	☐	
2	Tag_02	Signed 8-bit value	1	CharToSignedByte	S7-1200_co		DB100,DBB1	☐	
3	Tag_03	Signed 16-bit value	2	ShortToSignedWord	S7-1200_co		DB100,DBW2	☐	
4	Tag_04	Signed 32-bit value	4	LongToSignedDword	S7-1200_co		DB100,DD4	☐	
5	Tag_05	Floating-point number	4	FloatToFloat	S7-1200_co		DB100,DD8	☐	
6									
7									
8									

7.13.3.6 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

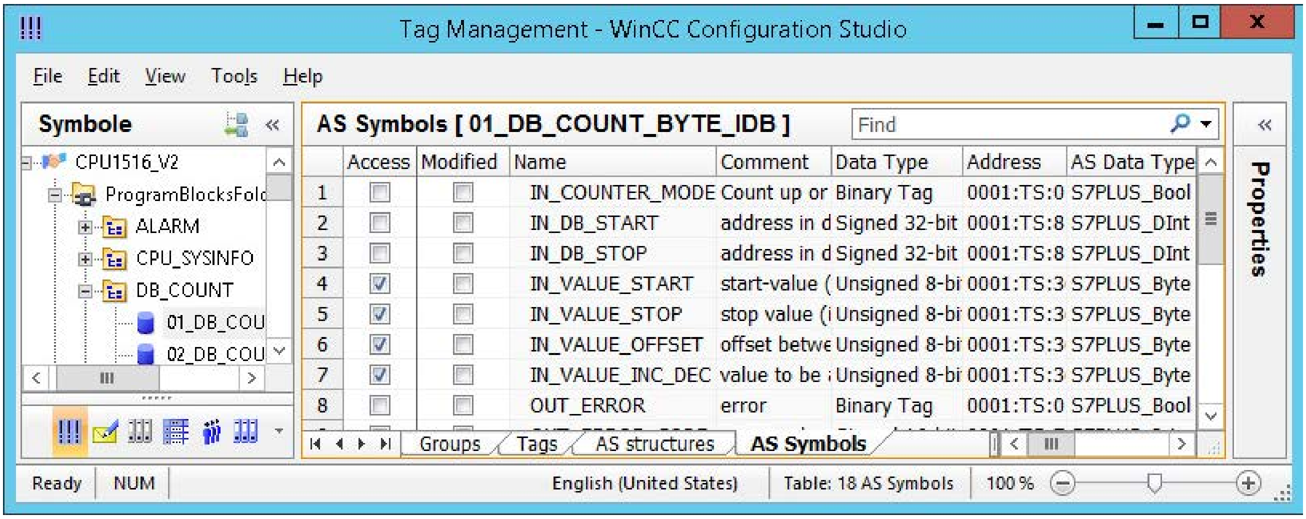
You cannot transfer tags that you have created as AS symbols via "Load to AS" in Runtime with load online changes.

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

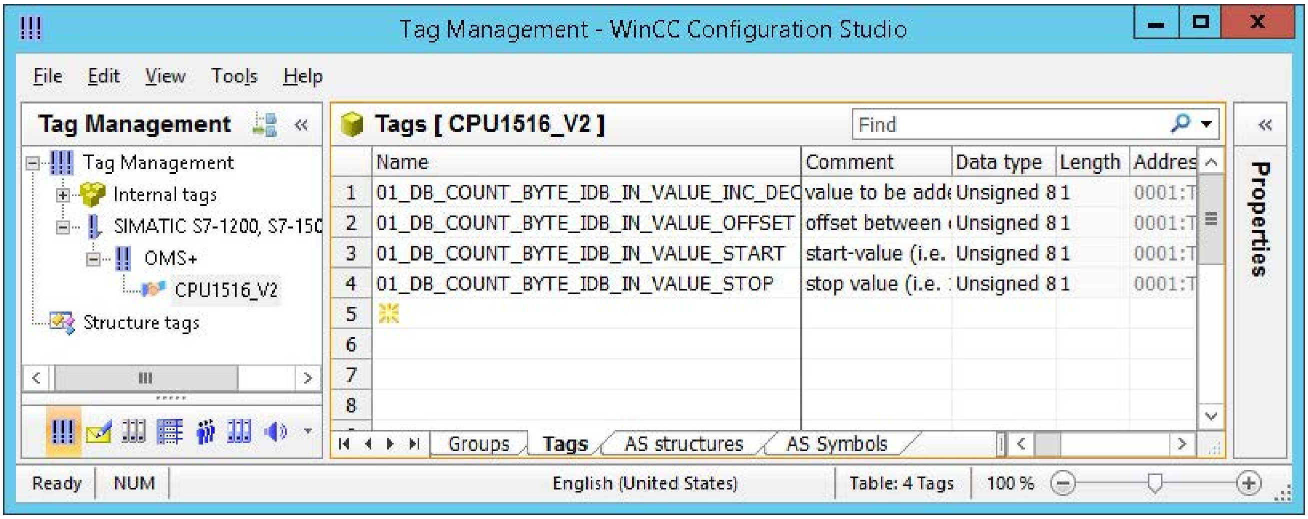
1. Select the required connection.
2. Select "AS Symbols > Read from AS" from the shortcut menu of the connection.
   The available controller data is loaded and the "Symbols" view opens.
   The loaded data is displayed in the table area in the "AS Symbols" tab.
   If the loaded data also contains structures, the "AS structures" tab is displayed additionally.



	Access	Modified	Name	Comment	Data Type	Address	AS Data Type
1	☐	☐	IN_COUNTER_MODE	Count up or	Binary Tag	0001:TS:0	S7PLUS_Bool
2	☐	☐	IN_DB_START	address in d	Signed 32-bit	0001:TS:8	S7PLUS_DInt
3	☐	☐	IN_DB_STOP	address in d	Signed 32-bit	0001:TS:8	S7PLUS_DInt
4	☑	☐	IN_VALUE_START	start-value (	Unsigned 8-bi	0001:TS:3	S7PLUS_Byte
5	☑	☐	IN_VALUE_STOP	stop value (i	Unsigned 8-bi	0001:TS:3	S7PLUS_Byte
6	☑	☐	IN_VALUE_OFFSET	offset betwe	Unsigned 8-bi	0001:TS:3	S7PLUS_Byte
7	☑	☐	IN_VALUE_INC_DEC	value to be	Unsigned 8-bi	0001:TS:3	S7PLUS_Byte
8	☐	☐	OUT_ERROR	error	Binary Tag	0001:TS:0	S7PLUS_Bool

Ready NUM English (United States) Table: 18 AS Symbols 100 %

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.



	Name	Comment	Data type	Length	Addres
1	01_DB_COUNT_BYTE_IDB_IN_VALUE_INC_DEC	value to be add	Unsigned 8	1	0001:T
2	01_DB_COUNT_BYTE_IDB_IN_VALUE_OFFSET	offset between	Unsigned 8	1	0001:T
3	01_DB_COUNT_BYTE_IDB_IN_VALUE_START	start-value (i.e.	Unsigned 8	1	0001:T
4	01_DB_COUNT_BYTE_IDB_IN_VALUE_STOP	stop value (i.e.	Unsigned 8	1	0001:T
5					
6					
7					
8					

Ready NUM English (United States) Table: 4 Tags 100 %

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 How to download AS symbols offline (Page 518).

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 525)

How to configure AS structures (Page 522)

7.13.3.7 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:

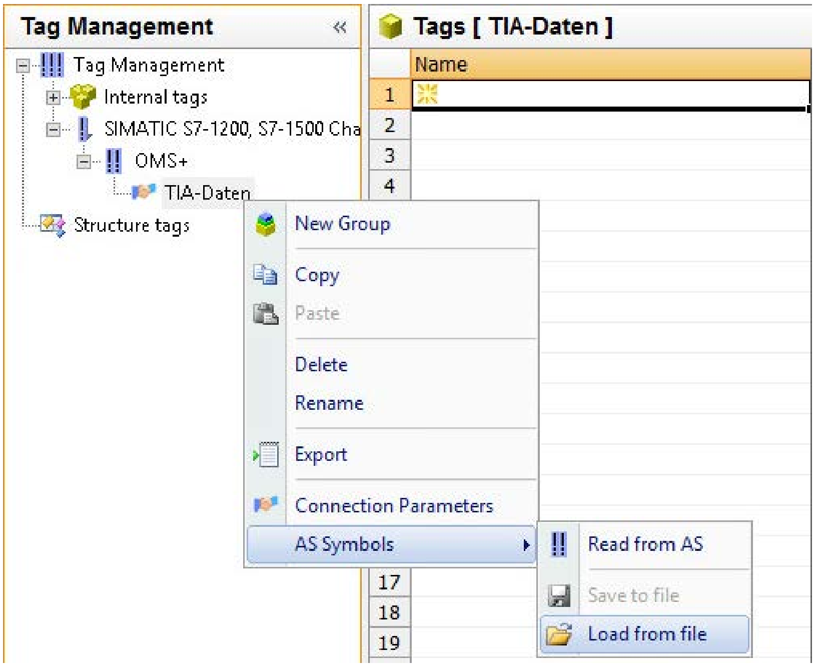
- Industry Online Support: Download "SIMATIC SCADA Export for TIA Portal" (ID 109748955) (https://support.industry.siemens.com/cs/ww/en/view/109748955)
- Industry Online Support: "SIMATIC SCADA Export" documentation (ID 101908495) (https://support.industry.siemens.com/cs/ww/en/view/101908495)

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.

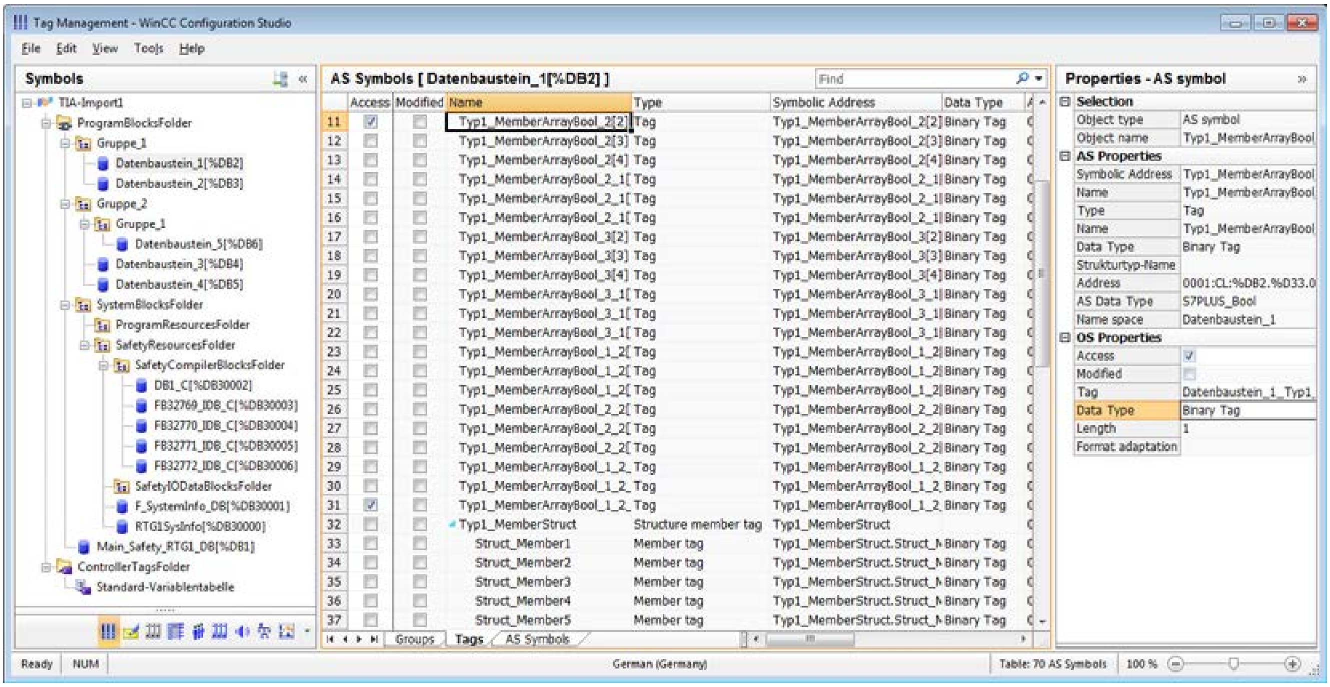
Result

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.



Symbols

- TIA-Import1
  - ProgramBlocksFolder
    - Gruppe_1
      - Datenbaustein_1[%DB2]
      - Datenbaustein_2[%DB3]
    - Gruppe_2
      - Gruppe_1
        - Datenbaustein_5[%DB6]
      - Datenbaustein_3[%DB4]
      - Datenbaustein_4[%DB5]
    - SystemBlocksFolder
      - ProgramResourcesFolder
      - SafetyResourcesFolder
        - SafetyCompilerBlocksFolder
          - DB1_C[%DB30002]
          - FB32769_IDB_C[%DB30003]
          - FB32770_IDB_C[%DB30004]
          - FB32771_IDB_C[%DB30005]
          - FB32772_IDB_C[%DB30006]
        - SafetyIODataBlocksFolder
        - F_SystemInfo_DB[%DB30001]
        - RTG1SysInfo[%DB30000]
    - Main_Safety_RTG1_DB[%DB1]
  - ControllerTagsFolder
    - Standard-Variablentabelle

AS Symbols [Datenbaustein_1[%DB2]]

	Access	Modified	Name	Type	Symbolic Address	Data Type
11	☑	☐	Typ1_MemberArrayBool_2[2]	Tag	Typ1_MemberArrayBool_2[2]	Binary Tag
12	☐	☐	Typ1_MemberArrayBool_2[3]	Tag	Typ1_MemberArrayBool_2[3]	Binary Tag
13	☐	☐	Typ1_MemberArrayBool_2[4]	Tag	Typ1_MemberArrayBool_2[4]	Binary Tag
14	☐	☐	Typ1_MemberArrayBool_2_1[	Tag	Typ1_MemberArrayBool_2_1[	Binary Tag
15	☐	☐	Typ1_MemberArrayBool_2_1[	Tag	Typ1_MemberArrayBool_2_1[	Binary Tag
16	☐	☐	Typ1_MemberArrayBool_2_1[	Tag	Typ1_MemberArrayBool_2_1[	Binary Tag
17	☐	☐	Typ1_MemberArrayBool_3[2]	Tag	Typ1_MemberArrayBool_3[2]	Binary Tag
18	☐	☐	Typ1_MemberArrayBool_3[3]	Tag	Typ1_MemberArrayBool_3[3]	Binary Tag
19	☐	☐	Typ1_MemberArrayBool_3[4]	Tag	Typ1_MemberArrayBool_3[4]	Binary Tag
20	☐	☐	Typ1_MemberArrayBool_3_1[	Tag	Typ1_MemberArrayBool_3_1[	Binary Tag
21	☐	☐	Typ1_MemberArrayBool_3_1[	Tag	Typ1_MemberArrayBool_3_1[	Binary Tag
22	☐	☐	Typ1_MemberArrayBool_3_1[	Tag	Typ1_MemberArrayBool_3_1[	Binary Tag
23	☐	☐	Typ1_MemberArrayBool_1_2[	Tag	Typ1_MemberArrayBool_1_2[	Binary Tag
24	☐	☐	Typ1_MemberArrayBool_1_2[	Tag	Typ1_MemberArrayBool_1_2[	Binary Tag
25	☐	☐	Typ1_MemberArrayBool_1_2[	Tag	Typ1_MemberArrayBool_1_2[	Binary Tag
26	☐	☐	Typ1_MemberArrayBool_2_2[	Tag	Typ1_MemberArrayBool_2_2[	Binary Tag
27	☐	☐	Typ1_MemberArrayBool_2_2[	Tag	Typ1_MemberArrayBool_2_2[	Binary Tag
28	☐	☐	Typ1_MemberArrayBool_2_2[	Tag	Typ1_MemberArrayBool_2_2[	Binary Tag
29	☐	☐	Typ1_MemberArrayBool_1_2_	Tag	Typ1_MemberArrayBool_1_2	Binary Tag
30	☐	☐	Typ1_MemberArrayBool_1_2_	Tag	Typ1_MemberArrayBool_1_2	Binary Tag
31	☑	☐	Typ1_MemberArrayBool_1_2_	Tag	Typ1_MemberArrayBool_1_2	Binary Tag
32	☐	☐	Typ1_MemberStruct	Structure member tag	Typ1_MemberStruct	
33	☐	☐	Struct_Member1	Member tag	Typ1_MemberStruct.Struct_M	Binary Tag
34	☐	☐	Struct_Member2	Member tag	Typ1_MemberStruct.Struct_M	Binary Tag
35	☐	☐	Struct_Member3	Member tag	Typ1_MemberStruct.Struct_M	Binary Tag
36	☐	☐	Struct_Member4	Member tag	Typ1_MemberStruct.Struct_M	Binary Tag
37	☐	☐	Struct_Member5	Member tag	Typ1_MemberStruct.Struct_M	Binary Tag

Properties - AS symbol

Selection	
Object type	AS symbol
Object name	Typ1_MemberArrayBool
AS Properties	
Symbolic Address	Typ1_MemberArrayBool
Name	Typ1_MemberArrayBool
Type	Tag
Name	Typ1_MemberArrayBool
Data Type	Binary Tag
Strukturtyp-Name	
Address	0001:CL:%DB2.%D33.0
AS Data Type	S7PLUS_Bool
Name space	Datenbaustein_1
OS Properties	
Access	☑
Modified	☐
Tag	Datenbaustein_1_Typ1_
Data Type	Binary Tag
Length	1
Format adaptation	

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.

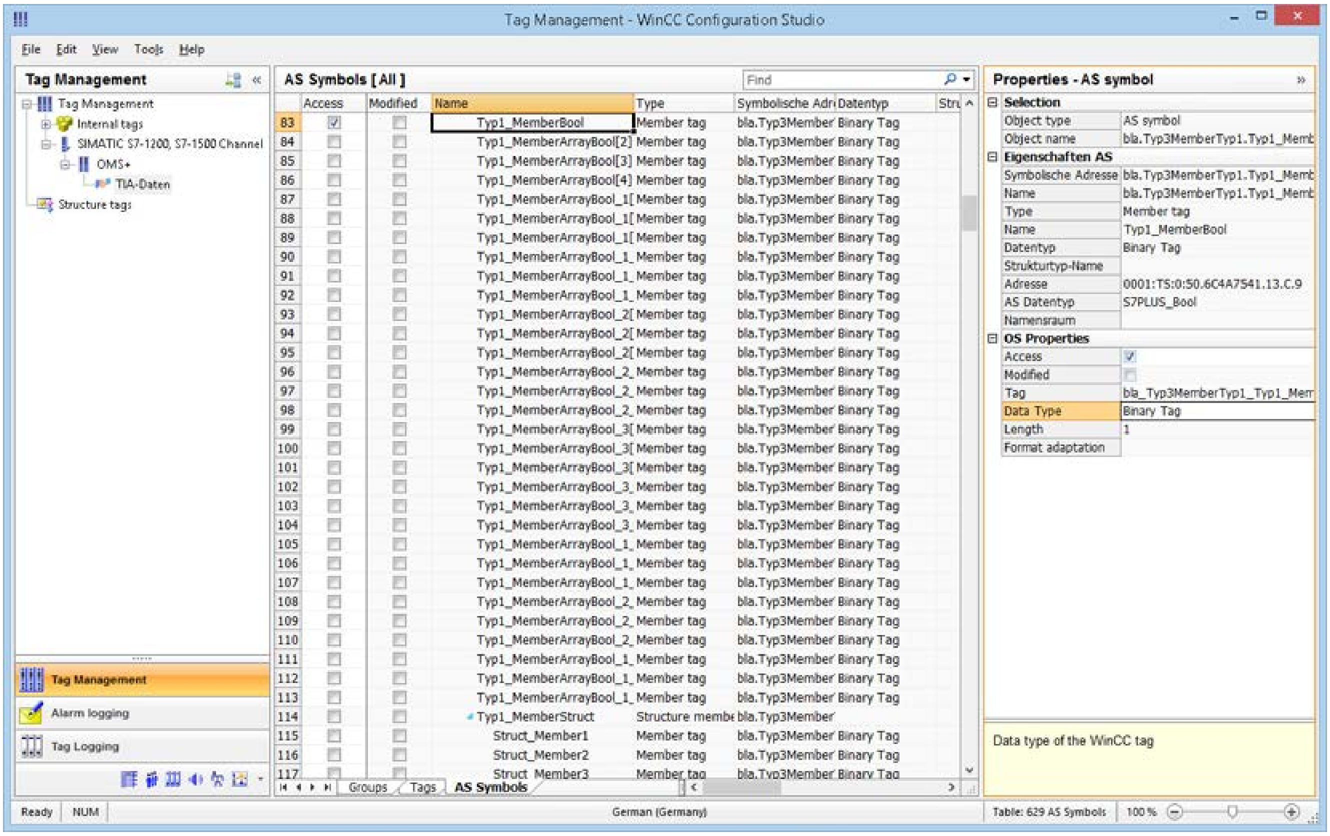
By selecting the check boxes in the "Access" column, you create a WinCC tag from the found AS tags.

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.



	Access	Modified	Name	Type	Symbolische Adr	Datentyp	Stru
83	☑	□	Typ1_MemberBool	Member tag	bla.Typ3Member	Binary Tag	
84	□	□	Typ1_MemberArrayBool[2]	Member tag	bla.Typ3Member	Binary Tag	
85	□	□	Typ1_MemberArrayBool[3]	Member tag	bla.Typ3Member	Binary Tag	
86	□	□	Typ1_MemberArrayBool[4]	Member tag	bla.Typ3Member	Binary Tag	
87	□	□	Typ1_MemberArrayBool_1[	Member tag	bla.Typ3Member	Binary Tag	
88	□	□	Typ1_MemberArrayBool_1[	Member tag	bla.Typ3Member	Binary Tag	
89	□	□	Typ1_MemberArrayBool_1[	Member tag	bla.Typ3Member	Binary Tag	
90	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
91	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
92	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
93	□	□	Typ1_MemberArrayBool_2[	Member tag	bla.Typ3Member	Binary Tag	
94	□	□	Typ1_MemberArrayBool_2[	Member tag	bla.Typ3Member	Binary Tag	
95	□	□	Typ1_MemberArrayBool_2[	Member tag	bla.Typ3Member	Binary Tag	
96	□	□	Typ1_MemberArrayBool_2_	Member tag	bla.Typ3Member	Binary Tag	
97	□	□	Typ1_MemberArrayBool_2_	Member tag	bla.Typ3Member	Binary Tag	
98	□	□	Typ1_MemberArrayBool_2_	Member tag	bla.Typ3Member	Binary Tag	
99	□	□	Typ1_MemberArrayBool_3[	Member tag	bla.Typ3Member	Binary Tag	
100	□	□	Typ1_MemberArrayBool_3[	Member tag	bla.Typ3Member	Binary Tag	
101	□	□	Typ1_MemberArrayBool_3[	Member tag	bla.Typ3Member	Binary Tag	
102	□	□	Typ1_MemberArrayBool_3_	Member tag	bla.Typ3Member	Binary Tag	
103	□	□	Typ1_MemberArrayBool_3_	Member tag	bla.Typ3Member	Binary Tag	
104	□	□	Typ1_MemberArrayBool_3_	Member tag	bla.Typ3Member	Binary Tag	
105	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
106	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
107	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
108	□	□	Typ1_MemberArrayBool_2_	Member tag	bla.Typ3Member	Binary Tag	
109	□	□	Typ1_MemberArrayBool_2_	Member tag	bla.Typ3Member	Binary Tag	
110	□	□	Typ1_MemberArrayBool_2_	Member tag	bla.Typ3Member	Binary Tag	
111	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
112	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
113	□	□	Typ1_MemberArrayBool_1_	Member tag	bla.Typ3Member	Binary Tag	
114	□	□	Typ1_MemberStruct	Structure membe	bla.Typ3Member		
115	□	□	Struct_Member1	Member tag	bla.Typ3Member	Binary Tag	
116	□	□	Struct_Member2	Member tag	bla.Typ3Member	Binary Tag	
117	□	□	Struct_Member3	Member tag	bla.Typ3Member	Binary Tag	

Properties - AS symbol

Selection	
Object type	AS symbol
Object name	bla.Typ3MemberTyp1.Typ1_Memb
Eigenschaften AS	
Symbolische Adresse	bla.Typ3MemberTyp1.Typ1_Memb
Name	bla.Typ3MemberTyp1.Typ1_Memb
Type	Member tag
Name	Typ1_MemberBool
Datentyp	Binary Tag
Strukturtyp-Name	
Adresse	0001:TS:0:50.6C4A7541.13.C.9
AS Datentyp	S7PLUS_Bool
Namensraum	
OS Properties	
Access	☑
Modified	□
Tag	bla_Typ3MemberTyp1_Typ1_Mem
Data Type	Binary Tag
Length	1
Format adaptation	

Data type of the WinCC tag

See also

How to export AS project data (Page 525)

How to configure AS structures (Page 522)

STEP 7 "Secure Communication" in the SIMATIC S7-1200, S7-1500 Channel. (Page 508)

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)

7.13.3.8 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".

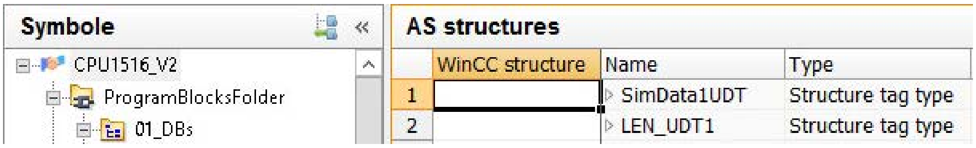
You change the name of the WinCC structure type and the structure type elements in the Tag Management. The assignment of the AS structure is automatically adjusted.

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

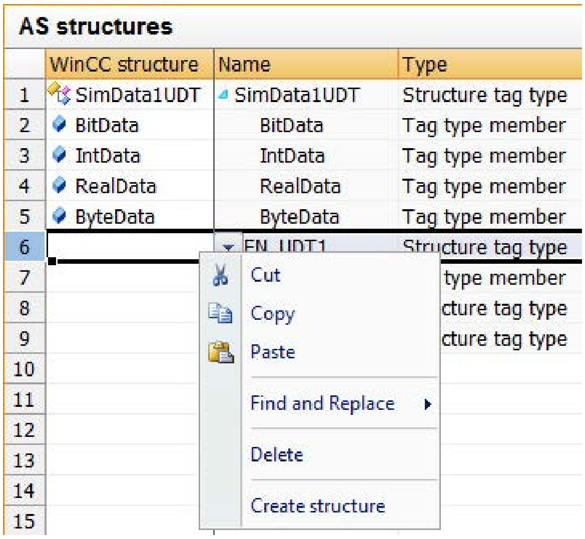
1. Load the AS symbols via "Read from AS" or "Load from file".
   The loaded messages are displayed in the "Symbols" Tag Management view.
   The loaded structures are displayed on the "AS structures" tab.
   The structure names are transferred when loading from the AS.



	WinCC structure	Name	Type
1		SimData1UDT	Structure tag type
2		LEN_UDT1	Structure tag type

2. Click "AS structures".
   To display the elements of a structure, click the arrow in front of the structure name.

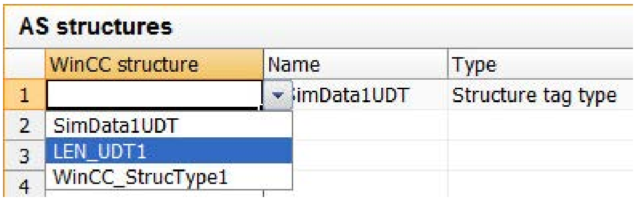
3. Select the entire row of a structure and select the "Create structure" entry in the shortcut menu.



AS structures

	WinCC structure	Name	Type
1	SimData1UDT	SimData1UDT	Structure tag type
2	BitData	BitData	Tag type member
3	IntData	IntData	Tag type member
4	RealData	RealData	Tag type member
5	ByteData	ByteData	Tag type member
6		EN_UDT1	Structure tag type
7			type member
8			cture tag type
9			cture tag type
10			
11			
12			
13			
14			
15			

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".



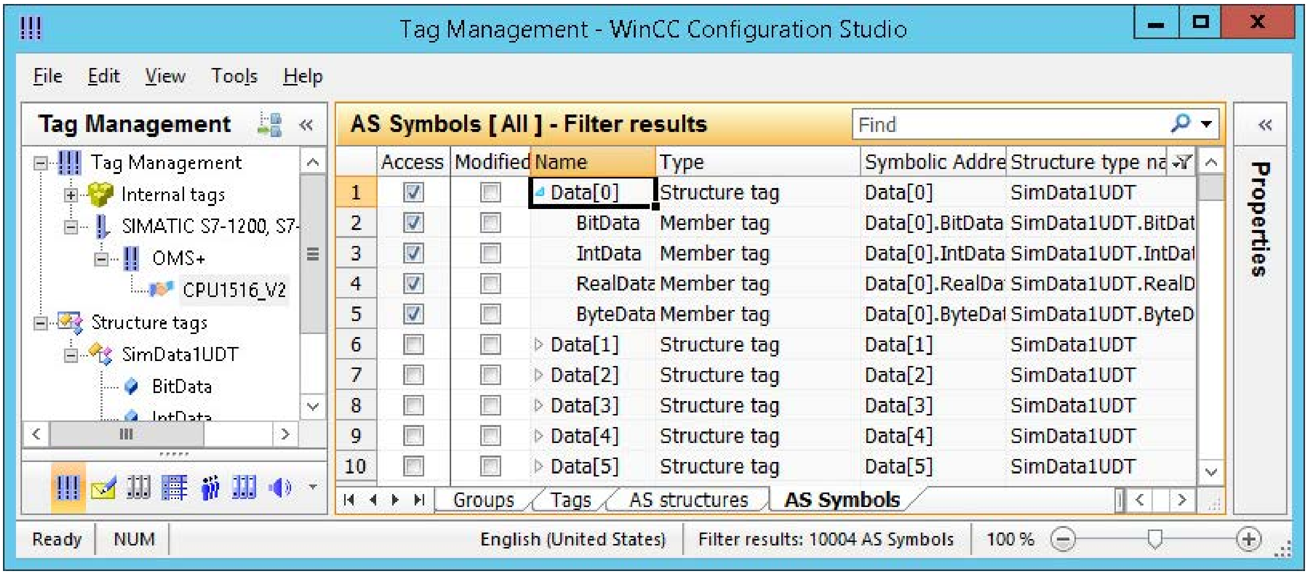
AS structures

	WinCC structure	Name	Type
1		imData1UDT	Structure tag type
2			
3			
4			

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.



	Access	Modified	Name	Type	Symbolic Addre	Structure type na
1	☑	□	Data[0]	Structure tag	Data[0]	SimData1UDT
2	☑	□	BitData	Member tag	Data[0].BitData	SimData1UDT.BitDat
3	☑	□	IntData	Member tag	Data[0].IntData	SimData1UDT.IntDat
4	☑	□	RealDat	Member tag	Data[0].RealDa	SimData1UDT.RealD
5	☑	□	ByteData	Member tag	Data[0].ByteDat	SimData1UDT.ByteD
6	□	□	Data[1]	Structure tag	Data[1]	SimData1UDT
7	□	□	Data[2]	Structure tag	Data[2]	SimData1UDT
8	□	□	Data[3]	Structure tag	Data[3]	SimData1UDT
9	□	□	Data[4]	Structure tag	Data[4]	SimData1UDT
10	□	□	Data[5]	Structure tag	Data[5]	SimData1UDT

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 518)

How to export AS project data (Page 525)

How to configure a tag with optimized block access (Page 515)

7.13.3.9 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 515)

How to download AS symbols offline (Page 518)

How to configure AS structures (Page 522)

7.13.3.10 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 528)

Software redundancy for S7-1500R/H (Page 531)

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)

7.13.3.11 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 531)"

See also

Redundant System S7-1500R/H (Page 526)

Software redundancy for S7-1500R/H (Page 531)

"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)

7.13.3.12 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. You can find additional information under: "Configuring redundant control systems (Page 528)"

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, e.g. every 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
  'set force connection state = 0
        C1ForceConnectionState.Value = 0
        C1ForceConnectionState.Write

  'switch between plc1 and plc2
        Select Case C1UseAlternativeAddress.Value

            Case 0
                    C1UseAlternativeAddress.Value = 1
                    C1UseAlternativeAddress.Write
            Case 1
                    C1UseAlternativeAddress.Value = 0
                    C1UseAlternativeAddress.Write

        End Select

  'reconnect
        C1ForceConnectionState.Value = 1
        C1ForceConnectionState.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

  'switch between plc1 and plc2
        Select Case C1UseAlternativeAddress.Value

            Case 0
                    C1UseAlternativeAddress.Value = 1
                    C1UseAlternativeAddress.Write
            Case 1
                    C1UseAlternativeAddress.Value = 0
                    C1UseAlternativeAddress.Write

        End Select

  'reconnect
        C1ForceConnectionState.Value = 1
        C1ForceConnectionState.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 526)

Configuring redundant control systems (Page 528)

7.14 SIMATIC TI Ethernet Layer 4

7.14.1 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).

7.14.2 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

7.14.3 Configuring the Channel

7.14.3.1 Configuring the channel "SIMATIC TI Ethernet Layer 4"

Introduction

The following steps are required for configuring the channel "SIMATIC TI Ethernet Layer 4".

7.14.3.2 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.

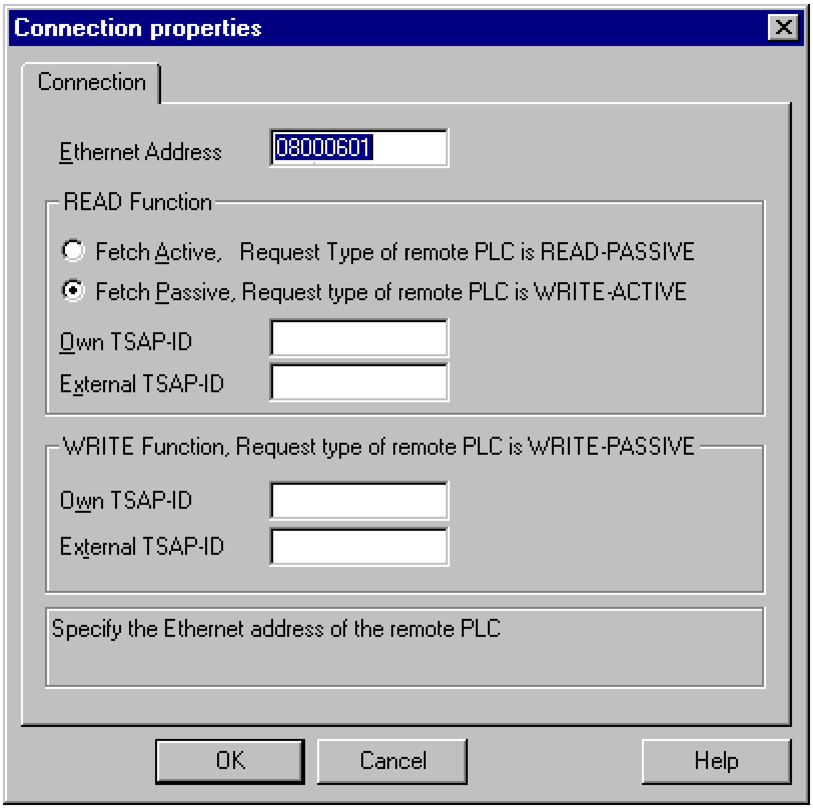
WinCC side	SIMATIC TI side
FETCH-Active (Request "READ-Active")	READ-Passive (Request "READ-Passive")
FETCH-Passive (Request "WRITE-Passive")	WRITE-Active (Request "WRITE-Active")

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.



Connection properties

Connection

Ethernet Address: 08000601

READ Function

○ Fetch Active, Request Type of remote PLC is READ-PASSIVE

● Fetch Passive, Request type of remote PLC is WRITE-ACTIVE

Own TSAP-ID:

External TSAP-ID:

WRITE Function, Request type of remote PLC is WRITE-PASSIVE

Own TSAP-ID:

External TSAP-ID:

Specify the Ethernet address of the remote PLC

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.

7.14.3.3 Configuring the tags

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.

How to configure the address of a tag

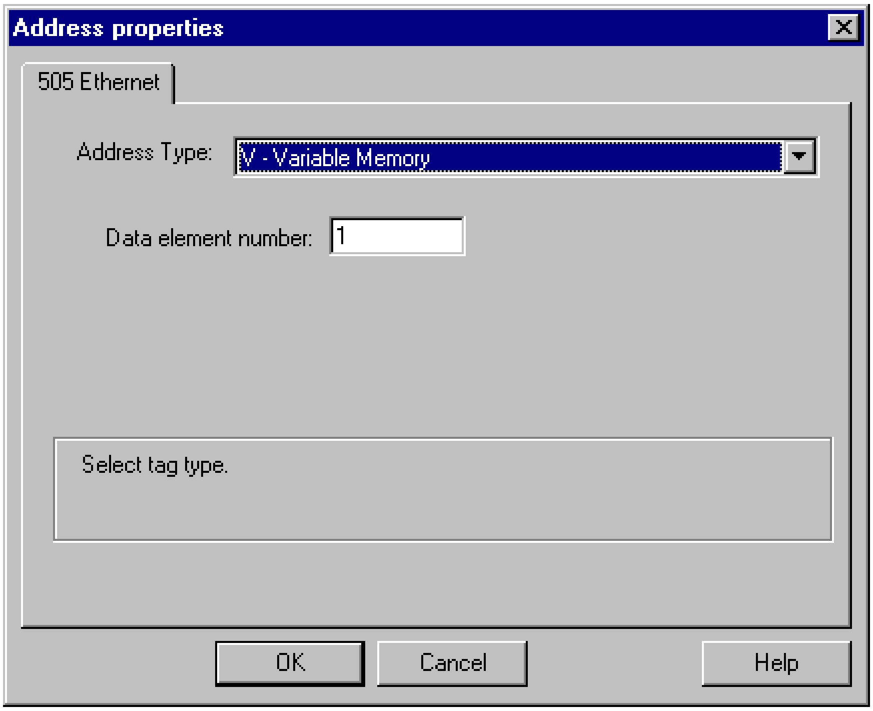
Introduction

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.



Address properties

505 Ethernet

Address Type: V - Variable Memory

Data element number: 1

Select tag type.

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.

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.

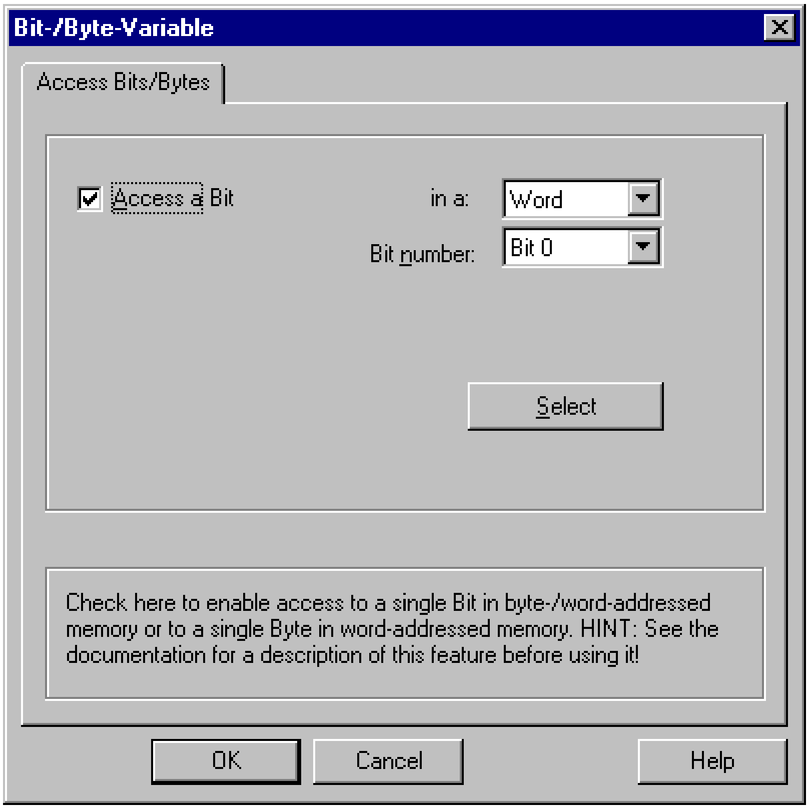
Note

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.



☑ Access a Bit

in a: Word

Bit number: Bit 0

Check here to enable access to a single Bit in byte-/word-addressed memory or to a single Byte in word-addressed memory. HINT: See the documentation for a description of this feature before using it!

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.

How to Configure a Tag with Byte by Byte Access

Introduction

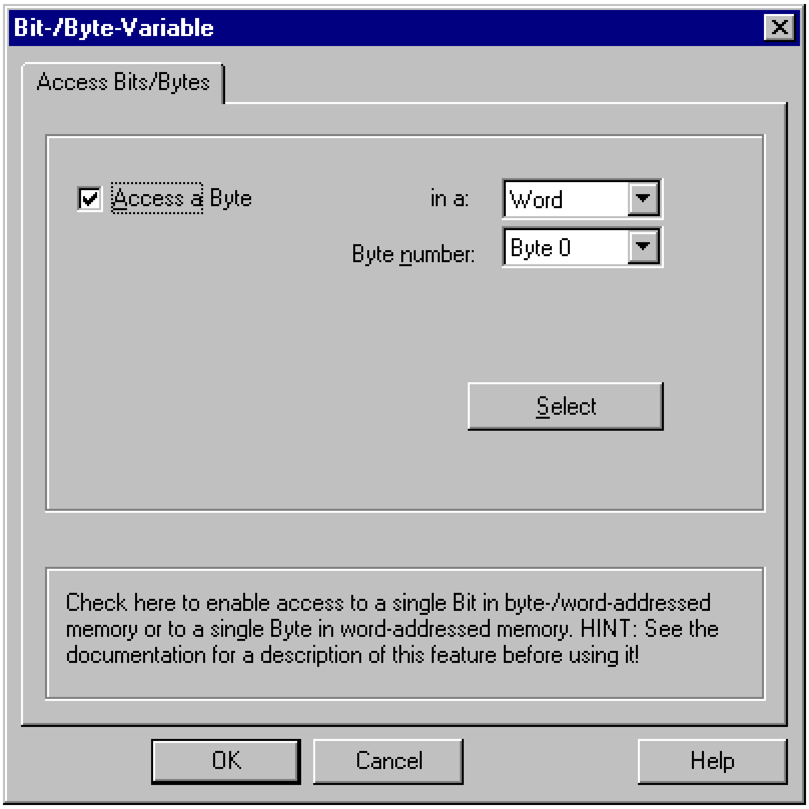
Write access to memory areas in the AS can only performed bit-wise or byte-wise in channel "TI Ethernet Layer 4". When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to define write access to individual bits or bytes. For 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.



Bit-/Byte-Variable

Access Bits/Bytes

☑ Access a Byte

in a: Word

Byte number: Byte 0

Check here to enable access to a single Bit in byte-/word-addressed memory or to a single Byte in word-addressed memory. HINT: See the documentation for a description of this feature before using it!

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.

7.14.3.4 System parameters

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.

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.

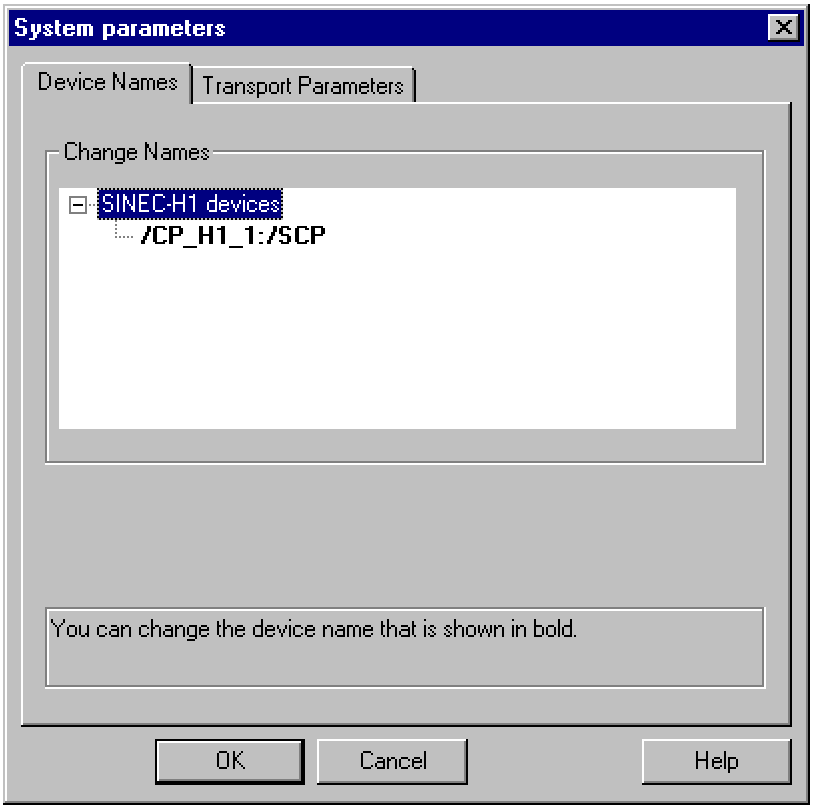
Because communication takes place via the ISO transport protocol, it is not necessary to configure the logical connection in the local database.

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.



SINEC-H1 devices
/CP_H1_1:/SCP

You can change the device name that is shown in bold.

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.

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.

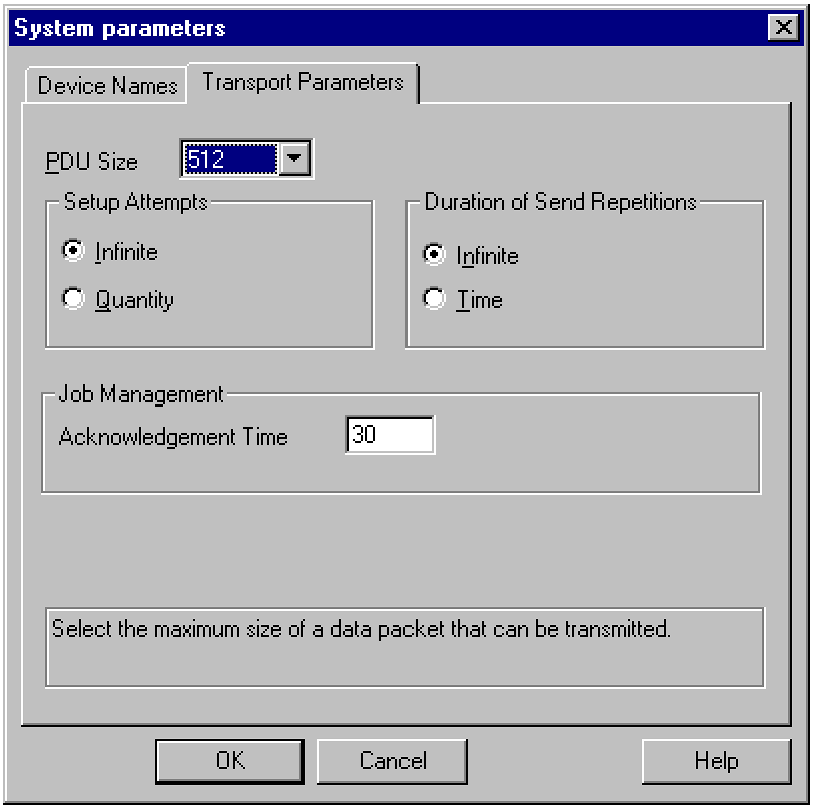
Because communication takes place via the ISO transport protocol, it is not necessary to configure the logical connection in the local database.

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.



System parameters

PDU Size: 512

Setup Attempts: ● Infinite ○ Quantity

Duration of Send Repetitions: ● Infinite ○ Time

Job Management

Acknowledgement Time: 30

Select the maximum size of a data packet that can be transmitted.

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).

7.15 SIMATIC TI Serial

7.15.1 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.

7.15.2 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

7.15.3 Configuring the Channel

7.15.3.1 Configuring the "SIMATIC TI Serial" channel

Introduction

The following steps are required for configuring the channel "SIMATIC TI Serial".

7.15.3.2 How to configure the connection

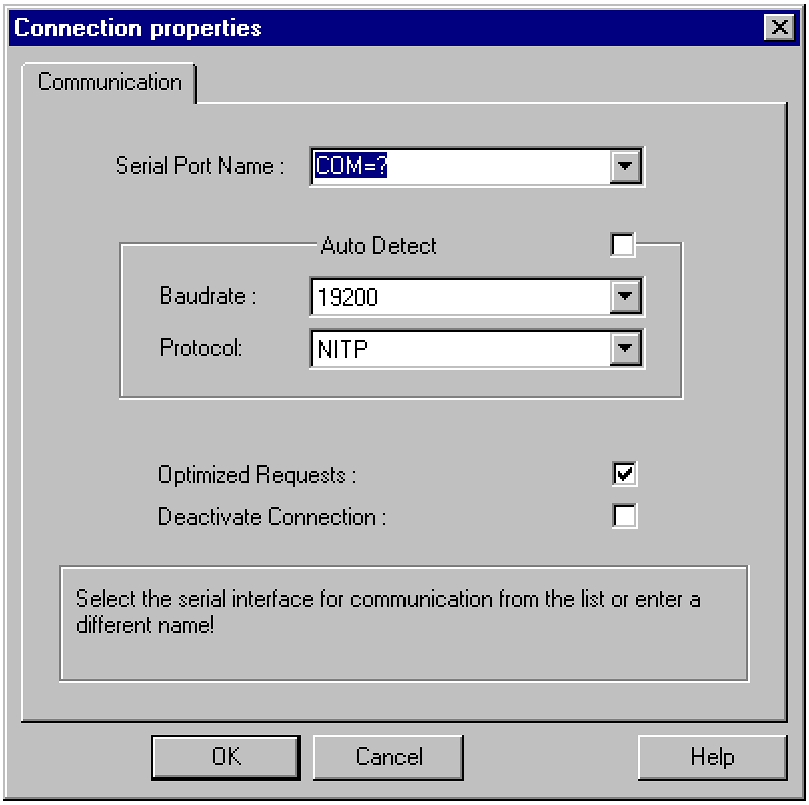
Introduction

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.



Connection properties

Communication

Serial Port Name : COM=?

Auto Detect □

Baudrate : 19200

Protocol: NITP

Optimized Requests : ☑

Deactivate Connection : □

Select the serial interface for communication from the list or enter a different name!

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.

7.15.3.3 Configuring the tags

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.

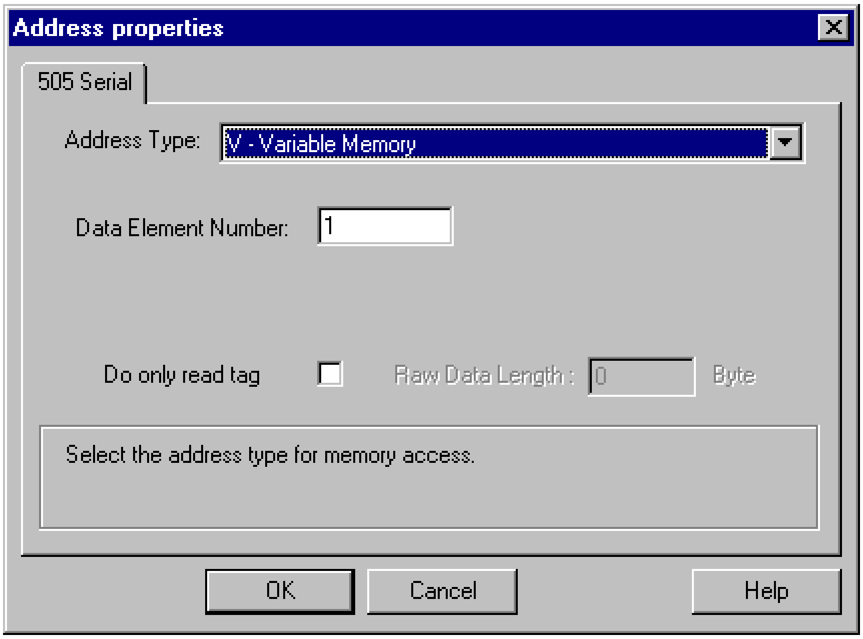
How to configure the address of a tag

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.



Address properties

505 Serial

Address Type: V - Variable Memory

Data Element Number: 1

Do only read tag □ Raw Data Length : 0 Byte

Select the address type for memory access.

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.

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.

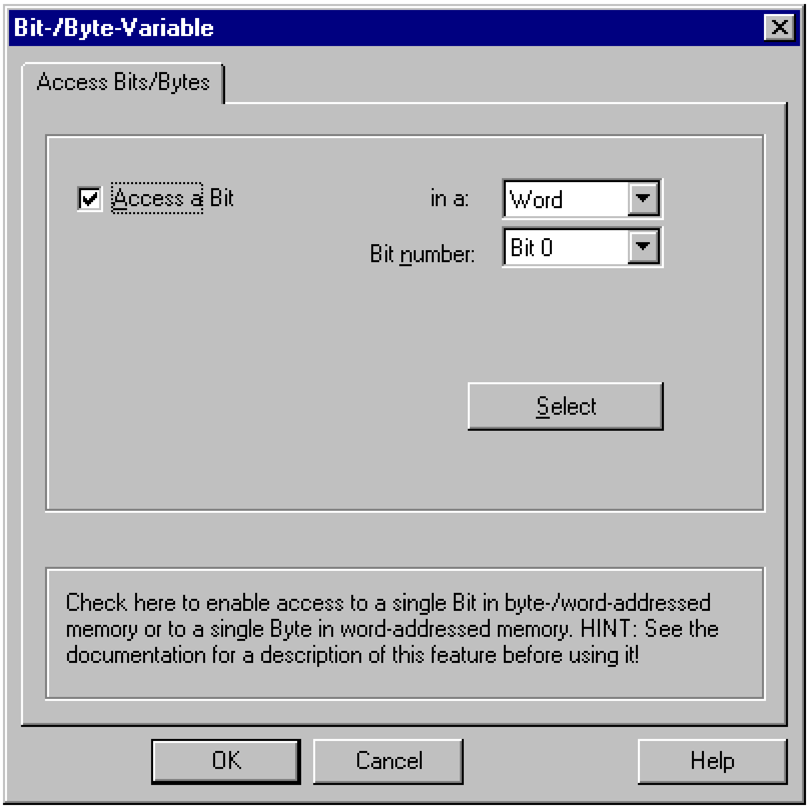
Note

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.



Bit-/Byte-Variable

Access Bits/Bytes

☑ Access a Bit

in a: Word

Bit number: Bit 0

Check here to enable access to a single Bit in byte-/word-addressed memory or to a single Byte in word-addressed memory. HINT: See the documentation for a description of this feature before using it!

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.

How to Configure a Tag with Byte by Byte Access

Introduction

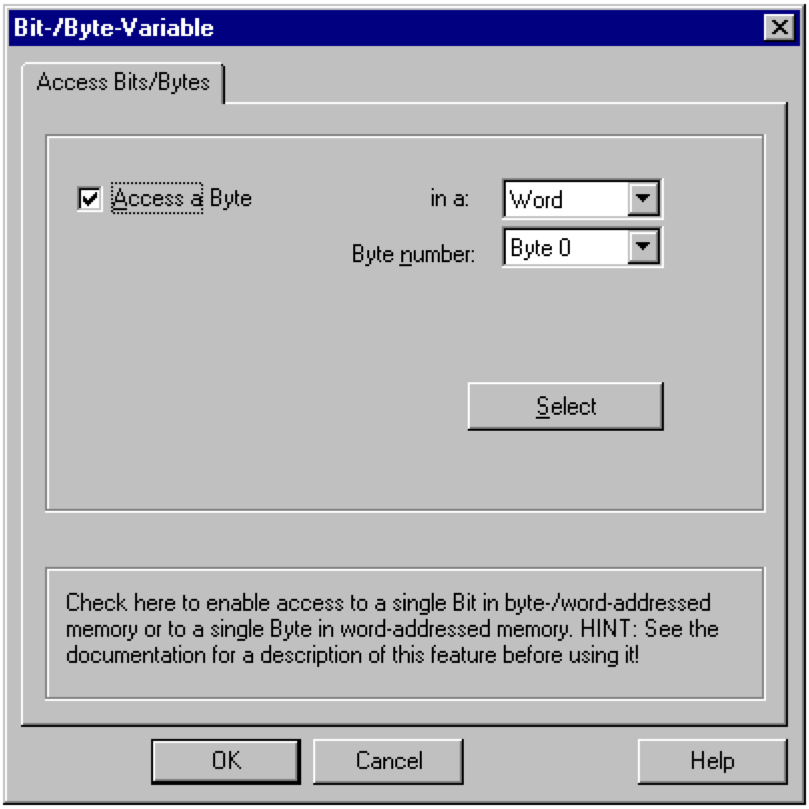
Write access to memory areas in the AS can only performed bit-wise or byte-wise in channel "SIMATIC TI Serial". When using binary and "8 Bit" tags, dialog "Bit-Byte-tag" is opened in addition to dialog "Address properties" and this can be used to define write access to individual bits or bytes. For this 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.



Bit-/Byte-Variable

Access Bits/Bytes

☑ Access a Byte

in a: Word

Byte number: Byte 0

Check here to enable access to a single Bit in byte-/word-addressed memory or to a single Byte in word-addressed memory. HINT: See the documentation for a description of this feature before using it!

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.

7.16 SIMOTION

7.16.1 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.

7.16.2 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

Data Types	Type conversion
Unsigned 32-bit value	Yes
Floating-point number 32-bit IEEE 754	Yes
Text tag, 8-bit font	No
Raw data type	No

7.16.3 Configuring the channel

7.16.3.1 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.

7.16.3.2 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.

7.16.3.3 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.

7.16.3.4 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.

7.16.3.5 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

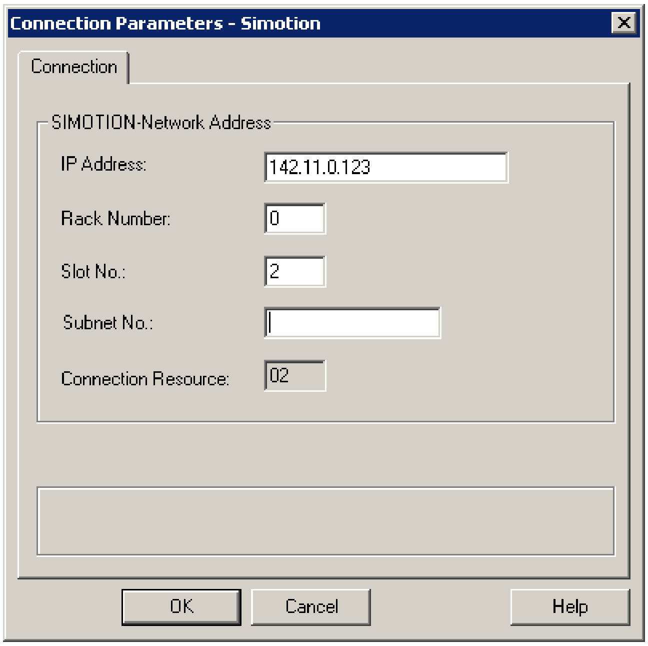
Change only the connection parameters listed here. Do not create new connection for the "SIMOTION" channel. Incorrectly set connections may result in control errors in the PLC. Configure new connections according to the description in the section "Configuration of the "SIMOTION" channel (Page 557)".

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.



Connection Parameters - Simotion

Connection

SIMOTION-Network Address

IP Address: 142.11.0.123

Rack Number: 0

Slot No.: 2

Subnet No.:

Connection Resource: 02

3. Change the connection parameters for the SIMOTION network address in the respective fields.
4. Close each open dialog box by clicking "OK."

7.16.3.6 How to change the tag address

Introduction

This section describes how to change a tag address in the "SIMOTION" channel.

Note

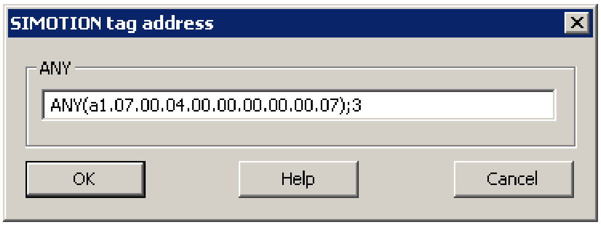
You must have very good knowledge of the use of ANY pointers to change the tag address in the "SIMOTION" channel. No communication connection may be established if the tag address is entered incorrectly.

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.



ANY(a1.07.00.04.00.00.00.00.00.07);3

2. Change the tag address.

7.16.3.7 System parameter configuration

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.

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

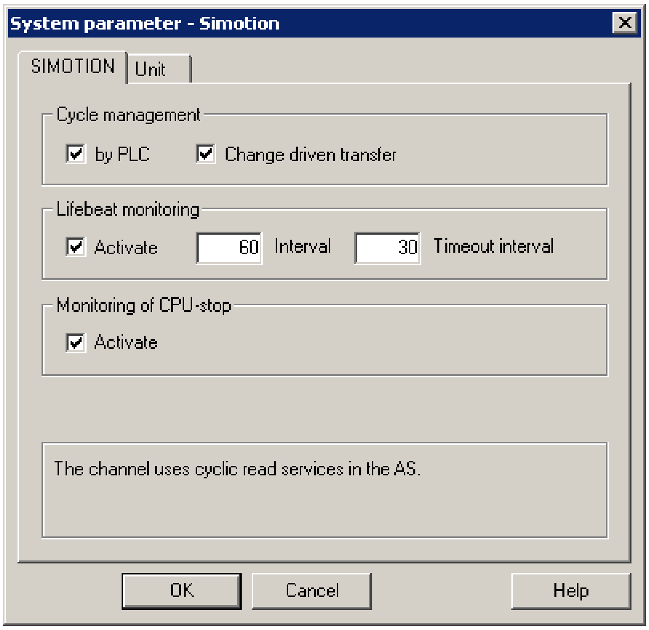
When the project is copied to another computer, the settings in the "Unit" tab are retained. The settings on the "SIMOTION" tab are deleted on the other hand.

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.



System parameter - Simotion

SIMOTION | Unit

Cycle management

☑ by PLC ☑ Change driven transfer

Lifebeat monitoring

☑ Activate 60 Interval 30 Timeout interval

Monitoring of CPU-stop

☑ Activate

The channel uses cyclic read services in the AS.

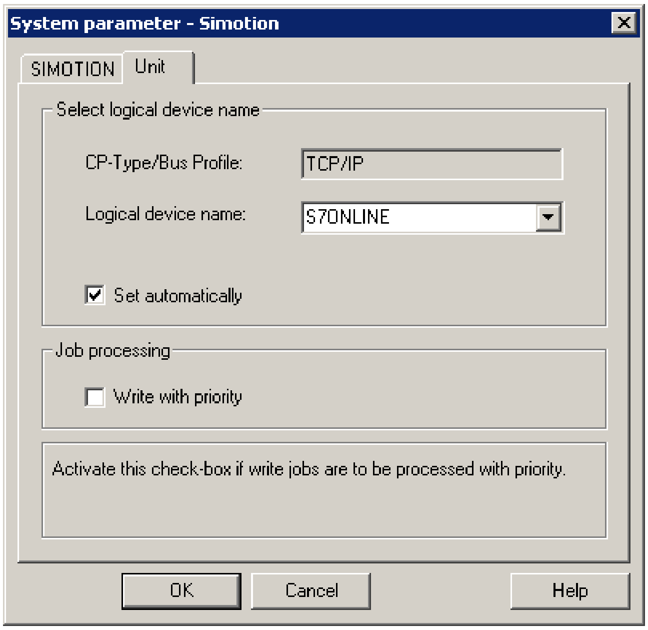
3. To enable cyclic reading of tags and change-driven transfer, select "by PLC" and "Change-driven transfer."

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.


System parameter - Simotion

Select logical device name

CP-Type/Bus Profile: TCP/IP

Logical device name: S7ONLINE

☑ Set automatically

Job processing

□ Write with priority

Activate this check-box if write jobs are to be processed with priority.

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.

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.

7.16.4 Diagnosis "SIMOTION" channel

7.16.4.1 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.

7.16.4.2 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

Message text	Meaning
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.

7.17 SINUMERIK

7.17.1 "SINUMERIK 840D" channel

Introduction

The "SinumerikNC" channel is used for communication between a WinCC station and CNCs of the type "SINUMERIK 840D".

Communication is via the Ethernet IP protocol.

Communication via MPI (Multi-Point Interface) is supported. Simultaneous use of both protocols is not possible.

Support: Complete documentation for downloading

You can find the "Communication channel SINUMERIK" manual with comprehensive information for configuring the connections and tags on the Internet at "SiePortal Knowledge Base":

- SiePortal: WinCC V8.0 SINUMERIK (Entry ID 109816693) (https://support.industry.siemens.com/cs/ww/en/view/109816693)

Channel units

The "SinumerikNC" channel has the channel unit "SINUMERIK 840D sl".

See also

SiePortal: WinCC V8.0 SINUMERIK (Entry ID 109816693) (https://support.industry.siemens.com/cs/ww/en/view/109816693)

7.17.2 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

Golbal User Data (GUD) tags of the SINUMERIK NC are supported and can be reached in Runtime.

7.17.3 Configuring the channel

7.17.3.1 Configuring the "SINUMERIK 840D" Channel

Introduction

WinCC needs a logical connection for communication of WinCC with the automation system (AS). This section describes the communication with the "SinumerikNC" 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.

7.17.3.2 How to configure a "SINUMERIK 840D" channel connection

Introduction

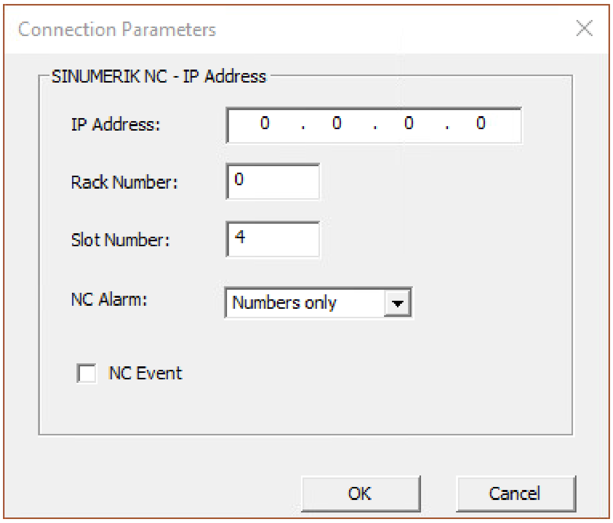
This section shows you how to configure a connection for the "SinumerikNC" channel unit.

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 8400 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.



Connection Parameters

SINUMERIK NC - IP Address

IP Address: 0 . 0 . 0 . 0

Rack Number: 0

Slot Number: 4

NC Alarm: Numbers only

□ NC Event

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.
8. Close the dialog with "OK".

See also

SiePortal: WinCC V8.0 SINUMERIK (Entry ID 109816693) (https://support.industry.siemens.com/cs/ww/en/view/109816693)

7.17.3.3 How to configure alarms

1. Open the "Connection Parameters" dialog.
2. To activate NC alarms, select under "NC Alarm" whether alarm numbers only, or alarm texts and alarm numbers are to be passed on.
   Default: No alarms are passed on.
3. To apply NC events, activate the "NC Event" option.
   Default: No NC events are transferred.

See also

How to configure a "SINUMERIK 840D" channel connection (Page 571)

7.17.3.4 Configuring the tags

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.

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).

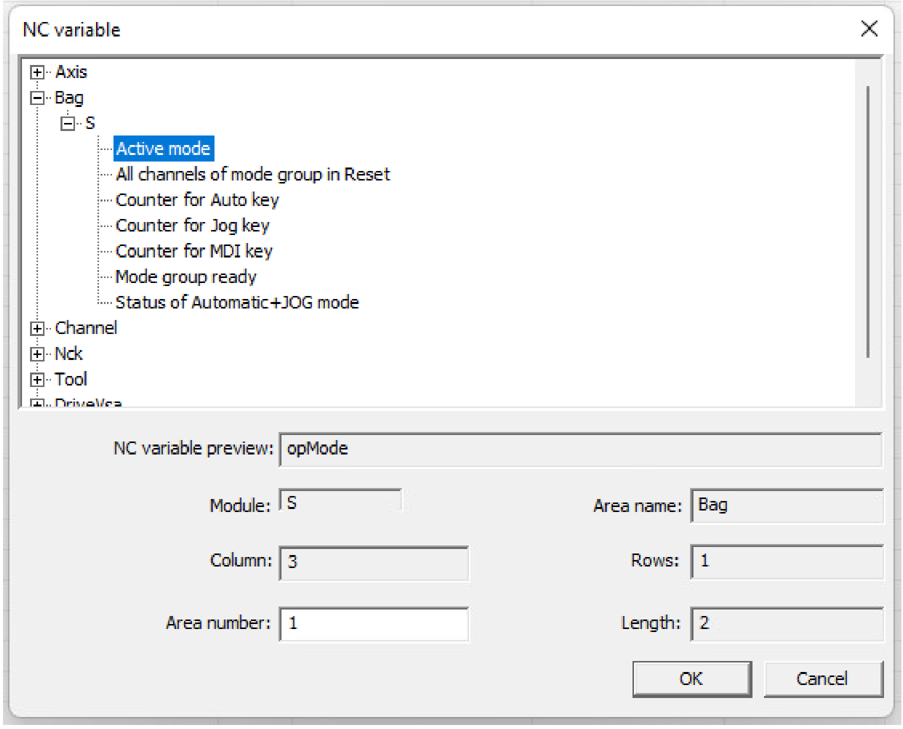
Requirements

- 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.



NC variable preview: opMode

Module: S

Area name: Bag

Column: 3

Rows: 1

Area number: 1

Length: 2

7. Select the corresponding NC variable in the tree view.

   Note

   After the address has been selected, the data type is automatically adapted.

   Change the area number if required.

8. Specify the other tags of the property, for example, limit values in the "Properties - Tag" area.

See also

Format adaptation sorted by WinCC data type (Page 199)

SiePortal: WinCC V8.0 SINUMERIK (Entry ID 109816693) (https://support.industry.siemens.com/cs/ww/en/view/109816693)

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.

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 GUD tags are imported and are available in the Tag Management.

7.17.4 Executing system functions

7.17.4.1 Supported system functions

The following system functions are available for configuration of the channel:

- LogonNC
- LogoffNC
- ChangeNCPassword
- ConfigureNCMachineData
- ResetNC
- AcknowledgeNCCancelAlarms
- SetNCUserFrame
- StartNCPIService

7.17.4.2 LogonNC

The function transfers a password to the SINUMERIK NC.

Parameter	Explanation
Connection	Connection name
Password	The password for the logon at the SINUMERIK NC

Example VB script:

```
Dim output
LogonNC "Connection", "Password", output
Hmiruntime.Trace output
```

7.17.4.3 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
```

7.17.4.4 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
```

7.17.4.5 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
```

7.17.4.6 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
```

7.17.4.7 AcknowledgeNCCancelAlarms

The function is used to acknowledge all pending alarms of a connection.

Parameter	Explanation
Connection	Connection name

Example VB script:

```
Dim output
AcknowledgeNCCancelAlarms "Connection", output
Hmiruntime.Trace output
```

7.17.4.8 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.

7.18 System Info

7.18.1 "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

7.18.2 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 596)

How To Call Up and Evaluate System Information (Page 585)

How to Configure the System Info Channel (Page 584)

Differences to Other Software Components (Page 584)

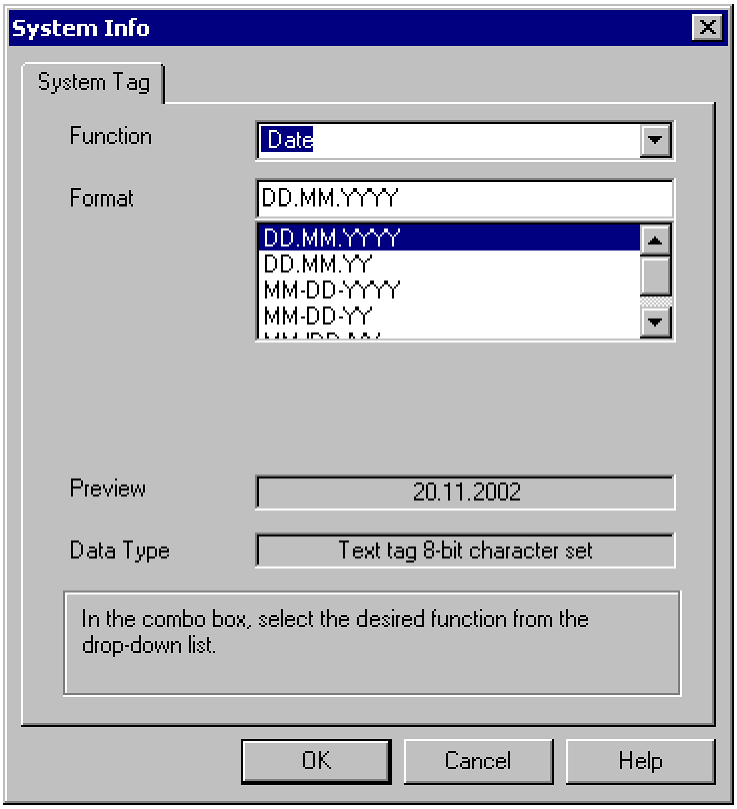
Overview of the Supported System Information (Page 580)

Diagnosis of Channels and Tags (Page 609)

7.18.3 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.



System Info

System Tag

Function: Date

Format: DD.MM.YYYY

DD.MM.YYYY
DD.MM.YY
MM-DD-YYYY
MM-DD-YY

Preview: 20.11.2002

Data Type: Text tag 8-bit character set

In the combo box, select the desired function from the drop-down list.

System Info Channel Supported System Information - Overview

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
Weekday	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
Milliseconds	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 load in % process load in %	0...100%
Timer	Signed 32-bit value	TTTT	0...9999
Free main memory	Floating-point number 32-bit IEEE 754	Free capacity 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 swap file	Floating-point number 32-bit IEEE 754	Used in kB Used in % Available in kB	0...n kB 0...100% 0...n kB

Counter

This function is useable for test purposes in scripts.

Timer

When this function is selected, the "System Info" dialog is extended with 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 starting and ending 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 in Runtime, a value is entered in the I/O field linked to the timer, the start and current timer values is 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 enter the CPU number beginning with "0".
For the format "Process load in %" enter the instance number, if there are several instances of a process.

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. From 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 assessed 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 for 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

Status	Error code
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 corresponds to the values in the Visual C-referenced file "Winspool.h" .

7.18.4 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.

7.18.5 Configuring the Channel

7.18.5.1 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.

7.18.6 Examples of Evaluating and Displaying System Information

7.18.6.1 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 593)

How to Configure a Message Regarding Free Disk Capacity (Page 590)

How to Display the CPU Load in a Trend Window (Page 589)

How to Display the Free Disk Capacity in a Bar Graph (Page 588)

How to Display the Time in an I/O Field (Page 587)

How to Configure a Tag in the System Info Channel (Page 585)

7.18.6.2 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
Trends	Sysinfo_CPU	CPU load	Number 0-100% (total load as %)	Floating-point number 32-bit IEEE 754
Printer status	Sysinfo_Printerstate	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 590)

How to Configure the System Info Channel (Page 584)

How to Display the Printer Status in a Status Display (Page 593)

How to Display the CPU Load in a Trend Window (Page 589)

How to Display the Free Disk Capacity in a Bar Graph (Page 588)

How to Display the Time in an I/O Field (Page 587)

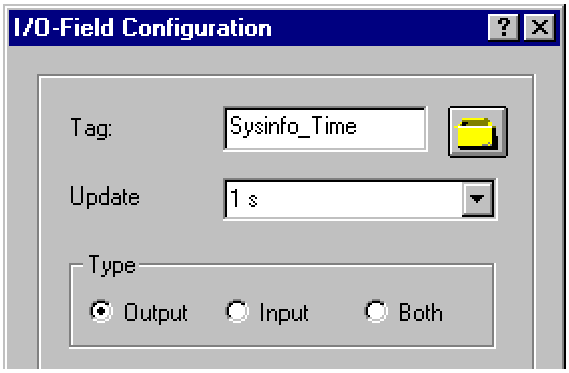
7.18.6.3 How to Display the Time in an I/O Field

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.



I/O-Field Configuration

Tag: Sysinfo_Time

Update 1 s

Type

● Output ○ Input ○ Both

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 596)

How to Insert an I/O Field (Page 596)

How to Configure a Tag in the System Info Channel (Page 585)

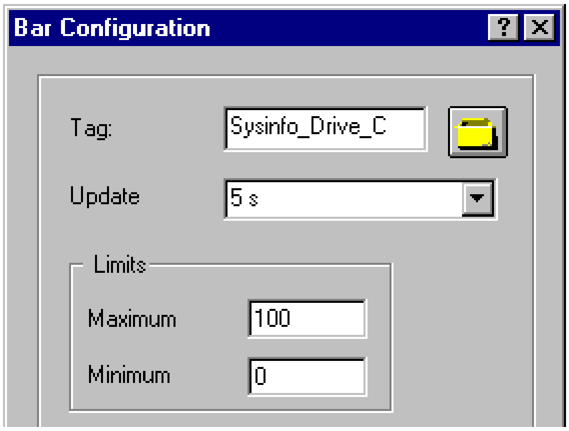
7.18.6.4 How to Display the Free Disk Capacity in a Bar Graph

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.



Bar Configuration

Tag: Sysinfo_Drive_C

Update: 5 s

Limits

Maximum: 100

Minimum: 0

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 596)

How to Insert a Bar Graph (Page 595)

How to Configure a Tag in the System Info Channel (Page 585)

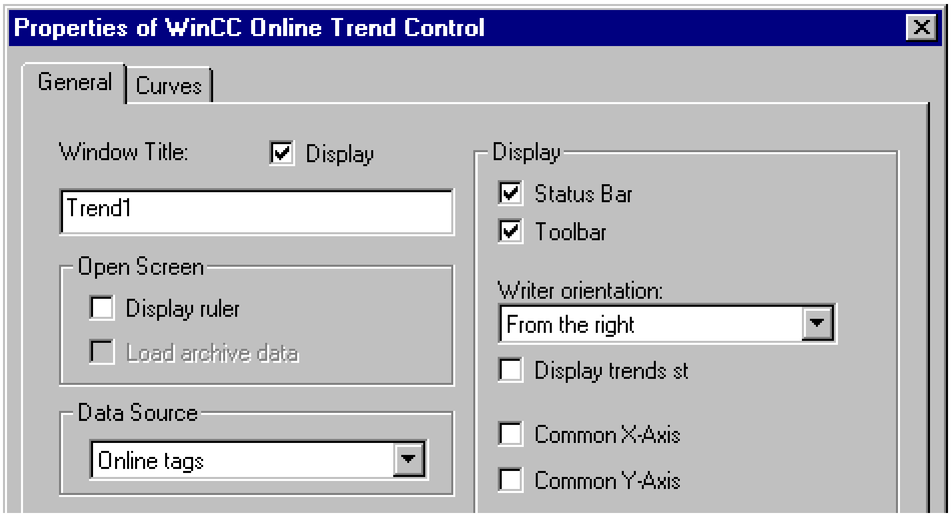
7.18.6.5 How to Display the CPU Load in a Trend Window

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.



Properties of WinCC Online Trend Control

General | Curves

Window Title: ☑ Display

Trend1

Open Screen
□ Display ruler
□ Load archive data

Data Source
Online tags

Display
☑ Status Bar
☑ Toolbar

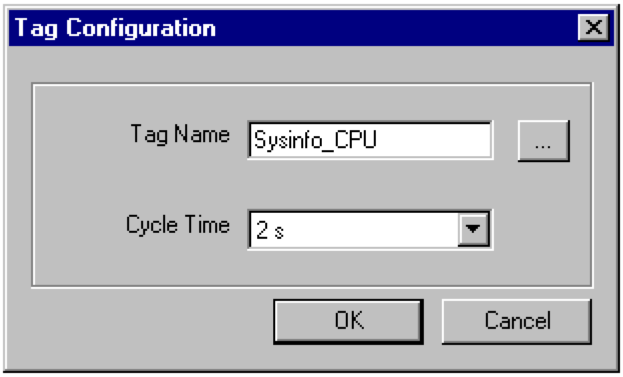
Writer orientation:
From the right

□ Display trends st

□ Common X-Axis
□ Common Y-Axis

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.



Tag Configuration

Tag Name Sysinfo_CPU

Cycle Time 2 s

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 596)

How to Configure a Tag in the System Info Channel (Page 585)

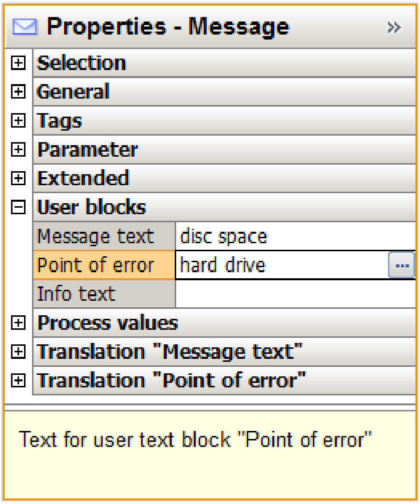
7.18.6.6 How to Configure a Message Regarding Free Disk Capacity

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"

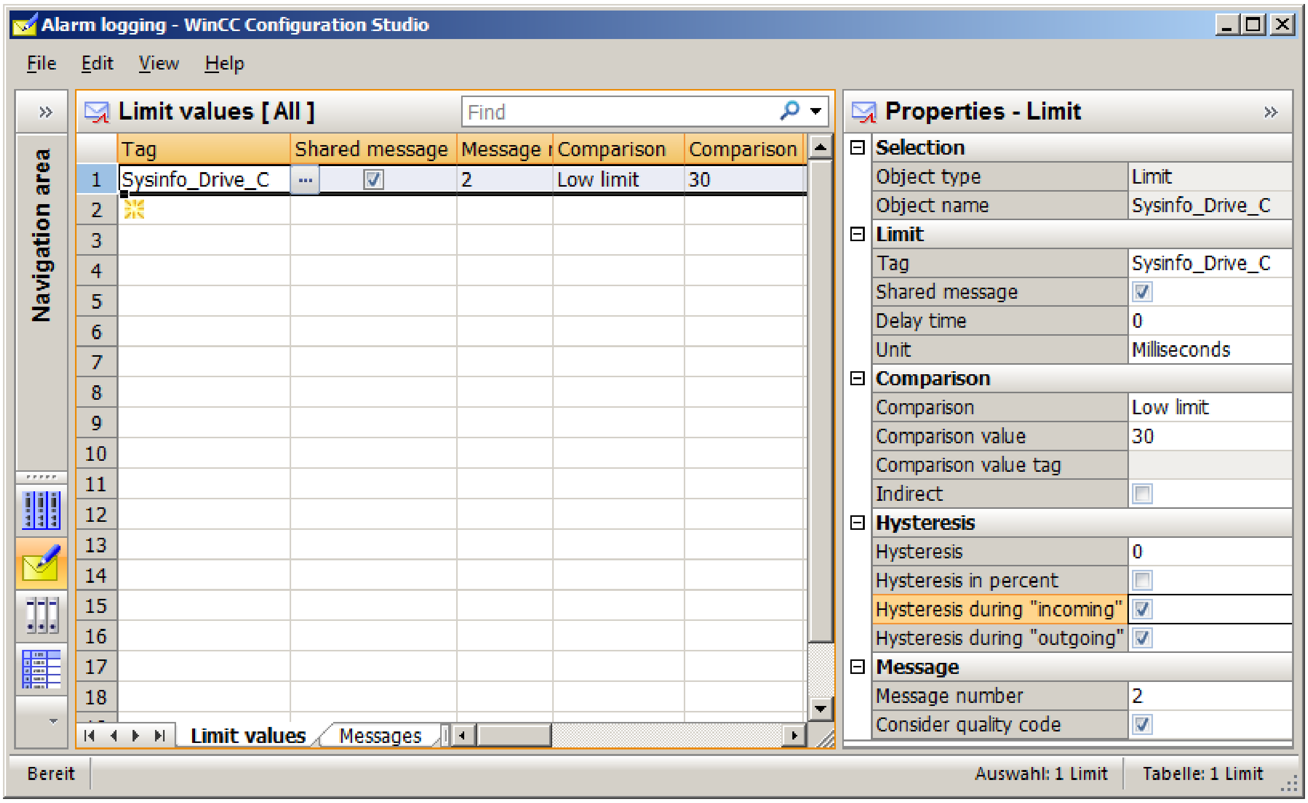


Message text: disc space
Point of error: hard drive
Info text:

Text for user text block "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.



Limit values [All]

	Tag	Shared message	Message	Comparison	Comparison
1	Sysinfo_Drive_C	☑	2	Low limit	30
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					

Properties - Limit

Selection	
Object type	Limit
Object name	Sysinfo_Drive_C
Limit	
Tag	Sysinfo_Drive_C
Shared message	☑
Delay time	0
Unit	Milliseconds
Comparison	
Comparison	Low limit
Comparison value	30
Comparison value tag	
Indirect	□
Hysteresis	
Hysteresis	0
Hysteresis in percent	□
Hysteresis during "incoming"	☑
Hysteresis during "outgoing"	☑
Message	
Message number	2
Consider quality code	☑

See also

How to Configure a Tag in the System Info Channel (Page 585)

How to Display a Message regarding the Available Disk Capacity (Page 592)

7.18.6.7 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 595)

How to Start Runtime (Page 596)

How to Configure a Tag in the System Info Channel (Page 585)

7.18.6.8 How to Display the Printer Status in a Status Display

Introduction

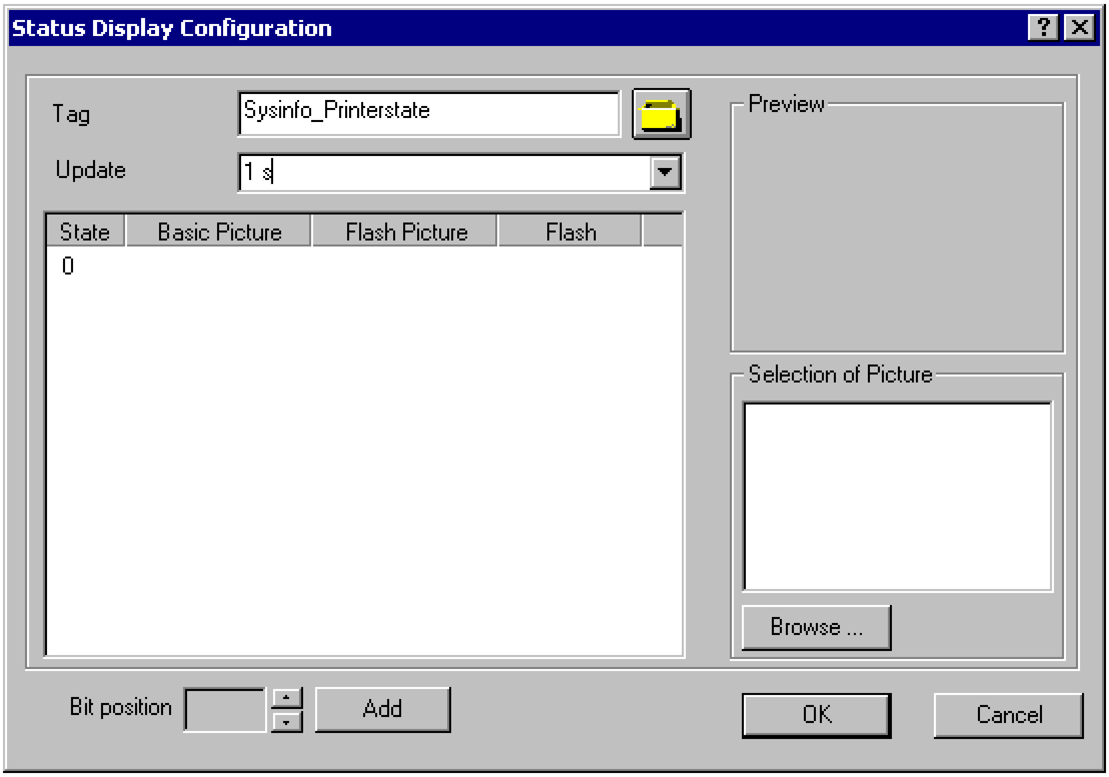
This example shows a possible evaluation of the "Printer monitoring" system information based on the "Printer status" display format. The printer or job status could also be evaluated by configuring messages that are triggered by single bits in the printer or job status.

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.



Status Display Configuration

Tag: Sysinfo_Printerstate

Update: 1 s

State	Basic Picture	Flash Picture	Flash
0			

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 585)

How to Start Runtime (Page 596)

7.18.6.9 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.

7.18.6.10 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.

7.18.6.11 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.

7.18.6.12 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.

7.18.7 Special Functions

7.18.7.1 Use in Multi-User and Client Systems

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 597)

7.18.7.2 Example of monitoring system information from multiple servers

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 602)

How to Configure the Process Picture on the WinCC Client (Page 601)

How to Import the Tags to the WinCC Client (Page 600)

How to Configure the Second Server (Page 599)

How to Configure the First Server (Page 597)

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 599)

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 600)

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 601)

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 602)

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.

7.19 WinCC Unified Channel

7.19.1 Channel "WinCC Unified Channel"

Introduction

The "WinCC Unified Channel" channel enables communication in 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".

You can find additional information here: 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)

7.19.2 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.

7.19.3 Configuring the channel

7.19.3.1 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

After duplicating a project, the collaboration data must be adjusted, as it must be unique among all collaboration systems involved.

See also

Introduction to the WinCC Certificate Manager (Page 149)

7.19.3.2 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.

Introduction

The following steps are required for configuring the "WinCC Unified Channel":

1. Configuring the channel
2. Configuring a connection
3. Configuring tags

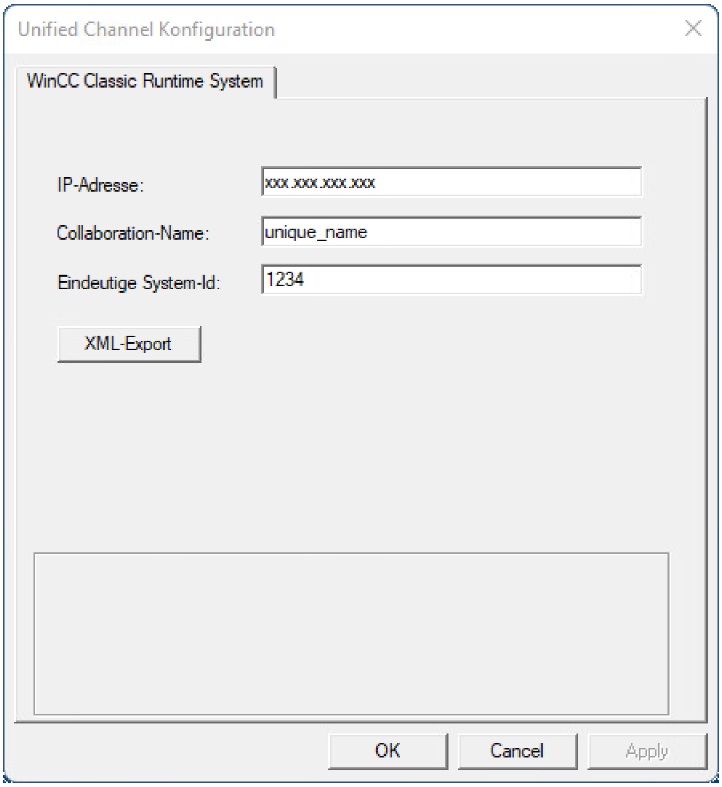
Requirements

- 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.



IP-Adresse: xxx.xxx.xxx.xxx

Collaboration-Name: unique_name

Eindeutige System-Id: 1234

4. Enter the IP address of the WinCC system.
5. Specify the unique "Collaboration name".
6. Specify the unique system ID.

Note

The system ID and collaboration name must be unique throughout the system among all participating collaboration systems.

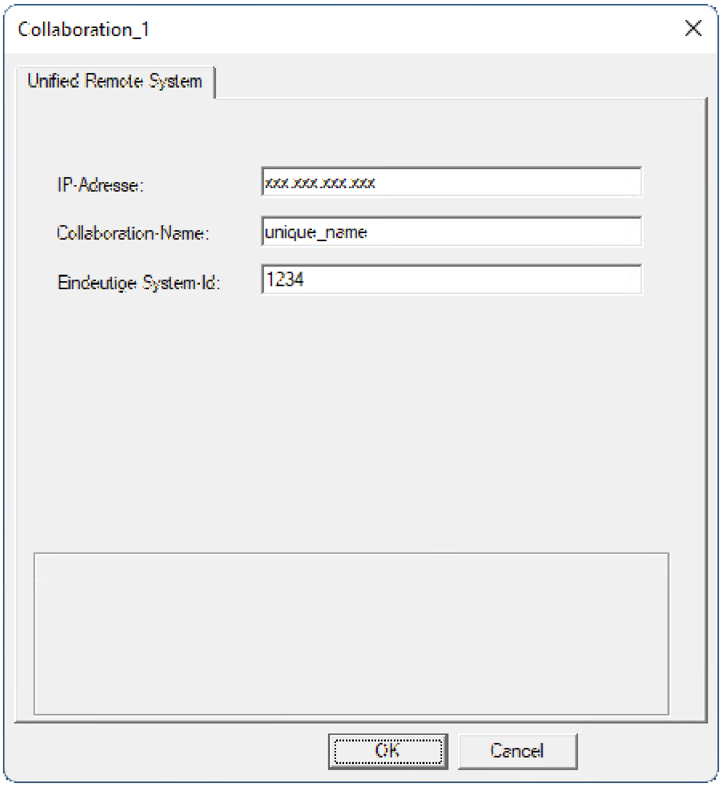
7.19.3.3 How to configure a connection

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.



IP-Adresse: xxx.xxx.xxx.xxx
Collaboration-Name: unique_name
Eindeutige System-Id: 1234

The data you enter 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. Specify the "Collaboration name".
7. Specify the unique system ID.

Note

The system ID and collaboration name must be unique throughout the system among all participating collaboration systems.

7.19.3.4 Configuring the tags

Collaboration allows you to share data with other systems.

Tags are read into WinCC by import. You can use imported tags like other tags in pictures, controls, or scripts, for example. The tags can be archived by WinCC.

Note

No structured tags can be used

Structure tags are read in flat without preserving the structure.

Note

Version compatibility

Tags of WinCC Unified Version V18 and V17 can be used.

How to configure tags

Introduction

This section describes how to configure the sharing of tags as part of collaboration.

Requirements

- Collaboration is activated and configured in WinCC Unified.
- In WinCC, the "WinCC Unified Channel" channel is created and a connection is configured.

Importing tags

1. In the shortcut menu of the connection, select the entry "Unified Import" under the "WinCC Unified Channel" channel.
   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.

Communication - Diagnostics 8

8.1 Diagnosis of Channels and Tags

This section describes the diagnostics of channels and their tags, as well as of internal tags.

These diagnostics can be used, for example, in the event of communication problems or unexpected tag values.

The following documentation describes:

- How to recognize communication errors.
- How to configure and use the following diagnostics tools:
  - "Status - Logical Connections"
  - "WinCC Channel Diagnosis Control"
  - System tags of the "Performance" tag group
- How to diagnose channels, connections and their tags.
- How to diagnose internal tags.
- How to check the WinCC communications hardware.

See also

Diagnostic Options for the "SIMATIC S5 PROFIBUS FDL" Channel (Page 649)

Quality Codes of Tags (Page 669)

Monitoring Tag Status Using Global Actions (Page 677)

Using the Tag Status to Monitor Connection Status (Page 675)

How to Check an Internal Tag (Page 678)

Channel diagnosis (Page 611)

General Information about Error Detection (Page 610)

Possibilities for Diagnosing the "OPC" Channel (Page 658)

"SIMATIC S7 Protocol Suite" Channel - Diagnostic Options (Page 638)

"System Info" Channel - Diagnostic Options (Page 625)

Diagnostics channel "SIMATIC S7-1200/S7-1500" (Page 630)

8.2 General Information about Error Detection

A fault or error in establishing a communication link is generally first detected in Runtime.

Objects dynamized using WinCC tags, which cannot be supplied with current process values, are displayed in the process picture as inactive. These could be e.g. I/O fields, slider objects or bar graphs.

If the fault does not affect some of a connection's WinCC tags, this indicates that one of the WinCC tags is the source of the trouble. In this case, you should for example check the addressing of the tags as well as their spelling when used in Graphics Designer.

If the fault affects all of a connection's WinCC tags, this indicates a fault in the connection itself.

The following sections describe which measures and means can be used to pinpoint the source of the error.

8.3 Channel Diagnosis

8.3.1 Channel diagnosis

The following functions are available for diagnostics of channels and their connections:

- "Status - Logical Connections" function
- System tags of the "Performance" tag group
- WinCC "Channel diagnosis"

See also

How to Use the "Status - Logical Connections" Function to Check a Channel (Page 614)

Principle of Channel Diagnosis (Page 616)

Check connection with performance tags (Page 611)

8.3.2 Check connection with performance tags

WinCC provides the following system tags for analyzing a communication channel:

- @PRF_DMRT_CHNCON_... (Tag Management)
- @PRF_ALGRT_CHNCON_... (Alarm Logging)

You can evaluate the time behavior of a connection with these performance tags.

Use the @<...>@ConnectionStateEx system tag to determine the status of a connection.
More information:

- "How to Use the "Status - Logical Connections" Function to Check a Channel (Page 614)"
- "Configuring tags for the connection status in Runtime (Page 193)"

Creating performance tags

As soon as you create a new connection under the communication driver, WinCC Tag Management creates the corresponding performance tags.

When you rename the connection, the performance tags are also automatically renamed.

The tags are located in the "Performance" tag group in the "Internal tags" area.

Additional information on WinCC performance tags:

- "Working with WinCC > Working with Projects > Making Settings for Runtime":
  - "System diagnostics with performance tags"
  - "Overview of performance tags"

Types of performance tags

The "Performance" tag group contains the following tag types:

Tags	Data type	Access	Description
Relative tags	Floating-point number 64-bit IEEE 754	Read	Values which apply relatively to the time of reading, e.g. currently pending values or values per second. The reset tag does not have an influence on these values. The tag name ends in: • ..._PENDING • ..._SECOND Update cycle: 1 second
Counter tags	Floating-point number 64-bit IEEE 754	Read	Absolute values since Runtime activation You can reset the value to "0" by using the reset tag. The tag name ends in: • ..._AVERAGE • ..._PEAK • ..._TOTAL Update cycle: 1 second
Reset tags	Unsigned 32-bit value	Read Write	You can set the value of the reset tags via scripts, for example: • 0: Disabled • 1: The value of all associated counter tags is reset to "0". The value of the reset tag itself is also reset to "0". The tag name ends in: • ...RESET

Overview of performance tags

System tag [1)	Description
Alarm Logging	
@PRF_ALGRT_RESET	The reset tag resets the values of the following performance tags: • @PRF_ALGRT_CHNCON_..._AVERAGE • @PRF_ALGRT_CHNCON_..._PEAK The @PRF_ALGRT_... counter tags without reference to the communication driver are also reset.
@PRF_ALGRT_CHNCON_<...>_ALARMS_PER_SECOND [2)	Number of generated messages per second sent over the connection
@PRF_ALGRT_CHNCON_<...>_ALARMS_PER_SECOND_AVERAGE	Average number of messages per second
@PRF_ALGRT_CHNCON_<...>_ALARMS_PER_SECOND_PEAK	Maximum number of messages per second

System tag [1]	Description
Tag Management	
@PRF_DMRT_CHNCON_<...>_RESET	The reset tag resets the values of the following performance tags: • @PRF_DMRT_CHNCON_<...>..._TOTAL • @PRF_DMRT_CHNCON_<...>_RESET The reset applies to all counter tags that were created for the same connection.
@PRF_DMRT_CHNCON_<...>_TAG_READ_BYTES_PER_SECOND [2]	Bytes read/second Bits are rounded up to a byte. Metadata, e.g. time stamps or SetValue callback data, is not included.
@PRF_DMRT_CHNCON_<...>_TAG_READ_BYTES_TOTAL	Bytes read since activation of Runtime Bits are rounded up to a byte. Metadata, e.g. time stamps or SetValue callback data, is not included.
@PRF_DMRT_CHNCON_<...>_TAG_READS_PENDING	Started read requests that have not yet been completed A constantly rising value indicates a system overload. Possible cause: • A data source or connection does not process the read requests quickly enough because it is overloaded or blocked.
@PRF_DMRT_CHNCON_<...>_TAG_READS_PER_SECOND [2]	Tags read/second
@PRF_DMRT_CHNCON_<...>_TAG_READS_TOTAL	Tags read since activation of Runtime
@PRF_DMRT_CHNCON_<...>_TAG_WRITES_PENDING	Started write requests that have not yet been completed A constantly rising value indicates a system overload. Possible cause: • A data source or connection does not process the write requests quickly enough because it is overloaded or blocked.
@PRF_DMRT_CHNCON_<...>_TAG_WRITES_PER_SECOND 2)	Tags written/second
@PRF_DMRT_CHNCON_<...>_TAG_WRITES_TOTAL	Tags written since activation of Runtime
@PRF_DMRT_CHNCON_<...>_TAG_WRITTEN_BYTES_PER_SECOND [2]	Bytes written/second Bits are rounded up to a byte. Metadata, e.g. time stamps or SetValue callback data, is not included.
@PRF_DMRT_CHNCON_<...>_TAG_WRITTEN_BYTES_TOTAL	Bytes written since activation of Runtime Bits are rounded up to a byte. Metadata, e.g. time stamps or SetValue callback data, is not included.

1) <...> stands for the name of the communication connection, e.g. "@PRF_DMRT_CHNCON_S7-417_TAG_READS_PER_SECOND".

2) The information "PER_SECOND" relates to the last second before tag update

See also

Configuring tags for the connection status in Runtime (Page 193)

How to Use the "Status - Logical Connections" Function to Check a Channel (Page 614)

8.3.3 How to Use the "Status - Logical Connections" Function to Check a Channel

"Logical connections status" function

With the "Logical connections status" function, WinCC Explorer provides an option for displaying the current status of all configured connections in a simple form.

However, the status display is only possible in Runtime.

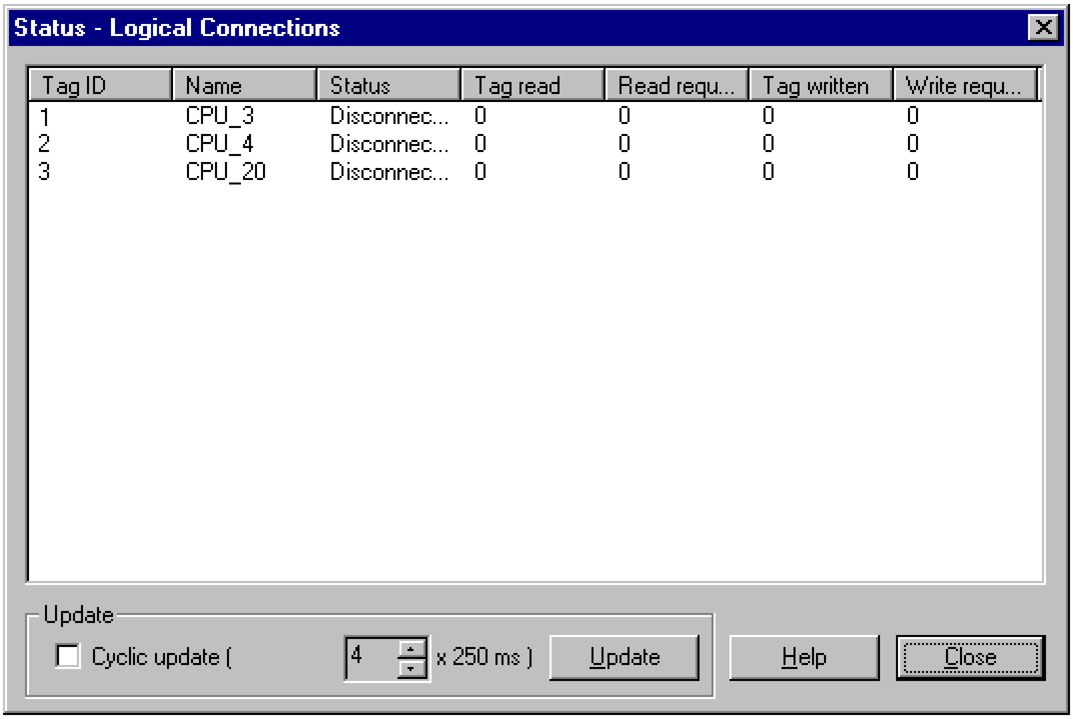
If you only want to query whether a particular connection is established or terminated, use the system tag "@<Connectionname>@ConnectionStateEx".

Requirements

- Configure a channel and create a tag in this channel.
- WinCC Runtime is activated.

Procedure

1. In the "Tools" menu on the WinCC Explorer menu bar, select the "Status of Driver Connections" entry.
   The "Status - Logical Connections" dialog opens.



Status - Logical Connections

Tag ID	Name	Status	Tag read	Read requ...	Tag written	Write requ...
1	CPU_3	Disconnec...	0	0	0	0
2	CPU_4	Disconnec...	0	0	0	0
3	CPU_20	Disconnec...	0	0	0	0

2. In the "Update" area, you can specify that an update is to be done cyclically.
   Otherwise, you can start an update of the display manually by clicking the "Update" button.
3. The configured connections are displayed in the "Name" column.
   The "Status" column displays the status of the respective connection.
4. Check the entries in the "Status" column.
   The "Disconnected" status indicates either a configuration error or hardware error.
   You can find additional information under "Channel diagnosis" of the associated channel.

See also

Configuring tags for the connection status in Runtime (Page 193)

Check connection with performance tags (Page 611)

8.3.4 Diagnosis of Channels with Channel Diagnosis

8.3.4.1 Principle of Channel Diagnosis

Introduction

The WinCC "Channel Diagnosis" enables WinCC users to gain a quick overview of the status of the active connections in Runtime. On the one hand, "Channel Diagnosis" provides status and diagnostic information on channel units and, on the other hand, it serves as a user interface for the configuration of the diagnostic output:

- The output of status/statistical information of the communication e.g. in a process picture
- Text output in a logbook file for fault analysis and correction by Service
- Text output in a trace file to assist the Hotline in pinpointing the cause of communication problems

The diagnostics module can be inserted into a process picture as an ActiveX control or be started as an independent application in Microsoft Windows.

The module only displays status information for channels that support diagnostics.

The diagnostics of a channel's tag can be found in the description of the channel-specific diagnostics.

Logbook file

"Channel Diagnosis" creates a logbook file called named <ChannelName.log> for every configured WinCC channel. Important information and errors is recorded in this. The exact text content depends on the channel.

The creation of the file and the output texts cannot be configured.

The logbook file contains information such as the start and end messages, version information, and information regarding communication errors.

Each entry in the file consists of a date and time stamp, the flag name and a description. The file is always saved immediately after an entry to ensure that in the event of a voltage drop, for example, as much of the information as possible is available.

Trace file

A trace file named <ChannelName.trc> to which additional information and errors are output can be created for every configured WinCC channel. You can select in Runtime whether to use a trace file. When the function is activated, a message is displayed warning that the link's runtime is affected.

Each entry in a trace file has a time stamp followed by a flag name and description.

When the trace function is enabled, all information recorded in the logbook is also written to the trace file.

The information recorded in a trace file is intended to assist the Hotline in pinpointing the cause of communication problems.

Note

The trace and logbook file entries are only recorded in English.

Both files are saved in the "Diagnostics" directory in the WinCC directory structure.

The current counter values are not output in these files.

See also

Channel Diagnosis with ActiveX Control (Page 617)

How to Check a Channel with Channel Diagnosis as an ActiveX Control (Page 617)

Diagnosing a Channel with "Channel Diagnosis" (Page 620)

How to Check a Channel with Channel Diagnosis (Page 621)

How to Configure the Trace Function of a Channel (Page 622)

How to Start Runtime (Page 623)

8.3.4.2 Channel Diagnosis with ActiveX Control

Introduction

The status information for a channel can also be displayed in a process picture by the "WinCC Channel Diagnosis Control" ActiveX control.

The ActiveX control is found in the "Controls" object palette in Graphics Designer and is simply inserted in a picture. The user can thus create e.g. a diagnostics process picture, in which he can view the status of the communication and other information in Runtime, without needing to reconfigure this arrangement every time.

See also

How to Check a Channel with Channel Diagnosis (Page 621)

Diagnosing a Channel with "Channel Diagnosis" (Page 620)

8.3.4.3 How to Check a Channel with Channel Diagnosis as an ActiveX Control

Introduction

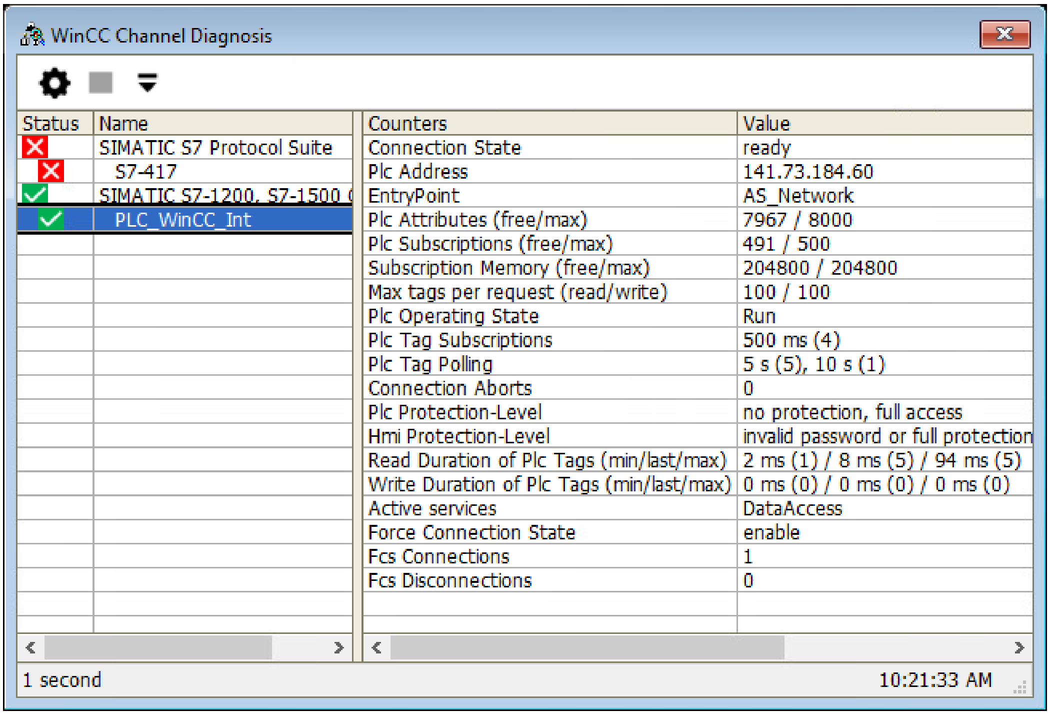
This section shows how to configure the diagnostics of a channel using the "WinCC Channel Diagnosis Control" ActiveX control.

Requirements

- Configure a channel and create a tag in this channel.

Procedure

1. Start Graphics Designer and open a picture.
2. Insert the "WinCC Channel Diagnosis Control" ActiveX control into the picture.
   To do this, select the ActiveX control from the "Controls" object palette, insert it into the picture, and stretch it to the desired size.
3. Save the picture.
4. Activate Runtime via the Graphics Designer toolbar.
5. Select the picture into which you inserted the ActiveX control.
   The status information for the channels appears in the "Channel Diagnosis" application window.



WinCC Channel Diagnosis

Status	Name
✗	SIMATIC S7 Protocol Suite
✗	S7-417
✓	SIMATIC S7-1200, S7-1500 (
✓	PLC_WinCC_Int

Counters	Value
Connection State	ready
Plc Address	141.73.184.60
EntryPoint	AS_Network
Plc Attributes (free/max)	7967 / 8000
Plc Subscriptions (free/max)	491 / 500
Subscription Memory (free/max)	204800 / 204800
Max tags per request (read/write)	100 / 100
Plc Operating State	Run
Plc Tag Subscriptions	500 ms (4)
Plc Tag Polling	5 s (5), 10 s (1)
Connection Aborts	0
Plc Protection-Level	no protection, full access
Hmi Protection-Level	invalid password or full protection
Read Duration of Plc Tags (min/last/max)	2 ms (1) / 8 ms (5) / 94 ms (5)
Write Duration of Plc Tags (min/last/max)	0 ms (0) / 0 ms (0) / 0 ms (0)
Active services	DataAccess
Force Connection State	enable
Fcs Connections	1
Fcs Disconnections	0

1 second 10:21:33 AM

6. Click the gear icon in the control.
   The "Configuration" dialog appears.
7. Select a channel from the "Channel name" drop-down list.
8. Specify a name for the log file.
9. Configure which error messages are to be recorded in the log file.

You can find more information about activating the Trace function under "How to Configure the Trace Function of a Channel (Page 622)".

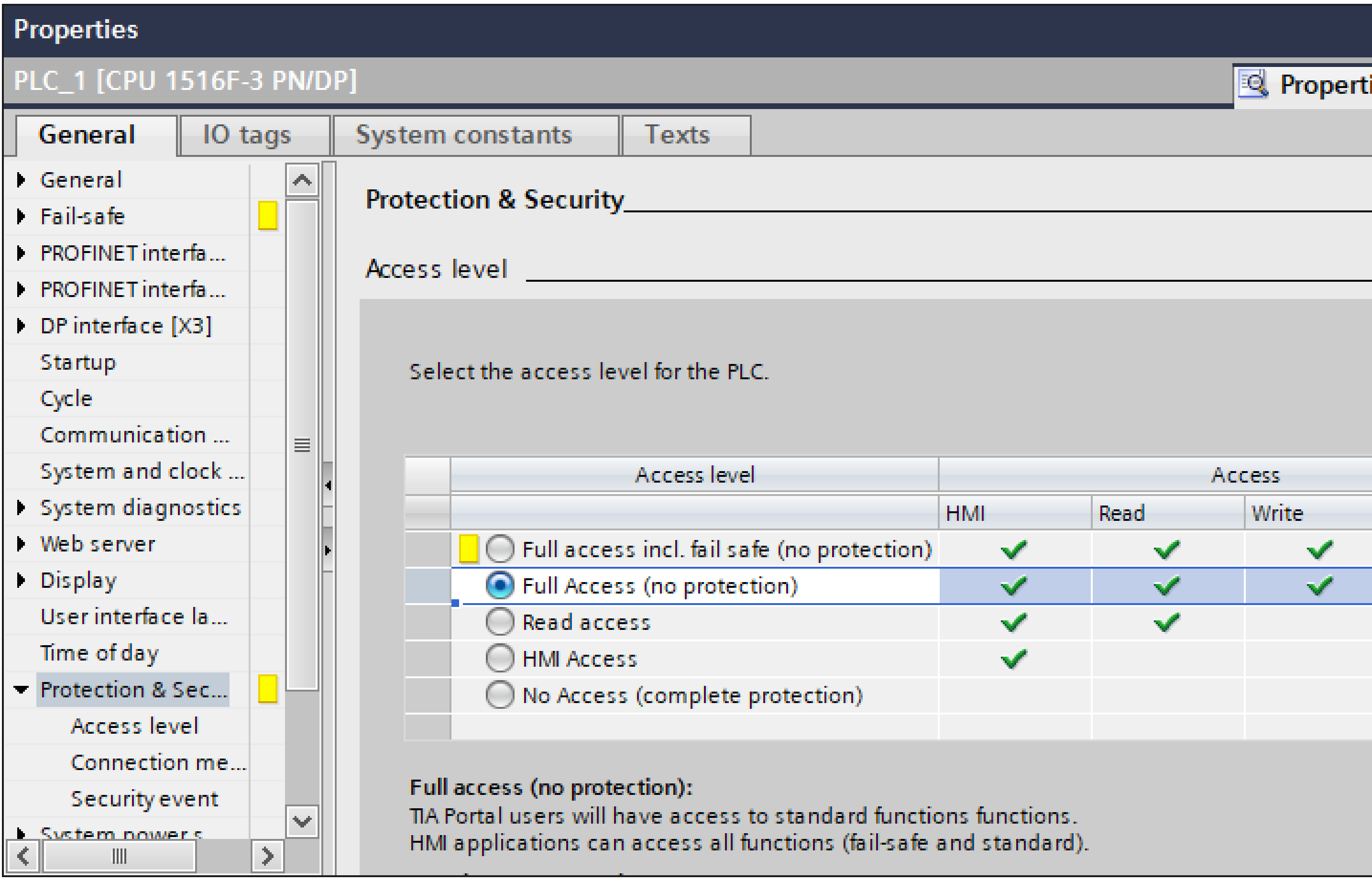
Note

"Channel Diagnosis" - scope

"Channel Diagnosis" only displays status information for channels that support channel diagnostics.

Protection and Security

You can specify the "Plc Protection-Level" settings when configuring the control in TIA Portal/ STEP 7.



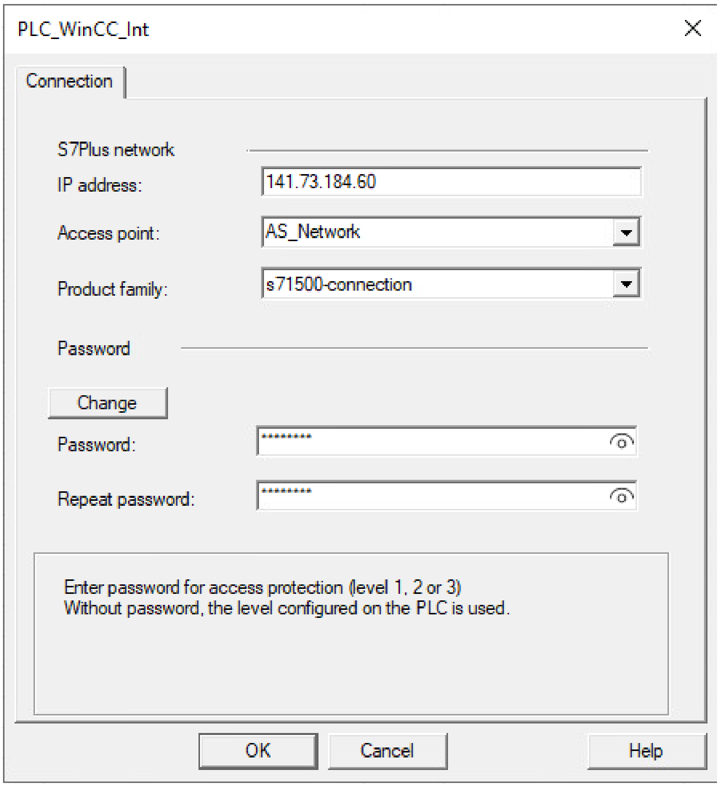
Select the access level for the PLC.

Access level	Access		
	HMI	Read	Write
○ Full access incl. fail safe (no protection)	✓	✓	✓
● Full Access (no protection)	✓	✓	✓
○ Read access	✓	✓	
○ HMI Access	✓		
○ No Access (complete protection)			

Full access (no protection):
TIA Portal users will have access to standard functions functions.
HMI applications can access all functions (fail-safe and standard).

More information on assigning a password for connection protection:

- "Communication > SIMATIC S7-1200, S7-1500 Channel > Configuring the channel > How to configure a connection (Page 507)"



PLC_WinCC_Int

Connection

S7Plus network

IP address: 141.73.184.60

Access point: AS_Network

Product family: s71500-connection

Password

Change

Password: ********

Repeat password: ********

Enter password for access protection (level 1, 2 or 3)
Without password, the level configured on the PLC is used.

If you do not assign a password, "Invalid password OR Complete protection" is displayed in the diagnostics.

The protection level of the PLC is used.

If you have assigned the same password for both communication partners, the status "No protection, full access" is displayed. This means that the connection to the PLC is protected by the password, although the HMI device has full access to the PLC.

This state can only be displayed for the indicated channel and its partners. This says nothing about the general protection of the HMI device.

See also

How to Configure the Trace Function of a Channel (Page 622)

How to configure a connection (Page 507)

8.3.4.4 Diagnosing a Channel with "Channel Diagnosis"

Introduction

Independently of WinCC, Channel Diagnosis can also be started as an application via the Windows program group "Siemens Automation".

"Channel Diagnosis" is thus always available and not dependent on the selection of a process picture, as is the case with "WinCC Channel Diagnosis Control".

The status information is only displayed by "Channel Diagnosis" when WinCC is in Runtime.

See also

How to Check a Channel with Channel Diagnosis (Page 621)

Channel Diagnosis with ActiveX Control (Page 617)

8.3.4.5 How to Check a Channel with Channel Diagnosis

Introduction

This section describes how to start "Channel Diagnosis" as an application from the Windows Start menu.

Note

"Channel Diagnosis" only displays status information for channels that support channel diagnosis.

Requirements

- Configure a channel and create a tag in this channel.

Procedure

1. In the "Siemens Automation" Windows program group, select the entry "Channel Diagnosis".
   The "Channel Diagnosis" application window opens.
   If no WinCC project is currently located in Runtime, the message "No connection to WinCC" is displayed.
2. Activate Runtime via the WinCC Explorer toolbar.
3. Select the picture into which you have inserted the ActiveX control.
   The status information for the channels appears in the "Channel Diagnosis" application window.
4. Click the gear icon in the "Channel Diagnosis" application window.
   The "Configuration" dialog appears.
5. Select a channel from the "Channel name" drop-down list.
6. Specify a name for the log file.

More information about the activation of the Trace function can be found under "Configuring a Channel's Trace Function".

See also

How to Configure the Trace Function of a Channel (Page 622)

How to Start Runtime (Page 623)

Channel Diagnosis with ActiveX Control (Page 617)

Diagnosing a Channel with "Channel Diagnosis" (Page 620)

8.3.4.6 How to Configure the Trace Function of a Channel

Introduction

This section describes how to configure and activate the Trace function of a channel in Runtime.

More information regarding communication status and errors is recorded in the trace file.

Trace file

The information recorded in a trace file is intended to assist the Hotline in pinpointing the cause of communication problems.

Further evaluation of the file is therefore not described in this section.

OPC Trace

You can find more information in the following FAQs:

- Entry ID 99412077:
"How do you configure and enable the trace of the WinCC OPC channel?" (https://support.industry.siemens.com/cs/ww/en/view/99412077)
- Entry ID 99412263:
"How do you configure and enable the trace of the WinCC OPC DA/HDA/A&E server?" (https://support.industry.siemens.com/cs/ww/en/view/99412263)

Requirements

- You have configured a channel and created a connection and a tag below it.
- Activate Runtime.

Standard Flags - Overview

Flag	Meaning
Fatal error	Fatal error (e.g. user intervention required)
Error	Error (e.g. telegram error)
Warning	Warning (e.g. reference to checksum error)
Information	Information (e.g. function call)
Successful	Successful execution (e.g. completing a function call)
Check User Flags	Enables the "User Flags" check boxes

Procedure

1. Start the WinCC Channel Diagnosis from the Start menu.
2. Click the gear icon in the "Channel Diagnosis" application window.
   The "Configuration" dialog appears.
3. Select the desired channel.
4. Under "Set flags", activate the status and error messages to be recorded in the trace file.
   A description of the standard flags can be found in the "Standard Flags - Overview" table.
5. Select the "Check User Flags" check box if the "User Flags" are to be recorded in the trace file.
   The number and significance of the "User Flags" depends on the channel.
6. Select the check boxes of the desired "User Flags".
   By clicking the "Set" or "Reset" buttons, you can set or reset all "User Flags".
7. In the "Trace File" section, select the "Enable" check box.
   This activates the other fields in this area.
8. In the "max. files" field, enter the maximum number of trace files.
9. In the "max. size" field, define the maximum size of the individual trace file.
10. Activate the "Overwrite" field if the channel's existing trace files are to be overwritten – beginning with the oldest – after the maximum number of files and file size has been reached.
11. Click "Save" to save the settings and activate the changes.

See also

How to Check the Configuration Data (Page 662)

How to Check the Channel and the Connection (Page 663)

Internet: Trace for the WinCC V7 OPC DA channel (https://support.industry.siemens.com/cs/ww/en/view/99412077)

Internet: Trace for OPC/OPC UA server (WinCC V7 / WinCC Professional) (https://support.industry.siemens.com/cs/ww/en/view/99412263)

8.3.4.7 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.

See also

How to Check a Channel with Channel Diagnosis (Page 621)

8.4 Diagnosis of "System Info" Channel

8.4.1 "System Info" Channel - Diagnostic Options

The following options for the diagnosis of the "System Info" channel or one of its tags are available:

Diagnosis of the Channel with "Channel Diagnosis"

"Channel Diagnosis" can query the status of the channel and connection in Runtime. Any errors that occur are displayed using "Error Codes".

Diagnosis of the Channel Tags

In tag management in Runtime, you can query the current value, the current quality code and the last time that the tag was changed.

See also

How to Check a Tag (Page 628)

How to Check the Channel and the Connection (Page 626)

8.4.2 Description of Log File Entries

Introduction

The channel records errors and important status changes in the log file. These entries can be used to analyze a communication problem.

Each entry in the file has a date and time stamp followed by a flag name and description.

Example of a logbook entry:

```
2000-03-10 12:00:21,050 INFO Log starting ...
2000-03-10 12:00:21,050 INFO | LogFileName : C:\Siemens\WinCC\Diagnose\SYSTEM_INFO_01.LOG
2000-03-10 12:00:21,050 INFO | LogFileCount : 3
2000-03-10 12:00:21,050 INFO | LogFileSize : 1400000
2000-03-10 12:00:21,050 INFO | TraceFlags : fa000001
2000-03-10 12:00:21,050 INFO start timer
2000-03-10 12:00:21,360 ERROR Illegal tag type! tag: "Format_0" correct type: "Text Tag 8-Bit Character Set"!
```

Entries for "INFO" Flag

Message text	Description
Log starting ...	Start message
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 : fa000001	Flags used by the channel in hexadecimal format
start timer	Start message

Entries for "ERROR" Flag

Message text	Description
Illegal tag type! tag: "tag" correct type: "data type"!	Incorrect data type of a tag tag= Name of tag with incorrect data type data type = Correct data type

8.4.3 Determining the Cause of Incorrect Tag Values

8.4.3.1 How to Determine the Cause of Incorrect Tags

If an unexpected tag value occurs in Runtime, proceed as follows to determine the cause:

1. Check the channel and connections
2. Check the tags of the channel

See also

How to Check a Tag (Page 628)

How to Check the Channel and the Connection (Page 626)

8.4.3.2 How to Check the Channel and the Connection

Introduction

This section describes how to check the "System Info" channel and its connection in Runtime.

Requirements

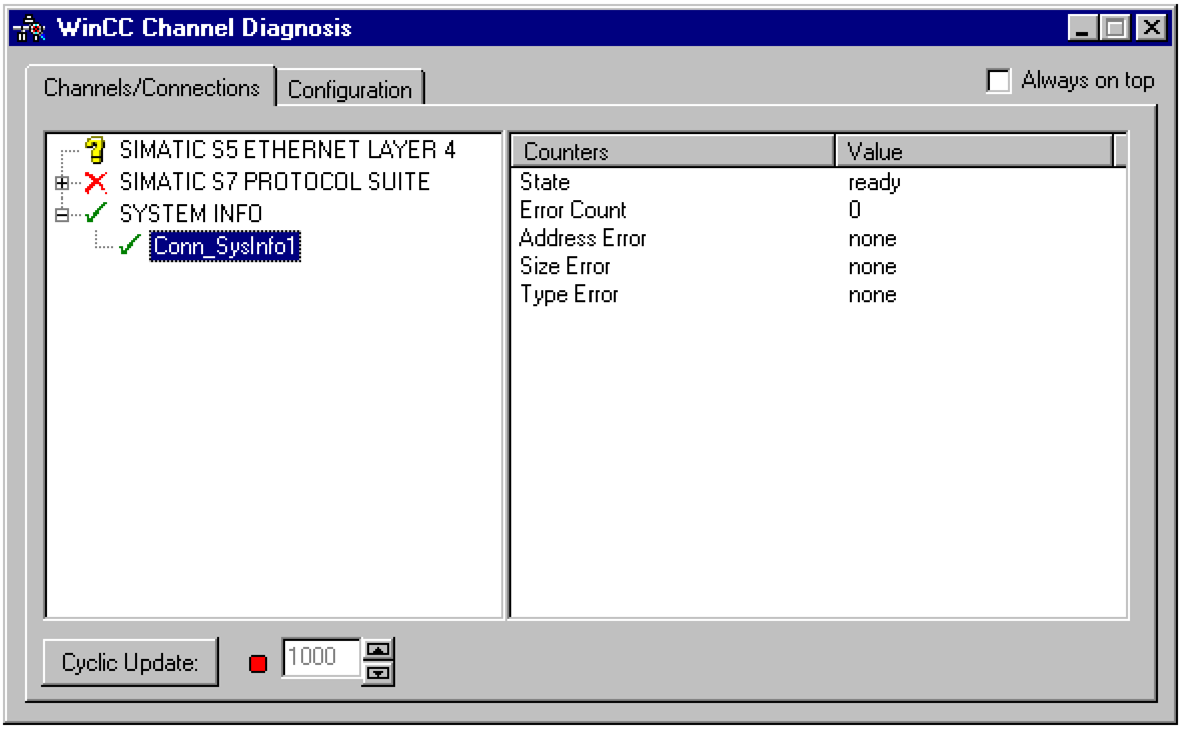
- Configure a connection and tag for the "System Info" channel.
- Activate the WinCC project.

Overview of Status Messages

Icon	Description
✓	Channel / connection unconditionally ready
⚠	Channel / connection ready with some restrictions
?	Channel / no statement possible regarding connection status
✗	Channel / connection failed

Procedure

1. Start the WinCC Channel Diagnosis from the Start menu.
2. The Channel Diagnosis application window opens. The status information for all installed channels and their connections is displayed on the left on the "Channels/Connections" tab.



- ? SIMATIC S5 ETHERNET LAYER 4
- ✗ SIMATIC S7 PROTOCOL SUITE
- ✓ SYSTEM INFO
  - ✓ Conn_SysInfo1

Counters	Value
State	ready
Error Count	0
Address Error	none
Size Error	none
Type Error	none

□ Always on top

Cyclic Update: 1000

3. Check the icons in front of the channel named "System Info" and its connection. If the status of the channel and connection are OK, a green check mark is displayed in front of each respective entry. For information on the significance of the individual icons refer to the "Overview of Status Messages" table.
4. If there is no green check mark in front of the channel's name and the connection, select the connection in the window on the left. In the window on the right, check the counter values for "Address Error", Size Error" and "Type Error". These values indicate the errors detected.

5. Check the channel-specific log file. To do this, use a text editor to open the file in the directory "Siemens\WinCC\Diagnose". Check the latest entries with the "ERROR" flag. For more information on this topic, refer to "Description of Log File Entries".
6. If you are still unable to pinpoint the error after checking the log file, please activate the Trace function and contact Customer Support.
   For more information on this topic, refer to "Configuring a Trace Function of a Channel".

See also

How to Configure the Trace Function of a Channel (Page 622)

Description of Log File Entries (Page 625)

How to Check a Tag (Page 628)

8.4.3.3 How to Check a Tag

Introduction

If an external tag does not have the expected value in Runtime, you can use the following procedure to check the tag.

Requirements

- Configure a connection and tag for the "System Info" channel.
- Activate the WinCC project.

Procedure

1. In WinCC Explorer in the tag management, select the "System Info" channel.
2. In the data window, select the external tag that you wish to check. To do this, open the directory structure until the tag is displayed in the table area.
3. Move the mouse pointer over the tag to be checked. A tooltip window opens showing the current tag value, the quality code and the last time that the value changed.
4. Check the quality code. If value "80" is displayed, the tag value is OK. A description of the other values can be found under "Tag quality codes".
5. If the quality code is not equal to "80", select the tag in the tag management and click "Properties" in the shortcut menu to open the "Tag Properties" dialog.
6. Check whether values have been configured for the high or low limits, the start or substitute values on the "Limits/Reporting" tab. These values can affect the display.
7. If the tag value is affected by one of the configured values, deactivate the project and change the limit or substitute value.

Note

Tag values, quality codes etc. are only displayed in Runtime.

See also

Quality Codes of Tags (Page 669)

8.5 Diagnostics channel "SIMATIC S7-1200/S7-1500"

8.5.1 System diagnostics with SysDiagControl

Overview

The system diagnostics displays faults and errors of the S7-1200 and S7-1500 controllers.

With the WinCC SysDiagControl, WinCC provides an overview for quick error localization in the "SIMATIC S7-1200, S7-1500 Channel" communication channel.

You can configure direct navigation from a message about the status of a controller to the diagnostics overview in the SysDiagControl. There, the details of the controller errors are displayed.

System diagnostics view

The following views are available in the WinCC SysDiagControl:

- Diagnostic overview
- Detail view
- Diagnostic buffer view

The system diagnostics display also offers a split view of the display. This allows you see the controllers and associated details at a glance.

The upper area shows the diagnostic overview and the diagnostic buffer view.

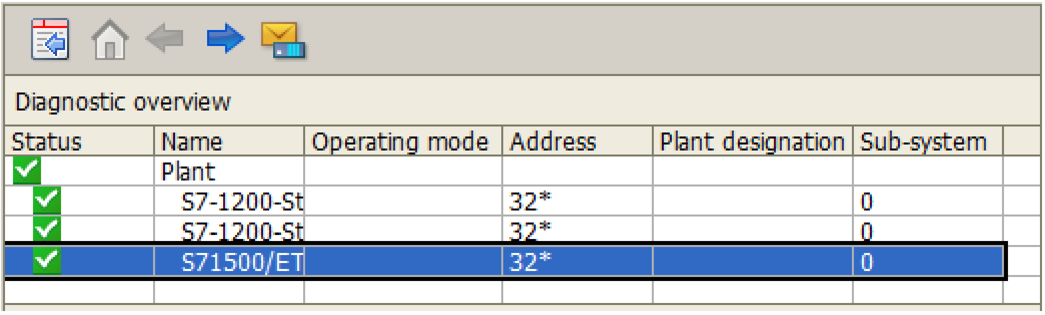
The lower area shows the detail view.

Diagnostic overview

The diagnostic overview displays all available S7-1200/1500 channels.

Double-clicking on a controller opens the detail view.

The symbols in the first column provide information about the current status of the controller.



Diagnostic overview

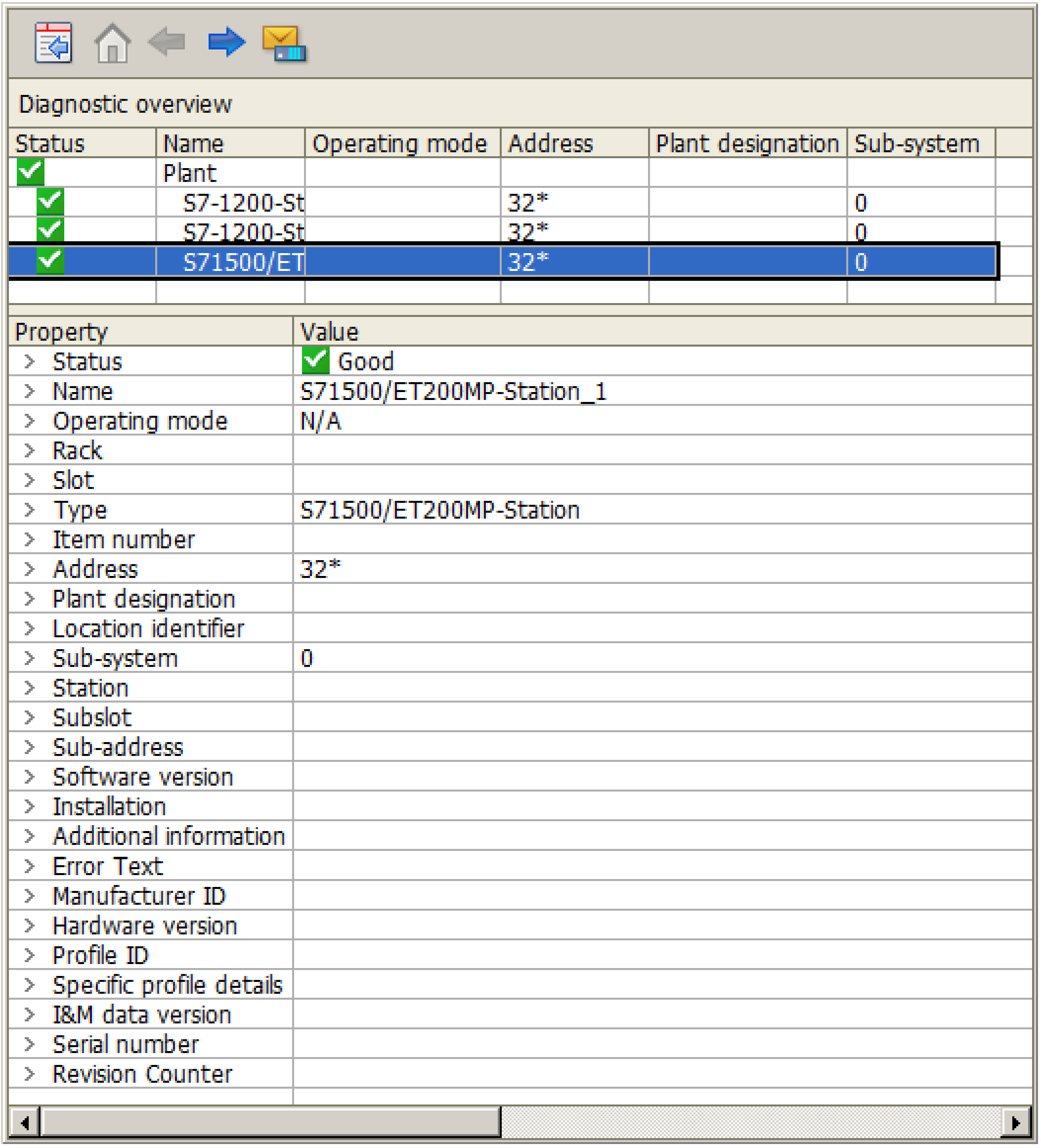
Status	Name	Operating mode	Address	Plant designation	Sub-system
✓	Plant				
✓	S7-1200-St		32*		0
✓	S7-1200-St		32*		0
✓	S71500/ET		32*		0

Detail view

The detail view gives detailed information about the selected controller.

Check whether the data is correct in the detail view. You can cannot sort error texts in the detail view.

The following figure shows the split view of the diagnostic overview and the detail view.



Diagnostic overview

Status	Name	Operating mode	Address	Plant designation	Sub-system
✓	Plant				
✓	S7-1200-St		32*		0
✓	S7-1200-St		32*		0
✓	S71500/ET		32*		0

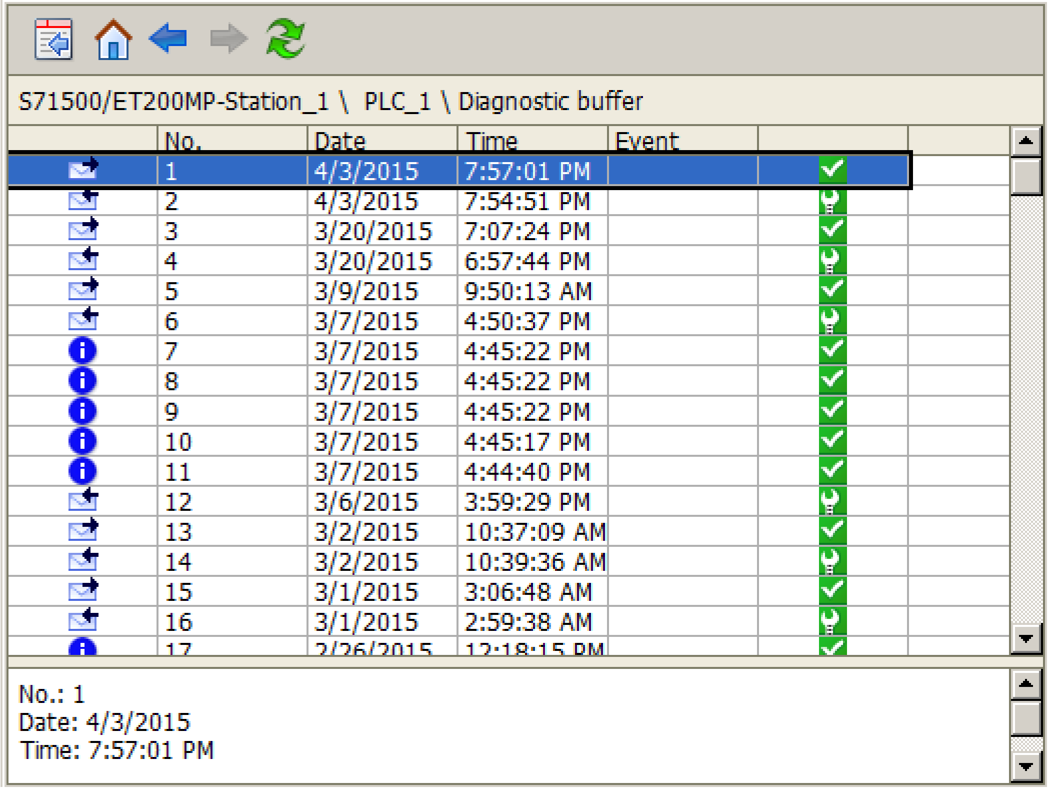
Property	Value
› Status	✓ Good
› Name	S71500/ET200MP-Station_1
› Operating mode	N/A
› Rack	
› Slot	
› Type	S71500/ET200MP-Station
› Item number	
› Address	32*
› Plant designation	
› Location identifier	
› Sub-system	0
› Station	
› Subslot	
› Sub-address	
› Software version	
› Installation	
› Additional information	
› Error Text	
› Manufacturer ID	
› Hardware version	
› Profile ID	
› Specific profile details	
› I&M data version	
› Serial number	
› Revision Counter	

Diagnostic buffer view

The diagnostic buffer view shows the current data from the diagnostic buffer of the controller.

The diagnostic buffer view can only be called in the diagnostic overview.

To update the diagnostic buffer view, select the "Update" button.



S71500/ET200MP-Station_1 \ PLC_1 \ Diagnostic buffer

	No.	Date	Time	Event	
✉	1	4/3/2015	7:57:01 PM		✓
✉	2	4/3/2015	7:54:51 PM		🔧
✉	3	3/20/2015	7:07:24 PM		✓
✉	4	3/20/2015	6:57:44 PM		🔧
✉	5	3/9/2015	9:50:13 AM		✓
✉	6	3/7/2015	4:50:37 PM		🔧
ℹ	7	3/7/2015	4:45:22 PM		✓
ℹ	8	3/7/2015	4:45:22 PM		✓
ℹ	9	3/7/2015	4:45:22 PM		✓
ℹ	10	3/7/2015	4:45:17 PM		✓
ℹ	11	3/7/2015	4:44:40 PM		✓
✉	12	3/6/2015	3:59:29 PM		🔧
✉	13	3/2/2015	10:37:09 AM		✓
✉	14	3/2/2015	10:39:36 AM		🔧
✉	15	3/1/2015	3:06:48 AM		✓
✉	16	3/1/2015	2:59:38 AM		🔧
ℹ	17	2/26/2015	12:18:15 PM		✓

No.: 1
Date: 4/3/2015
Time: 7:57:01 PM

Buttons in the system diagnostics view

Button	Function
📋	Opens the configuration dialog in which you can change the properties of the SysDiag-Control.
➡	Opens the child devices or the detail view if there are no child devices.
⬅	Opens the parent device or the diagnostic overview if there is no parent device.
🏠	Opens the diagnostic overview.
✉	Opens the diagnostic buffer view. Only visible in the diagnostic overview.
🔄	Updates the diagnostic buffer view.
A↓Z	Opens a dialog for setting user-defined sort criteria for the displayed diagnostic overview columns.

Button	Function
🖨	Starts the printout of the displayed values. The print job used for printing is defined in the configuration dialog on the "General" tab.
↩	This button is used for exporting all or the selected runtime data into a "CSV" file. If the option "Display dialog" is active, a dialog opens in which you can view the settings for exporting and can start the export. With the respective authorization, you are also allowed to select the file and the directory for the export. If a dialog is displayed, the export of the data to the predefined file starts immediately.

See also

"SIMATIC S7-1200, S7-1500 Channel" channel (Page 502)

How to configure the system diagnostics (Page 633)

8.5.2 How to configure the system diagnostics

Introduction

The faults and errors in the controllers are displayed in Runtime in various views of the system diagnostics.

In the Graphics Designer you configure a WinCC SysDiagControl for this.

Requirement

- A connection is created in the "OMS+" channel unit below the "SIMATIC S7-1200, S7-1500 Channel".
- Alarm Logging is activated in the startup list of the server.
- To display messages and texts of the S7-1500 channel in the diagnostic buffer view, additional requirements must be met:
  - The AS messages and AS text lists of the controller are loaded in the WinCC project.
  - The "Used" option must be selected for the AS text lists in Alarm Logging.
  - A defined acknowledgment philosophy must be configured in Alarm Logging for the diagnostic messages of message type "Notify_AP":
    The diagnostic messages must be assigned to a message type that does not require acknowledgment but can have "Went Out" status.
- Requirement for automatic update of S7-1500 messages:
  - The AS messages and AS text lists of the controller are loaded once in the WinCC project. As a result, the required texts are stored in the Text Library.

You can find more information under "Working with WinCC > Setting up a message system > Configuring the message system > AS messages".

Configuration steps

1. Insert the WinCC SysDiagControl in a process picture in the Graphics Designer.
2. Configure the basic properties of the SysDiagControl in the "General" tab.
   - The properties of the diagnostic window
   - The general properties of the control
   - The time base of the control
3. In the "Columns" tab, specify the controller data to be displayed as columns or rows in the views of the system diagnostics.
4. Use the sorting dialog to determine the columns in which the data is to be sorted.
   You can find more detailed information using the example of WinCC UserArchiveControl under:
   - "Options > User Archive > WinCC UserArchiveControl > Operation during runtime > How to sort the display of user archive data"
5. Configure the display and properties of the tables in the "Parameter", "Display" and "Marker" tabs.
   You can find more detailed information at:
   - "Working with WinCC > Setting up a Message System > Display of Messages during Runtime > Configuring the AlarmControl > How to configure the display for the table"
6. Configure the toolbar and the status bar of the table window in the respective tabs
   You can find more detailed information at:
   - "Working with WinCC > Setting up a Message System > Display of Messages during Runtime > Configuring the AlarmControl > How to configure the toolbar and the status bar"
7. Configure a button in the picture with a script for jumping from an AS message in WinCC AlarmControl directly to the WinCC SysDiagControl:
   - Insert a button in the picture.
     As an event, create a script that, for example, perform the action at a mouse click.
   - You can use the following script example when the WinCC AlarmControl "AlarmControl_1" and WinCC SysDiagControl "SysDiagControl_1" are in the same screen:
     In C:
     ```
     SetPropChar(lpszPictureName,"SysDiagControl_1","NavigateTo",
     GetPropChar(lpszPictureName,"AlarmControl_1","DiagnosticsContext"));
     ```
     In VBS:
     ```
     ScreenItems("SysDiagControl_1").NavigateTo =
     ScreenItems("AlarmControl_1").DiagnosticsContext
     ```
8. Save your configuration data.

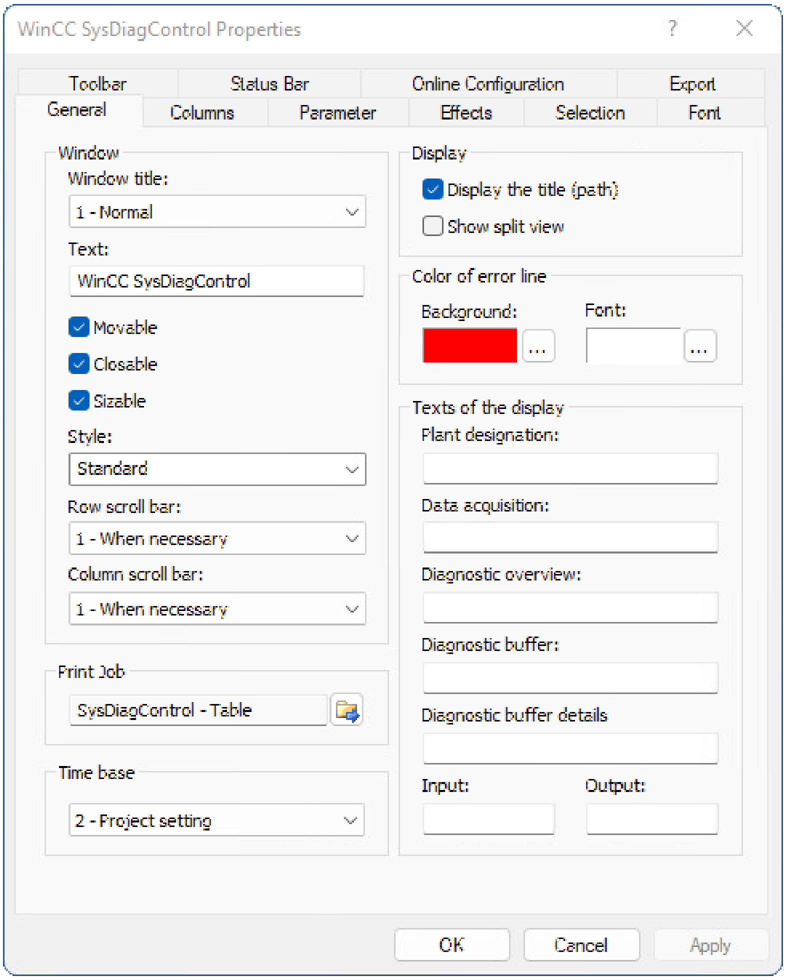
See also

System diagnostics with SysDiagControl (Page 630)

8.5.3 Information on configuring the SysDiagControl

Where does the control get its information?

The information displayed by the SysDiagControl is based on the configuration of the control. You create the configuration in the dialog "WinCC SysDiagControl Properties".



Window
Window title: 1 - Normal
Text: WinCC SysDiagControl
☑ Movable
☑ Closable
☑ Sizable
Style: Standard
Row scroll bar: 1 - When necessary
Column scroll bar: 1 - When necessary

Print Job
SysDiagControl - Table

Time base
2 - Project setting

Display
☑ Display the title (path)
☐ Show split view

Color of error line
Background: 🟥 ...
Font: ...

Texts of the display
Plant designation:
Data acquisition:
Diagnostic overview:
Diagnostic buffer:
Diagnostic buffer details
Input:
Output:

The dialog is displayed when you insert the control into a picture.

If the control has already been inserted, double-click the control to open the dialog.

More information: How to configure the system diagnostics (Page 633).

The controller must know which connections have to be taken into account for retrieving the data from the corresponding PLC.

You can find more information on this under "Requirement" in section "How to configure the system diagnostics".

The required information is retrieved from the configuration data when runtime is activated. The configuration data is used to establish the communication with the PLC. As soon as the connection to the PLC has been established, the SysDiagControl requests the necessary information from the PLC and displays it on the control.

If communication with the PLC fails during activation of runtime, individual data items corresponding to the data available in the configuration are displayed.

Information, such as the station name, is taken from the configuration data until a connection to the PLC has been established. As soon as the connection is established, the new data is available from the PLC and displayed automatically on the control.

In runtime, once the connection has been established, the data to be displayed on the control is retrieved from the PLC. Most of the information displayed on the control is retrieved directly from the PLC. An exception is the data that has to be extracted from the TextList.

Are existing test lists to be reloaded or is runtime to be restarted?

AS text lists must be updated either manually or via the "Automatic Update" setting. As soon as the AS text lists have been updated in the project, the picture with the SysDiagControl must be reloaded.

You can find details on the underlying settings in the following sections:

- AUTOHOTSPOT
- AUTOHOTSPOT
- AUTOHOTSPOT
- AUTOHOTSPOT
- AUTOHOTSPOT
- Configuring the Import Package (Page 46)
- How to Configure the Package Export (Page 35)

What steps are required to create a connection again when runtime is active?

When runtime is activated, the "CCSystemDiagnosticsHost" service reads the configuration file only once at the start.

When a new connection is created, the configuration data is changed, which is regarded as a "delta download". The system diagnostics does not currently support a "delta download". The configuration for the download of the AS messages must be followed.

Observe the requirements and follow the steps for updating the AS messages/text lists.
A restart of runtime is necessary in order to take the new connection into account in SysDiagControl.

What should be done if the connection is changed after loading runtime?

Scenario

- A new connection with loading from file has been created.
- All AS text lists have been enabled and loaded in runtime.
- The PLC is not yet available.
- This connection is to be changed from PLC 1516 to PLC 1516F.

What does "Changing the PLC connection from PLC 1516 to PLC 1516F" mean?

If the connection has been created and the PLC name is changed to PLC 1516F, a download to the PLC is required in order to this change into account. Following a successful download, such a change on the PLC is reported to SysDiagControl if the connection has already been established.

If the connection has not yet been established, the information from the configuration data, which still has the old name, is taken into account, since the configuration data is obtained at the start and additional changes are excluded. If the connection is now established, SysDiagControl gets the data directly from the PLC, and thus gets the up-to-date data.

A restart of runtime will reflect the change even if the connection is not reestablished, because the change already exists in the configuration data.

8.6 Diagnosis of the "SIMATIC S7 Protocol Suite" Channel

8.6.1 "SIMATIC S7 Protocol Suite" Channel - Diagnostic Options

The following options for the detection of errors and the diagnosis of the "SIMATIC S7 Protocol Suite" channel or one of its tags are available:

Checking the Communication Processor Configuration

Besides 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. Invalid tag values may also result from improperly addressing the tag in the AS.

Diagnosis of the Channel with "Channel Diagnosis"

"Channel Diagnosis" can query the status of the channel and connection in Runtime. Any errors that occur are displayed using "Error Codes".

Diagnosis of the Channel Tags

In tag management in Runtime, you can query the current value, the current quality code and the last time that the tag was changed.

See also

How to Check a Tag (Page 647)

How to Check the Channel and the Connection (Page 645)

How to Check the Configuration of the Connection and Tags (Page 644)

Checking the Communication Processor under SIMATIC NET (Page 643)

How to Check the Configuration of the Communication Processor (Page 641)

8.6.2 Description of Log File Entries

Introduction

The channel records errors and important status changes in the log file. These entries can be used to analyze a communication problem.

Each entry in the file has a date and time stamp followed by a flag name and description.

Example of a logbook entry:

```
1999-04-01 12:00:24,524 INFO Log starting ...

1999-04-01 12:00:24,524 INFO LogFileName :
C:\Siemens\WinCC\Diagnose\SIMATIC_S7_Protocol_Suite_01.LOG

1999-04-01 12:00:24,524 INFO LogFileCount : 3

1999-04-01 12:00:24,524 INFO LogFileSize : 1400000

1999-04-01 12:00:24,524 INFO TraceFlags : c4000000

1999-04-01 12:00:24,524 INFO S7 channel DLL started!

1999-04-01 12:00:26,096 ERROR Illegal tag address "nCPU3_1"!

1999-04-01 12:00:27,428 INFO S7DOS release: @(#)TIS-Block Library DLL Version C5.0.17.3-
REL5,0,17,47,3-BASIS

1999-04-01 12:00:27,428 INFO S7DOS version: V5.0 / 0

1999-04-01 12:00:27,428 INFO S7CHN version: V5.0 / Mar 1 1999 / 22:36:40

1999-04-01 12:00:27,428 INFO S7 channel unit "Industrial Ethernet" activated!

1999-04-01 12:00:27,468 ERROR Cannot connect to "CPU_4": Errorcode 0xFFDF 42C2!

1999-04-01 12:00:27,538 INFO S7 channel unit "MPI" activated!
```

Description of the Most Important Entries for the "INFO" Flag

Message text	Description
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
S7 channel DLL started!	Start message
S7 channel DLL terminated!	End message
S7 channel unit "unitname" activated!	Channel unit activated
S7 channel unit "unitname" deactivated!	Channel unit deactivated
S7DOS version: versionsstring	Version information
S7CHN version: versionsstring	Version information

Description of the Most Important Entries for the "ERROR" Flag

Message text	Description
Cannot connect to "connectionname": Errorcode 0xhhhh ffff!	Communication error Could not establish a connection to the AS immediately after activating WinCC. If the connection could be established without error at least once, the following message is output in the event of a later error. nnn = Number of disconnects for this connection connectionname = Name of the connection hhh = 1st error code in hex S7DOS / SAPI-S7 ffff = 2nd error code in hex S7DOS / SAPI-S7
Cannot connect to "connectionname": Errorcode 0xhhhh ffff!	Communication error Could not establish a connection to the AS immediately after activating WinCC. The connection was established at least once without error.
Channel API error: errorstring	Channel API error The channel passed the error string 'errorstring' to WinCC Explorer. Depending on the significance of the error, the error string may or may not be displayed in a notice box. For a description of the error strings, please see the API Error Text.
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
Max. logbooksize reached - Logbook deactivated	Message sent when log file has exceeded its maximum length. The logbook output is monitored for length. If the specified length is reached, the logbook is deactivated. The message is only output, when message output causes the max. file length to be exceeded. No message be output, if the file length is changed with an editor or the maximum file length is reduced in the INI file!

8.6.3 Determining the Cause of Incorrect Tag Values

8.6.3.1 How to Determine the Cause of Incorrect Tags

If an unexpected tag value occurs in Runtime, proceed as follows to determine the cause:

1. Checking the Configuration of the Communication Processor
2. Checking the Communication Processor under SIMATIC NET
3. Checking the Configuration of the Connection and Tags
4. Check the channel and connections
5. Check the tags of the channel

See also

How to Check a Tag (Page 647)

How to Check the Channel and the Connection (Page 645)

How to Check the Configuration of the Connection and Tags (Page 644)

Checking the Communication Processor under SIMATIC NET (Page 643)

How to Check the Configuration of the Communication Processor (Page 641)

8.6.3.2 How to Check the Configuration of the Communication Processor

Introduction

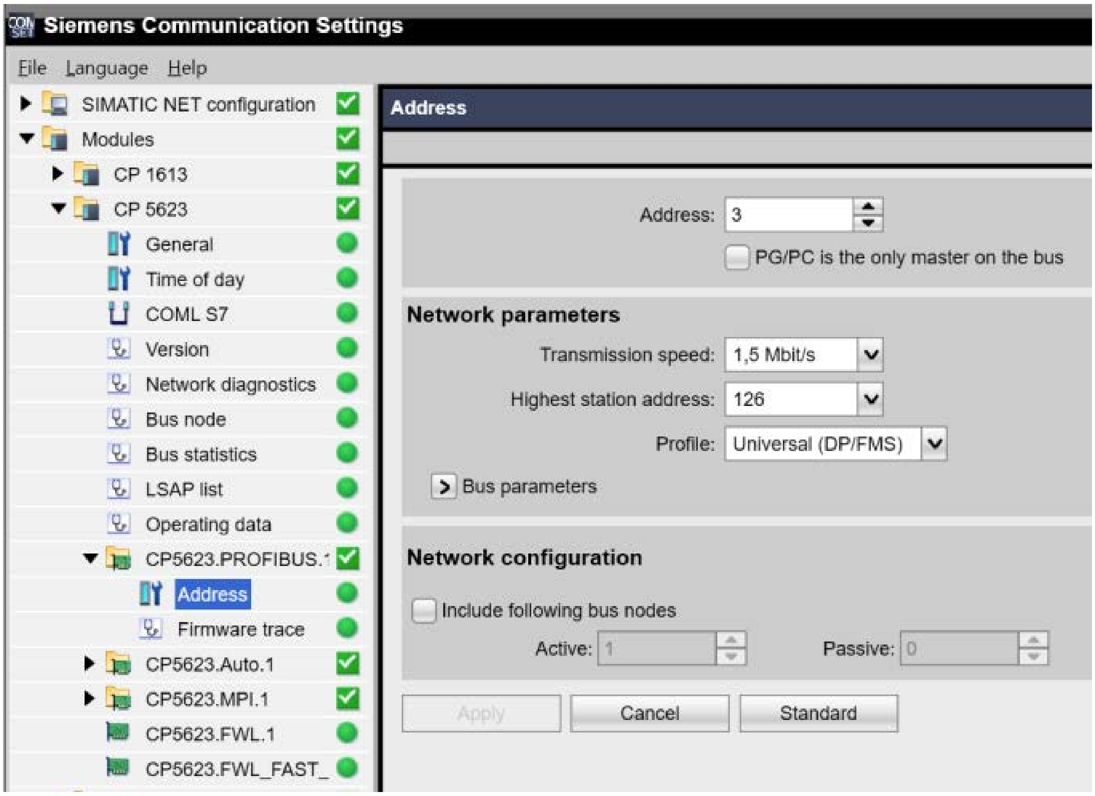
This section describes how to use the "PG/PC Port" program to check a communication processor. In this example, the "CP 5613 A3" type processor is used for the PROFIBUS communication.

Requirements

- Install the CP 5613 A3.
- Install the associated communication driver.
- Configure the CP 5613 A3.

Procedure

1. On the Control Panel, click the "Set PG/PC Port" icon. The "Set PG/PC Port" dialog is opened.
2. Check the entry for the access point. The access point "CP_L2_1:" for Profibus connection is automatically added when a CP 5613 A3 is installed. Select the entry for this access point. Click "Properties" to open the "Properties - CP5613A3.PROFIBUS.1" dialog.



Address: 3

☐ PG/PC is the only master on the bus

Transmission speed: 1,5 Mbit/s

Highest station address: 126

Profile: Universal (DP/FMS)

☐ Include following bus nodes

Active: 1

Passive: 0

Figure 8-1 Example configuration: CP 5623

3. Check the entry on the "PROFIBUS" tab.
4. Click the "Operational State" tab. Click "Test" to execute a function test on the CP 5613 A3. The test results is shown in the output field below. Depending on the test results, you can click the "Restart" button to perform a reset and a complete restart of the CP 5613 A3.
5. Click "OK" to close all open dialogs.

See also

How to Check the Configuration of the Connection and Tags (Page 644)

Checking the Communication Processor under SIMATIC NET (Page 643)

8.6.3.3 Checking the Communication Processor under SIMATIC NET

Introduction

This section explains how to check a communication processor using the "Set PC Station" program in the SIMATIC NET software.

In this example, the "CP 5613 A3" type is used for the PROFIBUS communication to the "SIMATIC S7 Protocol Suite" channel.

Requirements

- Install the CP 5613 A3.
- Install the SIMATIC NET software.
- Configure the CP 5613 A3 under SIMATIC NET.

Procedure

1. Open the menu item "Set PC station" in the SIMATIC NET settings.
   The "Configurations Console PC Station" dialog opens.
2. Check the entry for the access point. Select the "Access Point" directory in the navigation window. The existing access points are listed in the data window. During installation of the CP 5613 A3, access point "CP_L2_1:" is inserted automatically for the Profibus connection. Select this access point in the data window. Use the "Properties" menu item from the shortcut menu to open the "Properties of CP_L2_1:" dialog.
3. Check the entry in the "Assigned Interface Parameters" field. For a CP 5613 A3 in a PROFIBUS network, the entry "CP5613A3.PROFIBUS.1" should be selected.
4. Open the navigation window, select the "Components" directory and then the "CP5613 A3" subdirectory.
5. Select the "Network Diagnosis" directory. Click "Test" to execute a function test on the CP 5613 A3. The result is displayed in the output window. Depending on the test results, click "Restart" in the "General" directory to perform a reset and then a complete restart of the CP 5613 A3.
6. Check the list of participants connected to PROFIBUS in the list in the "Bus Participants" directory. Based on the display, it is possible to determine the function and status of your own station as well as other participants which are connected.
7. Close the dialog.
8. If a fault is detected in the configuration of the communication processor, modifications can only be made to the configuration using SIMATIC NET tools. Further information is available under SIMATIC NET.

See also

How to Check the Configuration of the Connection and Tags (Page 644)

8.6.3.4 How to Check the Configuration of the Connection and Tags

Introduction

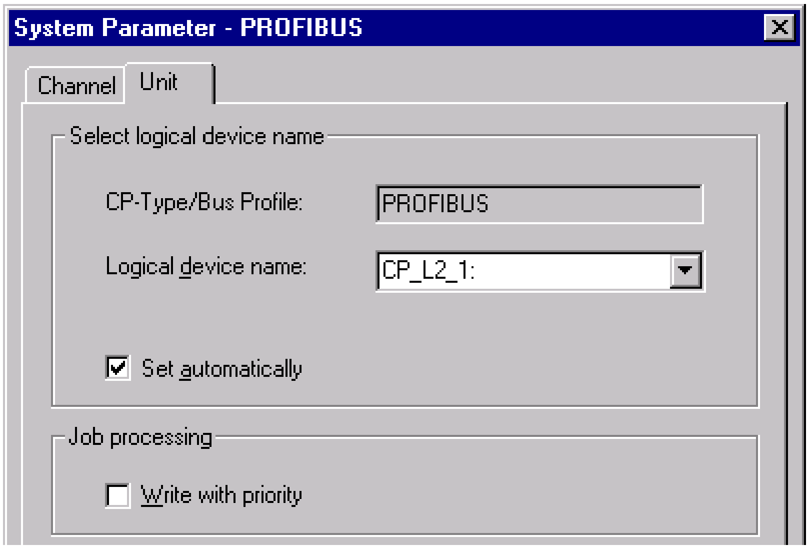
This section describes how to check the system parameters and the configuration of the connection and tags. In this example, a "CP 5613 A3" communication processor is used for the PROFIBUS communication.

Requirements

- Install the CP 5613 A3.
- Install the associated communication driver.
- Configure the CP 5613 A3.
- Configure a connection and tag for the "SIMATIC S7 Protocol Suite" channel.
- Activate the WinCC project.

Procedure

1. In WinCC Explorer in the tag management, select the "SIMATIC S7 Protocol Suite" channel. In the data window, select the "PROFIBUS" channel unit. In the channel unit shortcut menu, click "System Parameters". The "System Parameters - PROFIBUS" dialog opens.
2. On the "Unit" tab, check the entry in the "Logical Device Name" field. By default, this is set to the access point "CP_L2_1:" The access point is assigned during installation of the communication processor in the CP 5613 A3 system. Close the dialog.



System Parameter - PROFIBUS

Select logical device name

CP-Type/Bus Profile: PROFIBUS

Logical device name: CP_L2_1:

☑ Set automatically

Job processing

☐ Write with priority

3. In the tag management navigation window, select the "PROFIBUS" channel unit. In the data window, select the connection to be checked. In the shortcut menu, click "Properties" to open the "Connection Properties" dialog.
4. Click the "Properties" button to open the "Connection Parameters - PROFIBUS" dialog.
5. Check the settings on the "Connection" tab. Close the open dialogs.

6. In the navigation window, select the checked connection. In the data window, select the tag to be checked. In the shortcut menu, click "Properties" to open the "Tag Properties" dialog. Check the values in the "Type Conversion" and "Data Type" fields.
7. Click the "Select" button to open the "Address properties" dialog. Check the settings for addressing the tag in the AS.
8. Click "OK" to close all open dialogs.

See also

How to Check the Channel and the Connection (Page 645)

8.6.3.5 How to Check the Channel and the Connection

Introduction

This section describes how to check the "SIMATIC S7 Protocol Suite" channel and its connection in Runtime.

Requirements

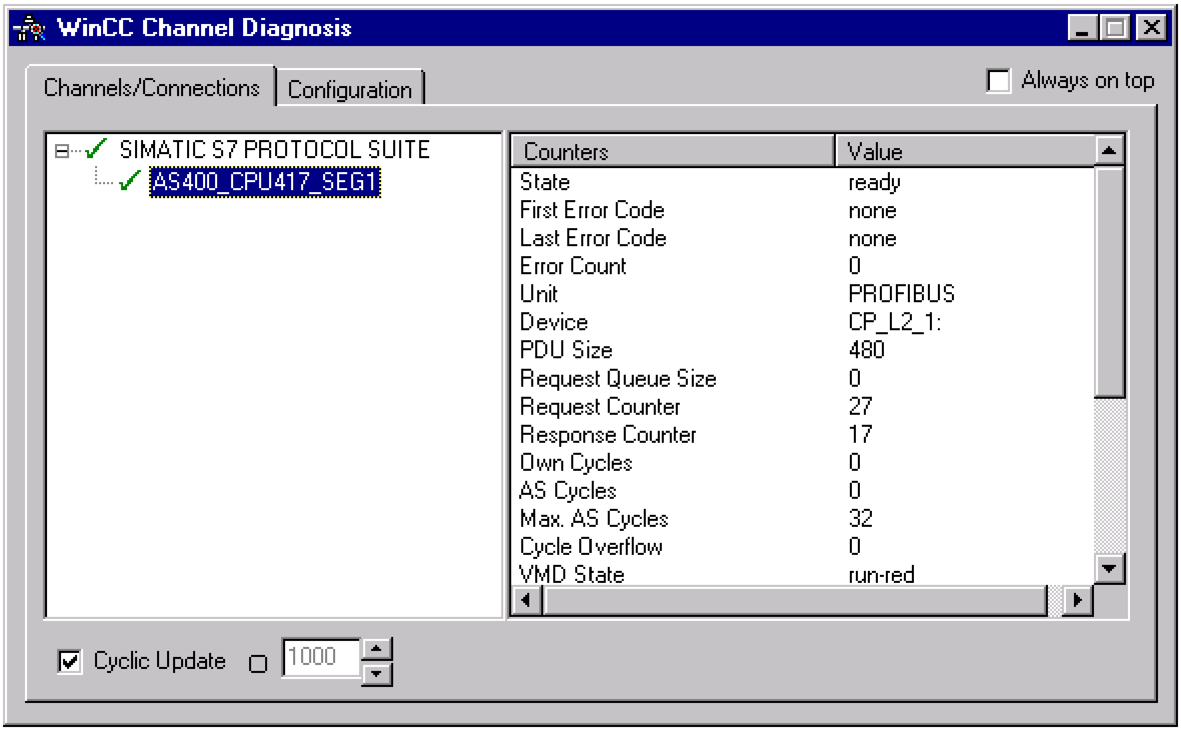
- Install a communication processor in a WinCC computer, for example the CP 5613 A3 for PROFIBUS communication.
- Install the associated communication driver.
- Configure the CP 5613 A3.
- Create a STEP7 project.
- Configure a connection and tag for the "SIMATIC S7 Protocol Suite" channel.
- Activate the WinCC project.

Overview of Status Messages

Icon	Description
✓	Channel / connection unconditionally ready
⚠	Channel / connection ready with some restrictions
?	Channel / no statement possible regarding connection status
✗	Channel / connection failed

Procedure

1. Start the WinCC Channel Diagnosis from the Start menu.
2. The Channel Diagnosis application window opens. The status information for all installed channels and their connections is displayed on the left on the "Channels/Connections" tab.



Counters	Value
State	ready
First Error Code	none
Last Error Code	none
Error Count	0
Unit	PROFIBUS
Device	CP_L2_1:
PDU Size	480
Request Queue Size	0
Request Counter	27
Response Counter	17
Own Cycles	0
AS Cycles	0
Max. AS Cycles	32
Cycle Overflow	0
VMD State	run-red

3. Check the icon in front of the channel named "SIMATIC S7 Protocol Suite" and the connection. If the channel and connection are OK, a green check mark is displayed in front of each respective entry. For information on the significance of the individual icons refer to the "Overview of Status Messages" table.
4. If there is no green check mark in front of the channel's name and the connection, select the connection in the window on the left. In the window on the right, check the entries for the counters "First Error Code" and "Last Error Code". These values indicate the errors detected. You can access Direct Help by clicking the shortcut menu of the value display.
5. On the "Configuration" tab, select the status and error messages, which are to be output to the log file. This is done by selecting "SIMATIC S7 Protocol Suite" and configuring the error display. For more information, please refer to "Configuring the Log File of a Channel".
6. Check the channel-specific log file. To do this, use a text editor to open the file in the directory "Siemens\WinCC\Diagnose". Check the latest entries with the "ERROR" flag. For more information on this topic, please see "Description of Log File Entries".
7. If you are still unable to pinpoint the error after checking the log file, please activate the Trace function and contact Customer Support.
   For more information on this topic, refer to "Configuring a Trace Function of a Channel".

See also

How to Configure the Trace Function of a Channel (Page 622)

Description of Log File Entries (Page 638)

How to Check a Tag (Page 647)

8.6.3.6 How to Check a Tag

Introduction

If an external tag does not have the expected value in Runtime, you can use the following procedure to check the tag.

In a "SIMATIC S7 Protocol Suite" channel, you can also use connection-specific internal tags. This procedure can also be used to check these tags.

Requirements

- Install a communication module on a WinCC computer, for example the CP 5613 A3 for MPI communication.
- Install the associated communication driver.
- Configure the CP 5613 A3.
- Create a STEP7 project.
- Configure a connection and tag for the "SIMATIC S7 Protocol Suite" channel.
- Activate the WinCC project.

Procedure

1. In WinCC Explorer in the tag management, select the "SIMATIC S7 Protocol Suite" channel.
2. In the data window, select the external tag that you wish to check. To do this, open the directory structure until the tag is displayed in the table area.
3. Move the mouse pointer over the tag to be checked. A tooltip window opens showing the current tag value, the quality code and the last time that the value changed.
4. Check the quality code. If value "80" is displayed, the tag value is OK. A description of the other values can be found under "Tag quality codes".
5. If the quality code is not equal to "80", select the tag in the tag management and click "Properties" in the shortcut menu to open the "Tag Properties" dialog.
6. Check whether values have been configured for the high or low limits, the start or substitute values on the "Limits/Reporting" tab. These values can affect the display.
7. If the tag value is affected by one of the configured values, deactivate the project and change the limit or substitute value.

Note

In Runtime, the current values of the connection-specific internal tags can be viewed in detail with "WinCC Channel Diagnosis". When the main connection is selected, the tags is shown in the "Counter" column.

Tag values, quality codes etc. are only displayed in Runtime.

See also

Quality Codes of Tags (Page 669)

8.7 Diagnosis of the "SIMATIC S5 Profibus FDL" Channel

8.7.1 Diagnostic Options for the "SIMATIC S5 PROFIBUS FDL" Channel

The following options for the detection of errors and the diagnosis of the "SIMATIC S5 PROFIBUS FDL" channel or one of its tags:

Checking the Communication Processor Configuration

Besides 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. Invalid tag values may also result from improperly addressing the tag in the AS.

Diagnosis of the Channel with "Channel Diagnosis"

"Channel Diagnosis" can query the status of the channel and connection in Runtime. Any errors that occur are displayed using "Error Codes".

Diagnosis of the Channel Tags

In tag management in Runtime, you can query the current value, the current quality code and the last time that the tag was changed.

See also

How to Check a Tag (Page 657)

How to Check the Channel and the Connection (Page 655)

How to Check the Configuration of the Connection and Tags (Page 653)

Checking the Communication Processor under SIMATIC NET (Page 652)

How to Check the Configuration of the Communication Processor (Page 651)

8.7.2 Description of Log File Entries

Introduction

The channel records errors and important status changes in the log file. These entries can be used to analyze a communications problem.

Each entry in the file has a date and time stamp followed by a flag name and description.

Example of a logbook entry:

```
2000-05-03 14:43:48,733 INFO Log starting ...
2000-05-03 14:43:48,733 INFO | LogFileName :
d:\Siemens\WinCC\Diagnose\SIMATIC_S5_PROFIBUS_FDL_01.LOG
2000-05-03 14:43:48,733 INFO | LogFileCount : 3
2000-05-03 14:43:48,733 INFO | LogFileSize : 1400000
2000-05-03 14:43:48,733 INFO | TraceFlags : fa017fff
```

Description of the "INFO" Flag Entries

Message text	Description
Log starting ...	Start message
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 : fa017fff	Displays the flags used by the Trace function as a hexadecimal number

8.7.3 Determining the Cause of Incorrect Tag Values

8.7.3.1 How to Determine the Cause of Incorrect Tags

If an unexpected tag value occurs in Runtime, proceed as follows to determine the cause:

1. Check the configuration of the communication processor
2. Check the communication processor under SIMATIC NET
3. Check the configuration of the connection and tags
4. Check the channel and connections
5. Check the tags of the channel

See also

How to Check the Configuration of the Connection and Tags (Page 653)

How to Check a Tag (Page 657)

How to Check the Channel and the Connection (Page 655)

Checking the Communication Processor under SIMATIC NET (Page 652)

How to Check the Configuration of the Communication Processor (Page 651)

8.7.3.2 How to Check the Configuration of the Communication Processor

Introduction

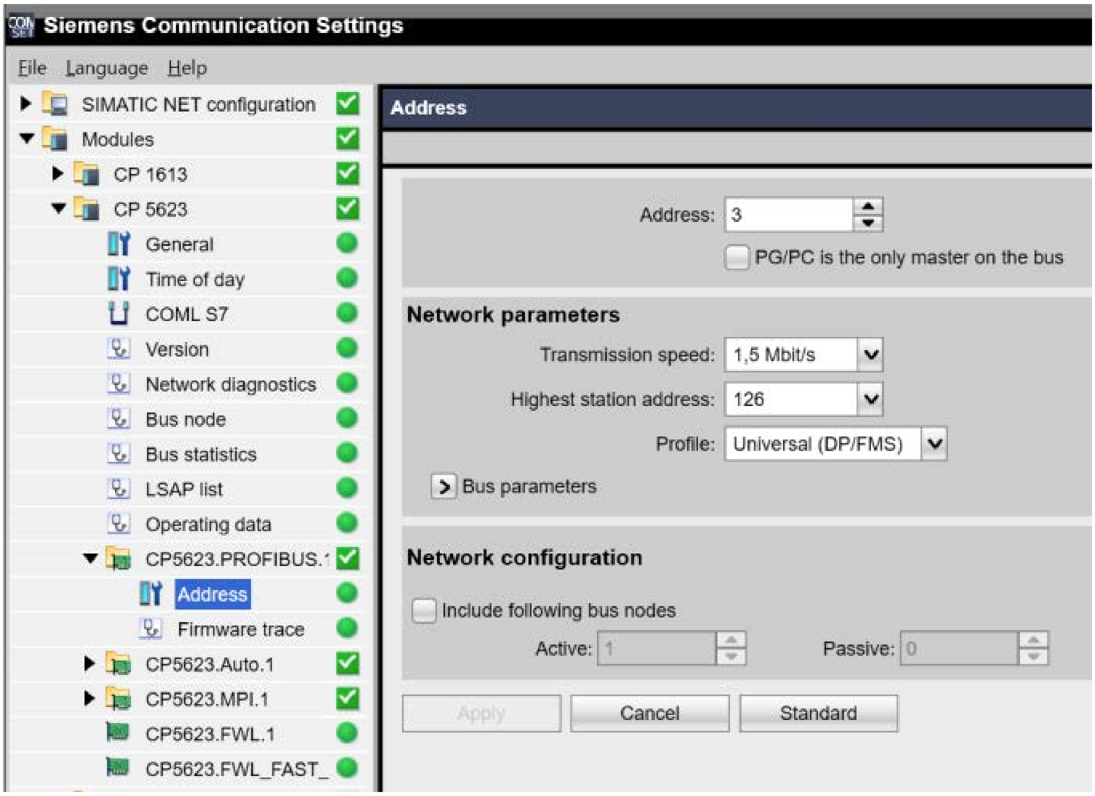
This section describes how to use the "PG/PC Port" program to check a communication processor. In this example, the "CP 5613 A3" type processor is used for the PROFIBUS communication.

Requirements

- Install the CP 5613 A3.
- Install the associated communication driver.
- Configure the CP 5613 A3.

Procedure

1. On the Control Panel, click the "Set PG/PC Port" icon. The "Set PG/PC Port" dialog is opened.
2. Check the entry for the access point. The access point "CP_L2_1:/SCP" for Profibus connection is automatically added when a CP 5613 A3 is installed. Select the entry for this access point. Click "Properties" to open the "Properties - CP5613A3.PROFIBUS.1" dialog.



Address

Address: 3

□ PG/PC is the only master on the bus

Network parameters

Transmission speed: 1,5 Mbit/s

Highest station address: 126

Profile: Universal (DP/FMS)

> Bus parameters

Network configuration

□ Include following bus nodes

Active: 1 Passive: 0

Figure 8-2 Example configuration: CP 5623

3. Check the entry on the "PROFIBUS" tab.

4. Click the "Operational State" tab. Click "Test" to execute a function test on the CP 5613 A3. The test results is shown in the output field below. Depending on the test results, you can click the "Restart" button to perform a reset and a complete restart of the CP 5613 A3.
5. Click "OK" to close all open dialogs.

See also

How to Check the Configuration of the Connection and Tags (Page 653)

Checking the Communication Processor under SIMATIC NET (Page 652)

8.7.3.3 Checking the Communication Processor under SIMATIC NET

Introduction

This section explains how to check a communication processor using the "Set PC Station" program in the SIMATIC NET software.

In this example, the "CP 5613 A3" type is used for the PROFIBUS communication to channel "SIMATIC S5 PROFIBUS FDL".

Requirements

- Install the CP 5613 A3.
- Install the SIMATIC NET software.
- Configure the CP 5613 A3 under SIMATIC NET.

Procedure

1. Open the menu item "Set PC station" in the SIMATIC NET settings.
   The "Configurations Console PC Station" dialog opens.
2. Check the entry for the access point. Select the "Access Point" directory in the navigation window. The existing access points are listed in the data window. During installation of the CP 5613 A3, access point "CP_L2_1:" is inserted automatically for the Profibus connection. Select this access point in the data window. Use the "Properties" menu item from the shortcut menu to open the "Properties of CP_L2_1:" dialog.
3. Check the entry in the "Assigned Interface Parameters" field. For a CP 5613 A3 in a PROFIBUS network, the entry "CP5613A3.PROFIBUS.1" should be selected.
4. Open the navigation window, select the "Components" directory and then the "CP5613 A3" subdirectory.
5. Select the "Network Diagnosis" directory. Click "Test" to execute a function test on the CP 5613 A3. The result is displayed in the output window. Depending on the test results, click "Restart" in the "General" directory to perform a reset and then a complete restart of the CP 5613 A3.

6. Check the list of participants connected to PROFIBUS in the list in the "Bus Participants" directory. Based on the display, it is possible to determine the function and status of your own station as well as other participants which are connected.
7. Close the dialog.
8. If a fault is detected in the configuration of the communication processor, modifications can only be made to the configuration using SIMATIC NET tools. Further information is available under SIMATIC NET.

See also

How to Check the Configuration of the Connection and Tags (Page 653)

8.7.3.4 How to Check the Configuration of the Connection and Tags

Introduction

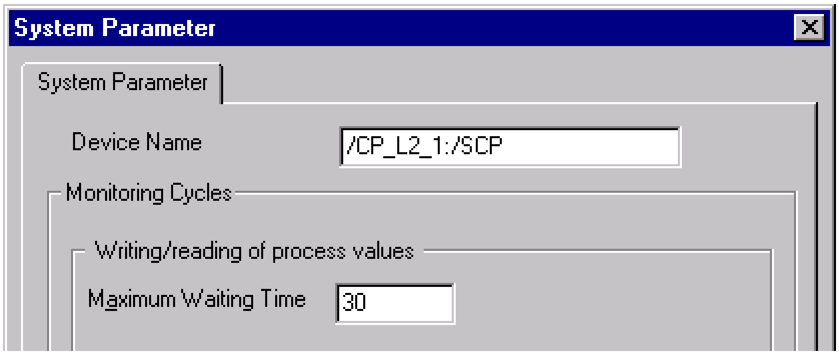
This section describes how to check the system parameters and the configuration of the connection and tags.

Requirements

- Install the CP 5613 A3.
- Install the communication driver.
- Configure the CP 5613 A3.
- Configure a connection and tag for the "SIMATIC S5 PROFIBUS FDL" channel.
- Activate the WinCC project.

Procedure

1. Click the plus sign in front of the "SIMATIC S5 PROFIBUS FDL" icon in WinCC Explorer navigation window. In the shortcut menu of the "FDL(CP5412/A2-1)" icon, click "System Parameters". The "System Parameters" dialog opens.



System Parameter

System Parameter

Device Name: /CP_L2_1:/SCP

Monitoring Cycles

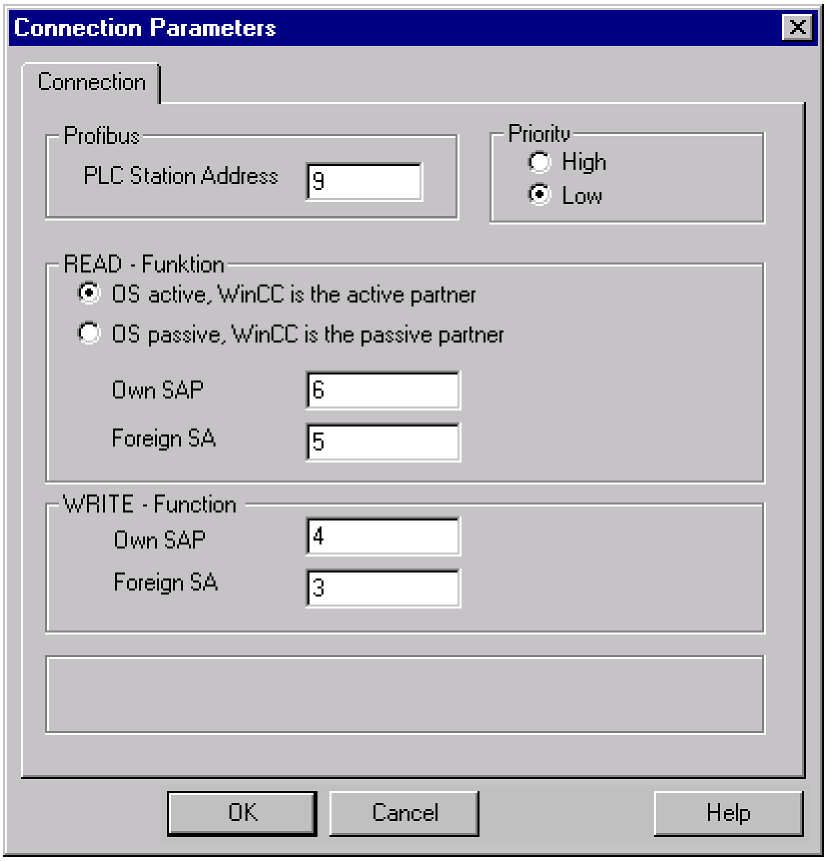
Writing/reading of process values

Maximum Waiting Time: 30

2. Check the entry in the "Device Name" field. By default, this is set to access point "CP_L2_1:/SCP". The access point is assigned during installation of the communication processor in the CP 5613 A3 system. Close the dialog.

3. Click the plus sign in front of the FDL(CP5412/A2-1)" icon. In the shortcut menu of the tag to be tested, select the "Properties" entry. The "Connection Properties" dialog is opened.

4. In the "Connection Properties" dialog, click the "Properties" button. The "Connection Parameters" dialog opens.



Connection Parameters

Connection

Profibus

PLC Station Address: 9

Priority

○ High

● Low

READ - Funktion

● OS active, WinCC is the active partner

○ OS passive, WinCC is the passive partner

Own SAP: 6

Foreign SA: 5

WRITE - Function

Own SAP: 4

Foreign SA: 3

5. Check the settings on the "Connection" tab. Close the open dialogs.
6. Click the plus sign in front of the icon of the connection. In the shortcut menu of the tag to be tested, click the "Properties" entry. The "Tag Properties" dialog opens. Check the entries in the "Type Conversion" and "Data Type" fields.
7. In the "Tag Properties" dialog, click the "Select" button. The "Address Properties" dialog opens. Check the settings.
8. Click "OK" to close all open dialogs.

See also

How to Check the Channel and the Connection (Page 655)

8.7.3.5 How to Check the Channel and the Connection

Introduction

This section describes how to check the "SIMATIC S5 PROFIBUS FDL" channel and its connection in Runtime.

Requirements

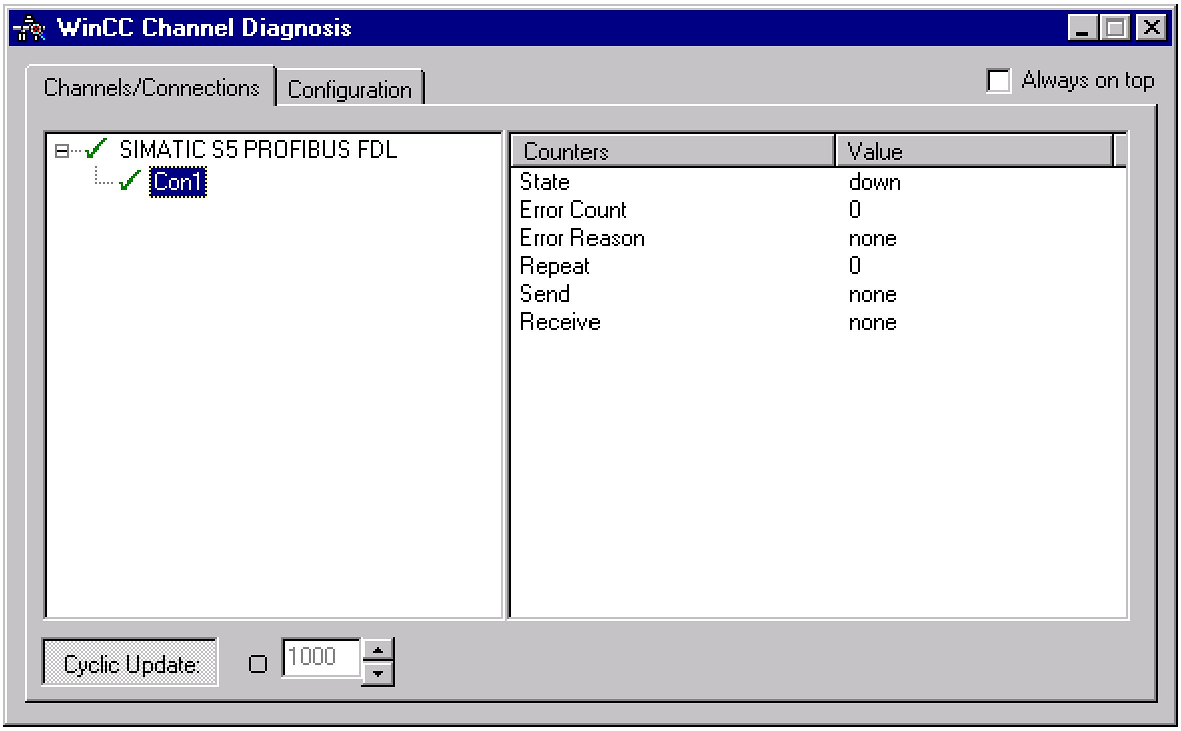
- Install the CP 5613 A3.
- Install the communication driver.
- Configure the CP 5613 A3.
- Create a STEP5 project.
- Configure a connection and tag for the "SIMATIC S5 PROFIBUS FDL" channel.
- Activate the WinCC project.

Overview of Status Messages

Icon	Description
✓	Channel / connection unconditionally ready
⚠	Channel / connection ready with some restrictions
?	Channel / no statement possible regarding connection status
✗	Channel / connection failed

Procedure

1. Start the WinCC Channel Diagnosis from the Start menu.
2. The Channel Diagnosis application window opens. The status information for all installed channels and their connections is displayed on the left on the "Channels/Connections" tab.



- ✓ SIMATIC S5 PROFIBUS FDL
  - ✓ Con1

Counters	Value
State	down
Error Count	0
Error Reason	none
Repeat	0
Send	none
Receive	none

□ 1000

3. Check the icon in front of the channel name "SIMATIC S5 PROFIBUS FDL" and the connection. If the status of the channel and connection are OK, a green check mark is displayed in front of each respective entry. For information on the significance of the individual icons refer to the "Overview of Status Messages" table.
4. If there is no green check mark in front of the channel's name and the connection, select the connection in the window on the left. In the window on the right, check the entries for the counters "State", "Error Count", "Error Reason", "Send" and "Receive". These values indicate the errors detected.
5. Check the channel-specific log file. To do this, use a text editor to open the file in the directory "Siemens\WinCC\Diagnose". Check the latest entries with the "ERROR" flag. For more information on this topic, please see "Description of Log File Entries".
6. If you are still unable to pinpoint the error after checking the log file, please activate the Trace function and contact Customer Support.
   For more information on this topic, refer to "Configuring a Trace Function of a Channel".

See also

How to Configure the Trace Function of a Channel (Page 622)

Description of Log File Entries (Page 649)

How to Check a Tag (Page 657)

8.7.3.6 How to Check a Tag

Introduction

If an external tag does not have the expected value in Runtime, you can use the following procedure to check the tag.

Requirements

- Install the CP 5613 A3.
- Install the communication driver.
- Configure the CP 5613 A3.
- Create a STEP5 project.
- Configure a connection and tag for the "SIMATIC S5 PROFIBUS FDL" channel.
- Activate the WinCC project.

Procedure

1. In WinCC Explorer in tag management, select the "SIMATIC S5 PROFIBUS FDL" channel.
2. In the data window, select the external tag that you wish to check. To do this, open the directory structure until the tag is displayed in the table area.
3. Move the mouse pointer over the tag to be checked. A tooltip window opens showing the current tag value, the quality code and the last time that the value changed.
4. Check the quality code. If value "80" is displayed, the tag value is OK. A description of the other values can be found under "Tag quality codes".
5. If the quality code is not equal to "80", select the tag in the tag management and click "Properties" in the shortcut menu to open the "Tag Properties" dialog.
6. Check whether values have been configured for the high or low limits, the start or substitute values on the "Limits/Reporting" tab. These values can affect the display.
7. If the tag value is affected by one of the configured values, deactivate the project and change the limit or substitute value.

Note

Tag values, quality codes etc. are only displayed in Runtime.

See also

Quality Codes of Tags (Page 669)

8.8 Diagnosis of the "OPC" Channel

8.8.1 Possibilities for Diagnosing the "OPC" Channel

There are the following possibilities for detecting errors and diagnosing the "OPC" channel or one of its tags:

Checking the Configuration of the Connection and Tags

There may be errors in the configuration of the system and connection parameters. Invalid tag values may also result from improperly addressing the tag in the AS.

Diagnosis of the Channel with "Channel Diagnosis"

"Channel Diagnosis" can query the status of the channel and connection in Runtime. Any errors that occur are displayed using "Error Codes".

Diagnosis of the Channel Tags

In tag management in Runtime, you can query the current value, the current quality code and the last time that the tag was changed.

See also

How to Check a Tag (Page 665)

How to Check the Channel and the Connection (Page 663)

How to Check the Configuration Data (Page 662)

Error Handling in the Event of Disturbed OPC Communication (Page 286)

8.8.2 Description of Log File Entries

8.8.2.1 Description of Log File Entries

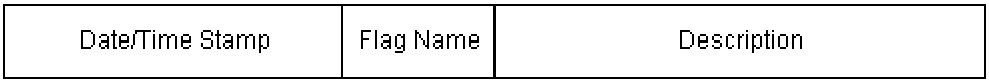
Introduction

The channel records errors and important status changes in the log file. The following sections cover only the most important entries. These entries can be used to analyze a communications problem.

A distinction must be made between two types of entries:

- INFO
- ERROR

Structure of an Entry



Date/Time Stamp	Flag Name	Description

Examples of entries in a logbook

```
2000-03-24 10:43:18,756 INFO Log starting ...
2000-03-24 10:43:18,756 INFO | LogFileName : C:\Siemens\WinCC\Diagnose\OPC.LOG
2000-03-24 10:43:18,756 INFO | LogFileCount : 3
2000-03-24 10:43:18,756 INFO | LogFileSize : 1400000
2000-03-24 10:43:18,756 INFO | TraceFlags : fa000007
000-03-24 10:43:18,756 INFO Process attached at 2000-03-24 09:43:18,746 UTC
2000-03-23 10:46:18,756 INFO Process detached at 2000-03-2410:46:18,746UTC
2000-03-27 13:22:43,390 ERROR ..FOPCData::InitOPC CoCreateInstanceEx- ERROR 800706ba
2000-03-27 13:22:43,390 ERROR - ChannelUnit::SysMessage("[OPC Groups (OPCHN
Unit #1)]![OPC_No_Machine]: CoCreateInstance for server "OPCServer.WinCC" on
machine OPC_No_Machine failed, Error=800706ba (HRESULT = 800706ba -
RPC_S_SERVER_UNAVAILABLE (Der RPC-Server ist nicht verfügbar.))")
```

See also

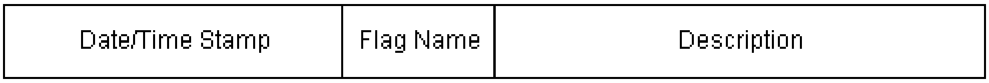
Entries for "ERROR" Flag (Page 660)

Entries for "INFO" Flag (Page 659)

8.8.2.2 Entries for "INFO" Flag

Introduction

Each entry in the file has a date and time stamp followed by a flag name and description.



Date/Time Stamp	Flag Name	Description

Examples of entries in a logbook

```
2000-03-24 10:43:18,756 INFO Log starting ...
2000-03-24 10:43:18,756 INFO | LogFileName : C:\Siemens\WinCC\Diagnose\OPC.LOG
2000-03-24 10:43:18,756 INFO | LogFileCount : 3
2000-03-24 10:43:18,756 INFO | LogFileSize : 1400000
2000-03-24 10:43:18,756 INFO | TraceFlags : fa000007
```

```
000-03-24 10:43:18,756 INFO Process attached at 2000-03-24 09:43:18,746 UTC
2000-03-23 10:46:18,756 INFO Process detached at 2000-03-2410:46:18,746UTC
```

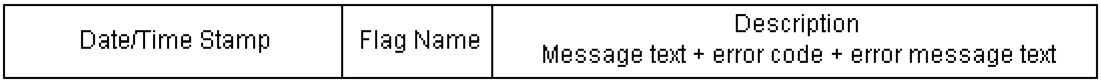
Description of the Mst Iportant Logbook Entries

Message text	Description
Log starting ...	Start message
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 : fa000007	Displays the flags used by the Trace function as a hexadecimal number
Process attached at 2000-03-24 09:43:18,746 UTC	The channel was loaded by the WinCC Data Manager.
Process detached at 2000-03-2410:46:18,746 UTC	The channel was unloaded by the WinCC Data Manager.
IOPCChnShutdown::ShutdownRequest was called... Reason: system going down" IOPCChnShutdown::ShutdownRequest	The WinCC OPC Server WinCC project was deactivated. The WinCC OPC clients are requested to disconnect from the WinCC OPC server.

8.8.2.3 Entries for "ERROR" Flag

Introduction

Each entry in the file has a date and time stamp followed by a flag name and description. In the case of the "Error" flag, the description consists of a message text, error code and the text of the error message. Some error codes do not have text for an error message.



Date/Time Stamp	Flag Name	Description Message text + error code + error message text

Examples of entries in a logbook

```
2000-03-27 13:22:43,390 ERROR ..FOPCData::InitOPC CoCreateInstanceEx- ERROR 800706ba

2000-03-27 13:22:43,390 ERROR - ChannelUnit::SysMessage("[OPC Groups (OPCHN
Unit #1)]![OPC_No_Machine]: CoCreateInstance for server "OPCServer.WinCC" on
machine OPC_No_Machine failed, Error=800706ba (HRESULT = 800706ba -
RPC_S_SERVER_UNAVAILABLE (RPC server not available.))")
```

Description of the Most Important Logbook Entries

Error Code	Error Message Text	Possible Causes
c0040004	Conversion between the "canonicalDatatype" and the "requestedDatatype" is not supported by the server.	Access to the WinCC tag on the OPC server failed. Conversion is possible but failed. The WinCC tag is not on the server or the configured data type does not match.
c0040007	The name does not exist in the name space of the server.	The error code is always returned by the server if the OPC client is accessed with a tag name that does not exist in the name space of the server. Examples: Browse, read tag, write tag, insert tag in a subscription.
00000001	AddItems	Access to the WinCC tag on the OPC server failed. The WinCC tag is not on the server or the configured data type does not match. Data Type WinCC Tag OPC Server = Data Type WinCC Tag OPC Client.
80004005	Could not resolve Server Name	The computer that is used as the WinCC OPC Server is not available in the network. The WinCC OPC server, which was accessed by the "OPC" channel, was not available.
80040154	Class not registered	The WinCC OPC Server is not properly registered in the system. The WinCC OPC Server's WinCC project is not activated.
80070057	Parameter wrong	The WinCC tag is not on the OPC Server, or the configured data type does not match.
800706ba	The RPC Server is not available.	The computer on which the OPC Server is to be started could not be found in the network.

8.8.3 Determining the Cause of Incorrect Tag Values

8.8.3.1 How to Determine the Cause of Invalid Tags

If an unexpected tag value occurs in Runtime, proceed as follows to determine the cause:

1. Check the configuration data
2. Check connection
3. Check the tags of the channel

See also

How to Check a Tag (Page 665)

How to Check the Channel and the Connection (Page 663)

How to Check the Configuration Data (Page 662)

8.8.3.2 How to Check the Configuration Data

Requirements

- A computer as WinCC OPC client with a WinCC project
- The "OPC" channel must be integrated in the WinCC project of the OPC client.
- Configure a WinCC tag in the WinCC project of the OPC server.
- On the OPC client, configure a connection and a WinCC tag which communicates with the created server tags.
- Activate Runtime on the OPC server and on the OPC client.

Procedure

1. On the OPC client, click the plus sign in front of the "OPC" icon in the WinCC Explorer navigation window.
   Click the plus sign in front of the "OPC Groups (OPCHN Unit#1)" icon.
2. In the shortcut menu of the connection to be tested, select the "Properties" entry.
   The "Connection Properties" dialog is opened.
3. In the OPC Connection tab, check the ProgID of the OPC server in the "OPC Server Name" field.
   - When connecting to a server with WinCC V5.0 or higher, "OPCServer.WinCC" must be entered.
   - When connecting to a server with WinCC V4.x or higher, "OE.Groups" must be entered.
4. Check that the name of the computer that is used as the OPC server is entered in the "Run the server on another computer" field.
5. To test the connection to the OPC server, click "Test Server".
   Close the dialog.
6. Click the plus sign in front of the icon of the connection.
7. In the shortcut menu of the tag to be tested, select the "Properties" entry.
   The "Tag Properties" dialog opens.
   - The same "Data Type" must be entered for this tag as for the tag on the OPC server.
8. In the "Tag Properties" dialog, click the "Select" button.
   The "Address Properties" dialog opens.
9. Check the entries in the fields "Item Name" and "Data Type".
   - The "Item Name" must match the tag name of the OPC server.
   - The "Data Type" must match the data type of the tag on the OPC server.

10. Check the channel-specific log file.
    To do this, use a text editor to open the file in the directory "Siemens\WinCC\Diagnose".
    Check the latest entries with the "ERROR" flag.
    For more information on this topic, refer to "Description of Log File Entries (Page 658)".
11. If you are still unable to pinpoint the error after checking the log file, activate the Trace function and contact Customer Support.
    You can find additional information under "How to Configure the Trace Function of a Channel (Page 622)".

See also

How to Configure the Trace Function of a Channel (Page 622)

Description of Log File Entries (Page 658)

How to Check the Channel and the Connection (Page 663)

8.8.3.3 How to Check the Channel and the Connection

Introduction

This section describes how to check the "OPC" channel and its connection in Runtime.

Requirements

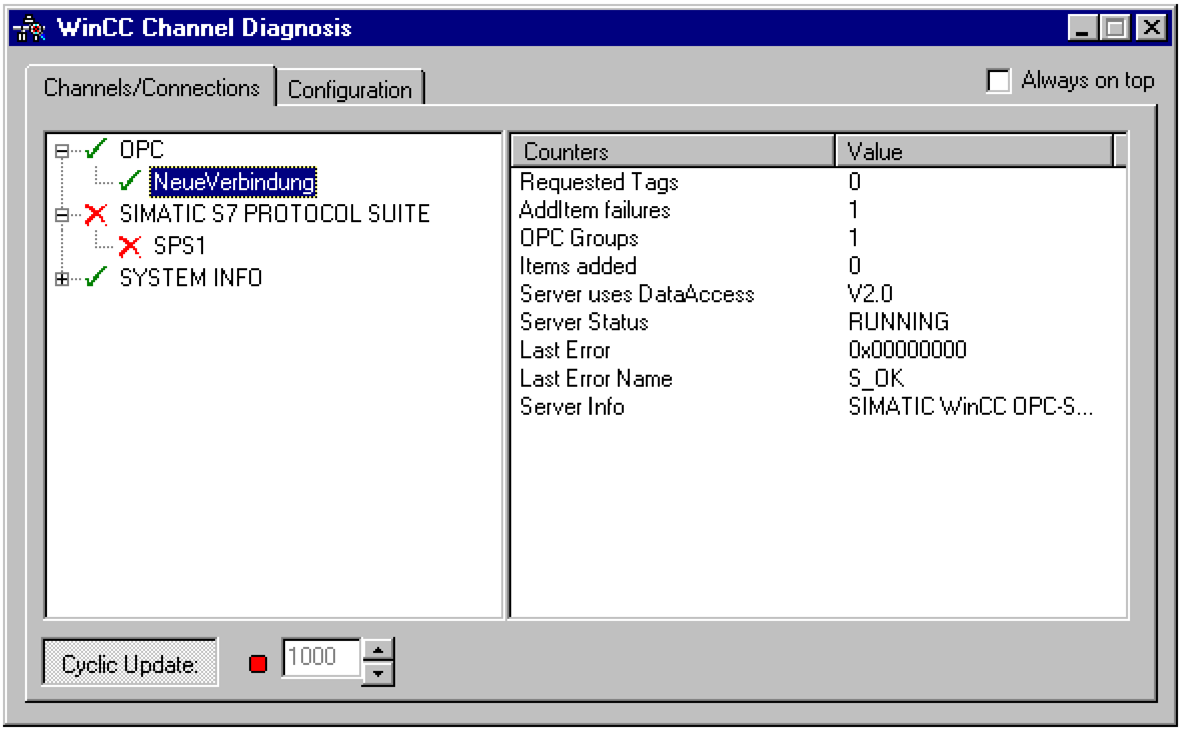
- A computer as WinCC OPC client with a WinCC project
- The "OPC" channel must be integrated in the WinCC project of the OPC client.
- Configure a WinCC tag in the WinCC project of the OPC server.
- On the OPC client, configure a connection and a WinCC tag which communicates with the created server tags.
- Enable the WinCC project on the OPC server and on the OPC client.

Overview of Status Messages

Icon	Description
✓	Channel / connection unconditionally ready
⚠	Channel / connection ready with some restrictions
?	Channel / no statement possible regarding connection status
✗	Channel / connection failed

Procedure

1. Start the WinCC Channel Diagnosis from the Start menu.
The Channel Diagnosis application window opens.
The status information for all installed channels and their connections is displayed on the left on the "Channels/Connections" tab.



Counters	Value
Requested Tags	0
AddItem failures	1
OPC Groups	1
Items added	0
Server uses DataAccess	V2.0
Server Status	RUNNING
Last Error	0x00000000
Last Error Name	S_OK
Server Info	SIMATIC WinCC OPC-S...

2. Check the icons in front of the OPC connection.
If the status of the connection is OK, a green check mark is displayed in front of the respective entry.
For information on the significance of the individual icons refer to the "Overview of Status Messages" table.

3. If there is no green check mark in front of the name of the connection, select the connection in the window on the left.

4. In the window on the right, check the entries for the counters "AddItemFailures", "Server Status", "Last Error" and "Last Error Name".
These values indicate the errors detected.

5. Check the channel-specific log file.
To do this, use a text editor to open the file in the directory "Siemens\WinCC\Diagnose".
Check the latest entries with the "ERROR" flag.
For more information on this topic, refer to "Description of Log File Entries (Page 658)".

6. If you are still unable to pinpoint the error after checking the log file, please activate the Trace function and contact Customer Support.
You can find additional information under "How to Configure the Trace Function of a Channel (Page 622)".

See also

How to Configure the Trace Function of a Channel (Page 622)

Description of Log File Entries (Page 658)

How to Check a Tag (Page 665)

8.8.3.4 How to Check a Tag

Introduction

If an external tag does not have the expected value in Runtime, you can use the following procedure to check the tag.

Requirements

- A computer as WinCC OPC Client with a WinCC Project.
- The "OPC" channel must be integrated in the OPC client's WinCC project.
- Configure a WinCC tag in the OPC server's WinCC project.
- On the OPC client configure a connection and a WinCC tag, which communicates with the created server tags.
- Activate the WinCC project on the OPC Server and Client.

Procedure

1. In WinCC Explorer in tag management, select the "OPC" channel.
2. In the data window, select the external tag that you wish to check. To do this, open the directory structure until the tag is displayed in the table area.
3. Move the mouse pointer over the tag to be checked. A tooltip window opens with the current tag value, the quality value and the time of the most recent change.
4. Check the quality value. If the value "C0" is displayed, the tag value is OK. A description of the other values can be found under "Tag quality codes".
5. If the quality code is not equal to "C0", select the tag in tag management and click "Properties" in the shortcut menu to open the "Tag Properties" dialog.
6. Check whether values have been configured for the high or low limits, the start or substitute values on the "Limits/Reporting" tab. These values can affect the display.
7. If the tag value is affected by one of the configured values, deactivate the project and change the limit or substitute value.

Note

Tag values, quality codes etc. are only displayed in Runtime.

See also

Quality Codes of Tags (Page 669)

8.9 Quality of Tags

8.9.1 Quality of Tags

Introduction

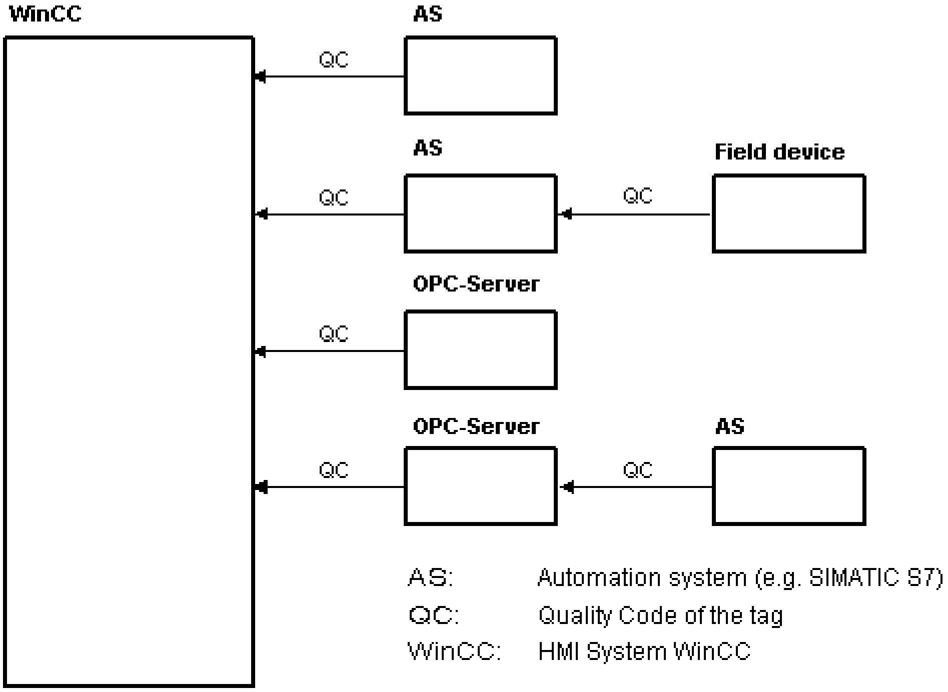
In WinCC, there are two quality indicators that allow you to evaluate the quality of tags. These two indicators are tag status and quality code.

The tag status is formed in WinCC and informs of the quality of configuration settings within the OS. The tag status informs additionally of the connection status to the WinCC communication partner. This may be an automated system or the server computer.

The quality code contains the same information as the tag status. In addition to this information, the quality status contains quality statements on partners which assess or process tags. Possible partners are:

- Automation systems
- Automation systems with field devices
- OPC server
- OPC server with subordinate automation systems

Therein the quality code is forwarded within the processing chain. If at one point in the processing chain several quality codes are pending for a tag, the worst code is forwarded.



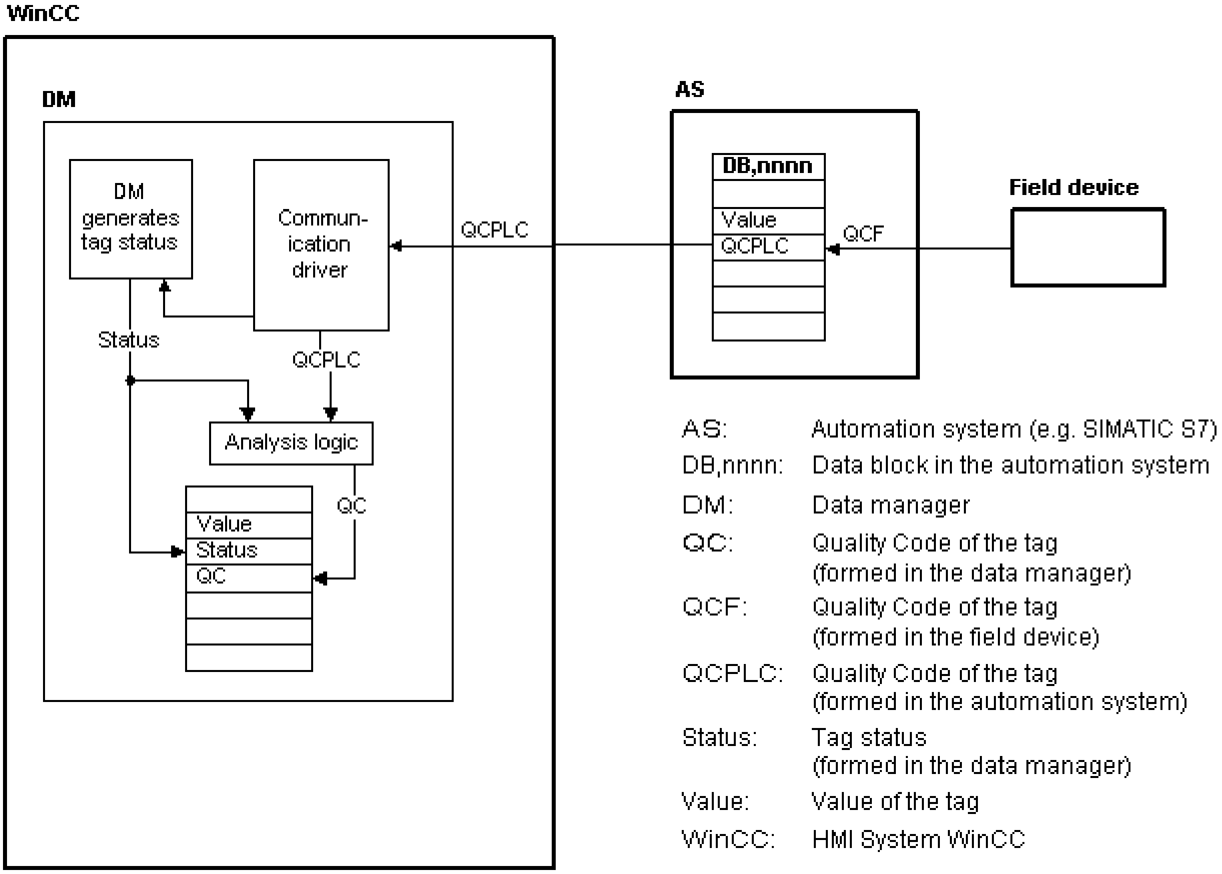
The quality code informs of the quality of a tag independent of where this code was formed.

Cascading of Quality Code

By using the example of an automation system with field device connected, the cascading of quality codes shall be outlined.

The automation system reads the quality codes generated by the field device. Using an analysis logic, quality codes pending for the same tag concurrently are evaluated by priority. The quality code with the worst status is assigned to the tag. This quality code must be saved in a data block directly behind the associated tag value.

You may initiate the analysis logic using the channel modules of the PCS7 Library. If the PCS7 Library is not available to you, you must configure the analysis logic in the automation system yourself.



AS:	Automation system (e.g. SIMATIC S7)
DB,nnnn:	Data block in the automation system
DM:	Data manager
QC:	Quality Code of the tag (formed in the data manager)
QCF:	Quality Code of the tag (formed in the field device)
QCPLC:	Quality Code of the tag (formed in the automation system)
Status:	Tag status (formed in the data manager)
Value:	Value of the tag
WinCC:	HMI System WinCC

Using one of the communication drivers, WinCC reads the tags from the automation device in Runtime, including the associated quality codes. For each tag, the tag status is formed in the data manager. It contains, for example, violations of configured measurement range limits as well as the status of linkage between WinCC and the automation device.

Using the analysis logic in the data manager, the quality code is generated from the tag status of the data manager and the quality code of the automation device. Here too, the code with the worst status is passed on and saved as quality code by WinCC. For tags that do not have a quality code in the automation system, the quality code is always identical with the tag status.

8.9.2 Quality Codes of Tags

Introduction

The quality code is needed to check status and quality of a tag. The displayed quality code summarizes the quality of the entire value transmission and value processing for the respective tag. Thus with the quality code you can for example see whether the current value is a start value or a substitute value.

The quality codes are prioritized. If several codes occur at the same time, the code with the worst status is displayed.

Evaluation of Quality Codes

Quality codes can be evaluated in a number of different ways:

- Evaluation with VB scripts
- Evaluation with C scripts
- Evaluation through the dynamic dialog
- Evaluation of the "Quality Code Change Tag" result of an I/O field

Note

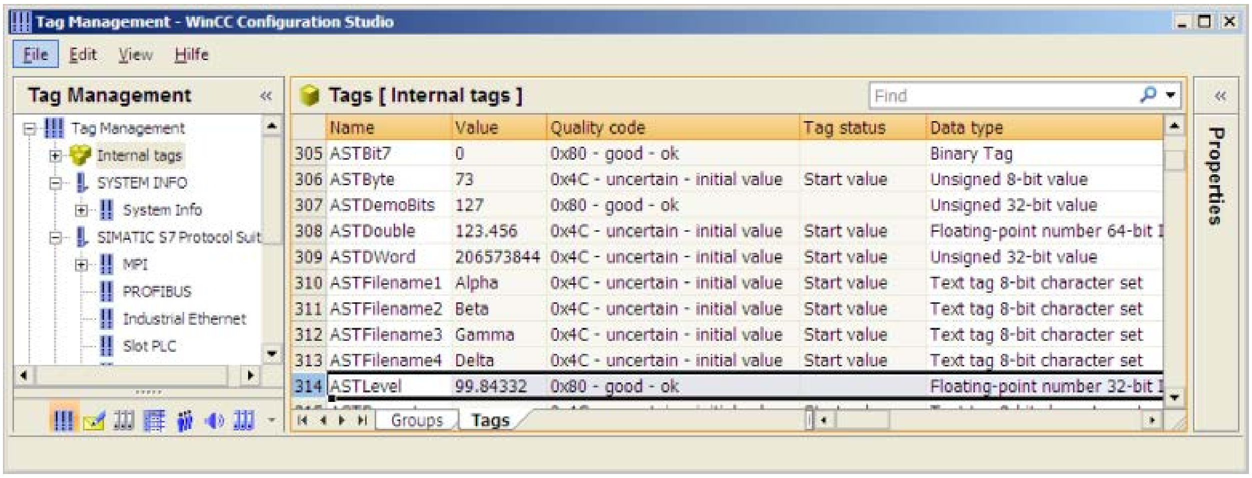
In order to include the entire value transfer and value processing in the quality code for a process tag, the connected automation system must support the quality code. When configuring the tags in the AS, make sure there is enough memory space for the quality code. In an AS from the S7 family, for example, the quality code needs an additional byte that is appended to the process value. To prevent errors, this byte must be taken into account when configuring a tag, for instance at the end of a data block.

Display of Quality Codes in Tag Management

You can view the quality code of a tag in Tag Management.

Requirements:

- The WinCC project is activated.
- The "Quality Code" column in the Tag Management data area is displayed.



	Name	Value	Quality code	Tag status	Data type
305	ASTBit7	0	0x80 - good - ok		Binary Tag
306	ASTByte	73	0x4C - uncertain - initial value	Start value	Unsigned 8-bit value
307	ASTDemoBits	127	0x80 - good - ok		Unsigned 32-bit value
308	ASTDouble	123.456	0x4C - uncertain - initial value	Start value	Floating-point number 64-bit I
309	ASTDWord	206573844	0x4C - uncertain - initial value	Start value	Unsigned 32-bit value
310	ASTFilename1	Alpha	0x4C - uncertain - initial value	Start value	Text tag 8-bit character set
311	ASTFilename2	Beta	0x4C - uncertain - initial value	Start value	Text tag 8-bit character set
312	ASTFilename3	Gamma	0x4C - uncertain - initial value	Start value	Text tag 8-bit character set
313	ASTFilename4	Delta	0x4C - uncertain - initial value	Start value	Text tag 8-bit character set
314	ASTLevel	99.84332	0x80 - good - ok		Floating-point number 32-bit I

Display of Quality Codes in Process Pictures

For the display of tag values in graphic objects with process connection, the quality code may affect the display. If the quality code has a value of 0x80 (good) or 0x4C (initial value), the display of the tag value is not grayed out. For all other values, the display is grayed out. In addition, a yellow warning triangle is displayed for the following objects depending on the set WinCC design:

- I/O field
- Bar, 3D bar
- Check box, radio box
- Group display, status display
- Slider object

Structure

The quality code has the following binary structure:

QQSSSSLL

Q: Quality

S: Substatus of the quality

L: Limits. This value is optional.

Note

The quality codes shown in the "Quality" table are basic values of the quality stages. Making use of the substatus and limit elements gives rise to intermediate values over and above the quality stage concerned.

Quality

The first two digits specify the quality of the tag.

	Q	Q	S	S	S	S	L	L
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Bad - The value is not useful	0	0	-	-	-	-	-	-
Uncertain - The quality of the value is less than normal, but the value may still be useful.	0	1	-	-	-	-	-	-
Good (non-cascade) - The quality of the value is good. Possible alarm conditions may be indicated by the sub-status.	1	0	-	-	-	-	-	-
Good (cascade) - The value may be used in control.	1	1	-	-	-	-	-	-

Substatus

The quality alone is not enough. Individual qualities are divided into substatuses. The quality code is binary coded. In order to analyze quality codes their values must be converted into their hexadecimal representation.

Quality Codes of Tags

Possible quality codes are listed in the following table. At the top of the list, you find the poorest quality code, while the best quality code is shown at the bottom of the list. The best quality code is assigned the lowest priority, while the poorest quality code is assigned the highest priority. If several statuses occur for one tag in the process, the poorest code is passed on.

Code (Hex)	**Quality**		**Q**	**Q**	**S**	**S**	**S**	**S**	**L**	**L**
0x23	Bad	Device passivated - Diagnostic alerts inhibited	0	0	1	0	0	0	1	1
0x3F	Bad	Function check - Local override	0	0	1	1	1	1	1	1
0x1C	Bad	Out of Service - The value is not reliable because the block is not being evaluated, and may be under construction by a configuration planner. Set if the block mode is O/S.	0	0	0	1	1	1	-	-
0x73	Uncertain	Simulated value - Start	0	1	1	1	0	0	1	1
0x74	Uncertain	Simulated value - End	0	1	1	1	0	1	-	-
0x84	Good (non-cascade)	Active Update event - Set if the value is good and the block has an active Update event.	1	0	0	0	0	1	-	-
0x24	Bad	Maintenance alarm - More diagnostics available.	0	0	1	0	0	1	-	-
0x18	Bad	No Communication, with no usable value - Set if there has never been any communication with this value since it was last "Out of Service".	0	0	0	1	1	0	-	-
0x14	Bad	No Communication, with last usable value - Set if this value had been set by communication, which has now failed.	0	0	0	1	0	1	-	-
0x0C	Bad	Device failure - Set if the source of the value is affected by a device failure.	0	0	0	0	1	1	-	-
0x10	Bad	Sensor failure	0	0	0	1	0	0	-	-

Code (Hex)	Quality		Q	Q	S	S	S	S	L	L
0x08	Bad	Not Connected - Set if this input is required to be connected and is not connected.	0	0	0	0	1	0	-	-
0x04	Bad	Configuration error - Set if the value is not useful because there is some inconsistency regarding the parameterization or configuration, depending on what a specific manufacturer can detect.	0	0	0	0	0	1	-	-
0x00	Bad	Non-specific - There is no specific reason why the value is bad. Used for propagation.	0	0	0	0	0	0	-	-
0x28	Bad	Process related - Substitute value	0	0	1	0	1	0	-	-
0x2B	Bad	Process related - No maintenance	0	0	1	0	1	0	1	1
0x68	Uncertain	Maintenance demanded	0	1	1	0	1	0	-	-
0x60	Uncertain	Simulated value - Set when the process value is written by the operator while the block is in manual mode.	0	1	1	0	0	0	-	-
0x64	Uncertain	Sensor calibration	0	1	1	0	0	1	-	-
0x5C	Uncertain	Configuration error	0	1	0	1	1	1	-	-
0x58	Uncertain	Subnormal	0	1	0	1	1	0	-	-
0x54	Uncertain	Engineering unit range violation - Set if the value lies outside of the set of values defined for this parameter. The limits define which direction has been exceeded.	0	1	0	1	0	1	-	-
0x50	Uncertain	Sensor conversion not accurate	0	1	0	1	0	0	-	-
0x4B	Uncertain	Substitute (constant)	0	1	0	0	1	0	1	1
0x78	Uncertain	Process related - No maintenance	0	1	1	1	1	0	-	-
0x4C	Uncertain	Initial value - Value of volatile parameters during and after reset of the device or of a parameter.	0	1	0	0	1	1	-	-
0x48	Uncertain	Substitute value - Predefined value is used instead of the calculated one. This is used for fail safe handling.	0	1	0	0	1	0	-	-
0x44	Uncertain	Last usable value - Whatever was writing this value has stopped doing so. This is used for fail safe handling.	0	1	0	0	0	1	-	-
0x40	Uncertain	Non-specific - There is no specific reason why the value is uncertain.	0	1	0	0	0	0	-	-
0xE0	Good (cascade)	Initiate fail safe (IFS) - The value is from a block that wants its downstream output block (e.g. AO) to go to fail safe.	1	1	1	0	0	0	-	-
0xD8	Good (cascade)	Local override (LO) - The value is from a block that has been locked out by a local key switch or is a Complex AO/DO with interlock logic active. The failure of normal control must be propagated to a function running in a host system for alarm and display purposes. This also implies "Not Invited".	1	1	0	1	1	0	-	-
0xD4	Good (cascade)	Do not select (DNS) - The value is from a block which should not be selected, due to conditions in or above the block.	1	1	0	1	0	1	-	-
0xCC	Good (cascade)	Not invited (NI) - The value is from a block which does not have a target mode that would use this input.	1	1	0	0	1	1	-	-
0xC8	Good (cascade)	Initialization request (IR) - The value is an initialization value for a source (back calculation input parameter), because the lower loop is broken or the mode is wrong.	1	1	0	0	1	0	-	-
0xC4	Good (cascade)	Initialization acknowledge (IA) - The value is an initialized value from a source (cascade input, remote-cascade in, and remote-output in parameters).	1	1	0	0	0	1	-	-

Code (Hex)	Quality		Q	Q	S	S	S	S	L	L
0xC0	Good (cascade)	OK - No error or special condition is associated with this value.	1	1	0	0	0	0	-	-
0xA0	Good (non-cascade)	Initiate fail safe	1	0	1	0	0	0	-	-
0x98	Good (non-cascade)	Unacknowledged critical alarm - Set if the value is good and the block has an unacknowledged alarm with a priority greater than or equal to 8.	1	0	0	1	1	0	-	-
0x94	Good (non-cascade)	Unacknowledged advisory alarm - Set if the value is good and the block has an unacknowledged alarm with a priority less than 8.	1	0	0	1	0	1	-	-
0x90	Good (non-cascade)	Unacknowledged update event - Set if the value is good and the block has an unacknowledged update event.	1	0	0	1	0	0	-	-
0x8C	Good (non-cascade)	Active critical alarm - Set if the value is good and the block has an active alarm with a priority greater than or equal to 8.	1	0	0	0	1	1	-	-
0x88	Good (non-cascade)	Active advisory alarm - Set if the value is good and the block has an active alarm with a priority less than 8.	1	0	0	0	1	0	-	-
0xA8	Good (non-cascade)	Maintenance demanded	1	0	1	0	1	0	-	-
0xA4	Good (non-cascade)	Maintenance required	1	0	1	0	0	1	-	-
0xBC	Good (non-cascade)	Function check - Local override	1	0	1	1	1	1	-	-
0x80	Good (non-cascade)	OK - No error or special condition is associated with this value.	1	0	0	0	0	0	-	-

Limit

Quality codes can be further subdivided by limits. Limits are optional.

	Q	Q	S	S	S	S	L	L
O.K. - The value is free to move.	-	-	-	-	-	-	0	0
Low limited - The value has acceded its low limits.	-	-	-	-	-	-	0	1
High limited - The value has acceded its high limits.	-	-	-	-	-	-	1	0
Constant (high and low limited) - The value cannot move, no matter what the process does.	-	-	-	-	-	-	1	1

Quality Codes in Communication with OPC

In the communication via the "OPC" channel, the quality codes that the OPC does not support are converted.

Quality Code in WinCC	Quality Code in OPC
0x48	0x40
0x4C	0x40
0x5C	0x40
0x60	0x40
0x80...0xD4	0xC0
0xD8	0xC0

8.9.3 Tag Status

Introduction

The tag status of individual WinCC tags can be monitored in Runtime. The tag status contains, among other information, data regarding violations of the configured measurement range limits as well as the status of linkage between WinCC and automation device.

The quality code informs of the quality of a tag independent of where this code was formed. Thereby, the status of the entire value transfer and value processing are taken into consideration.

For example, if a violation occurs of the measurement range at the lower limit, quality code "0x55" is communicated. This violation of the measurement range might have occurred in the WinCC data manager or in the field device. The tag status allows you to find out if this measurement range violation occurred in WinCC or prior to passing the value to WinCC.

For example, if the tag status reports a limit violation with code 0x0010, it indicates that the values remained below the lower range limit configured in WinCC. If the tag status does not report any limit violation, the quality code passed on to WinCC already contained the limit violation.

Evaluation of Quality Codes

Quality codes can be evaluated in a number of different ways:

- Evaluation with C scripts
- Evaluation through the dynamic dialog
- Evaluation of the "Quality Code Change Tag" result of an I/O field

WinCC Status Flags

Possible tag statuses are contained in the following table.

Name of flag	Value	Description
	0x00 00	No error
DM_VARSTATE_NOT_ESTABLISHED	0x00 01	Connection to partner not established
DM_VARSTATE_HANDSHAKE_ERROR	0x00 02	Handshake error
DM_VARSTATE_HARDWARE_ERROR	0x00 04	Network module defective
DM_VARSTATE_MAX_LIMIT	0x00 08	Configured upper limit exceeded
DM_VARSTATE_MIN_LIMIT	0x00 10	Configured lower limit exceeded
DM_VARSTATE_MAX_RANGE	0x00 20	Format upper limit exceeded
DM_VARSTATE_MIN_RANGE	0x00 40	Format lower limit exceeded
DM_VARSTATE_CONVERSION_ERROR	0x00 80	Display conversion error (in connection with format limit xxx exceeded)
DM_VARSTATE_STARTUP_VALUE	0x01 00	Tag initialization value
DM_VARSTATE_DEFAULT_VALUE	0x02 00	Tag replacement value
DM_VARSTATE_ADDRESS_ERROR	0x04 00	Channel addressing error
DM_VARSTATE_INVALID_KEY	0x08 00	Tag not found / not available
DM_VARSTATE_ACCESS_FAULT	0x10 00	Access to tag not permitted
DM_VARSTATE_TIMEOUT	0x20 00	Timeout / no check-back message from the channel
DM_VARSTATE_SERVERDOWN	0x40 00	Server not available.

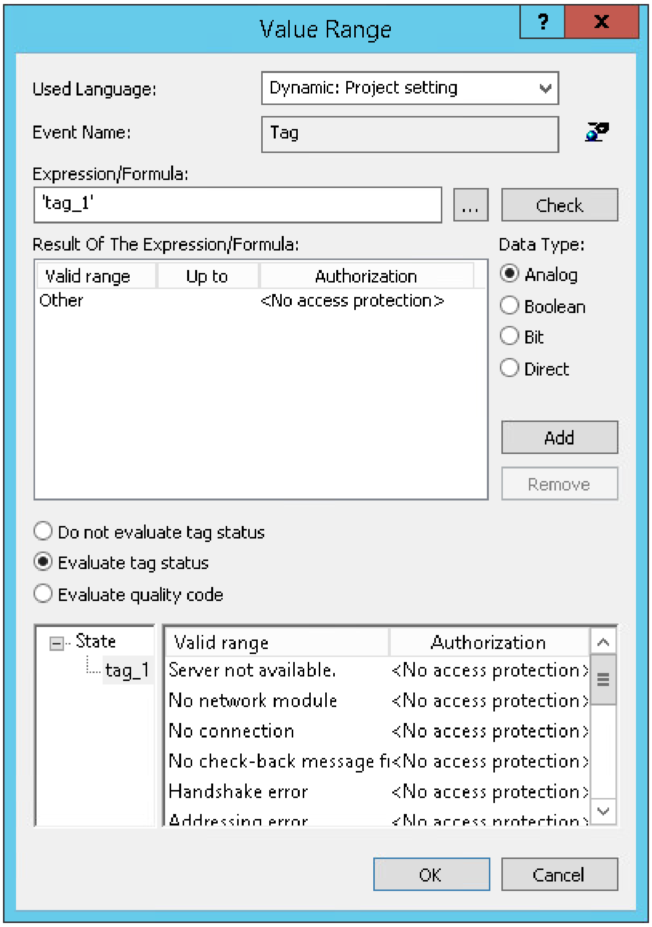
8.9.4 Using the Tag Status to Monitor Connection Status

You can monitor the status of individual WinCC tags in runtime, providing information about the status of the associated connection.

Configure status monitoring

Configure the monitoring as an object property in Graphics Designer.

1. Select the required property in the "Object Properties" window.
2. Select the "Dynamic dialog" entry from the shortcut menu of the "Dynamic" column.
   The "Value range" dialog opens.



Value Range

Used Language: Dynamic: Project setting

Event Name: Tag

Expression/Formula: 'tag_1'

Result Of The Expression/Formula:

Valid range	Up to	Authorization
Other		<No access protection>

Data Type:
● Analog
○ Boolean
○ Bit
○ Direct

○ Do not evaluate tag status
● Evaluate tag status
○ Evaluate quality code

State
tag_1

Valid range	Authorization
Server not available.	<No access protection>
No network module	<No access protection>
No connection	<No access protection>
No check-back message fr	<No access protection>
Handshake error	<No access protection>
Addressing error	<No access protection>

3. Specify the settings:
   – Monitored tag.
   – Tag value: Assignment of valid range and status display
   – Status evaluation of the tags
   – Status: Assignment of valid range and corresponding status text

One of the entered status texts, corresponding to the current tag status, is displayed in the configured object in runtime.

See also

Configuring tags for the connection status in Runtime (Page 193)

8.9.5 Monitoring Tag Status Using Global Actions

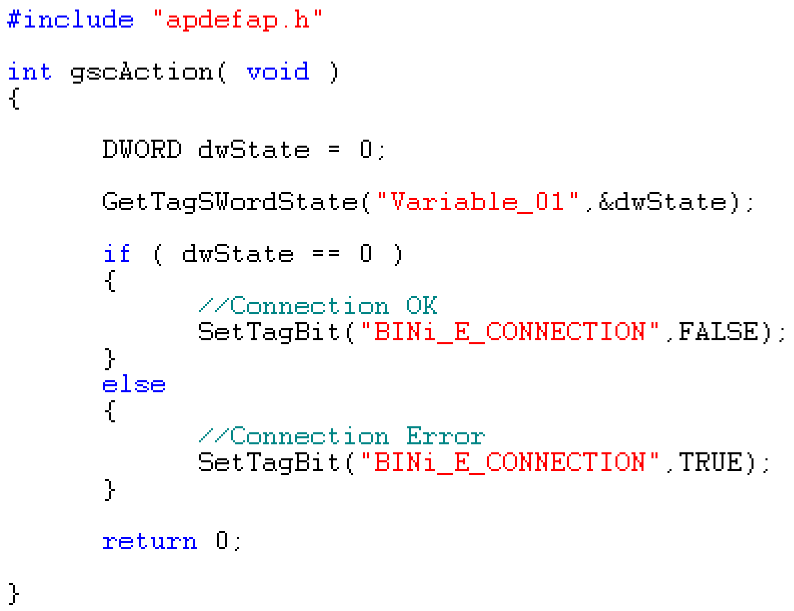
Introduction

One way to monitor the status of a tag is to make use the internal functions "GetTagState" and "GetTagStateWait" in the Global Script editor. In contrast to the "GetTag" and "GetTagWait" functions, these not only return the tag's value but also its status. This status value can be evaluated and then used to trigger various events. It can also be used to assess the status of the associated connection.

In the global action, the status value of the monitored tag is determined using the "GetTagState" function for this tag type. There is such a function for each tag type. The status value "0" indicates a good connection with no errors. This status can now be evaluated as desired.

Example:

This example illustrates the monitoring of a WinCC tag of the type "Signed 16-Bit Value". The "GetTagSWordState" function is used to determine the status of this tag. The first function parameter is the name of the WinCC tag to be monitored. The second parameter gives where the returned status value is to be written.



```
#include "apdefap.h"

int gscAction( void )
{

      DWORD dwState = 0;

      GetTagSWordState("Variable_01",&dwState);

      if ( dwState == 0 )
      {
            //Connection OK
            SetTagBit("BINi_E_CONNECTION",FALSE);
      }
      else
      {
            //Connection Error
            SetTagBit("BINi_E_CONNECTION",TRUE);
      }

      return 0;

}
```

The tag status is output in the internal tag BINi_E_CONNECTION. In the event of an error, the value of this tag is set to TRUE. In the error handling, the tag can, for example, be used to trigger an alarm or display an error message.

8.9.6 How to Check an Internal Tag

Introduction

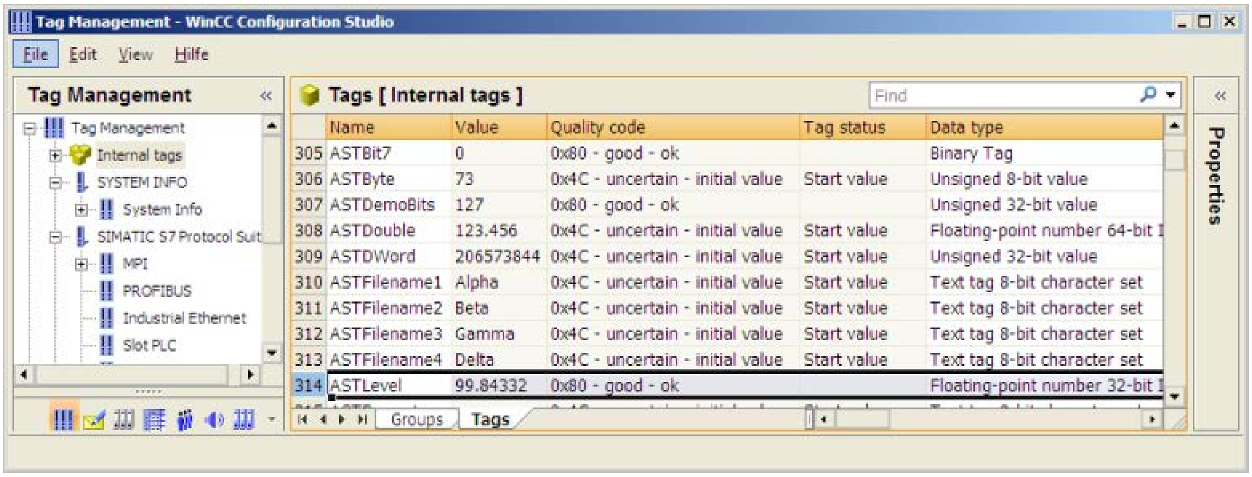
If an internal tag does not have the expected value in Runtime, you can use the following procedure to check the tag.

Requirements

- An internal tag has been configured.
- The WinCC project is activated.

Procedure

1. Open Tag Management in the WinCC Explorer.
2. Select the entry "Internal tags" and the tag to be checked in the navigation area.
3. To display the "Quality Code" and "Value" columns in the data area, you may have to go to "Show" and select these columns in the shortcut menu of a column header.
4. Check the quality code. If the value "80" is displayed, the tag value is OK. A description of the other values can be found under "Tag Quality Codes".
5. If the quality code is not equal to "80", check the settings under properties on the right.
6. Check whether values have been configured for the upper and lower limits or start value. These values can affect the display.
7. If the tag value is affected by one of the configured values, deactivate the project and change the limit or substitute value.



Tags [Internal tags]

	Name	Value	Quality code	Tag status	Data type
305	ASTBit7	0	0x80 - good - ok		Binary Tag
306	ASTByte	73	0x4C - uncertain - initial value	Start value	Unsigned 8-bit value
307	ASTDemoBits	127	0x80 - good - ok		Unsigned 32-bit value
308	ASTDouble	123.456	0x4C - uncertain - initial value	Start value	Floating-point number 64-bit I
309	ASTDWord	206573844	0x4C - uncertain - initial value	Start value	Unsigned 32-bit value
310	ASTFilename1	Alpha	0x4C - uncertain - initial value	Start value	Text tag 8-bit character set
311	ASTFilename2	Beta	0x4C - uncertain - initial value	Start value	Text tag 8-bit character set
312	ASTFilename3	Gamma	0x4C - uncertain - initial value	Start value	Text tag 8-bit character set
313	ASTFilename4	Delta	0x4C - uncertain - initial value	Start value	Text tag 8-bit character set
314	ASTLevel	99.84332	0x80 - good - ok		Floating-point number 32-bit I

Note

The tag value and the quality codes are only displayed in Runtime.

See also

Quality Codes of Tags (Page 669)

OPC - Open Connectivity

9

9.1 OPC - Open Connectivity

Contents

The OPC standardized software interface allows you to combine devices and applications from various manufacturers in a uniform manner.

WinCC can be used as an OPC server or an OPC client. The "OPC" channel represents the OPC client application of WinCC.

This section shows you:

- which OPC servers WinCC has.
- how to use OPC in WinCC.
- how to set up various OPC DA links.
- how to configure the access to the WinCC message system.
- how the WinCC message system is mapped on the OPC A&E.
- how to set up access to the WinCC archive system.

9.2 Functionality of OPC

OPC is a standardized manufacturer-independent software interface for data exchange in automation engineering.

OPC interfaces allow the standard linking of devices and applications from different manufacturers.
OPC is based on the Windows COM (Component Object Model) and DCOM (Distributed Component Object Model) technologies.

OPC XML DA provides an additional software interface that is based on the XML, SOAP and HTTP Internet standards.

OPC UA (Unified Architecture) is the successor technology to OPC. OPC UA is platform-independent and supports different protocols as communication medium.

9.3 OPC specifications and compatibility

Overview

OPC specifies interfaces for access to the following objects in WinCC:

- Process values (OPC Data Access 2.05a, 3.0; OPC XML Data Access 1.01; OPC UA 1.03)
- Archived process values (OPC Historical Data Access 1.20; OPC UA Historical Access 1.03)
- Chronological messages (OPC Historical Alarms and Events 1.10)
- Messages (OPC Alarms and Events 1.10; OPC UA Alarms and Conditions 1.03)

For more information about individual OPC specifications, refer to the OPC Foundation (http://www.opcfoundation.org) website.

Compatibility

Support of these specifications is regularly monitored by the "Compliance Test Tool" (CTT) of the OPC Foundation. Interoperability with OPC products from other manufacturers is guaranteed by participation in "OPC Interoperability Workshops".

The test results submitted are published on the OPC Foundation website. To view the results, enter the search term "OPC Self-Certified Products".

9.4 Using OPC in WinCC

Introduction

In WinCC, servers are available for the following OPC interfaces:

- OPC Data Access / OPC XML Data Access: Access to the WinCC body of data
- OPC Historical Data Access: Access to the WinCC archive system
- OPC Alarms&Events: Access to the WinCC message system
- OPC Unified Architecture: Access to the WinCC body of data and archive system

WinCC contains an OPC channel by default. The OPC channel can access the relevant OPC servers as client via OPC DA , OPC XML DA or OPC UA.

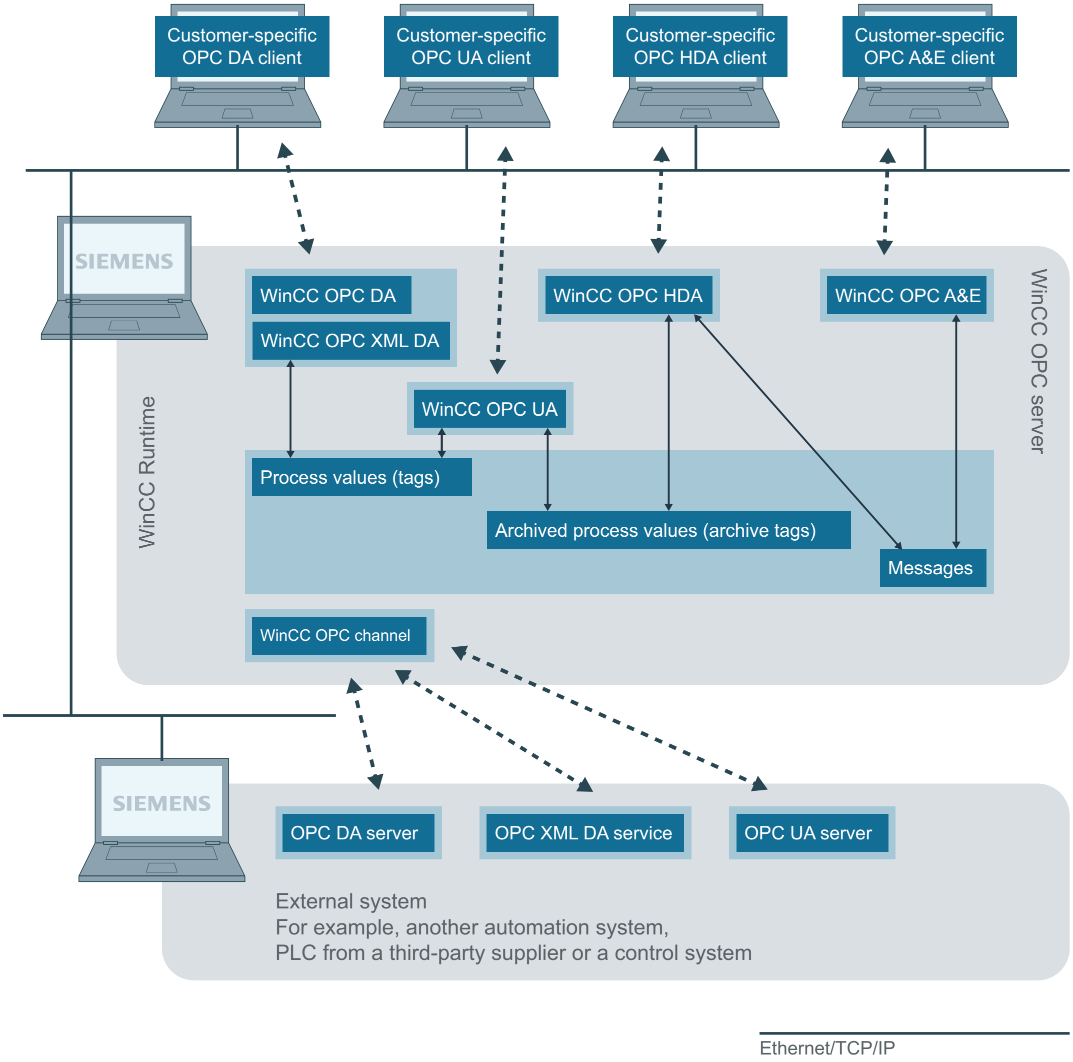
WinCC OPC communications concept

Data exchange between a WinCC OPC server and OPC client is completed via DCOM. After installation of WinCC, the DCOM settings of the WinCC OPC server are correctly configured.

If a WinCC OPC server or client communicates with an external OPC system, corresponding adaptations must be performed. The "Local access" and "Remote access" authorizations must be entered for the user in "DCOM/Workplace/COM Security/Access rights/Edit default" of User Administration on the client.

The OPC XML server of WinCC is implemented as a web service. This gives you access to your PC via the Internet. You therefore need to define appropriate access rights.

The following shows the WinCC OPC communication concept:


Customer-specific OPC DA client
Customer-specific OPC UA client
Customer-specific OPC HDA client
Customer-specific OPC A&E client
SIEMENS
WinCC Runtime
WinCC OPC server
WinCC OPC DA
WinCC OPC XML DA
WinCC OPC HDA
WinCC OPC A&E
WinCC OPC UA
Process values (tags)
Archived process values (archive tags)
Messages
WinCC OPC channel
SIEMENS
OPC DA server
OPC XML DA service
OPC UA server
External system
For example, another automation system,
PLC from a third-party supplier or a control system
Ethernet/TCP/IP

Licensing

OPC server	Licensing
WinCC OPC DA server	A valid RT license for WinCC
WinCC OPC XML DA Server WinCC OPC UA Server WinCC OPC HDA server WinCC OPC A&E Server	A valid RT license for WinCC WinCC Option Connectivity Pack

9.5 How to configure Windows for the use of WinCC OPC

Introduction

The OPC client and the OPC server are DCOM applications. A distributed DCOM application can only be run under the same user account. Therefore the OPC server must recognize the OPC client's user account and vice-versa. If the WinCC OPC servers are used with WinCC OPC clients, the correct configuration is already warranted by the installation.

Declaration of the user account, if an external OPC server or client is used

For additional information on the granting of user rights, refer to the Windows documentation.

Requirements

Log on as the administrator to both the WinCC OPC server and OPC client workstations to configure the user permissions.

Procedure

1. Go to "Control Panel > System and Security > Administrative Tools > Computer Management > Local Users and Groups".
2. In the "Users" shortcut menu, select "New User".
   In the "New User" dialog, enter the user account details of the communication partner. Click "Create" and close the dialog.
3. Click the "Users" icon. Double-click the relevant user. The "Properties" dialog for this user is displayed.
4. Click the "Member Of" tab. Click "Add". The "Select group" dialog is opened.
5. Add the group "Users".
   If you are on a computer that has WinCC installed, also add the group "SIMATIC HMI". Click "OK" to close all open dialogs.

How to adapt the Windows firewall settings

After installation of WinCC, the Windows firewall settings of the WinCC OPC servers are correctly configured.

If OPC clients access OPC servers in different subnets, you must adapt the configuration of the permitted network areas to the OPC servers.

9.6 WinCC OPC DA server

9.6.1 Functionality of the WinCC OPC DA Server

Introduction

The WinCC OPC-DA-Server supports OPC Data Access specifications 2.05a and 3.00. This has been confirmed by the compliance test.

The WinCC OPC DA server is a DCOM application. This interface is used by the WinCC OPC DA server to make the required information about WinCC tag available to the WinCC client.

The WinCC OPC DA server is active, if the WinCC OPC DA client is accessing it via a connection. To establish successful OPC communication, the following must be observed:

- The WinCC project of the WinCC OPC DA server must be enabled.
- The computer on which the WinCC OPC DA server runs must be accessible via its IP address.

Installation

The WinCC OPC DA server can be selected during the installation of WinCC. After installation, the WinCC OPC DA server is immediately usable without any further configuration.

The WinCC OPC DA server can be implemented on a WinCC server or a WinCC client.

Notes on configuration

- If the WinCC-OPC-DA server is used, the application "OPC-DA server, OPC-A&E server, OPC-HDA server" must be activated.
  You can activate the application in the Editor "Computer" of the WinCC Configuration Studio in the "Processes when starting WinCC Runtime" tab.
- You can assemble tags into tag groups for structuring in the WinCC project. The tags should not have the same name as the group.
- Each write request initiated in WinCC, for example via VBScript or the object "IO field", is always treated as a synchronous "Write" call. The "IOPCSyncIO::Write" interface is used by the WinCC OPC DA server for this. The asynchronous write mechanism is not implemented in the WinCC OPC DA channel.

Note

If the Internet options on a computer are set to automatically detect settings under "Connections -> LAN Settings", access to OPC DA via the web service will take significantly longer.

See also

Querying the OPC DA Server Name (Page 689)

Using Multiple OPC DA Servers (Page 688)

Example of WinCC to WinCC Connection (Page 690)

Example of WinCC - SIMATIC NET FMS OPC Server Connection (Page 694)

Example of a WinCC - SIMATIC NET S7 OPC Server Connection (Page 696)

Example of the WinCC - Microsoft Excel Connection (Page 701)

Overview of the Supported WinCC Data Types (Page 273)

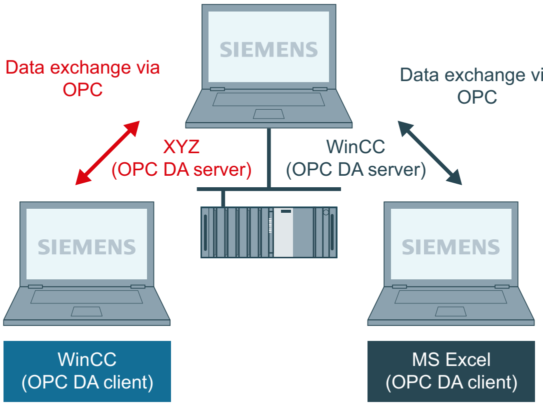
www.opcfoundation.org (http://www.opcfoundation.org)

9.6.2 Using Multiple OPC DA Servers

Introduction

More than one OPC DA server may be installed on a computer, and any number may work in parallel.
In this way, the OPC DA server of WinCC and the OPC DA server of another (third-party) provider may be operated independently of one another on the same computer.

The WinCC OPC DA client can access the process data of the automation device via the OPC server of the third-party provider. The OPC DA client of Microsoft Excel can use the WinCC OPC DA server to access the WinCC data.



There are a number of OPC DA servers available from various manufacturers. Each of these OPC DA servers has a unique name (ProgID) for identification. OPC DA clients must use this name to address the OPC server.

The OPC Item Manager can be used to query the name of the OPC DA server. The OPC DA server of WinCC V 7 is named: "OPCServer.WinCC".

See also

Querying the OPC DA Server Name (Page 689)

9.6.3 Querying the OPC DA Server Name

Introduction

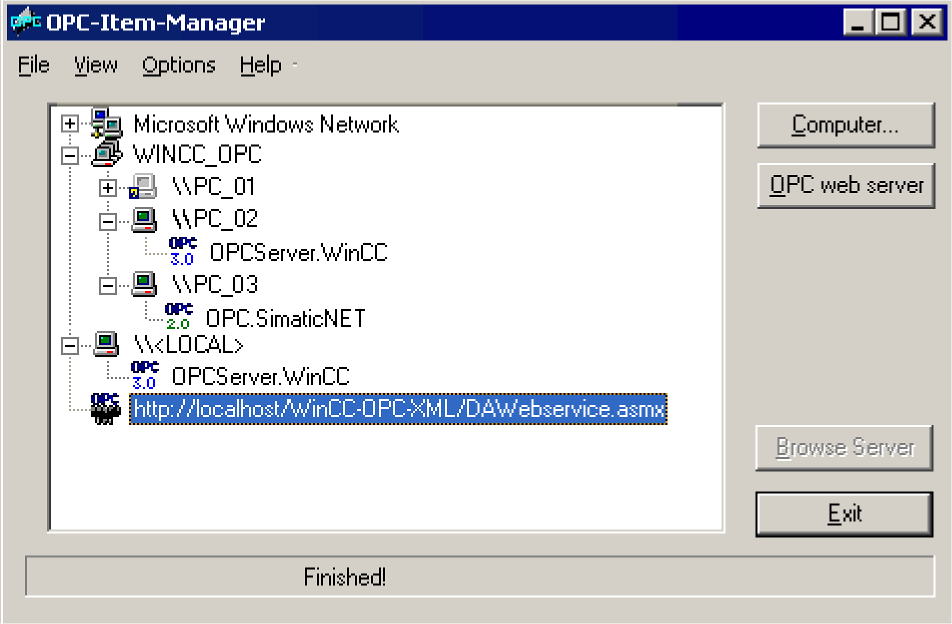
Multiple OPC DA servers can be installed on a single computer. The OPC Item Manager displays the names of the OPC DA servers available to the workstation in a selection window. These OPC DA servers can be run on the same computer or on computers in the network environment.

Requirement

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

Procedure

1. In the shortcut menu of the channel unit "OPC Groups(OPCHN Unit#1)" on the WinCC OPC DA client, select "System Parameters". The "OPC Item Manager" is opened.
2. In the navigation window of the OPC Item Manager, select the name of the computer you wish to access.
3. The OPC Item Manager displays the names of the OPC DA servers that available to your computer in a selection window.



See also

OPC Item Manager (Page 270)

9.6.4 Examples of OPC DA Connections

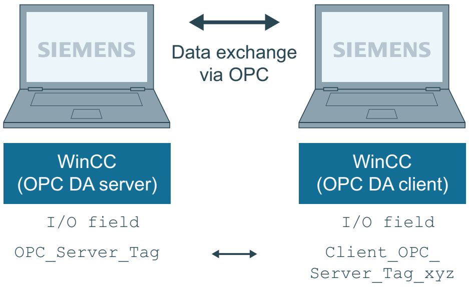
9.6.4.1 WinCC - WinCC Connection

Example of WinCC to WinCC Connection

Introduction

When establishing a WinCC - WinCC connection, data are exchanged between the WinCC OPC DA server and client by means of the "OPC_Server_Tag" WinCC tag. The "Client_OPC_Server_Tag_xyz" WinCC tag on the client reads the "OPC_Server_Tag" WinCC tag on the server. If the value of the "OPC_Server_Tag" tag on the WinCC OPC server changes, the value of the "Client_OPC_Server_Tag_xyz" WinCC tag on the WinCC OPC DA client also changes. Changes on the client are also reflect on the server.

Tag values are displayed in I/O fields on both computers.



Requirements

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

Configuration Procedure

The following configurations are required to establish a WinCC - WinCC connection:

1. Configuring a WinCC Project on a WinCC OPC DA Server
2. Configuring a WinCC Project on a WinCC OPC DA Client

See also

How to Configure a WinCC Project on a WinCC OPC DA Server (Page 691)

Configuring the WinCC Project on the WinCC OPC DA Client (Page 691)

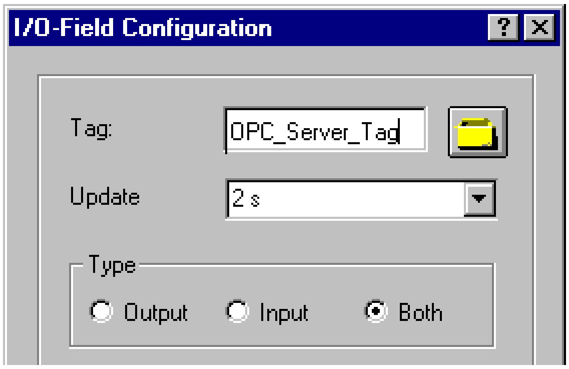
How to Configure a WinCC Project on a WinCC OPC DA Server

Introduction

In this section, a WinCC tag is created in the WinCC project of the WinCC OPC DA server and displayed in an I/O field.

Procedure

1. Select "New Tag" from the shortcut menu of the "Internal Tags" icon on the WinCC OPC DA server. Create a new tag called "OPC_Server_Tag" of the "signed 16-bit value" type.
2. Launch the Graphics Designer and open a new picture.
3. Add an I/O field to the picture. Select the "I/O field" object from the object list under "Smart Objects". The "I/O Field Configuration" dialog is opened.



Tag: OPC_Server_Tag
Update 2 s
● Both

4. Enter the name "OPC_Server_Tag" in the "Tag" field.
5. Set the update to "2s" and the field type to "I/O field".
6. Click "OK" to close the dialog and save the picture.
7. Enable the WinCC project by clicking the "Activate" button in the Graphics Designer.

See also

Configuring the WinCC Project on the WinCC OPC DA Client (Page 691)

Configuring the WinCC Project on the WinCC OPC DA Client

Introduction

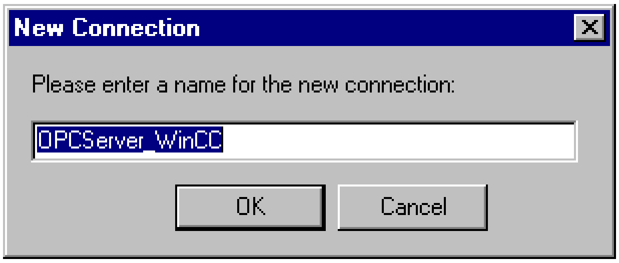
In this section, a WinCC tag is created on the WinCC OPC DA client, in order to read a WinCC tag on the WinCC OPC DA server. The tag value is displayed in an I/O field.

Requirements

- Add the "OPC" channel to the WinCC project of the WinCC OPC DA client.
- Configure an internal tag named "OPC_Server_Tag" of the data type "signed 16-bit value" in the WinCC project of the WinCC OPC DA server.
- Enable the WinCC project of the WinCC OPC DA server.

Procedure

1. In the shortcut menu of the channel unit "OPC Groups(OPCHN Unit#1)" on the WinCC OPC DA client, select "System Parameters". The OPC Item Manager is opened.
2. Choose the name of the computer to be used as the OPC DA server from the selection dialog. Select "OPCServer.WinCC" from the list. Click the "Browse Server" button. The "Filter Criteria" dialog is opened.
3. Click the "Next->" button in the "Filter Criteria" dialog. Select the "OPC_Server_Tag" tag in the "OPCServer.WinCC ..." dialog. Click the "Add Items" button.
4. If a connection to the OPC DA server already exists, continue with step 5.
   If no connection has been configured, a corresponding message is displayed.
   Click "Yes". The "New Connection" dialog is displayed.

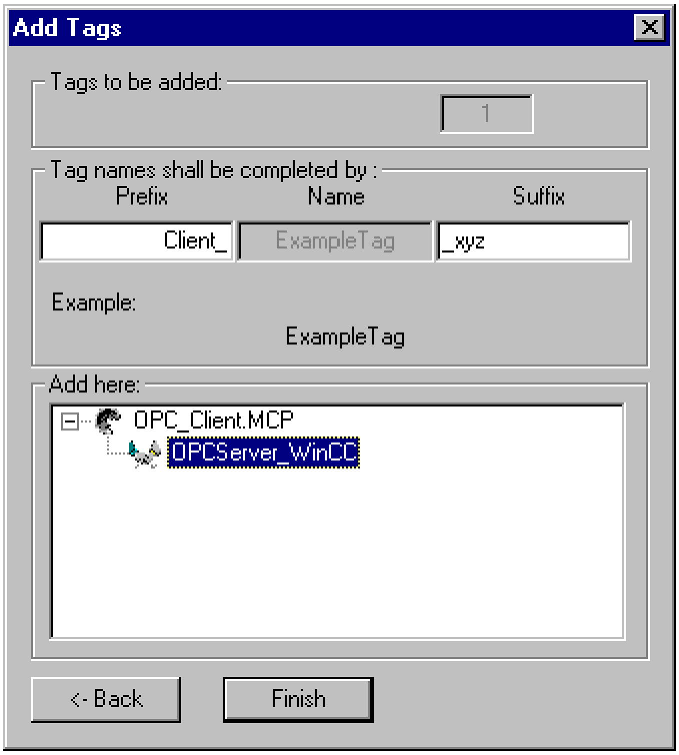

New Connection

Please enter a name for the new connection:

OPCServer_WinCC

Enter "OPCServer_WinCC" as the name of the connection. Click "OK".

5. The "Add Tags" dialog is displayed.
   Enter "Client_" in the prefix field and "_xyz" in the suffix field. Select connection "OPCServer_WinCC". Click "Finish".



Add Tags

Tags to be added: 1

Tag names shall be completed by :

Prefix	Name	Suffix
Client_	ExampleTag	_xyz

Example:
ExampleTag

Add here:
- OPC_Client.MCP
  - OPCServer_WinCC

6. Click the "<- Back" button in the "OPCServer.WinCC ..." dialog. In the "OPC Item Manager", click "Exit" to close the OPC Item Manager.
7. Launch the Graphics Designer and open a new picture. Add an I/O field to the picture. Select the "I/O field" object from the object list under "Smart Objects". The "I/O Field Configuration" dialog is opened.
8. Enter the name "Client_OPC_Server_Tag_xyz" in the "Tag" field. Set the update to "2 s". Set the field type to "I/O field". Close the dialog and save the picture. Enable the WinCC project by clicking the "Activate" button in the Graphics Designer.
9. The value of the configured tags is displayed in the I/O field on both the WinCC OPC DA server and the client. Enter a new value in the I/O field on the WinCC OPC DA server. The new value is displayed in the I/O field on the WinCC OPC DA client.

See also

How to Configure a WinCC Project on a WinCC OPC DA Server (Page 691)

Configuring the OPC Channel on the WinCC OPC DA Client (Page 276)

9.6.4.2 WinCC - SIMATIC NET FMS OPC Server Connection

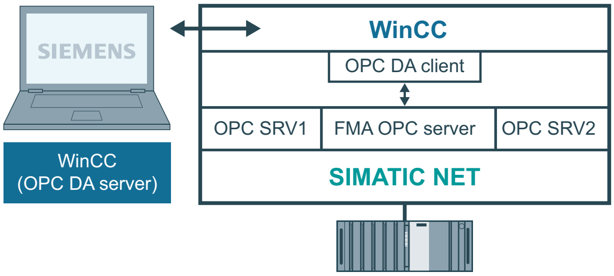
Example of WinCC - SIMATIC NET FMS OPC Server Connection

Introduction

During the installation of SIMATIC NET, you can select the OPC server to be installed. In the following example, a connection between WinCC and SIMATIC NET FMS OPC server is configured. Data from the automation device is made available to WinCC through the SIMATIC NET FMS OPC server.

In this example, WinCC is used as the WinCC OPC DA client. The OPC Item Manager displays the indexes of the object list configured for the automation device.

The current value of the tag is displayed in an I/O field. As soon as the value of the tags on the SIMATIC NET FMS OPC server changes, the new value is reflected on the process picture on the WinCC OPC DA client. Conversely, a value entered in the I/O field is sent to the automation device.



Requirements

- A computer with WinCC, SIMATIC NET software.
- A configured SIMATIC NET FMS OPC server. For additional information regarding the setup of SIMATIC NET S7 OPC servers refer to the SIMATIC NET documentation.

Configuration steps

The following configuration is required in the WinCC project of the WinCC OPC DA client:

1. Configuring a WinCC - SIMATIC NET FMS OPC server connection

Communication Manual

The communication manual contains additional information and extensive examples for the channel configuration. This manual is available for download on the Internet:

- http://support.automation.siemens.com/

Search by item number:

- A5E00391327

How to Configure the WinCC - SIMATIC NET FMS OPC Server Connection

Introduction

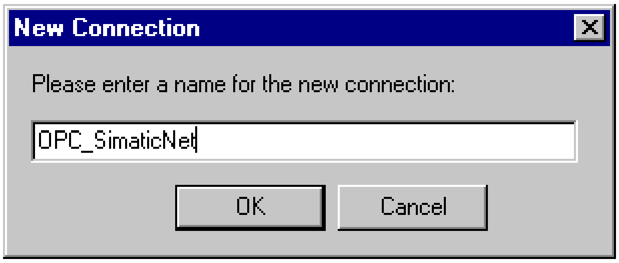
In this section, a WinCC tag that accesses an FMS index is configured in the WinCC project of the WinCC OPC DA client. The tag value is displayed in an I/O field.

Requirement

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

Procedure

1. In the shortcut menu of the channel unit "OPC Groups(OPCHN Unit#1)" on the WinCC OPC DA client, select "System Parameters". The OPC Item Manager is opened.
2. Choose the name of the computer to be used as the OPC DA server from the selection dialog. Select "OPC.SIMATICNet" from the list.
   Click the "Browse Server" button. The "Filter Criteria" dialog is opened.
3. Click the "Next->" button in the "Filter Criteria" dialog. The "OPC.SIMATICNet.." dialog is opened. All FMS indexes configured are displayed in a selection list. Select an index. Click the "Add Items" button.
4. If a connection to the SIMATIC NET FMS OPC server already exists, continue with step 5.
   If no connection has been configured, a corresponding message is displayed.
   Click "Yes". The "New Connection" dialog is displayed.

   
   New Connection
   Please enter a name for the new connection:
   OPC_SimaticNet

   Enter "OPC_SlimaticNET" as the name of the connection. Click "OK".
5. The "Add Tags" dialog is opened.
   Enter "Client_" in the prefix field and "_xyz" in the suffix field. Select the connection "OPC_SimaticNET". Click "Finish".
6. Click the "<- Back" button in the "OPC.SIMATICNet .." dialog. In the "OPC Item Manager", click "Exit" to close the OPC Item Manager.
7. Launch the Graphics Designer and open a new picture. Add an I/O field to the picture. Select the "I/O field" object from the object list under "Smart Objects". The "I/O Field Configuration" dialog is opened.
8. Enter the name of the tags in the "Tag" field. Set the update to "2s". Set the field type to "I/O field".

9. Click "OK" to close the dialog and save the picture. Enable the WinCC project by clicking the "Activate" button in the Graphics Designer.
10. The current value of the FMS index is shown in the I/O field. The value is updated every two seconds. Enter a value in the I/O field. The changed value is passed to the automation device.

See also

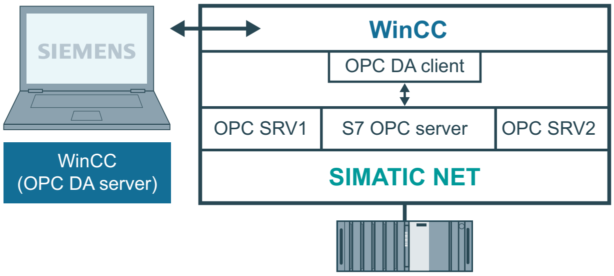
Configuring the OPC Channel on the WinCC OPC DA Client (Page 276)

9.6.4.3 WinCC - SIMATIC NET S7-OPC Server Connection

Example of a WinCC - SIMATIC NET S7 OPC Server Connection

During the installation of SIMATIC NET, you can select the OPC server to be installed. In the following example, a WinCC - SIMATIC NET S7 OPC server is configured. Data from the automation device is made available to the WinCC client via the SIMATIC NET S7 OPC server.

The current value of the tag is displayed in an I/O field on the WinCC OPC client. As soon as the value of the tags on the SIMATIC NET S7 OPC server changes, the changed value is shown on the process picture. Conversely, a value entered in the I/O field is sent to the automation device.



Requirements

- A computer with WinCC, SIMATIC NET software.
- A configured SIMATIC NET S7 OPC Server. For additional information regarding the setup of SIMATIC NET S7 OPC servers refer to the SIMATIC NET documentation.

Configuration steps

The following configurations are required to establish a WinCC - SIMATIC NET S7 OPC server connection:

1. Adding Tags to a SIMATIC NET S7 OPC Server
2. Configuring Access to the Tags on a SIMATIC NET S7 OPC Server

Communication Manual

The communication manual contains additional information and extensive examples for the channel configuration. This manual is available for download on the Internet:

- http://support.automation.siemens.com/

Search by item number:

- A5E00391327

Adding Tags to the SIMATIC NET S7 OPC Server

Introduction

In order for the OPC Item Manager to display the tags, they must be added to the address space of the SIMATIC NET S7 OPC server. The "OPC Scout" program is used for the configuration. OPC Scout is set up by the SIMATIC NET installer. For this example, the marker word "0" in the automation device is addressed.

Table of Parameters Used

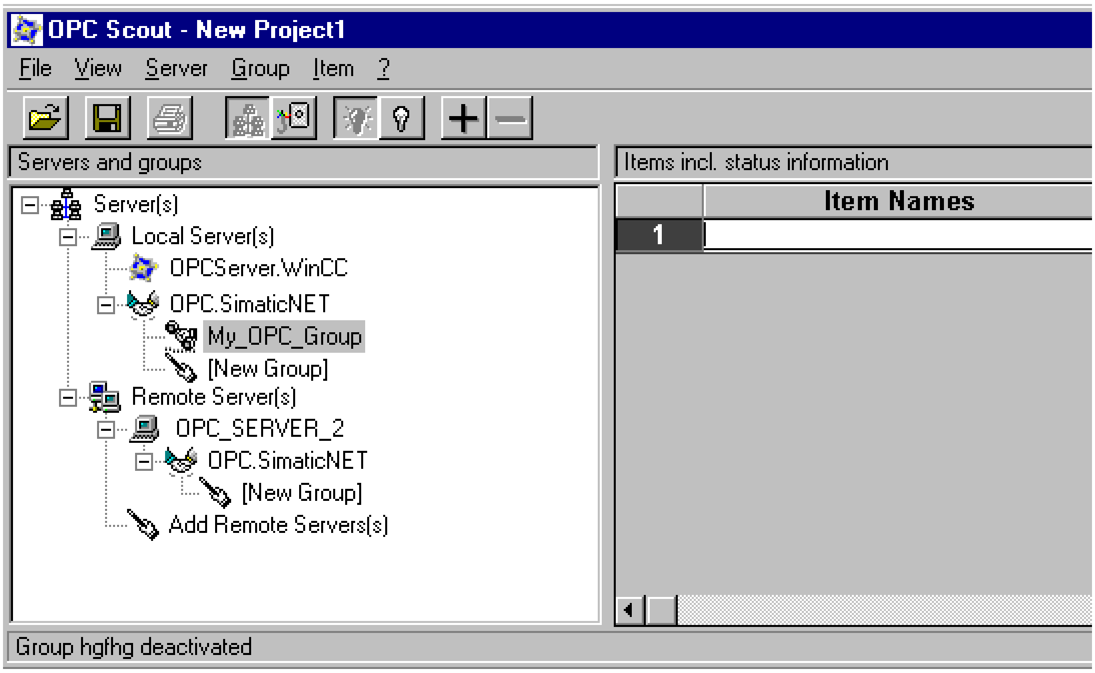
Parameter	Value
Data type	W
Range byte	0
No. values	1
Item alias	MW0

Requirements

- Configure an S7 connection in the SIMATIC NET software. For more information, refer to the SIMATIC NET documentation.

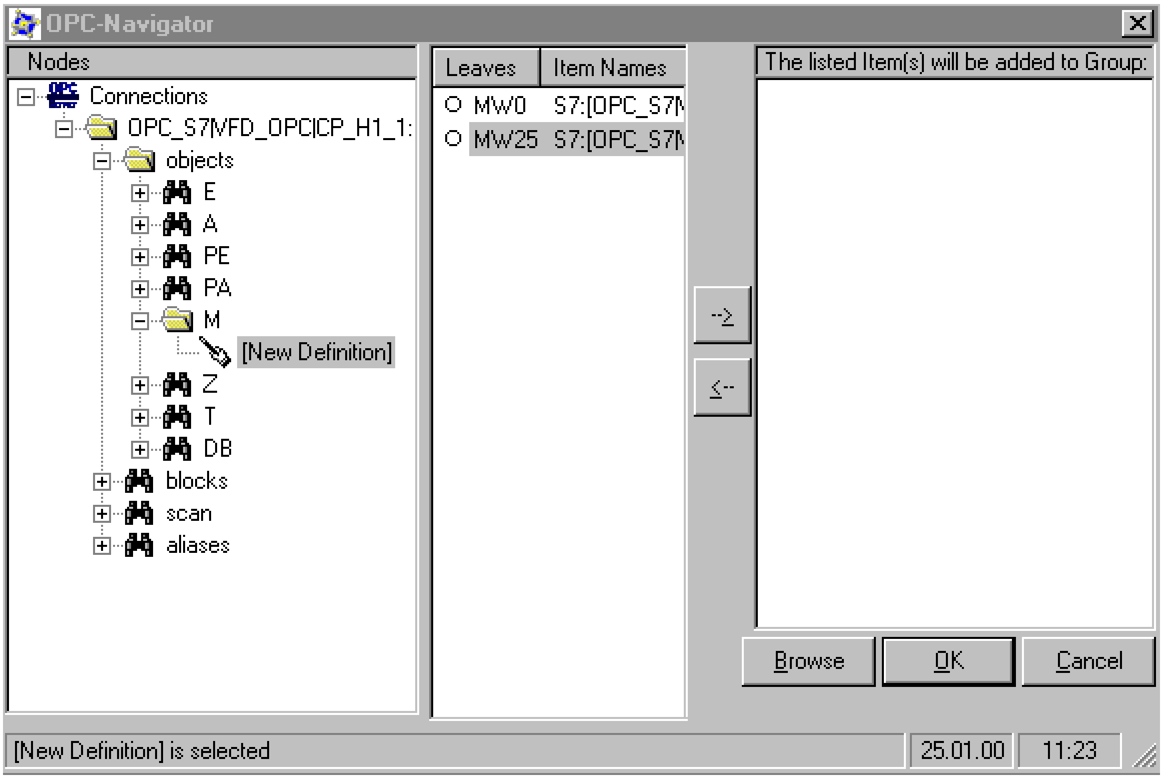
Procedure

1. Open the "OPC Scout" via Start → "Programs" → "SimaticNet" → "OPCServer" → "OPCScout" .



2. Select "OPC.SimaticNet" under "Local Server(s)". If the SIMATIC S7 OPC server is not run on the same computer, select "Add Remote Server(s)" in the "Server(s)" shortcut menu. Enter the name of the computer used as the OPC server in the "Add Remote Server(s)" dialog, then click "OK" to close the dialog.
3. Select "Connect" in the "OPC.SimaticNet" shortcut menu. The "Add Group" dialog is displayed. Enter a name for the group. Click "OK" to close the dialog.

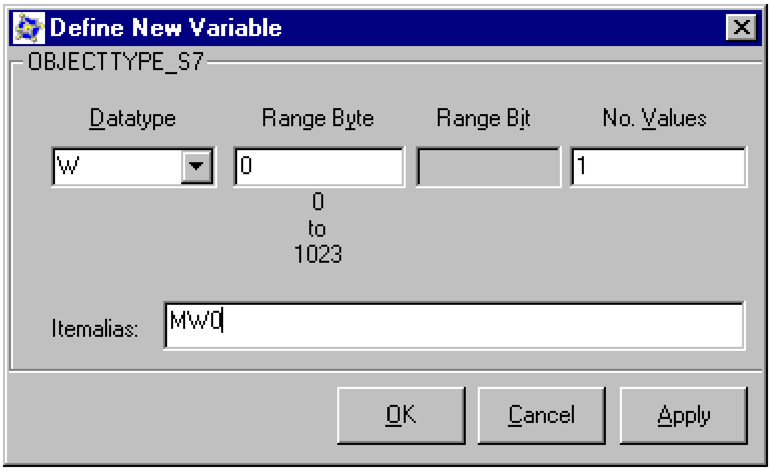
4. Select "Add Item" from the shortcut menu of the added group. The "OPC Navigator" is opened.



[New Definition] is selected | 25.01.00 | 11:23

5. Select "M" (marker) under "Objects" in the "OPC Navigator". Double-click "(New Definition)" to open the "Define New Tag" dialog.

6. Enter the parameters from the table in the "Define New Tag" dialog.



Datatype: W
Range Byte: 0
No. Values: 1
Itemalias: MW0

Click "OK" to close the "Define New Tag" dialog.

7. Mark the tag "MW0" in the "Leaves" area of the OPC Navigator. Click the "--> " button. Click "OK" in the OPC Navigator.

See also

Configuring Access to the Tags of the SIMATIC NET S7 OPC Server (Page 700)

Configuring Access to the Tags of the SIMATIC NET S7 OPC Server

Introduction

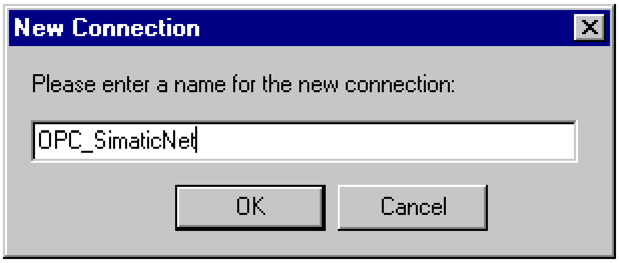
In this section, a WinCC tag is configured in the WinCC project of the WinCC OPC DA client. This tag accesses the tag "MW0" in the address space of the SIMATIC NET S7 OPC server. The tag value is displayed in an I/O field.

Requirements

- Create the tag "MW0" using the OPC Scout.
- Add the "OPC" channel to the WinCC project of the WinCC OPC DA client.

Procedure

1. Select "System Parameters" in the shortcut menu of "OPC Groups(OPCHN Unit#1)". The OPC Item Manager is opened.
2. Choose the name of the computer to be used as the OPC server from the selection dialog. Select "OPC.SIMATICNet" from the list.
   Click the "Browse Server" button. The "Filter Criteria" dialog is opened.
3. Click the "Next->" button in the "Filter Criteria" dialog. The "OPC.SIMATICNet.." dialog is opened. Select the "MW0" tag. Click the "Add Items" button.
4. If a connection to the SIMATIC NET FMS OPC server already exists, continue with step 5.
   If no connection has been configured, a corresponding message is displayed.
   Click "Yes". The "New Connection" dialog is displayed.

   
   New Connection

   Please enter a name for the new connection:

   OPC_SimaticNet

   Enter "OPC_SlimaticNET" as the name of the connection. Click "OK".
5. The "Add Tags" dialog is opened.
   Enter "Client_" in the prefix field and "_xyz" in the suffix field. Select the connection "OPC_SimaticNET". Click "Finish".
6. Click the "<- Back" button in the "OPC.SIMATICNet .." dialog. In the "OPC Item Manager", click "Exit" to close the OPC Item Manager.

7. Start Graphics Designer and open a picture. Add an I/O field to the picture. Select the "I/O field" object from the object list under "Smart Objects". The "I/O Field Configuration" dialog is opened.
8. Enter the name "Client_MW0_xyz" in the "Tag" field. Set the update to "2s". Set the field type to "I/O field".
9. Close the dialog and save the picture. Enable the WinCC project by clicking the "Activate" button in the Graphics Designer.
10. The I/O field on the WinCC OPC DA client displays the current value of the S7 tags. The value is updated every two seconds. Enter a value in the I/O field. The changed value is passed to the automation device.

See also

Adding Tags to the SIMATIC NET S7 OPC Server (Page 697)

Configuring the OPC Channel on the WinCC OPC DA Client (Page 276)

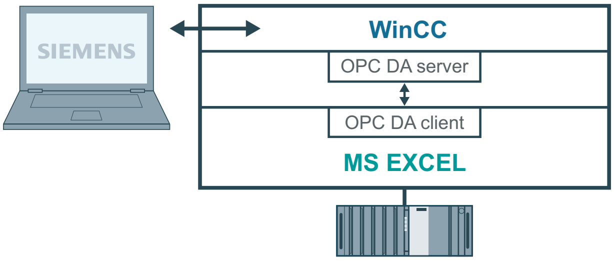
9.6.4.4 WinCC - Microsoft Excel Connection

Example of the WinCC - Microsoft Excel Connection

Introduction

In this example, an OPC DA client is created in Microsoft Excel using the Visual Basic Editor. The OPC DA client reads a WinCC tag in the WinCC project of the WinCC OPC DA server and writes the value into a cell. If a new value is entered in the cell, the value is passed to the WinCC OPC DA server.

A computer on which both WinCC and Microsoft Excel are installed is used for the connection.



Configuration steps

The following configurations must be made in Microsoft Excel:

1. Creating an OPC DA client in Visual Basic Editor of Microsoft Excel
2. Configuring access to a WinCC tag in Microsoft Excel

See also

How to Configure the Access to a WinCC Tag in Microsoft Excel (Page 704)

Creating an OPC DA Client in Microsoft Excel (Page 702)

Creating an OPC DA Client in Microsoft Excel

Introduction

To use Microsoft Excel as an OPC DA client, a special script must be created in the Visual Basic Editor of Microsoft Excel.

Requirements

Basic knowledge of Visual Basic Editor in Microsoft Excel.

Procedure

1. Open Microsoft Excel with a new workbook.
2. In the "Tools" menu of the Visual Basic Editor, → click "Macro". The Visual Basic Editor for Microsoft Excel is opened.
3. In the "Tools" menu of the Visual Basic Editor, select "References...". The "References - VBAProject" dialog is displayed. Locate entry "Siemens OPC DAAutomation 2.0" in the list of available references. Select the corresponding check box. Click "OK".
4. Copy the script shown below. This script is only available in the online help.
5. Open a new code window by double-clicking "Sheet1" in the project window of the Visual Basic Editor.
6. Paste the script into the code window.
7. Select "Save" from the "File" menu. Select "Close and Return to Microsoft Excel" from the "File" menu.

Example Script

```
Option Explicit
Option Base 1

Const ServerName = "OPCServer.WinCC"

Dim WithEvents MyOPCServer As OpcServer
Dim WithEvents MyOPCGroup As OPCGroup
Dim MyOPCGroupColl As OPCGroups
Dim MyOPCItemColl As OPCItems
Dim MyOPCItems As OPCItems
Dim MyOPCItem As OPCItem

Dim ClientHandles(1) As Long
Dim ServerHandles() As Long
Dim Values(1) As Variant
Dim Errors() As Long
Dim ItemIDs(1) As String
Dim GroupName As String
Dim NodeName As String

'---------------------------------------------------------------------
' Sub StartClient()
' Purpose: Connect to OPC_server, create group and add item
'---------------------------------------------------------------------
Sub StartClient()
  ' On Error GoTo ErrorHandler
  '----------- We freely can choose a ClientHandle and GroupName
  ClientHandles(1) = 1
  GroupName = "MyGroup"
  '----------- Get the ItemID from cell "A1"
  NodeName = Range("A1").Value
  ItemIDs(1) = Range("A2").Value
  '----------- Get an instance of the OPC-Server
  Set MyOPCServer = New OpcServer
  MyOPCServer.Connect ServerName, NodeName

  Set MyOPCGroupColl = MyOPCServer.OPCGroups
  '----------- Set the default active state for adding groups
  MyOPCGroupColl.DefaultGroupIsActive = True
  '----------- Add our group to the Collection
  Set MyOPCGroup = MyOPCGroupColl.Add(GroupName)

  Set MyOPCItemColl = MyOPCGroup.OPCItems
  '----------- Add one item, ServerHandles are returned
  MyOPCItemColl.AddItems 1, ItemIDs, ClientHandles, ServerHandles, Errors
  '----------- A group that is subscribed receives asynchronous notifications
  MyOPCGroup.IsSubscribed = True
  Exit Sub

ErrorHandler:
  MsgBox "Error: " & Err.Description, vbCritical, "ERROR"
End Sub

'---------------------------------------------------------------------
' Sub StopClient()
' Purpose: Release the objects and disconnect from the server
'----------------------------------------------------------------------
Sub StopClient()
  '----------- Release the Group and Server objects
  MyOPCGroupColl.RemoveAll
  '----------- Disconnect from the server and clean up
  MyOPCServer.Disconnect
  Set MyOPCItemColl = Nothing
  Set MyOPCGroup = Nothing
  Set MyOPCGroupColl = Nothing
  Set MyOPCServer = Nothing
End Sub

'----------------------------------------------------------------------
' Sub MyOPCGroup_DataChange()
' Purpose: This event is fired when a value, quality or timestamp in our Group has changed
'----------------------------------------------------------------------
'----------- If OPC-DA Automation 2.1 is installed, use:
Private Sub MyOPCGroup_DataChange(ByVal TransactionID As Long, ByVal NumItems As Long,
ClientHandles() As Long, ItemValues() As Variant, Qualities() As Long, TimeStamps() As
Date)
 '----------- Set the spreadsheet cell values to the values read
  Range("B2").Value = CStr(ItemValues(1))
  Range("C2").Value = Hex(Qualities(1))
  Range("D2").Value = CStr(TimeStamps(1))
End Sub

'----------------------------------------------------------------------
' Sub worksheet_change()
' Purpose: This event is fired when our worksheet changes, so we can write a new value
'----------------------------------------------------------------------
Private Sub worksheet_change(ByVal Selection As Range)
  '----------- Only if cell "B3" changes, write this value
  If Selection <> Range("B3") Then Exit Sub
  Values(1) = Selection.Cells.Value
  '----------- Write the new value in synchronous mode
  MyOPCGroup.SyncWrite 1, ServerHandles, Values, Errors
End Sub
```

See also

How to Configure a WinCC Project on a WinCC OPC DA Server (Page 691)

How to Configure the Access to a WinCC Tag in Microsoft Excel

Introduction

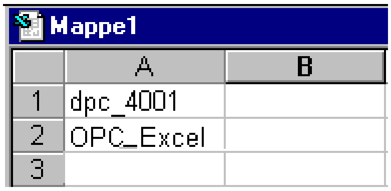
The Excel OPC DA client reads a WinCC tag of the WinCC OPC DA server and writes the value of the tag into a cell. In the WinCC project of the WinCC OPC DA server, the value of the tag is displayed in an I/O field. If the tag value in a cell is changed, this alters the value in the I/O field of the WinCC OPC DA server.

Requirements

- Configure an internal tag named "OPC_Excel" with data type "signed 16-bit value" in the WinCC project of the WinCC OPC DA server.
- Write the value of the "OPC_Excel" tag to an I/O field on the WinCC project of the WinCC OPC DA server.
- Enable the WinCC project of the WinCC OPC DA server.

Procedure

1. In Microsoft Excel, enter the name of the computer used as the OPC server in cell A1. In cell A2, enter the tag name "OPC_Excel".



Mappe1

	A	B
1	dpc_4001	
2	OPC_Excel	
3		

2. In the "Tools" menu in Excel, select "Macro" → "Macros". The "Macro" dialog is opened. Select the entry "Sheet1.StartClient" from the list of macros. Click "Run" to start the OPC client.
3. The value of the tag is written into cell B2, the quality code into C2 and the timestamp into D2.
4. Enter a new value in cell B3. The changed value is displayed in the I/O field on the WinCC OPC server.
5. In the "Tools" menu in Excel, select "Macro" → "Macros". The "Macro" dialog is opened. Select the entry "Sheet1.StopClient" from the list of macros. Click "Run" to stop the OPC client.

9.7 WinCC OPC HDA server

9.7.1 Functionality of the WinCC OPC HDA server

Introduction

The WinCC OPC HDA server is a DCOM application makings data needed from the archive system available to the OPC HDA client. Access the data using Item Handles. Read or write access is enabled. The data can also be analyzed.

The WinCC OPC HDA server supports the OPC Historical Data Access 1.20 specification. This has been confirmed by the compliance test.

The following chapter explains the design of the data structure, as well as the attributes, aggregates and functions supported by the WinCC OPC HDA server. This is not a detailed description, but rather a summary of the most important information. For more information, refer to the "OPC Historical Data Access 1.20" specification.

Installation

The WinCC OPC HDA server can be selected during the installation of WinCC. It is possible to select whether access is made to the WinCC archive system with or without write function . After installation, the WinCC OPC DA server is immediately available for use without any additional configuration.

In the case of installation without write access, the data in the WinCC archive system can only be read and analyzed. In the case of write access, data in the WinCC archive system can be analyzed, added, deleted and updated.

The WinCC OPC HDA server can be implemented on a WinCC server or a WinCC client.

Licensing

In order to operate the WinCC OPC HDA server, the following licenses must be installed on each WinCC computer implemented as an OPC HDA server:

- A valid RT license for WinCC
- WinCC Option Connectivity Pack

Notes on configuration

If the WinCC OPC HDA server is used, the application "OPC DA server, OPC A&E server, OPC HDA server" must be activated.

You can activate the application in the Editor "Computer" of the WinCC Configuration Studio in the "Processes when starting WinCC Runtime" tab.

OPC HDA Client

All OPC HDA clients that conform to the OPC Historical Data Access 1.20 specification can access the WinCC OPC HDA server. You can also create the OPC HDA client yourself. By creating proprietary OPC HDA clients, most user-specific requirements can be met.

Examples of how an OPC HDA client can be used include:

- Analysis and evaluation of archived data
- Statistical process control of archives from different OPC HDA servers

To request for historical values using OPC HDA client, you need to take care of the following during configuration:

- Select a query cycle in such a way that the client can receive the requested data before the next query is sent. Too short cycles can lead to high time delays while receiving data.
- CPU load of the WinCC server depends on the number of tags per query.

Write access to cyclic archive with configured swapping out

In runtime, the data is modified in the cyclic archives on the WinCC server.

Changes are accepted into the swapped-out archive only when the data is changed almost immediately after being created.

If the concerned archive segment of the circulation archive has already been swapped out, then the change is not done subsequently in the swapped-out archive. Even the modified data is deleted when you delete the archive segment on the WinCC server.

See also

Quality codes (Page 712)

Data Structure of a WinCC OPC HDA Server (Page 707)

www.opcfoundation.org (http://www.opcfoundation.org)

9.7.2 Data Structure of a WinCC OPC HDA Server

9.7.2.1 Data Structure of a WinCC OPC HDA Server

Introduction

The data on the WinCC OPC HDA server are structured. The available data structures are listed below. This is not a detailed description, but rather a summary of the most important information. For more information, refer to the "OPC Historical Data Access 1.20" specification.

Data structure

	Description
Attributes	Provide additional quality characteristics for the raw data. Attributes include data type, specifications re. archiving, etc. For more information, see the overview of supported attributes.
Assemblies	Summarize raw data of a specified time interval. Aggregates include average value, minimum, maximum, etc. For more information, see overview of supported aggregates.
StartTime/ EndTime	Set the beginning and end point for the time interval.
Bounding values	Values recorded at the beginning and end. If no bounding values are available, the values closest to these times are used as bounding values.
Raw data	Data from the WinCC archive system of a particular time interval. These data include a time stamp and quality rating.
Item handle	Unique assignment to a WinCC archive tag.
ItemID	Unique identifier of the WinCC archive tag. The ItemID can be used to get an item handle.

See also

Overview of the supported functions (Page 710)

Time Format of a WinCC OPC HDA Server (Page 710)

Overview of the supported attributes (Page 708)

Overview of the supported assemblies (Page 709)

www.opcfoundation.org (http://www.opcfoundation.org)

9.7.2.2 Overview of the supported attributes

Introduction

The following table contains the attributes supported by the WinCC OPC HDA server.

For more information, refer to the "OPC Historical Data Access 1.20" specification.

Attributes

Attribute	Attribute ID	Description
ItemID	OPCHDA_ITEMID	Indicates the WinCC archive tag to be accessed.
Item data type	OPCHDA_DATA_TYPE	Indicates the data type of the WinCC archive tag.
Description	OPCHDA_DESCRIPTION	Returns a description of the WinCC archive tag. The description is defined in WinCC Tag Logging.
Engineering units	OPCHDA_ENG_UNITS	Sets the display of measurement units. The labeling is defined in WinCC Tag Logging.

See also

Data Structure of a WinCC OPC HDA Server (Page 707)

www.opcfoundation.org (http://www.opcfoundation.org)

9.7.2.3 Overview of the supported assemblies

Introduction

The following table lists the aggregates supported by the WinCC OPC HDA server. For more information, refer to the "OPC Historical Data Access 1.20" specification.

Assemblies

Assembly	Description
OPCHDA_COUNT	Returns the raw data count for the specified time interval.
OPCHDA_START	Returns the initial value of the raw data at the beginning of the time interval.
OPCHDA_END	Returns the final value of the raw data at the end of the time interval.
OPCHDA_AVERAGE	Returns the average value of the raw data for the specified time interval.
OPCHDA_TIMEAVERAGE	Returns the time-weighted average of the raw data for the specified time interval.
OPCHDA_TOTAL	Returns the sum total value for the specified time interval.
OPCHDA_STDEV	Returns the standard deviation of the raw data for the specified time interval.
OPCHDA_MINIMUMACTUALTIME	Returns the minimum value of the raw data and its time stamp for the specified time interval.
OPCHDA_MINIMUM	Returns the minimum value of the raw data for the specified interval.
OPCHDA_MAXIMUMACTUALTIME	Returns the maximum value of the raw data and its time stamp for the specified time interval.
OPCHDA_MAXIMUM	Returns the maximum value of the raw data for the specified interval.
OPCHDA_DELTA	Returns the difference between the first and last value in the raw data for the specified time interval.
OPCHDA_REGSLOPE	Returns the slope of the regression line of the raw data for the specified time interval.
OPCHDA_REGCONST	Returns the regression value of the raw data at the starting point.
OPCHDA_REGDEV	Returns the standard deviation of the regression of the raw data in the specified time interval.
OPCHDA_VARIANCE	Returns the variance of the raw data for the specified time interval.
OPCHDA_RANGE	Returns the difference between OPCHDA_MAXIMUM and OPCHDA_MINIMUM of the raw data for the specified time interval.
OPCHDA_DURATIONGOOD	Returns the period of time in which the quality of the raw data was good. The period is indicated in seconds.
OPCHDA_DURATIONBAD	Returns the period of time in which the quality of the raw data was bad. The period is indicated in seconds.

Assembly	Description
OPCHDA_PERCENTGOOD	Returns the percentage of the raw data of good quality.
OPCHDA_PERCENTBAD	Returns the percentage of the raw data of bad quality.
OPCHDA_WORSTQUALITY	Returns the worst quality of the raw data for the specified time interval.

See also

Data Structure of a WinCC OPC HDA Server (Page 707)

Functionality of the WinCC OPC HDA server (Page 706)

www.opcfoundation.org (http://www.opcfoundation.org)

9.7.2.4 Overview of the supported functions

Introduction

The following tables list the functions supported by the WinCC OPC HDA server. These functions can be used by the OPC HDA client for data exchange. For more information, refer to the "OPC Historical Data Access 1.20" specification.

Read

Function	Description
ReadRaw	Returns the raw data, its quality and time stamp for the specified time interval.
ReadProcessed	Returns the calculated value, the quality of the value and the time stamp for the specified time interval. The calculated value is determined by the selected aggregate.
ReadAtTime	Returns the raw data, its quality and time stamp for a particular time interval. If no value is available, the value for this point is interpolated.
ReadAttribute	Returns the item attributes and time stamp for the specified time interval.

See also

Functionality of the WinCC OPC HDA server (Page 706)

www.opcfoundation.org (http://www.opcfoundation.org)

9.7.2.5 Time Format of a WinCC OPC HDA Server

Introduction

The time interval is specified on the WinCC OPC HDA server by setting the starting and ending times. The specified time interval determines the observation period for the historical data. When specifying the times, certain formats must be maintained.

The following options are available for the specification of times:

- Absolute based on UTC
- Relative to the local time of the server

Absolute Value According to UTC

By default, the WinCC OPC HDA server uses the coordinated world time (UTC) as its time base. This time corresponds to the Greenwich Mean Time (Central European Time minus an hour).

Time format

YYYY/MM/DD hh:mm:ss.msmsms

Parameters

YYYY = year

MM = month

DD = day

hh = hours

mm = minutes

ss = seconds

ms = milliseconds

Input example

2002/06/10 09:27:30.000

Specification of Time Relative to Local Time

For this option, the time is entered relative to the local time of the server. The local time zone is set on the computer's "Date/Time" control panel.

Time format

keyword +/-offset1 +/-offset(n)

The offset is the deviation from the local time of the server.

Keywords

NOW = current local time on the server

SECOND = current second

MINUTE = current minute

HOUR = current hour

DAY = current day

WEEK = current week

MONTH = current month

YEAR = current year

Offset

+/-S = deviation in seconds

+/-M = deviation in minutes

+/-H = deviation in hours

+/-D = deviation in days

+/-W = deviation in weeks

+/-MO = deviation in months

+/-Y = deviation in years

Example:

DAY - 1D = previous day

DAY-1D + 7H30 = previous day at 7:30

MO-1D+5H = last day of the previous month at 5:00.

NOW-1H15M = one hour and 15 minutes ago

YEAR+3MO= April of this year

See also

Functionality of the WinCC OPC HDA server (Page 706)

www.opcfoundation.org (http://www.opcfoundation.org)

9.7.3 Quality codes

Introduction

Quality codes are used to evaluate the status and quality of the raw data. The quality codes for OPC are described under "6.8 OPC Quality flags" of the "Data Access Custom Interface Standard Version 3.00" specifications.

Quality Codes of the WinCC OPC HDA Server

Code	OPC	Description	Quality
0x00040000	OPCHDA_RAW	Indicates the quality of raw data transmission.	GOOD BAD UNCERTAIN
0x00080000	OPCHDA_CALCULATED	Indicates the quality of calculated data transmission.	GOOD BAD UNCERTAIN
0x00100000	OPCHDA_NOBOUND	No bounding values were found at the starting or ending point.	BAD

Code	OPC	Description	Quality
0x00200000	OPCHDA_NODATA	No raw data were found for the specified time interval.	BAD
0x00400000	OPCHDA_DATALOST	The raw data in the selected interval were not completely archived.	BAD

See also

www.opcfoundation.org (http://www.opcfoundation.org)

9.7.4 Supported Write-Accesses

Introduction

The following table shows the write accesses supported by the WinCC OPC HDA server.

Table element:

	Description
Cyclic archive	The process values to be archived are stored in a cyclic archive. The cyclic archive consists of a configurable number of data buffers. The size and a period of time (e.g. in days) for the data buffer are defined. If all data buffers are full, the process data in the first data buffer is overwritten.
Cyclic archive after swapping	In order to protect process data in the data buffers from being overwritten process, it can be swapped (exported).
☺	Supported by WinCC.
☹	Not supported by WinCC.

Write Accesses

Adding process values later

Cyclic archive	Cyclic archive after swapping	Supported by WinCC	Description
Yes	No	☺	When the time period is contained in the cyclic archive, a process value can be added later.
Yes	Yes	☹	The data buffer of the corresponding time period is swapped to an archive backup. Process values cannot be added later to an archive backup.
No	No	☹	The cyclic archive is not available. The process value cannot be stored.
No	Yes	☹	The cyclic archive is not available. The process value cannot be stored.

Adding process values in Runtime

Cyclic archive	Cyclic archive after swapping	Supported by WinCC	Description
Yes	No	☺	The process value is added in the data buffer currently valid for the cyclic archive.

Inserting future process values

Cyclic archive	Cyclic archive after swapping	Supported by WinCC	Description
YES	No	☹	During write access, no values can be added in the future.
No	No	☹	With write access, no values can be added in the future.

Deleting process values

Cyclic archive	Cyclic archive after swapping	Supported by WinCC	Description
Yes	No	☺	When the time period is contained in the cyclic archive, a process value can be deleted.
Yes	Yes	☹	The data buffer of the corresponding time period is swapped to an archive backup. Process values can be deleted from an archive backup.
No	No	☹	The cyclic archive is not available. The process value cannot be stored.
No	Yes	☹	The cyclic archive is not available. The process value cannot be stored.

Editing process values

Cyclic archive	Cyclic archive after swapping	Supported by WinCC	Description
Yes	No	☺	When the time period is contained in the cyclic archive, a process value can be edited.
Yes	Yes	☹	The data buffer of the corresponding time period is swapped to an archive backup. Process values cannot be edited in an archive backup.
No	No	☹	The cyclic archive is not available. The process value cannot be stored.
No	Yes	☹	The cyclic archive is not available. The process value cannot be stored.

9.7.5 Example of an OPC HDA Connection

9.7.5.1 Example of an OPC HDA Connection

Introduction

In the example below, a connection between WinCC and the OPC HDA client is configured. Data from the WinCC archive system are made available via the WinCC OPC HDA server. The OPC HDA client accesses the data via item handles. To simplify the configuration process, the OPC HDA browser is used.

The OPC HDA client from the OPC Foundation is used. All OPC HDA clients conforming to the OPC Historical Data Access 1.20 specification can access the WinCC OPC HDA server.

Requirements

- Create an internal tag named "OPC_HDA" with data type "unsigned 16-bit value" in the WinCC project of the WinCC OPC HDA server.
- Create a process value archive called "HDA_ProcessValueArchive" in the WinCC archive system.
- Create an WinCC archive tag called "OPC_HDA_Tag" in the "HDA_ProcessValueArchive" process value archive. Link the WinCC archive tag to the internal tag "OPC_HDA".
- In the Runtime list, launch Tag Logging Runtime and disable Graphics Runtime.
- Launch the WinCC project of the WinCC OPC HDA server.

Configuration steps

The following configurations are required to connect WinCC to the OPC HDA client:

1. Configuring access to a WinCC archive tag using the HDA server browser
2. Reading values from the WinCC archive tags

See also

How to Configure Access to a WinCC Archive Tag Using the HDA Server Browser (Page 717)

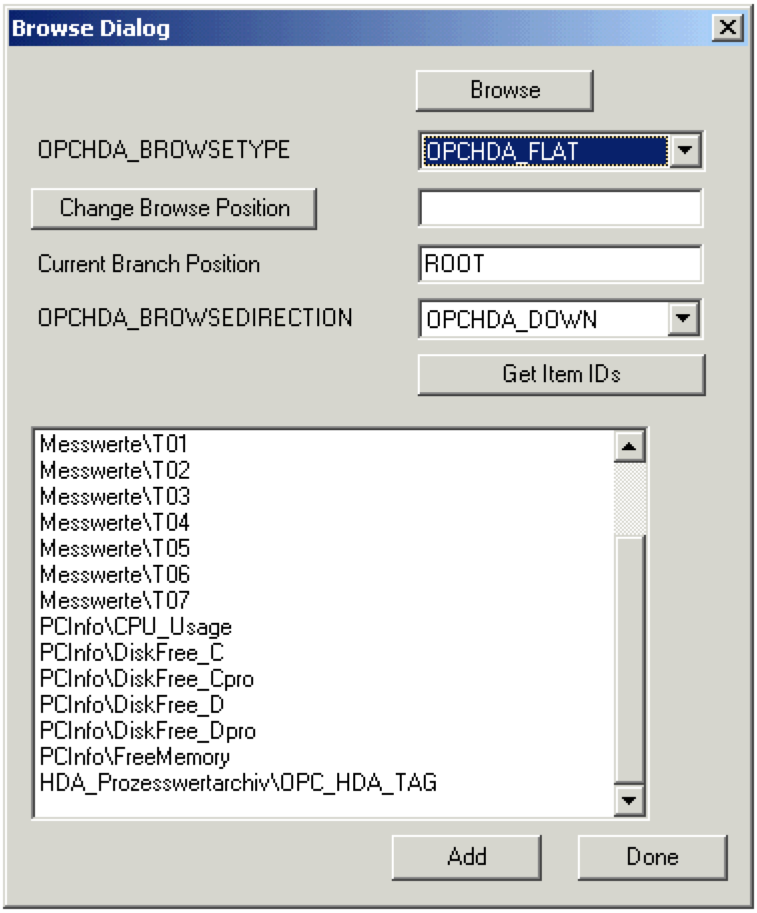
HDA server browser (Page 716)

Reading Values of WinCC Archive Tags (Page 718)

9.7.5.2 HDA server browser

Introduction

The OPC HDA client accesses the tag values via item handles. For ease of configuration, the WinCC OPC HDA server supports the browser functionality. The OPC HDA client can use the HDA server browser to search the address space of the WinCC OPC HDA server. The data are listed hierarchically by process value archive.



OPCHDA_BROWSETYPE: OPCHDA_FLAT

Current Branch Position: ROOT

OPCHDA_BROWSEDIRECTION: OPCHDA_DOWN

```
Messwerte\T01
Messwerte\T02
Messwerte\T03
Messwerte\T04
Messwerte\T05
Messwerte\T06
Messwerte\T07
PCInfo\CPU_Usage
PCInfo\DiskFree_C
PCInfo\DiskFree_Cpro
PCInfo\DiskFree_D
PCInfo\DiskFree_Dpro
PCInfo\FreeMemory
HDA_Prozesswertarchiv\OPC_HDA_TAG
```

Note

Access to a WinCC archive tag without the HDA server browser requires manual configuration of the item ID.

When addressing WinCC archive tags, the computer name (server prefix) is included in the path. The ItemID has the following syntax: Server-prefix::process_value_archive\WinCC_archive_tag.

See also

How to Configure Access to a WinCC Archive Tag Using the HDA Server Browser (Page 717)

www.opcfoundation.org (http://www.opcfoundation.org)

9.7.5.3 How to Configure Access to a WinCC Archive Tag Using the HDA Server Browser

Introduction

In this section, the OPC HDA client is used to access a WinCC archive tag.

The OPC HDA client from the OPC Foundation is used.

The HDA server browser is used to configure access.

Note

Demo client

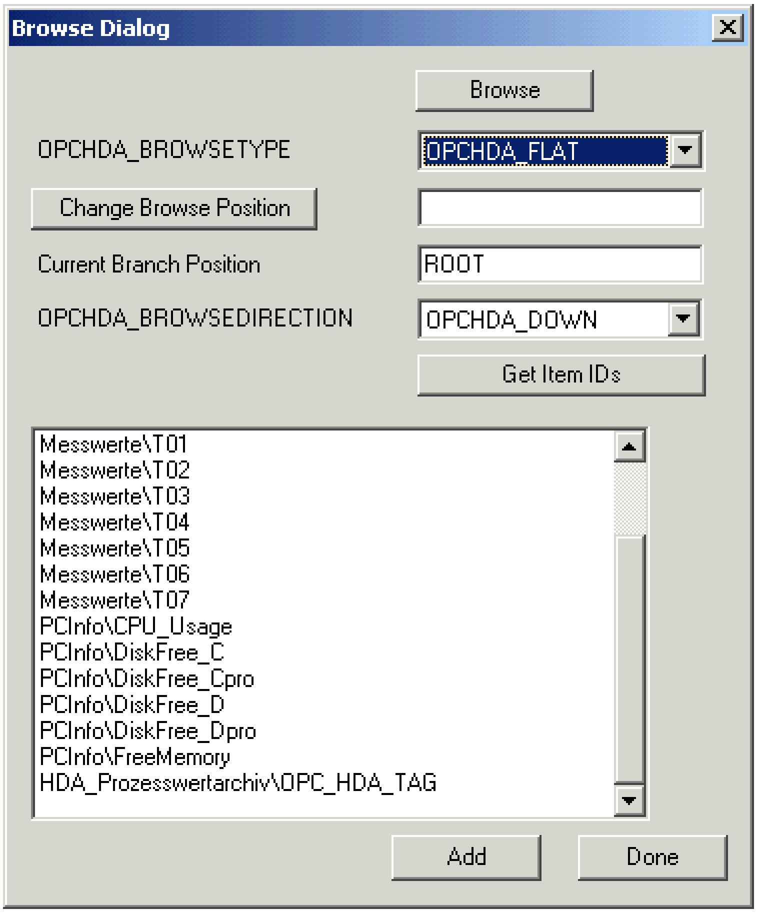
The OPC HDA client described here is the demo client from the OPC Foundation.

You can find the sources for this on the Internet at http://www.opcfoundation.org.

Procedure

1. Copy the "SampleClientHDA.exe" file from the WinCC installation path "Siemens\WinCC\documents\English" to a folder of your choice.
2. Double-click the "SampleClientHDA.exe" file.
   The "HDA client" program opens.
3. In the "Server Name" area, select entry "OPCServerHDA.WinCC.1".
4. Click "Connect".
   Confirm the next dialog.

5. Click "Browse" in the HDA client.
   The "Browse Dialog" dialog opens.



OPCHDA_BROWSETYPE: OPCHDA_FLAT

Current Branch Position: ROOT

OPCHDA_BROWSEDIRECTION: OPCHDA_DOWN

```
Messwerte\T01
Messwerte\T02
Messwerte\T03
Messwerte\T04
Messwerte\T05
Messwerte\T06
Messwerte\T07
PCInfo\CPU_Usage
PCInfo\DiskFree_C
PCInfo\DiskFree_Cpro
PCInfo\DiskFree_D
PCInfo\DiskFree_Dpro
PCInfo\FreeMemory
HDA_Prozesswertarchiv\OPC_HDA_TAG
```

6. Select "OPCHDA_FLAT" in the "OPCHDA_BROWSETYPE" field.
7. In the selection window, select entry "HDA_ProcessValueArchive_HDA_TAG".
8. Click "Add" and then "Done" to close the dialog.

For more information, refer to http://www.opcfoundation.org.

See also

Reading Values of WinCC Archive Tags (Page 718)

www.opcfoundation.org (http://www.opcfoundation.org)

9.7.5.4 Reading Values of WinCC Archive Tags

Introduction

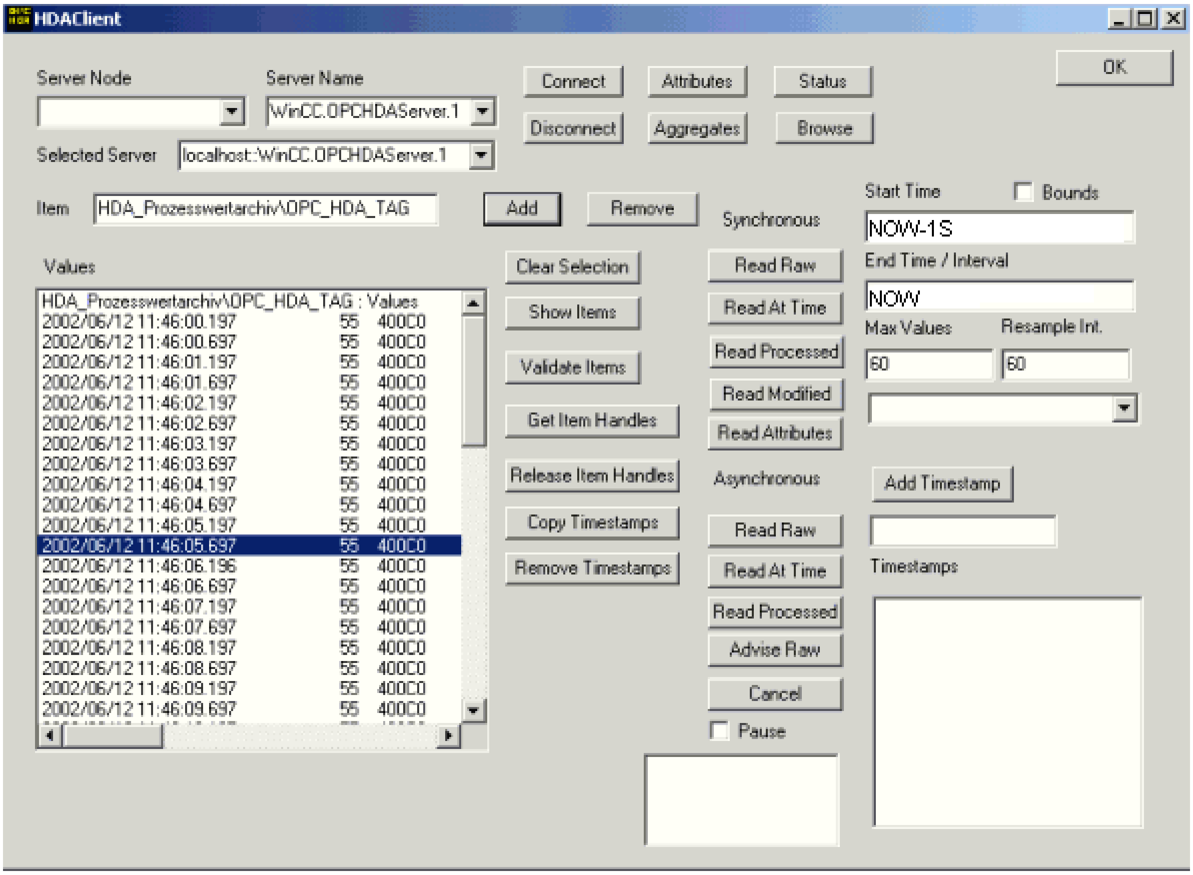
This section explains how you can access and read WinCC archive tags.

Requirement

- The OPC HDA client must be running.

Procedure

1. Click "Show Items" in the HDA client.
2. Click "Get Item Handles" in the HDA client.
3. Double-click "HDA_ProcessValueArchive_HDA_Tag" in the selection field "Value" selection field.
4. Enter "NOW-10S" in the "Start Time" field. Enter "NOW" in the "End Time" field.



Server Name: WinCC.OPCHDAServer.1

Selected Server: localhost::WinCC.OPCHDAServer.1

Item: HDA_Prozesswertarchiv\OPC_HDA_TAG

Start Time: NOW-1S

End Time / Interval: NOW

Max Values: 60

Resample Int.: 60

```
HDA_Prozesswertarchiv\OPC_HDA_TAG : Values
2002/06/12 11:46:00.197    55   400C0
2002/06/12 11:46:00.697    55   400C0
2002/06/12 11:46:01.197    55   400C0
2002/06/12 11:46:01.697    55   400C0
2002/06/12 11:46:02.197    55   400C0
2002/06/12 11:46:02.697    55   400C0
2002/06/12 11:46:03.197    55   400C0
2002/06/12 11:46:03.697    55   400C0
2002/06/12 11:46:04.197    55   400C0
2002/06/12 11:46:04.697    55   400C0
2002/06/12 11:46:05.197    55   400C0
2002/06/12 11:46:05.697    55   400C0
2002/06/12 11:46:06.196    55   400C0
2002/06/12 11:46:06.697    55   400C0
2002/06/12 11:46:07.197    55   400C0
2002/06/12 11:46:07.697    55   400C0
2002/06/12 11:46:08.197    55   400C0
2002/06/12 11:46:08.697    55   400C0
2002/06/12 11:46:09.197    55   400C0
2002/06/12 11:46:09.697    55   400C0
```

5. Click "Read Raw". The values, their quality codes and time stamps are shown in the "Values" selection field.

9.7.6 Special features of the OPC HDA server in WinCC for acyclic logging

Introduction

Tag logging is performed in WinCC cyclically or acyclically. The WinCC HDA OPC server works differently depending on the logging method for tags:

- For all cyclically logged values, the OPC HDA server operates in conformity to the HDA specification of the OPC foundation. The OPC aggregates are linearly interpolated.
- Acyclically logged tags are not included in the HDA specification of the OPC Foundation. The OPC aggregates are interpolated incrementally. Especially when a tag experiences no change for a long period of time, no data is available during a time period. The following should be taken into consideration to nevertheless obtain valid data.

Note

The OPC HDA server is not OPC-compliant for acyclically logged tags. The HDA specification of the OPC Foundation does not recognize acyclically logged tags and, therefore, no archive server can handle acyclically logged tags. The supported aggregates are calculated in conformity to the OPC HDA specification. No non-explicitly called functions are supported.

Note

If write access to process value archives is enabled, no future values may be added.

Configuration of acyclically logged tags

For the configuration of acyclically logged tags, the "Archive after segment change" setting needs to be enabled for the tags. This enters the most recent valid value in the the new log when a segment changes.

Supported aggregates of the WinCC OPC HDA server for acyclically logged tags

The OPC HDA server supports the following aggregates:

- OPCHDA_MINIMUM
- OPCHDA_MAXIMUM
- OPCHDA_AVERAGE
- OPCHDA_END
- OPCHDA_INTERPOLATIVE
- OPCHDA_TIMEAVERAGE
- OPCHDA_TOTAL
- OPCHDA_DURATIONGOOD
- OPCHDA_PERCENTGOOD

Supported functions of the WinCC OPC HDA server for acyclically logged tags

- ReadRaw with "boundings" only. ReadRaw for a tag must always be performed with "boundings", in order to find the last real stored value for an area without logged value change.
- ReadProcessed
- DeleteRaw
- DeleteAtTime
- Insert
- InsertReplace
- Replace

Calculating the aggregates for acyclically logged tags

Calculation of the aggregates is based on the extended "RawData" data record, which contains virtual data points for the calculation in addition to real stored values. The WinCC OPC HDA server prepares the contained "RawData" corresponding to the requirements of the "ReadProcessed". The virtual data points needed for the calculation are formed from the bordering real data points. The following significant points are included for the virtual data points:

- Value for the "StartTime"
- Value for the "EndTime"
- Value for interval limits

Example

The values for "00:59:00", "01:02:00" and "01:03:00" are stored for an acyclic tag logging tag. An OPC HDA client postulates with "ReadProcessed" an aggregate with the following parameters:

- StartTime = 01:00:00
- EndTime = 01:04:00
- Interval = 00:02:00

Note

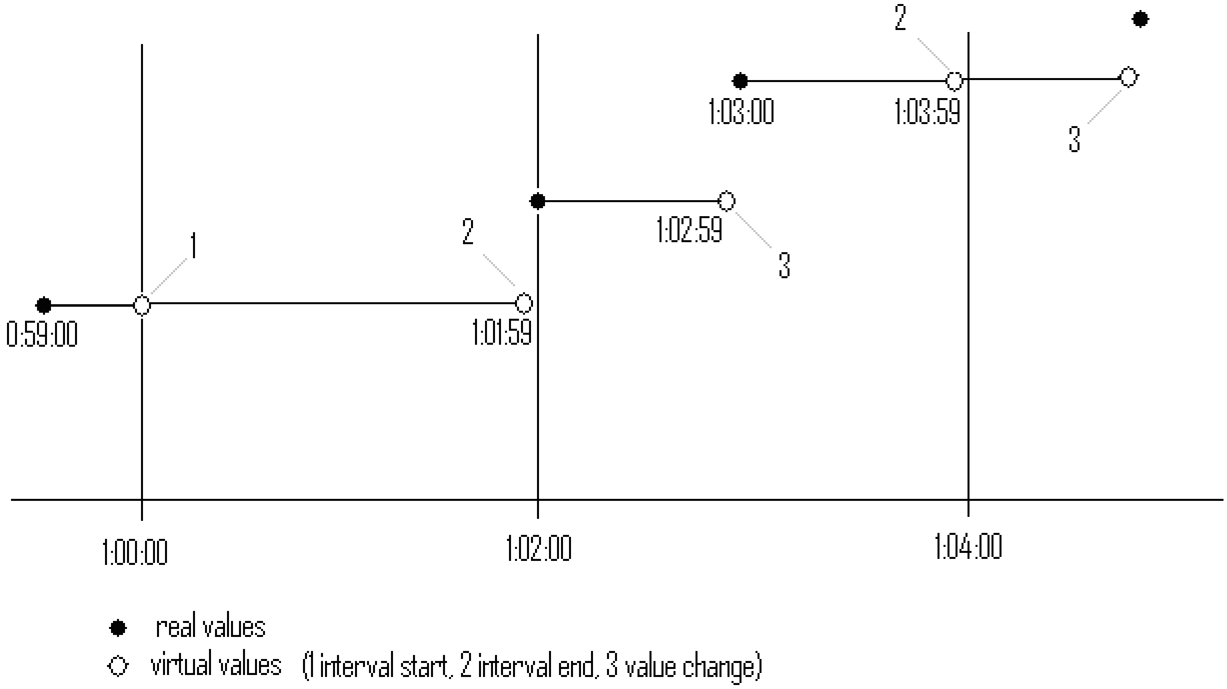
The time period is always 1 µs less than the time stamp at the limit for the calculation when generating virtual values at limits ("EndTime"/"Interval").

A delta of 1 seconds is used in the following table to provide a better overview. The following graphic illustrates the example.

The OPC server uses the following "RawData" for the calculation of the aggregate:

Number	Time stamp	Real stored values	Generated virtual values
1	00:59:00	1.00	
2	01:00:00		1.00
3	01:01:59		1.00

Number	Time stamp	Real stored values	Generated virtual values
4	01:02:00	2.00	
5	01:02:59		2.00
6	01:03:00	3.00	
7	01:03:59		3.00



9.8 WinCC OPC A&E Server

9.8.1 Functionality of the WinCC OPC A&E server

Introduction

The WinCC OPC A&E server is a DCOM application. The OPC A&E client is kept informed of status changes for WinCC messages by means of subscriptions. The OPC A&E client can apply a filter to the subscription. This filter determines which messages and attributes are displayed.

The WinCC OPC A&E server supports the specification OPC Alarm&Event 1.10. This has been confirmed by the compliance test.

The following chapter explains the display of the WinCC message system on OPC A&E, as well as the attributes supported by the WinCC OPC A&E server. This is not a detailed description, but rather a summary of the most important information. For more information, refer to the "OPC Alarms & Events 1.10" specification.

Installation

The WinCC OPC A&E server can be selected during the installation of WinCC. After installation, the WinCC OPC A&E server is immediately available for use without any additional configuration.

The WinCC OPC A&E server can be implemented on a WinCC server and a WinCC client.

Licensing

In order to operate the WinCC OPC A&E server, the following licenses must be installed on each WinCC server implemented as an OPC A&E server:

- A valid RT license for WinCC
- WinCC Option Connectivity Pack

Notes on configuration

If the WinCC OPC HDA server is used, the application "OPC DA server, OPC A&E server, OPC HDA server" must be activated.

You can activate the application in the Editor "Computer" of the WinCC Configuration Studio in the "Processes when starting WinCC Runtime" tab.

Server types

The WinCC OPC A&E server supports conditional events and simple events. In addition, there are tracking events.

Condition-related event server

With a condition-related event server, the event is associated with a condition. A condition might, for example, be a limit value violation of a tag. A message is generated in WinCC as soon as the bounding value is exceeded. This message is shown as an alarm in OPC A&E.

Simple event server

Simple events are messages that inform the OPC A&E client about events. Simple events include, for example, starting or exiting programs.

Note

Note the following when using redundant systems:

Simple events interconnected to internal tags are sent twice when tags are updated.

The first message is triggered by the master, the second by the standby.

Tracking event server

If a change in a process occurs, the OPC A&E client receives a message. Such a change might for example be a regulator adjustment.

OPC A&E client

All OPC A&E clients conforming to the OPC Alarms & Events 1.10 specification can access the WinCC OPC A&E server. You can also create the OPC A&E client yourself. By creating proprietary OPC clients, most user-specific requirements can be met. An OPC A&E client can, for example, be used for the analysis and common archiving of alarms from multiple OPC A&E servers.

See also

Quality Codes for OPC A&E (Page 730)

Mapping of the WinCC Message System on OPC A&amp;E (Page 725)

www.opcfoundation.org (http://www.opcfoundation.org)

9.8.2 Mapping of the WinCC Message System on OPC A&E

9.8.2.1 Mapping of the WinCC Message System on OPC A&amp;E

Introduction

During the configuration of the WinCC message system, settings are made to determine which process events generate a message. This message is shown as an alarm in OPC A&E. The table below lists the most important parameters of the alarm. It also describes how the information is made available by the WinCC message system. For more information, refer to "Alarm Structure".

Overview

OPC	WinCC message system
Source	Indicates the source of the message. The source has the format "<server prefix>::@LOCALMACHINE::".
Time	Issues a time stamp for received, sent and acknowledged messages. Issues a time stamp in UTC (Universal Time Coordinated).
Type	Indicates whether the event is a simple, tracking or condition-related event. WinCC - POC A&E server supports simple, condition-related and tracking events.
Severity	Indicates the priority of the WinCC message.
EventCategory	Returns the category of the message. For more information on this topic, refer to "Displaying Message Classes and Types".
Message	Indicates the message text of the corresponding message number.
ConditionName	Indicates the message number.
ChangeMask	Indicates the changed status of the message. For more information, refer to "Acknowledgement Theory".
NewState	Returns the message status. For more information, refer to "Acknowledgement Theory".
ConditionQuality	Returns the quality of the message. For more information, refer to "Quality Codes".
AckRequired	Indicates whether the message requires acknowledgement (receipt).
ActiveTime	Returns the time stamp for received messages.
EventAttribute	Lists the attributes required for the respective message. For more information, refer to "Attributes of the WinCC Message System".
Quality	Returns the quality code of the message.
Cookie	Returns the cookie from the OPC A&E server. The cookie corresponds to the message number in the WinCC alarm system

See also

Acknowledgement theory (Page 728)

Attributes of the WinCC Message System (Page 727)

Mapping the WinCC message classes and message types (Page 726)

9.8.2.2 Mapping the WinCC message classes and message types

Introduction

The WinCC message system informs the user of disturbances and operating conditions in the process. A WinCC message always belongs to a specific message class and message type that is related to the event category.

The mapping of the WinCC message system on OPC is configured via the "CcAeProvider.ini" file.

Event Category

An event category is created on the WinCC OPC A&E server for every combination of a message class and type.

An event category is determined by a category ID and a descriptive "Category Description". The category ID is composed of the WinCC internal IDs for the message class and the message type; the category description is composed of the message class and message type.

Note

If the OPC A&E server is run on a WinCC client of a connectivity station, the OS servers linked to it must have an identical configuration of message classes and message types. If this is not the case, the OPC client used must access the OS server directly.

The names of the message classes and message types can be ascertained exactly via the alarm attributes "CLASSNAME" and "TYPENAME".

9.8.2.3 Mapping the WinCC message priority

Introduction

The priority of WinCC messages is displayed by the OPC server to the attribute "Severity".

When configuring alarms in the WinCC messaging system, you can configure a priority between 0 and 16. The OPC A&E specification defines a value range from 1 to 1000 for the severity where 1 stands for the lowest and 1000 for the highest severity.

Therefore, the values of the WinCC priority are suitably displayed to the OPC severity. In the standard mapping, the WinCC priority 0 becomes OPC severity 1. All other priority values are interpolated in a linear manner up to severity 1000. Other priority mapping rules can be configured in the CcAeProvider.ini file.

9.8.2.4 Attributes of the WinCC Message System

Introduction

The following table lists the OPC attributes of the WinCC message system.

The attributes are configured in the WinCC message system.

Some attributes are intended for internal use in WinCC only and are therefore not relevant to an OPC A&E client. These attributes are not contained in the table.

Attributes

OPC attributes	WinCC message system	Data type
CLASSNAME	Outputs the message class name.	VT_BSTR
TYPE NAME	Outputs the message type name.	VT_BSTR
FOREGROUND COLOR	Outputs the text color for activated, deactivated and acknowledged messages.	VT_I4
BACKCOLOR	Outputs the background color for activated, deactivated and acknowledged messages.	VT_I4
FLASHCOLOR	Outputs the flash color.	VT_I4
FLAGS	Indicates whether the message requires acknowledgment.	VT_I4
TEXT01	Outputs the content of UserTextBlock01.	VT_BSTR
TEXT02	Outputs the content of UserTextBlock02.	VT_BSTR
TEXT03	Outputs the content of UserTextBlock03.	VT_BSTR
TEXT04	Outputs the content of UserTextBlock04.	VT_BSTR
TEXT05	Outputs the content of UserTextBlock05.	VT_BSTR
TEXT06	Outputs the content of UserTextBlock06.	VT_BSTR
TEXT07	Outputs the content of UserTextBlock07.	VT_BSTR
TEXT08	Outputs the content of UserTextBlock08.	VT_BSTR
TEXT09	Outputs the content of UserTextBlock09.	VT_BSTR
TEXT10	Outputs the content of UserTextBlock10.	VT_BSTR
PROCESSVALUE01	Outputs the content of ProcessValueBlock01.	VT_VARIANT
PROCESSVALUE02	Outputs the content of ProcessValueBlock02.	VT_VARIANT
PROCESSVALUE03	Outputs the content of ProcessValueBlock03.	VT_VARIANT
PROCESSVALUE04	Outputs the content of ProcessValueBlock04.	VT_VARIANT
PROCESSVALUE05	Outputs the content of ProcessValueBlock05.	VT_VARIANT
PROCESSVALUE06	Outputs the content of ProcessValueBlock06.	VT_VARIANT
PROCESSVALUE07	Outputs the content of ProcessValueBlock07.	VT_VARIANT
PROCESSVALUE08	Outputs the content of ProcessValueBlock08.	VT_VARIANT
PROCESSVALUE09	Outputs the content of ProcessValueBlock09.	VT_VARIANT
PROCESSVALUE10	Outputs the content of ProcessValueBlock10.	VT_VARIANT
STATETEXT	Outputs the status message.	VT_BSTR
INFO TEXT	Outputs the information text for the message.	VT_BSTR
LOOPINALARM	Indicates whether LoopInAlarm is configured.	VT_I4
CLASSID	Outputs the message class ID.	VT_I4

OPC attributes	WinCC message system	Data type
TYPEID	Outputs the message type ID.	VT_I4
MODIFYSTATE	Outputs the value of the status tag of the message.	VT_I4
AGNR	Outputs the number of the AS that generated the message.	VT_I2
CPUNR	Outputs the number of the CPU that generated the message.	VT_I2
DURATION	Outputs the interval between the activation, deactivation and acknowledgment of a message.	VT_I4
COUNTER	Outputs the number of messages after the start of Runtime.	VT_I4
QUITSTATETEXT	Indicates whether the message has been acknowledged.	VT_BSTR
QUITCOUNT	Outputs the number of active, unacknowledged messages.	VT_I4
PARAMETER	Outputs the parameters of message (image of the message configuration).	VT_BSTR
BLOCKINFO	Outputs the current content of the message block.	VT_BSTR
ALARMCOUNT	Outputs the number of messages pending.	VT_I4
LOCKCOUNT	Outputs the number of locked messages.	VT_I4
PRIORITY	Indicates the configured priority of the message.	VT_I4
APPLICATION	Outputs the application which triggered the message.	VT_BSTR
COMPUTER	Outputs the name of the PC that processed the message.	VT_BSTR
USER	Outputs the name of the user who processed the message.	VT_BSTR
COMMENT	Outputs the message comment.	VT_BSTR

9.8.2.5 Acknowledgement theory

Introduction

For WinCC, the acknowledgment philosophy is how a message is displayed and processed from "came in" to "went out". On the WinCC OPC A&E server, this message status is managed in parameters "ChangeMask" and "NewState".

Conditional, Simple and Tracking Events

Typically, messages from the WinCC system are sent to the client as conditional events. In order for a message to be treated as a simple event, the following conditions must be met during configuration of the message class:

- "Acknowledgment Came In" is not activated.
- "Message Without Status Went Out" is activated.

Depending on the mapping configuration, the messages of the message class "System without Acknowledgement" and of the message type "Operations message" are transferred as OPC Tracking Events.

ChangeMask

The "ChangeMask" parameter keeps track of where the message status was changed.

Parameter values:

- OPC_CHANGE_ACTIVE_STATE
- OPC_CHANGE_ENABLE_STATE
- OPC_CHANGE_ACK_STATE

NewState

The "NewState" parameter indicates the message status after a change.

Parameter values:

- OPC_CONDITION_ACTIVE
- OPC_CONDITION_ENABLED
- OPC_CONDITION_ACKED

Overview

WinCC	NewState	ChangeState
Received message	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Sent message with receipt	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Sent message without receipt	OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Acknowledged messages (message pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Acknowledged messages (message no longer pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Locked message	--------------------------------------	OPC_CHANGE_ENABLED_STATE
Unlocked message	OPC_CONDITION_ENABLED	OPC_CHANGE_ENABLED_STATE
Received, acknowledged message	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Received, sent message with receipt	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Received, sent message without receipt	OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Message acknowledged by the system (message pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Message acknowledged by the system (message no longer pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE

WinCC	NewState	ChangeState
Emergency-acknowledged message (message pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Emergency-acknowledged message (message no longer pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE

See also

www.opcfoundation.org (http://www.opcfoundation.org)

9.8.3 Quality Codes for OPC A&E

Introduction

Quality codes are used to evaluate the status and quality of a message. The quality codes for OPC are described under "6.8 OPC Quality flags" of the "Data Access Custom Interface Standard Version 3.00" specifications.

Quality codes

Code	Quality	Status
0xC0	OPC_GOOD	OK
0x40	OPC_UNCERTAIN	Returned in cases of uncertainty, e.g. in the event of delayed acknowledgement (receipt).
0x00	OPC_BAD	Returned if the connection to the source is interrupted.

9.8.4 Example of an OPC A&E Connection

9.8.4.1 Example of an OPC A&E Connection

Introduction

In the example below, a connection between WinCC and an OPC A&E client is configured. Data from the WinCC message system are made available via the WinCC OPC A&E server.

The OPC A&E client is kept informed of status changes of WinCC messages by means of a subscription.

All OPC A&E clients conforming to the OPC Alarms&Events 1.10 specifications can access the WinCC OPC A&E server.

Configuration Step

The following configurations are required for connection between WinCC and the OPC A&E client:

1. Configuring access to the WinCC message system

See also

How to Configure Access to the WinCC Message System (Page 731)

www.opcfoundation.org (http://www.opcfoundation.org)

9.8.4.2 How to Configure Access to the WinCC Message System

Introduction

In this section, the OPC A&E client of the OPC foundation accesses the WinCC message system.

Note

The OPC A&E client described here is the demo client from the OPC Foundation. The source code for it is found on the Internet at http://www.opcfoundation.org.

Requirement

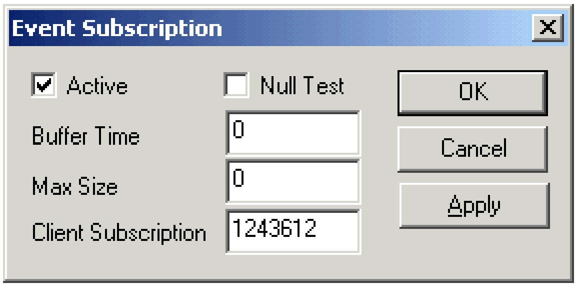
- Create several internal tags of the "binary" data type in the WinCC project of the WinCC OPC A&E server.
- Configure the WinCC message system in the WinCC project of the WinCC OPC A&E server. Link the messages to the internal tags.
- Configure a picture with the Graphics Designer. Add the WinCC alarm control and an I/O field to the picture. Link the message tags to the graphic objects.
- Enable the "Alarm Logging Runtime" in the start list.
- Enable the WinCC project of the WinCC OPC A&E server.

Procedure

1. Copy the "SampleClientAE.exe" file from the folder "Siemens\WinCC\documents\english" to a folder of your choice. This application is only available in the online help.
2. Select "OPC" >"Connect..." in the menu bar. Select "OPC.WinCC-AlarmsEvent" in the "OPC Alarm Server" dialog. Click "OK" to close the dialog.

3. Select "OPC" >"Event Subscription..." from the menu bar. The "Event Subscription" dialog is opened.

4. Select the check box labeled "Active" in the dialog. Enter "1000" in the "Buffer Time" and "Max Size" fields. Click "OK" to close the "Event Subscription" dialog.



Event Subscription

☑ Active

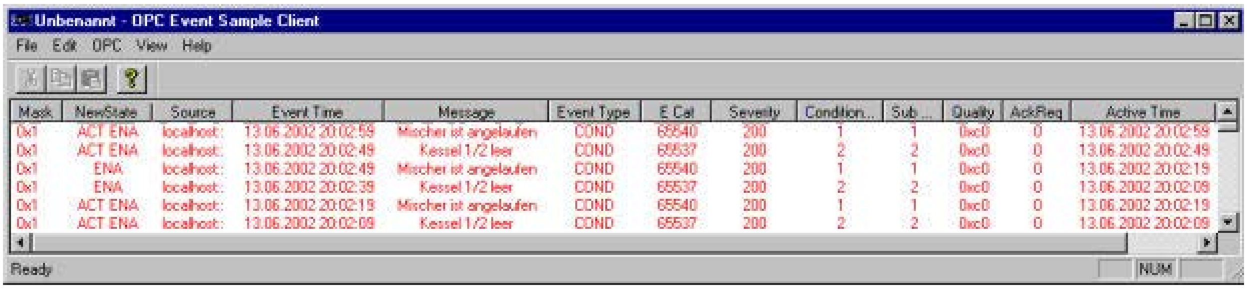
☐ Null Test

Buffer Time: 0

Max Size: 0

Client Subscription: 1243612

5. The messages from the WinCC message system are displayed in the OPC Event Sample Client.



Unbenannt - OPC Event Sample Client

Mask	NewState	Source	Event Time	Message	Event Type	E Cat	Severity	Condition...	Sub ...	Quality	AckReq	Active Time
0x1	ACT ENA	localhost:	13.06.2002 20:02:59	Mischer ist angelaufen	COND	65540	200	1	1	0xc0	0	13.06.2002 20:02:59
0x1	ACT ENA	localhost:	13.06.2002 20:02:49	Kessel 1/2 leer	COND	65537	200	2	2	0xc0	0	13.06.2002 20:02:49
0x1	ENA	localhost:	13.06.2002 20:02:49	Mischer ist angelaufen	COND	65540	200	1	1	0xc0	0	13.06.2002 20:02:19
0x1	ENA	localhost:	13.06.2002 20:02:39	Kessel 1/2 leer	COND	65537	200	2	2	0xc0	0	13.06.2002 20:02:09
0x1	ACT ENA	localhost:	13.06.2002 20:02:19	Mischer ist angelaufen	COND	65540	200	1	1	0xc0	0	13.06.2002 20:02:19
0x1	ACT ENA	localhost:	13.06.2002 20:02:09	Kessel 1/2 leer	COND	65537	200	2	2	0xc0	0	13.06.2002 20:02:09

Ready

6. Select "OPC" >"Filter" from the menu bar. The "Filter" dialog is opened. Select a category from the "Event Category" field. Click "OK" to close the "Filter" dialog.

7. The messages meeting the filter criteria are displayed in the OPC Event Sample Client.

"Buffer Time" and "Max Size" Parameters

According to OPC specification, the "Buffer Time" and "Max Size" parameters are configured in WinCC as follows:

OPC Client demands return value	WinCC uses
Buffer time < 100 OPC_S_INVALIDBUFFERTIME	Revised buffer time = 100
100 <= buffer time <= 600000 S_OK	Revised buffer time = buffer time
Buffer time > 600000 OPC_S_INVALIDBUFFERTIME	Revised buffer time = 600000
Max size = 0 OPC_S_INVALIDMAXSIZE	Revised max size = 1000
0 < max size < 10 OPC_S_INVALIDMAXSIZE	Revised max size = 10
10 <= max size <= 1000 S_OK	Revised max size = max size
Max Size = 1000 OPC_S_INVALIDMAXSIZE	Revised max size = 1000

Parameters may be set while creating a subscription. However, you cannot change an existing subscription using SetState() after the fact.

For more information, refer to http://www.opcfoundation.org.

See also

www.opcfoundation.org (http://www.opcfoundation.org)

9.8.5 OPC A&E server with hierarchical access

9.8.5.1 Functionality of the OPC A&E server

Introduction

The OPC-A&E server uses DCOM services for transferring messages between OPC-capable applications. The OPC A&E server supports the specification OPC Alarm&Event 1.10.

The following chapter explains the mapping of the WinCC message system on OPC A&E with hierarchical access and the attributes supported by the OPC A&E server. This documentation includes an overview of the specific information. For more information, refer to the "OPC Alarms & Events 1.10" specification.

Principle of operation

The OPC-A&E client receives WinCC messages via subscription. You can use the subscription filter to reduce the number of events that will be transferred with a subscription. The OPC-A&E client can be set for every event category that displays message attributes.

Installation

The WinCC OPC A&E server can be selected during the installation of WinCC. After installation, the WinCC OPC A&E server is immediately available for use without any additional configuration.

The WinCC OPC A&E server can be implemented on a WinCC server and a WinCC client.

Licensing

In order to operate the WinCC OPC A&E server, the following licenses must be installed on each WinCC server implemented as an OPC A&E server:

- A valid RT license for WinCC
- WinCC Option Connectivity Pack

Event types

The OPC-A&E server with hierarchical access supports conditional events, simple events and tracking events.

Condition related events

With a condition related event, the event is associated with a condition. A condition might, for example, be a limit value violation of a tag. This limit violation generates a message that is shown as an alarm with OPC A&E.

Simple events

Simple events are messages that inform the OPC A&E client about events. Simple events include, for example, starting or exiting programs.

Note

Note the following when using redundant systems:

Simple events interconnected to internal tags are sent twice when tags are updated.

The first message is triggered by the master, the second by the standby.

Tracking events

A tracking event is sent with a operator input message to the OPC A&E client. An operator input message is triggered by manual intervention in the process.

OPC A&E client

All OPC A&E clients conforming to the OPC Alarms & Events 1.10 specification can access the OPC A&E server. You can also create the OPC A&E client yourself. By creating proprietary OPC clients, most user-specific requirements can be met. An OPC A&E client, for example, may be used for analysis and joint archiving of alarms from different OPC A&E servers. The acknowledgment of archived messages is not possible; only current alarms and events can be acknowledged.

If you are using the OPC A&E with hierarchical access and want to use all functions, you may need to adapt the OPC A&E client currently used.

Note

Documentation on OPC

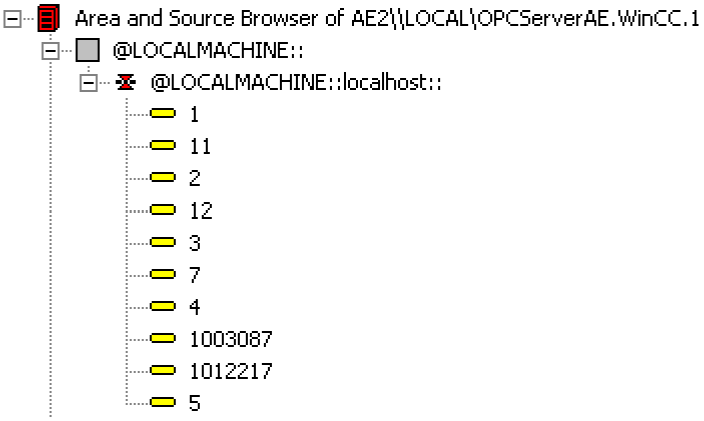
You can find additional information on OPC in the Chapter "Interfaces > OPC - OLE for Process Control".

9.8.5.2 Differences between OPC A&E and OPC A&E with hierarchical access

Displaying messages with OPC A&E

The OPC A&E server supports "conditional events" and "simple events" for accessing the message system. With "conditional events", the message numbers are shown for each source. Since an WinCC server can hold many more message numbers, it is difficult to maintain an overview of the messages.

The following figure shows an example of the display in an OPC browser:



```
Area and Source Browser of AE2\\LOCAL\OPCServerAE.WinCC.1
  @LOCALMACHINE::
    @LOCALMACHINE::localhost::
      1
      11
      2
      12
      3
      7
      4
      1003087
      1012217
      5
```

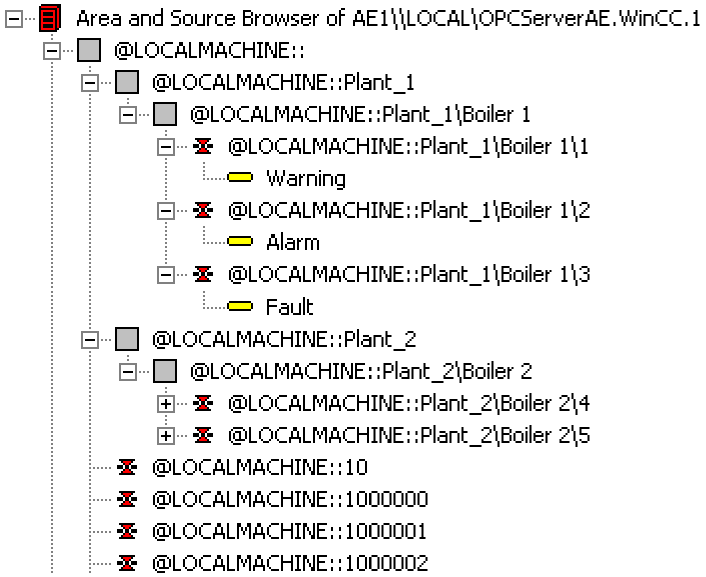
Displaying the messages with OPC A&E and hierarchical access

The OPC A&E server with hierarchical access supports the event types, conditional events, simple events and tracking events.

The user text block 2 determines the source of the messages for "conditional events". With the default setting, user text block 2 corresponds to the fault location. In order to present messages hierarchically, they must be combined in user-defined group messages in alarm logging messages. The structure of group messages is determined by the areas in OPC A&E.

Tracking events occur when operator input messages are triggered in the system.

The following figure shows an example of the display of conditional events in an OPC browser. The "Condition" is shown in addition to "Area" and "Source":



```
Area and Source Browser of AE1\\LOCAL\OPCServerAE.WinCC.1
  @LOCALMACHINE::
    @LOCALMACHINE::Plant_1
      @LOCALMACHINE::Plant_1\Boiler 1
        @LOCALMACHINE::Plant_1\Boiler 1\1
          Warning
        @LOCALMACHINE::Plant_1\Boiler 1\2
          Alarm
        @LOCALMACHINE::Plant_1\Boiler 1\3
          Fault
    @LOCALMACHINE::Plant_2
      @LOCALMACHINE::Plant_2\Boiler 2
        @LOCALMACHINE::Plant_2\Boiler 2\4
        @LOCALMACHINE::Plant_2\Boiler 2\5
    @LOCALMACHINE::10
    @LOCALMACHINE::1000000
    @LOCALMACHINE::1000001
    @LOCALMACHINE::1000002
```

Switching to OPC A&E with hierarchical access

Use an OPC A&E server with hierarchical access when creating a new project.

In an existing project, the OPC A&E server can be used as before or be converted for hierarchical access. The conversion can be undone again without any loss of data.

1. Copy the "CcAeProvider.ini" file into the project folder. The file is located in the WinCC installation path in the folder "OPC\AlarmEvent\Hierarchical-Access".
2. Update the clients or perform a complete download for the OS servers.

9.8.5.3 Mapping the WinCC Message System on OPC A&E

Mapping the WinCC message system

Introduction

The WinCC message system resulting from the configuration defines which event in the process will generate a message. This message is shown as an event notification in OPC A&E.

Mapping the WinCC message system on OPC A&E with hierarchical access

The OPC source of the WinCC user text block "2" and the OPC message of WinCC user text block "1" are used in WinCC as a default setting for mapping the WinCC message systems.

Overview

The following table shows the most important attributes of the event notifications and the respective information from the WinCC message system.

The events that use the configured attributes are shown in the third column of the table:

- "S" means a simple event
- "C" means a conditional event
- "T" means a tracking event

OPC	WinCC message system	Event type
Area	The structure of the group messages determine the areas in OPC A&E. If there is no group message configured for the message, only the OPC area corresponding to the server prefix is available.	S, C, T
Source	Indicates the source of a message. The source has the format "<server prefix>::Area\user text block 2". The server prefix of a local computer is "@LOCALMACHINE". The server prefix always shows the top Areas in the hierarchy of the server.	S, C, T
Time	Issues a time stamp for received, sent and acknowledged messages. Issues a time stamp in UTC (Universal Time Coordinated).	S, C, T
Type	Indicates whether the event is a simple, tracking or conditional event.	S, C, T
Severity	Returns the priority of the message.	S, C, T
EventCategory	Indicates the message class. "Event Category" is made up of the "CategoryID" and the "Category Description". "CategoryID" corresponds to the internal ID of the message class. "Category Description" corresponds to the name of the message class.	S, C, T
Message	Indicates the message text of the corresponding message number.	S, C, T
Condition	Indicates the message type.	C
Sub-condition	Corresponds with the "Condition" parameter.	C
ChangeMask	Specifies the change of the condition. For more information, refer to "Acknowledgment Theory".	C
NewState	Indicates the current status of the condition. For more information, refer to "Acknowledgment Theory".	C
ConditionQuality	Returns the quality of the message. For more information, refer to "Quality codes".	C
AckRequired	Indicates whether the message requires acknowledgment.	C
EventAttribute	Lists the attributes required for the respective message. For more information, refer to "Attributes of the WinCC message system".	C
Quality	Returns the quality code of the message.	C
Cookie	Does not include any usable information for the client	C
ActorID	Indicates which user acknowledged the message.	T

Note

If text without wild cards are specified as a filter for the area, only the messages of the area are returned. If you want to include sources that are located in areas outside the specified area, you need to use wild cards.

Note

The message classes and message types **must** be configured identically on the connected OS servers, if you run the OPC A&E server as follows:

- On a WinCC Client
- On a Connectivity station

If the OS server is not configured identically, the employed OPC client must access the respective OS server directly.

Mapping the message priority

Introduction

The priority of messages is mapped by the OPC A&E server to the attribute "Severity".

When configuring alarms in the messaging system, you can configure a priority between "0" and "16". The OPC A&E specification defines a value range of "1" to "1000" for the severity. In this case, "1" stands for the lowest and "1000" for the highest severity.

Therefore, the values of the priority are suitably displayed to the OPC severity. In the standard mapping, priority "0" is assigned to OPC severity "1" and priority "16" to OPC severity "1000". All other priority values are interpolated linearly between "0" and "1000".

Attributes of the WinCC Message System

Introduction

The following table lists the OPC attributes of the WinCC message system.

The attributes are configured in the WinCC message system.

Some attributes are intended for internal use in WinCC only and are therefore not relevant to an OPC A&E client. These attributes are not contained in the table.

Attributes

OPC attributes	WinCC message system	Data type
CLASSNAME	Outputs the message class name.	VT_BSTR
TYPE NAME	Outputs the message type name.	VT_BSTR
FOREGROUND COLOR	Outputs the text color for activated, deactivated and acknowledged messages.	VT_I4
BACKCOLOR	Outputs the background color for activated, deactivated and acknowledged messages.	VT_I4
FLASHCOLOR	Outputs the flash color.	VT_I4

OPC attributes	WinCC message system	Data type
FLAGS	Indicates whether the message requires acknowledgment.	VT_I4
TEXT01	Outputs the content of UserTextBlock01.	VT_BSTR
TEXT02	Outputs the content of UserTextBlock02.	VT_BSTR
TEXT03	Outputs the content of UserTextBlock03.	VT_BSTR
TEXT04	Outputs the content of UserTextBlock04.	VT_BSTR
TEXT05	Outputs the content of UserTextBlock05.	VT_BSTR
TEXT06	Outputs the content of UserTextBlock06.	VT_BSTR
TEXT07	Outputs the content of UserTextBlock07.	VT_BSTR
TEXT08	Outputs the content of UserTextBlock08.	VT_BSTR
TEXT09	Outputs the content of UserTextBlock09.	VT_BSTR
TEXT10	Outputs the content of UserTextBlock10.	VT_BSTR
PROCESSVALUE01	Outputs the content of ProcessValueBlock01.	VT_VARIANT
PROCESSVALUE02	Outputs the content of ProcessValueBlock02.	VT_VARIANT
PROCESSVALUE03	Outputs the content of ProcessValueBlock03.	VT_VARIANT
PROCESSVALUE04	Outputs the content of ProcessValueBlock04.	VT_VARIANT
PROCESSVALUE05	Outputs the content of ProcessValueBlock05.	VT_VARIANT
PROCESSVALUE06	Outputs the content of ProcessValueBlock06.	VT_VARIANT
PROCESSVALUE07	Outputs the content of ProcessValueBlock07.	VT_VARIANT
PROCESSVALUE08	Outputs the content of ProcessValueBlock08.	VT_VARIANT
PROCESSVALUE09	Outputs the content of ProcessValueBlock09.	VT_VARIANT
PROCESSVALUE10	Outputs the content of ProcessValueBlock10.	VT_VARIANT
STATETEXT	Outputs the status message.	VT_BSTR
INFO TEXT	Outputs the information text for the message.	VT_BSTR
LOOPINALARM	Indicates whether LoopInAlarm is configured.	VT_I4
CLASSID	Outputs the message class ID.	VT_I4
TYPEID	Outputs the message type ID.	VT_I4
MODIFYSTATE	Outputs the value of the status tag of the message.	VT_I4
AGNR	Outputs the number of the AS that generated the message.	VT_I2
CPUNR	Outputs the number of the CPU that generated the message.	VT_I2
DURATION	Outputs the interval between the activation, deactivation and acknowledgment of a message.	VT_I4
COUNTER	Outputs the number of messages after the start of Runtime.	VT_I4
QUITSTATETEXT	Indicates whether the message has been acknowledged.	VT_BSTR
QUITCOUNT	Outputs the number of active, unacknowledged messages.	VT_I4
PARAMETER	Outputs the parameters of message (image of the message configuration).	VT_BSTR
BLOCKINFO	Outputs the current content of the message block.	VT_BSTR
ALARMCOUNT	Outputs the number of messages pending.	VT_I4
LOCKCOUNT	Outputs the number of locked messages.	VT_I4
PRIORITY	Indicates the configured priority of the message.	VT_I4
APPLICATION	Outputs the application which triggered the message.	VT_BSTR
COMPUTER	Outputs the name of the PC that processed the message.	VT_BSTR
USER	Outputs the name of the user who processed the message.	VT_BSTR

OPC attributes	WinCC message system	Data type
COMMENT	Outputs the message comment.	VT_BSTR
HIDDEN COUNT	Outputs the number of hidden messages.	VT_I4
BIG COUNTER	Outputs the number of messages after the start of Runtime.	VT_CY
OS-HIDDEN	Outputs the hidden status of the message.	VT_BOOL
OS-EVENTID	Outputs the message number configured for the message.	VT_I4

Acknowledgement Theory

Introduction

The acknowledgment policy in WinCC is how a message from "came in" to "went out" is displayed and processed . On the OPC A&E server, this message status is displayed in the "ChangeMask" and "NewState" parameters.

Conditional events, simple events and tracking events

Messages from the system are sent to the client as conditional events with acknowledgment.

In order for a message to be handled as a simple event, the message class of the message must meet the following conditions:

- "Acknowledgment came in" is not activated.
- "Message without status went out" is activated.

In WinCC, messages of message class "System, does not require acknowledgment" with "Operator input message" message type are transferred as tracking events.

Note

Messages with "System, does not require acknowledgment" message class and "Process control system" message type are transferred as simple events with the "System message" event category.

ChangeMask

The "ChangeMask" parameter keeps track of where the message status was changed.

Parameter values:

- OPC_CHANGE_ACTIVE_STATE
- OPC_CHANGE_ENABLE_STATE
- OPC_CHANGE_ACK_STATE

NewState

The "NewState" parameter indicates the message status after a change.

Parameter values:

- OPC_CONDITION_ACTIVE
- OPC_CONDITION_ENABLED
- OPC_CONDITION_ACKED

Overview

WinCC	NewState	ChangeState
Received message	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Went out message with acknowledgment	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Went out message without acknowledgment	OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Acknowledged messages (message pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Acknowledged messages (message no longer pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Locked message	-------------------------------------	OPC_CHANGE_ENABLED_STATE
Unlocked message	OPC_CONDITION_ENABLED	OPC_CHANGE_ENABLED_STATE
Came in, acknowledged message	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Came in, went out message with acknowledgment	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Came in, went out message without acknowledgment	OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Message acknowledged by the system (message pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Message acknowledged by the system (message no longer pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Emergency-acknowledged message (message pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Emergency-acknowledged message (message no longer pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE

Note

Historical alarms and events are not acknowledged. The OPC A&E historical events interface only has read access.

9.8.5.4 Quality Codes for OPC A&E

Introduction

Quality codes are used to evaluate the status and quality of a message. The quality codes for OPC are described under "6.8 OPC Quality flags" of the "Data Access Custom Interface Standard Version 3.00" specifications.

Quality codes

Code	Quality	Status
0xC0	OPC_GOOD	OK
0x40	OPC_UNCERTAIN	Returned in cases of uncertainty, for example in the event of delayed acknowledgment (receipt).
0x00	OPC_BAD	Returned if the connection to the source is interrupted.

9.8.6 Reading archived messages

9.8.6.1 Accessing archived events

Introduction

You can access the archived messages via the OPC A&E server using an OPC client. Two methods are supported for accessing archived messages:

- Output archived messages from a time period in the past
- Output archived messages from a time period in the past without mentioning end of period. After the output of archived messages, all other newly generated messages are automatically sent to the OPC client.

Note

After reading archived messages, you cannot use the returned "ActiveTime" of a message for acknowledging the message or tracing transitions of the message. To ensure this, the OPC A&E client must check the "EventType" of a message with the extra flag "OPC_HAE_HISTORICAL_EVENTFLAG". The "ActiveTime" is incorrect on archived messages. You can find information on the additional flag under "Identifying archived messages".

Querying the "Historic Alarms and Events" functionalities

In addition to the standard filters, the following filters are offered with the expanded OPC A&E server of WinCC:

Filter	Filter Values	Description
OPC_HAE_FILTER_BY_TIMEFRAME	0x80000000	Matches "ReadRaw" function for OPC Historical Data Access
OPC_HAE_FILTER_BY_STARTTIME	0x40000000	Matches "AdviseRaw" function for OPC Historical Data Access

Source filter and historical alarm request

To request the archive messages, the OPC client must support the "SetFilter" to a subscription functionality. The OPC server will also send archived messages if you also insert keyword "OPCHAEServer" the array of the "Source Filter" of a subscription. In addition to this keyword, you can use other parameters to define which messages are to be read:

- Method
- Time period
- With or without limits

The lists of sources that are assigned in the filter can include other source names besides the "OPCHAEServer" source. In such a case, the subscription delivers only the historic events of the given sources. The sequence of the source names is inconsequential.

After configuring the source filter, the selected time period can be called up from the client with a "Refresh" call.

9.8.6.2 Syntax for accessing archived messages using OPC

Syntax

```
OPCHAEServer hMode=(read|advise) htStartTime=szTime
[hEndTime=szTime] [bBounds=(TRUE|FALSE)]
```

Parameter

hMode = [read|advise]

This parameter is required. Defines how the archived messages and events are to be read.

Read: Outputs archived messages and events of a definite period from the past (comparable to ReadRaw in case of OPC Historical Data Access).

The following is an example for setting a filter for reading over the last 30 minutes:

```
OPCHAEServer hMode=read htStartTime=NOW-30M bBounds=TRUE
```

Advise: Outputs archived messages and events from a definite period, After receiving all archived messages, new messages are sent in the same way as in the case of an active subscription (comparable to AdviseRaw in case of OPC Historical Data Access).

In the following example, the messages of the last 30 minutes are read (subscription must be active):

```
OPCHAEServer hMode=advise htStartTime=NOW-30M
```

Note

The following notation is supported for parameters "htStartTime" and "htEndTime":

- Relative notations, for example NOW
- Symbolic values, for example NOW, YEAR, MONTH
- Specification of absolute UTC data/time values according to XML notation: 2006-09-01T10:00:00.000Z

Using the symbolic notation corresponds to the syntax from OPC Historical Data Access.

htStartTime =

This parameter is required. Defines the time from when the messages and events are to be read from the archive.

htEndTime =

This parameter is optional. Defines the time up to which the messages and events are to be read from the archive. With "hMode = read", the default setting "NOW" is used.

bBounds = [TRUE|FALSE]

This parameter is optional. Defines how messages close to the start and end time are to be handled. The function is identical to OPC Historical Data Access

bBounds=FALSE:

- The time stamp of the first transferred message >= htStartTime
- The time stamp of the last transferred message >= htEndTime

bBounds=TRUE:

- The time stamp of the first transferred message <= htStartTime
- The time stamp of the last transferred message >= hEndTime

Default setting is FALSE.

9.8.6.3 Read methods for archived messages

Introduction

You can use one of two read modes to read archived messages:

- read
- advise

Read mode "read"

Archived messages from a defined period in the past are read with "read" mode. The order of the read messages is always chronological with regard to each OS server from which alarms are being read. By setting the start and end times, you can specify whether the oldest message is to be output first or last. If the start time is earlier than the end time, the oldest message is output last.

If you want to use "read" mode, run the following functions on the subscription:

1. SetFilter
2. Refresh

Event packets with Refresh identifier contain only historical events. The events can also be in queue.

The last Refresh packet of the historical messages contains the "Last Refresh" identifier.

A "SetFilter" during the "Refresh" will be rejected. If you activate the subscription during the "Refresh", it has no effect on the refresh process.

The historical events will continue to be transmitted with the Refresh identifier.

The newly generated events are transmitted according to the standard behavior of an active subscription:

- Taking into account the set filter values with the exception of the "historical" source "OPCHAEServer"
- Without the Refresh identifier

This enables the client to differentiate the received events based on the Refresh identifier. An event packet never contains historical and new events at the same time.

- Event packets with Refresh identifier contain only historical events. These events can also be in queue.
- Event packets without the Refresh identifier contain only newly generated events.

Read mode "advise"

Archived messages starting from a defined period in the past are read with "advise" mode. After all archived messages are read, new messages are sent the same as when a subscription is active. The archived messages are transferred chronologically with respect to each OS server: The archived messages starting from the start time are transmitted first. The newly archived messages are transmitted afterwards.

Note that you must not specify an end time for "advise".

An active subscription is used for "advise" mode. If you run the "SetFilter" function on an active subscription, the historical alarms are transmitted immediately.

If you run the "SetFilter" function on an inactive subscription, the archived messages are only transmitted after activation of the subscription. If you want to use "advise" mode with an inactive subscription, proceed as follows:

1. SetFilter
2. Set subscription to active using SetState

The transmission is ended when you set the subscription to "inactive". A "SetFilter" is rejected while the subscription is active.

A "Refresh" on an active "historical" subscription in "advise" mode functions in the same way as on a standard subscription:

All queued condition related events are transmitted in packets with Refresh identifier.

A "Refresh" call has no effect on the reading of historical alarms in "advise" mode.

9.8.6.4 Identifying archived messages

General procedure

Archived messages are distinguished using an additional flag in EventType. This flag is linked to the real EventType via a OR link.

Name	EventType	EventType (archived message)
OPC_SIMPLE_EVENT	0x01	0x81
OPC_CONDITION_EVENT	0x04	0x84
OPC_TRACKING_EVENT	0x02	0x82
OPC_HAE_HISTORICAL_EVENTFLAG		0x80

Examples

Example 1

The following source filter is used to output archived messages and events of the last 30 minutes in "read" mode. The oldest message for each OS server is output as the first one. The low limit value is also sent.

```
OPCHAEServer hMode=read htStartTime=NOW-30M bBounds=TRUE
```

Example 2

The following source filter is used to output archived events on September 1, 2006 from 10:00 to 12:00 hours in "read" mode. The newest message for each OS server is output as the first one. The limits for this time period are also sent.

```
OPCHAEServer hMode=read htStartTime=2006-09-01T12:00:00.000Z
htEndTime=2006-09-01T10:00:00.000Z bBounds=TRUE
```

Example 3

The following source filter is used to output archived messages and events of the last 30 minutes in "advise" mode. After reading the archived messages, newly generated messages are sent in the same way as for an active subscription.

```
OPCHAEServer hmode=advise htStartTime=NOW-30M
```

9.9 WinCC OPC UA Server

9.9.1 Principle of operation the WinCC OPC UA Server

How it works

The WinCC OPC UA Server provides the following values:

- Process values
- Values from tag archives
- WinCC messages

The WinCC OPC UA server is installed as Windows service and started automatically. The WinCC OPC UA server supports only the "UA-TCP UA-SC UA Binary" communication profile.
The used port number is adjustable.

Supported specifications

OPC Unified Architecture is a specification for the transmission of process values, archive data and messages. The WinCC OPC UA server supports OPC UA Specification 1.03. For additional information about supported UA functions, refer to "Supported OPC UA services and profiles (Page 756)".

Installation

After WinCC is installed, the WinCC OPC UA server can be used immediately without the need for any further configuration.
The WinCC OPC UA server can be used on a WinCC server or a WinCC client.

URL of the WinCC OPC UA server

You access the WinCC OPC UA server via the following URL:

- "opc.tcp://[HostName]:[Port]"

Parameter	Description
HostName	Placeholder for the computer name. Is used automatically
Port	Port number. The default setting is "4862".

Discovery Server

The "Discovery Server" is available by the OPC foundation. The "Discovery Server" is by default installed on the HMI device as Windows service.

On the "Discovery Server" via OPC UA server UA clients information is available that is registered on the "Discovery Server".

Depending on the configuration, the WinCC OPC UA server registers on no, on one or on multiple configured and available "Discovery servers" upon runtime startup. Registration is then repeated cyclically. If you end Runtime, the WinCC OPC UA server is automatically logged off from the "Discovery server".

Supported languages in the WinCC address area

The WinCC OPC A&E Server supports the WinCC address area in the following languages:

- German
- English
- French
- Italian
- Spanish

9.9.2 Security concept of OPC UA

Introduction

The OPC UA security concept is based largely on:

- Authentication and authorization of applications and users involved
- Ensuring the integrity and confidentiality of messages exchanged between the applications

Certificates are the method used for authentication of the OPC UA applications.

Each application has its own instance certificate with which it identifies itself in the public key infrastructure. The instance certificate is also called the "application certificate".

Certificate of the WinCC OPC UA Server

For secure operation, each WinCC OPC UA server requires its own certificate with a private key, a server certificate.

The certificate is only valid on the corresponding computer and may only be used by the WINCC OPC UA server installed on that computer.

A self-signed certificate of the server is created and stored in the certificate folder of the server.

The private key for this server certificate is also stored in the certificate folder. You must restrict access to the folder with the private key to:

- the server itself
- the system administrator

NOTICE
Access to the folder with the private key
For security reasons, no other users or applications apart from the server and the system administrator may have access to the private key of the WINCC OPC UA server.

The administrator of the plant can replace the server certificate and the corresponding private key generated during the installation.

In accordance with the applicable security concept for the system, the new server certificate can be either self-signed or issued by a certification authority.

The certificates used by the WINCC OPC UA server are determined by the settings in the "OpcUaServerWinCC.xml" configuration file: You can find additional information under "Configuration file of the WinCC OPC UA Server (Page 769)".

Storage of server certificates

The "WinCC OPC UA server" application is stored in the following path:

Storage path	Application	Configuration file
<Installation directory>WinCC\opc\UA-Server\	OpcUaServerWinCC.exe	OpcUaServerWinCC.xml

The WinCC OPC UA certificates are stored in the following folders of the WinCC installation path:

WinCC OPC UA server	Certificates	opc\UAServer\PKI\CA\certs
	Private key	opc\UAServer\PKI\CA\private

You can change the storage location in the configuration file.

Trusted client certificates

The WinCC OPC UA server supports secure communication with trusted clients only. A client is trusted:

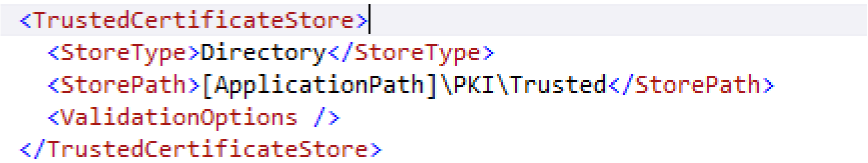
- If the client has a valid self-signed certificate which is stored in the trusted certificates certificate memory of the WinCC OPC UA server
- or if the valid client certificate was issued by a certification authority.
  The valid certificate from the certification authority must be located in the trusted certificates certificate memory of the WinCC OPC UA server. In this case, only the certificate from the certification authority is required. The client certificate does not need to be located in the certificate store for trusted certificates.

Storage of client certificates

You specify storage settings for trusted certificates using the WINCC OPC UA server configuration file:

Parameter	Meaning
StoreType	Type of certificate storage. The storage location can be either "Directory" or "Windows".
StorePath	The certificates of trusted clients are stored under this folder.

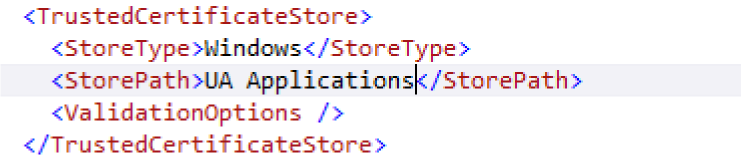
Example of configuration with "Directory" storage



```
<TrustedCertificateStore>
  <StoreType>Directory</StoreType>
  <StorePath>[ApplicationPath]\PKI\Trusted</StorePath>
  <ValidationOptions />
</TrustedCertificateStore>
```

In this case, the WINCC OPC UA server trusts all clients whose server certificates are located in the "...PKI\TrustList\Certs" folder.

Example of configuration with "Windows" storage



```
<TrustedCertificateStore>
  <StoreType>Windows</StoreType>
  <StorePath>UA Applications</StorePath>
  <ValidationOptions />
</TrustedCertificateStore>
```

For this storage option, the certificates of the clients must be located in the certificate store of the operating system under "<Local Computer>\UA Applications".

Certificates from certification authorities that are required for verifying a client certificate chain are stored in the certificate store of the certification authorities. Here too, you specify storage settings using the WINCC OPC UA server configuration file:

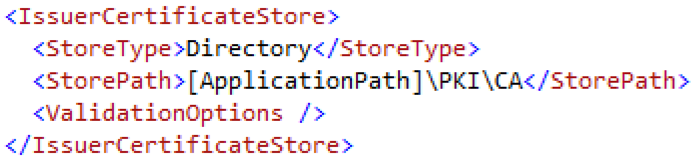
Parameter	Meaning
StoreType	Type of certificate storage. The storage location can be either "Directory" or "Windows".
StorePath	The certificates of trusted certification authorities are stored under this folder.

Note

Certificates from the memory of the certification authorities are not automatically trusted.

For a certification authority to be trusted, its certificate must be located in the memory for trusted certificates.

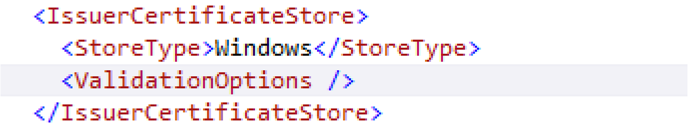
Example of configuration with "Directory" storage



```
<IssuerCertificateStore>
  <StoreType>Directory</StoreType>
  <StorePath>[ApplicationPath]\PKI\CA</StorePath>
  <ValidationOptions />
</IssuerCertificateStore>
```

The certificates of trusted certification authorities are in this case located in the "...\PKI\CA\Certs" folder.

Example of configuration with "Windows" storage



```
<IssuerCertificateStore>
  <StoreType>Windows</StoreType>
  <ValidationOptions />
</IssuerCertificateStore>
```

The "StorePath" parameter is not relevant. The certificates from certification authorities must be stored in the Windows certificate memory in accordance with the operating system requirements.

Certificates are trusted if they are located in one of these two locations:

- <Local computer>\Trusted root certification authorities
- <Local computer>\Third-party root certification authorities

Note

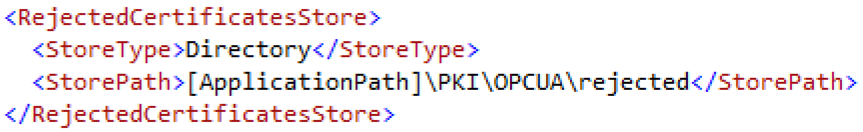
Important for storage

- The storage location for the server certificate must be "Directory".
- The two storage locations for trusted client certificates and for certificates from certification authorities must have the same StoreType, i.e. both must either be "Directory" or "Windows".

Client certificates not accepted

If a UA client accesses the WINCC OPC UA server without having a trusted certificate, the WINCC OPC UA server does not allow secure communication and copies the client certificate to the folder for rejected certificates.

You specify storage settings for rejected certificates using the WINCC OPC UA server configuration file, for example



```
<RejectedCertificatesStore>
  <StoreType>Directory</StoreType>
  <StorePath>[ApplicationPath]\PKI\OPCUA\rejected</StorePath>
</RejectedCertificatesStore>
```

Note

Here too, only the StoreType "Directory" is supported.

To enable secured communication with this client, you will have to move the rejected certificate to the certificate store for trusted certificates.

See also

Setting up authentication via certificates. (Page 303)

Configuration file of the WinCC OPC UA Server (Page 769)

Establishing a trust relationship (Page 175)

9.9.3 Configuring the security mechanisms

Introduction

The following is ensured at the communication level:

- UA application authenticity
- The confidentiality of messages exchanged
- The integrity of messages exchanged

The security mechanisms used, for example algorithms for encrypting and signing, are defined by standardized security policies.

The security policies supported by the WinCC OPC UA server are set using the server configuration file in "ServerConfiguration" and "SecuredApplication".

ServerConfiguration

The XML element "SecurityPolicies" under "ServerConfiguration" contains the list of all available "Security Profile" and "Message Security Mode" combinations for the server.

Security Profile	Message Security Mode	Description
http://opcfoundation.org/UA/SecurityPolicy#None	None	Unsecured communication
http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15	Sign or SignAndEncrypt	Secure communication, signed or encrypted and signed messages
http://opcfoundation.org/UA/SecurityPolicy#Basic256	Sign or SignAndEncrypt	Secure communication, signed or encrypted and signed messages
http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256 [1)]	Sign or SignAndEncrypt	Secure communication, signed or encrypted and signed messages

Security Profile	Message Security Mode	Description
http://opcfoundation.org/UA/SecurityPolicy#Aes128_Sha256_RsaOaep [1)]	Sign or SignAndEncrypt	Secure communication, signed or encrypted and signed messages
http://opcfoundation.org/UA/SecurityPolicy#Aes256_Sha256_RsaPss [1)]	Sign or SignAndEncrypt	Secure communication, signed or encrypted and signed messages

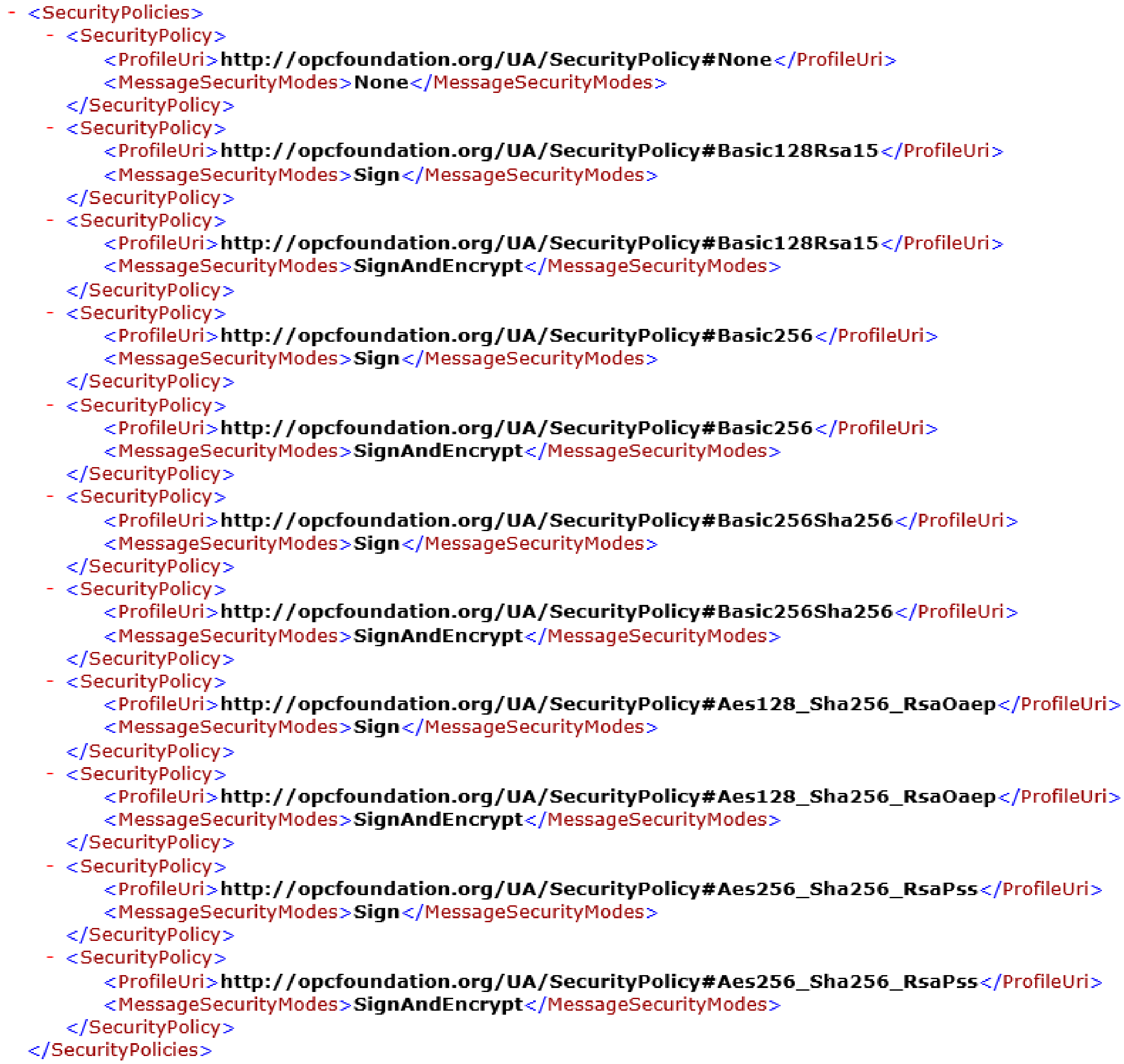
1) Requirement for the Use of Security Polices "Basic256Sha256", "Aes128_Sha256_RsaOaep" and "Aes256_Sha256_RsaPss": Instance certificate with signature algorithm "Sha256" and minimum key length = 2048.

Note

Ensuring secure communication

Secure communication requires server certificates for server and client and a correctly configured certificate store.

Example of a configuration file with maximum functional scope

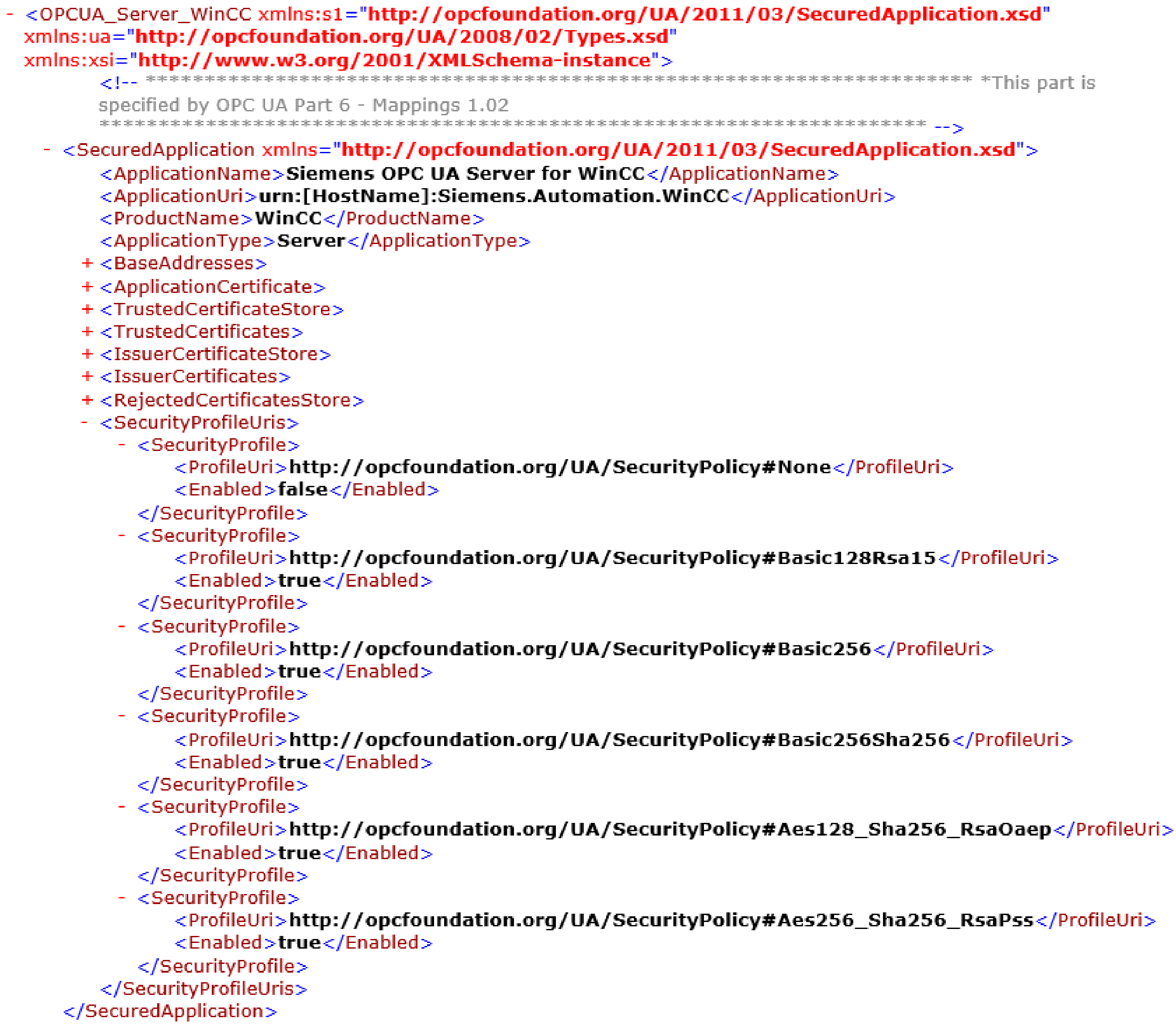


```
- <SecurityPolicies>
  - <SecurityPolicy>
      <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#None</ProfileUri>
      <MessageSecurityModes>None</MessageSecurityModes>
    </SecurityPolicy>
  - <SecurityPolicy>
      <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15</ProfileUri>
      <MessageSecurityModes>Sign</MessageSecurityModes>
    </SecurityPolicy>
  - <SecurityPolicy>
      <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15</ProfileUri>
      <MessageSecurityModes>SignAndEncrypt</MessageSecurityModes>
    </SecurityPolicy>
  - <SecurityPolicy>
      <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic256</ProfileUri>
      <MessageSecurityModes>Sign</MessageSecurityModes>
    </SecurityPolicy>
  - <SecurityPolicy>
      <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic256</ProfileUri>
      <MessageSecurityModes>SignAndEncrypt</MessageSecurityModes>
    </SecurityPolicy>
  - <SecurityPolicy>
      <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256</ProfileUri>
      <MessageSecurityModes>Sign</MessageSecurityModes>
    </SecurityPolicy>
  - <SecurityPolicy>
      <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256</ProfileUri>
      <MessageSecurityModes>SignAndEncrypt</MessageSecurityModes>
    </SecurityPolicy>
  - <SecurityPolicy>
      <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Aes128_Sha256_RsaOaep</ProfileUri>
      <MessageSecurityModes>Sign</MessageSecurityModes>
    </SecurityPolicy>
  - <SecurityPolicy>
      <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Aes128_Sha256_RsaOaep</ProfileUri>
      <MessageSecurityModes>SignAndEncrypt</MessageSecurityModes>
    </SecurityPolicy>
  - <SecurityPolicy>
      <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Aes256_Sha256_RsaPss</ProfileUri>
      <MessageSecurityModes>Sign</MessageSecurityModes>
    </SecurityPolicy>
  - <SecurityPolicy>
      <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Aes256_Sha256_RsaPss</ProfileUri>
      <MessageSecurityModes>SignAndEncrypt</MessageSecurityModes>
    </SecurityPolicy>
  </SecurityPolicies>
```

SecuredApplication

In accordance with the OPC UA specification, the security mechanisms and explicitly enabled and disabled with the "SecurityProfileUris" element under "SecuredApplication".

The diagram below shows a SecuredApplication in which unsecured communication is disabled:



```
- <OPCUA_Server_WinCC xmlns:s1="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd"
  xmlns:ua="http://opcfoundation.org/UA/2008/02/Types.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <!-- ***************************************************************** *This part is
        specified by OPC UA Part 6 - Mappings 1.02
        *************************************************************** -->
  - <SecuredApplication xmlns="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
        <ApplicationName>Siemens OPC UA Server for WinCC</ApplicationName>
        <ApplicationUri>urn:[HostName]:Siemens.Automation.WinCC</ApplicationUri>
        <ProductName>WinCC</ProductName>
        <ApplicationType>Server</ApplicationType>
      + <BaseAddresses>
      + <ApplicationCertificate>
      + <TrustedCertificateStore>
      + <TrustedCertificates>
      + <IssuerCertificateStore>
      + <IssuerCertificates>
      + <RejectedCertificatesStore>
      - <SecurityProfileUris>
        - <SecurityProfile>
            <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#None</ProfileUri>
            <Enabled>false</Enabled>
          </SecurityProfile>
        - <SecurityProfile>
            <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15</ProfileUri>
            <Enabled>true</Enabled>
          </SecurityProfile>
        - <SecurityProfile>
            <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic256</ProfileUri>
            <Enabled>true</Enabled>
          </SecurityProfile>
        - <SecurityProfile>
            <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256</ProfileUri>
            <Enabled>true</Enabled>
          </SecurityProfile>
        - <SecurityProfile>
            <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Aes128_Sha256_RsaOaep</ProfileUri>
            <Enabled>true</Enabled>
          </SecurityProfile>
        - <SecurityProfile>
            <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Aes256_Sha256_RsaPss</ProfileUri>
            <Enabled>true</Enabled>
          </SecurityProfile>
        </SecurityProfileUris>
    </SecuredApplication>
```

The WinCC OPC UA server therefore supports the following security strategies at runtime:

- "Basic128Rsa15"
- "Basic256"
- "Basic256Sha256"
- "Aes128_Sha256_RsaOaep"
- "Aes256_Sha256_RsaPss"

With "Message Security Modes Sign" and "SignAndEncrypt", but not unsecured communication.

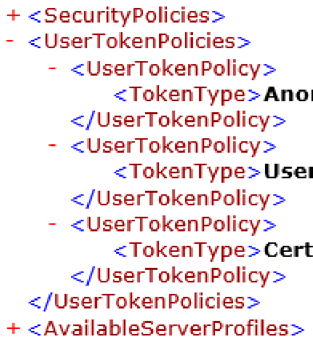
When communication is established, the UA clients select the required Policy from this list.

User identity

In addition to the security mechanisms of the communication level, the WinCC OPC UA server also supports user authentication for the client applications using the UserTokenPolicy "UserName" and "Certificate".

The client application must provide a valid combination of user certificate and private key when communication is established. The WinCC OPC UA server checks the combination and allows communication if the user certificate is trusted.

The UserTokenPolicy is set in the configuration file of the WinCC OPC UA server.



```
+ <SecurityPolicies>
- <UserTokenPolicies>
  - <UserTokenPolicy>
      <TokenType>Anonymous</TokenType>
    </UserTokenPolicy>
  - <UserTokenPolicy>
      <TokenType>UserName</TokenType>
    </UserTokenPolicy>
  - <UserTokenPolicy>
      <TokenType>Certificate</TokenType>
    </UserTokenPolicy>
  </UserTokenPolicies>
+ <AvailableServerProfiles>
```

With this configuration, the WinCC OPC UA server supports both anonymous users and the Policy "UserName" and "Certificate".

9.9.4 Supported OPC UA services and profiles

OPC UA services

The WinCC OPC A&E Server supports the following described functionality.

The following table summarizes the functionality supported by the OPC UA server 1.0.10:

OPC UA Service Sets	Services	Comment
Discovery Service Set	FindServers GetEndpoints	-
Secure Channel Service Session Service Set	All	-
View Service Set	Browse BrowseNext RegisterNodes UnregisterNodes	Determination of the mapped WinCC data: Process values and archived data
Attribute Service Set	Read Write HistoryRead HistoryUpdate*)	only WinCC tags only WinCC tags Only archived tags Only archived tags

OPC UA Service Sets	Services	Comment
Subscription Service Set	CreateSubscription SetPublishingMode Publish RePublish DeleteSubscription	
MonitoredItem Service Set	CreateMonitoredItems SetMonitoringMode DeleteMonitoredItems	Only "Value" attribute of WinCC tags Event Notifier during access to WinCC messages
Method Service Set	Call	Acknowledge ConditionRefresh
*): With restrictions, see "Supported Write-Accesses (Page 713)"		

OPC UA profile and Conformance Units

The WinCC OPC UA server supports the following OPC UA profiles 1.04 without restrictions:

- 6.6.6 Base Server Behavior Facet
- 6.6.24 Standard Event Subscription Server Facet
- 6.6.26 A & C Base Condition Server Facet
- 6.6.39 Method Server Facet
- 6.6. 48 Historical Raw Data Server Facet
- 6.6.53 Historical Data Insert Server Facet
- 6.6.54 Historical Data Update Server Facet
- 6.6.56 Historical Data Delete Server Facet
- 6.6.143 UA-TCP UA-SC UA Binary
- 6.6.161 SecurityPolicy - None
- 6.6.162 SecurityPolicy - Basic128Rsa15
- 6.6.163 SecurityPolicy - Basic256 #
- 6.6.164 SecurityPolicy [A] - Aes128-Sha256-RsaOaep
- 6.5.165 SecurityPolicy [B] - Basic256Sha256
- 6.6.166 SecurityPolicy - Aes256-Sha256-RsaPss
- 6.6.167 User Token – Anonymous Facet
- 6.6.168 User Token – User Name Password Server Facet
- 6.6.169 User Token – X509 Certificate Server Facet

The WinCC OPC A&E Server supports the following OPC UA profiles shown in the following table, however with restrictions:

Profile	"Group"	Not supported "Conformance Unit"
6.6.2 Core Server Facet	Attribute Services	Attribute Write Index
6.6.16 Standard DataChange Subscription Server Facet	Monitored Item Services	DeadBand Filter
6.6.18 Enhanced DataChange Subscription Server Facet	Monitored Item Services	-
6.6.21 Data Access Server Facet	Data Access	Data Access Analog Data Access Multistate Data Access PercentDeadBand Data Access Semantic Changes Data Access Two State
6.6.70 Standard UA Server Profile	Attribute Services	Attribute Write StatusCode & Timestamp

9.9.5 Name area of the WinCC OPC UA server

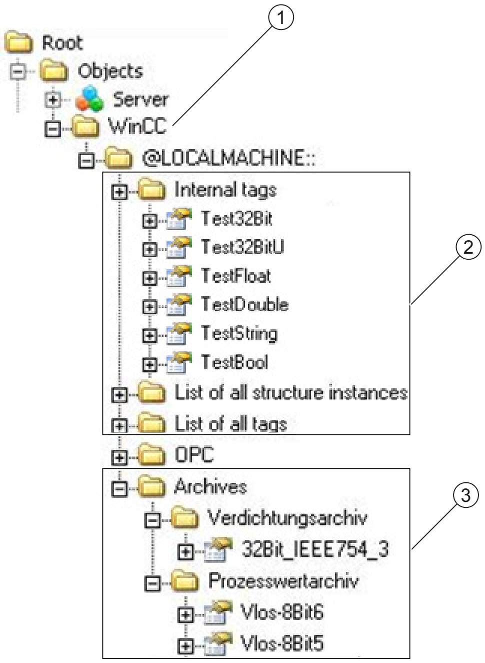
Introduction

The WinCC OPC UA server provides OPC UA clients with a hierarchical name area and access to the following runtime data:

- Process values (WinCC tags and WinCC tag groups)
- Data log inclusive logging tags
- WinCC messages

The name area of the WinCC OPC UA server is attached in the "Objects" default folder.

The following screen shows the name area of the WinCC OPC UA server of an active WinCC project on the local PC ("@LOCALMACHINE::"):



① Start node of the specific name area of WinCC.

② Display of the WinCC tags; the structure corresponds to the structure of the tags in WinCC.

③ Display of the data log

Display of the WinCC tags

Tag groups, communication drivers and connections are displayed by OPC UA objects of the "FolderType" type. Each of these folders has references of the "Organizes" type to the subordinate objects and tags.

Internal and external WinCC tags are displayed by OPC UA tags of the "DataItemType" type. If a WinCC tag is additionally logged, the displayed OPC UA tag has additionally a reference of the "HasHistoricalConfiguration" type for a log configuration. The "Historizing" and "AccessLevel" attributes are respectively set.

The following table shows the most important attributes of the OPC UA tags that represent a WinCC tag. You can find the complete list of attributes in the "OPC UA Part 3 - Address Space Model 1.03 Specification" document under "§5.6":

Attribute	Description	Comment
NodeId	Unique designation of the WinCC tag	-
BrowseName	WinCC tag name	-
DisplayName	WinCC tag name	-
Value	Tag value and status	-
DataType	OPC UA data type that corresponds to the WinCC tag type, for example: • Int32; signed 32 bit value • UInt32; unsigned 32 bit value	-

Attribute	Description	Comment
AccessLevel	"CurrentRead" / "CurrentWrite" "HistoryRead" / "HistoryWrite"	correspondingly to the WinCC tag configuration
ValueRank	Always "Scalar"	-

Write protection and read protection

You can protect the WinCC OPC UA server tags against access by clients.

In the Tag Management of the WinCC project, you activate the following setting in the property area of the tags in the "Options" group:

Property	Behavior in Runtime
OPC write protection	Clients only have read access to the tag value.
OPC read protection	Clients can neither read nor write the tag value.

No mapping of WinCC structure types

WinCC structures cannot be mapped as types on the OPC UA server.

You can only link OPC UA types with WinCC structure tags.

Display of the logging tags

Process values and compressed logs are displayed by OPC UA objects of the "FolderType" type. Each of these folders has references of the "Organizes" type to the related logging tags.

Logging tags from process value or compressed logs are displayed by OPC UA tags of the "BaseDateVariableType" type. A logging tag always has a reference of the "HasHistoricalConfiguration" type for a log configuration.

The following table shows the most important attributes of the OPC UA tags that represent a WinCC logging tag. You can find the complete list of attributes in the "OPC UA Part 3 - Address Space Model 1.03 Specification" document under "§5.6":

Attribute	Description	Comment
NodeId	Unique designation of a logging tag	-
BrowseName	Name of the archive tag	-
DisplayName	Name of the archive tag	-
Description	Node description	-
Value	Not available	For a logging tag, this attribute cannot be read nor changed.
DataType	OPC UA data type that corresponds to the WinCC tag type, for example: • Double; 64-bit floating-point number • UInt32; unsigned 32 bit value	-
AccessLevel	"HistoryRead" / "HistoryWrite"	-
ValueRank	Always "Scalar"	-

Access to WinCC messages

The start node of the WinCC namespace is an Event Notifier with which the OPC UA clients can receive status changes for WinCC messages in Runtime via Subscriptions.

9.9.6 OPC UA Data Access

Internal and external WinCC tags are displayed by OPC UA tags of the "DataItemType" type. Other DataAccess tag types as "AnalogItem" or "DiscreteType" are not supported.

The WinCC OPC A&E Server supports the reading access on the OPC UA tag attributes as "DataType" or "AccessLevel". Writing access and subscriptions are only supported for the "Value" attribute.

9.9.7 OPC UA Historical Access

Introduction

"OPC Historical Access" enables access to archives and includes the "Historical Data" and "Historical Events" services. The WinCC OPC UA server supports only the "Historical Data" service.

The WinCC OPC UA Server offers the OPC clients access to the raw data of tag archives via "Services".

- HistoryRead (READRAW)
- HistoryUpdate (INSERTDATA, REPLACEDATA, UPDATEDATA, DELETE_RAW)

You can read and limitedly write with an OPC UA client the values of archive tags in the tag archives. Depending on the configuration of the tag archive, the archive tag can contain either raw data or already processed process values.

Characteristics of archive tags

A process tag in WinCC can be located in multiple tag archives. In this case the process tag is linked to one of the corresponding archive tags.

Properties / Properties of archive configurations

The following table shows the Properties of an OPC UA tag configuration of the "HistoricalConfigurationType" type: In the "Description" property, the archive tag comment configured in WinCC is displayed. You can find the complete list of properties in the "OPC UA Part 11 - Historical Access 1.03 Specification" document under "§5.2.2":

Property	Description / Value	Comment
Definition	WinCC process tag name	For a process value archive
Stepped	True	-

The following optional Properties are not supported:

- MaxTimeInterval
- MinTimeInterval
- ExceptionDeviation
- ExceptionDeviationFormat

Limitations for Service "HistoryUpdate"

You can use the Service "HistoryUpdate" only on process value archives.

The following table lists the functions supported by the WinCC OPC UA server: Which functions are supported depends on the configuration of the WinCC OPC UA server as well as the process value archive configuration. You will find additional information in the "OPC UA Part 11 - Historical Access 1.03 Specification" document under "§5.5":

Service	Function	Description
HistoryUpdate	INSERTDATA	Insert new archive values
	REPLACEDATA	Replace existing archive values
	UPDATEDATA	Replace of insert archive values
	DELETE_RAW	Delete archive values

9.9.8 OPC UA Alarm & Conditions

Introduction

As of WinCC 7.3, the OPC UA server provides access to the messages of the WinCC message system.

The OPC UA server forwards WinCC message status changes to OPC UA clients with WinCC-Event-Notifications via Subscriptions and Monitored Event Items but does not maintain a Condition instance in its name space.

The Event Notifier node to be used is the start node of the WinCC name area.

The UA client can filter the messages and define the list of message attributes returned.

The OPC UA server supports the "OPC UA Alarms & Conditions 1.03" specification.

The following section outlines the mapping of the WinCC message system to OPC UA. You can find additional information in the specification in "Part 9: Alarms and Conditions 1.03 Specification".

WinCC message system mapping to OPC UA event types

BaseEventType

"BaseEventType" is the basic type from which the OPC UA Event types "WinCCEventType" and "WinCCAlarmConditionType" are derived.

Note

Filter shows all WinCC messages

When you filter for "BaseEventType", you receive all WinCC messages.

WinCC messages are mapped to the following OPC UA event types:

WinCCEventType

This type is based on "BaseEventType" and maps "simple" WinCC messages with the following acknowledgment theory:

- "Message without status went out" is activated
- "Acknowledgment came in" is not activated

Examples of this type of message are starting and stopping motors**.**

WinCCAlarmConditionType

This type is based on "AlarmConditionType" and maps all messages which cannot be mapped on "WinCCEventType", for example, acknowledgeable messages and messages with the status "came in" and "went out".

At a message of the "WinCCAlarmConditionType" type, the event is linked to a condition. For example, WinCC generates a message as soon as a tag limit is violated. This message in OPC UA is equivalent to an Alarm Condition.

WinCC message attributes

The two Event types add WinCC-specific message attributes to the basic type. The attributes are mapped 1:1 as UA Event Properties and are described in more detail in the section "Attributes of the WinCC message system".

Message class and message type

The WinCC message system informs the user of disturbances and operating conditions in the process. A WinCC message always belongs to a specific message class and message type, which are specified in the "CLASSID", "TYPEID", "CLASSNAME" and "TYPENAME" attributes of the corresponding UA Events.

Priority

When configuring messages in the WinCC message system, you can configure a priority between "0" and "16". The OPC UA specification defines a value range of "1" to "1000" for the Severity. "1" stands for the lowest and "1000" for the highest Severity.

The values of the priority must therefore be suitably mapped to the OPC severity. In standard mapping, a priority of "0" is assigned to OPC-Severity "1" and a priority of "16" to OPC-Severity "1000". All other values are interpolated linearly between "0" and "1000".

OPC UA mapping rules

During the configuration of the WinCC message system, settings are made to determine which process events generate a message. This message is generally shown as an Event in OPC UA.

The following table shows the most important Properties of Events and how the WinCC message system provides the information.

OPC UA property	Mapping in the WinCC message system
For all event types:	
EventID	Unique message designation
EventType	Event type: Node ID of the WinCCAlarmConditionType node or WinCCEventType node
SourceNode	Not relevant
SourceName	Indicates the source of the message. Mapping is described in more detail below.
Message	Message text for the corresponding message number.
Time	Time of the event. The time stamp is given in UTC
Severity	Priority of the WinCC message
Only with WinCCAlarmConditionType:	
ConditionName	Set text that is output as well as the message. The text output depends on the mapping rule set: • "Mode 1" and "Mode 2": Message number • "Mode 3": Message class, for example "Process control message"
Quality	Returns the quality of the message
ConditionClassId	Node ID of the "ProcessConditionClassType" node
ConditionClassName	"ProcessConditionClassType"
Retain	"TRUE" with pending messages
NodeId	ConditionId: Designates a UA-Condition uniquely, for example an alarm. Required for acknowledgment, even if no Condition instances are supported
EnabledState	"TRUE" if the message has been enabled
ActiveState/Id	"TRUE" if the message has come in
AckedState/Id	"TRUE" if the message has been acknowledged
ClientUserId	Indicates the user that is logged on

Note

The following OPC UA Condition and Alarm Properties are not supported by the OPC UA server:

- BranchId
- LastSeverity
- InputNode
- ConfirmedState
- SuppressedState
- ShelvingState
- SuppressedOrShelved
- MaxTimeShelved

Message statuses / acknowledgment statuses

The following table shows WinCC message status mapping to the corresponding WinCCAlarmConditionType - Properties:

Message status	EnabledState/Id	ActiveState/Id	AckedState/Id
Locked message	FALSE	-	-
Enabled message	TRUE		
Received message	TRUE	TRUE	FALSE
Sent message with acknowledgment	TRUE	FALSE	TRUE
Sent message without acknowledgment	TRUE	FALSE	FALSE
Acknowledged messages (message pending)	TRUE	TRUE	TRUE
Acknowledged messages (message no longer pending)	TRUE	FALSE	TRUE
Received, acknowledged message	TRUE	TRUE	TRUE
Received, sent message with acknowledgment	TRUE	FALSE	TRUE
Received, sent message without acknowledgment	TRUE	FALSE	FALSE
Message acknowledged by the system (message pending)	TRUE	TRUE	TRUE
Message acknowledged by the system (message no longer pending)	TRUE	FALSE	TRUE
Emergency-acknowledged message (message pending)	TRUE	TRUE	TRUE
Emergency-acknowledged message (message no longer pending)	TRUE	FALSE	TRUE

Settings for mapping the WinCC message system

The configuration of the OPC UA server also applies to the OPC UA server as regards the mapping of the Properties "SourceName" and "Message" of a message.

- With OPC A&E server with hierarchical access:

SourceName	Indicates the source of a message. The Source has the format "<Server prefix>::Area\UserTextBlock 2". The server prefix of the local computer is "@LOCALMACHINE".
Message	Returns the message text of the corresponding message number

- With OPC A&E server without hierarchical access:

SourceName	Indicates the source of a message. The Source has the format "<Server prefix>::localhost::". The server prefix of the local computer is "@LOCALMACHINE".
Message	Returns the message text of the corresponding message number

Alarm groups

In WinCC 7.3, the WinCC alarm groups are not displayed in the name area.

Supported event methods

Acknowledgment

A WinCC message is acknowledged using the "Acknowledge" method of the "AcknowledgeableConditionType" node in the standard OPC UA info model.

Only messages of the "WinCCAlarmConditionType" type can be acknowledged.

ConditionRefresh

Messages still pending are established using the "ConditionRefresh" method of the "ConditionType" node in the standard OPC UA info model.

Filters

The OPC UA client can defined a filter for Monitored Event Items .

The following operators are, however, not supported by the OPC UA server:

- FilterOperator_Cast
- FilterOperator_BitwiseAnd
- FilterOperator_BitwiseOr
- FilterOperator_RelatedTo
- FilterOperator_InView

See also

Attributes of the WinCC message system (Page 767)

9.9.9 Attributes of the WinCC message system

Overview

The following table lists the configurable attributes of the WinCC message system. The attributes are mapped 1:1 as UA Event Properties .

WinCC message attribute	Meaning	Data type
CLASSNAME	Name of message class	String
TYPENAME	Name of message type	String
FORECOLOR	Foreground color for incoming, outgoing and acknowledged messages.	Int32
BACKCOLOR	Background color for incoming, outgoing and acknowledged messages.	Int32
FLASHCOLOR	Flash color	Int32
FLAGS	Indicates whether the message requires acknowledgment.	Int32
TEXT01...TEXT10	Content of user text block #1....#10	String
PROCESSVALUE01...PROCESSVALUE10	Content of process value block #1....#10	
STATETEXT	Status message	String
INFOTEXT	Information text for the message	String
LOOPINALARM	Indicates whether LoopInAlarm was configured	Int32
CLASSID	Message class ID	Int32
TYPEID	Message type ID	Int32
MODIFYSTATE	Value of message status tag	Int32
AGNR	Outputs the number of the automation system that generated the message	Int16
CPUNR	Outputs the number of the CPU that generated the message	Int16
DURATION	Outputs the time period between the incoming state, outgoing state and acknowledgment of a message	Int32
COUNTER	Number of messages after the start of runtime	Int32
QUITSTATETEXT	Indicates whether the message has been acknowledged	String
QUITCOUNT	Number of open, unacknowledged messages	Int32
PARAMETER	Configuration parameter of the message	Int32
BLOCKINFO	Current content of the message block	String
ALARMCOUNT	Number of pending messages	Int32
LOCKCOUNT	Number of locked messages	Int32
PRIORITY	Priority of the message	Int32
APPLICATION	Outputs the application which triggered the message	String
COMPUTER	Outputs the name of the computer which processed the message	String

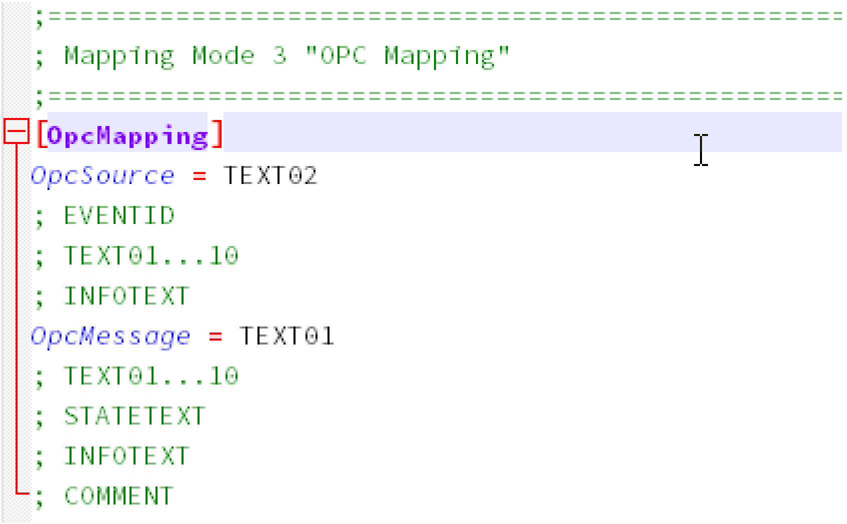
WinCC message attribute	Meaning	Data type
USER	Outputs the name of the user who processed the message	String
COMMENT	Message comment	String
HIDDEN_COUNT	Number of hidden messages	Int32
OS_HIDDEN	Indicates that the message is hidden	Boolean
OS_EVENTID	WinCC message number	Int32
BIG_COUNTER	Number of messages after the start of Runtime	Int64

Configuration for special WinCC message attributes

Mapping of the WinCC message system on OPC is configured via the "CcAeProvider.ini" file. The configuration file contains three different mapping modes. "Mapping Mode 1" is activated by default. In "Mapping Mode 3" the special message attributes BIG_COUNTER, HIDDEN_COUNT, OS_EVENTID and OS_HIDDEN are additionally active.

Proceed as follows to activate "Mapping Mode 3":

1. Use a text editor to open the file "CcAeProvider.ini" in the WinCC installation path in the folder "OPC\AlarmEvent\bin".
2. In the section "Mapping Mode 3", remove the comment markings at the beginning of the lines "[OpcMapping]", "OpcSource ..." and "OpcMessage ...":



```
;=======================================================
; Mapping Mode 3 "OPC Mapping"
;=======================================================
[OpcMapping]
OpcSource = TEXT02
; EVENTID
; TEXT01...10
; INFOTEXT
OpcMessage = TEXT01
; TEXT01...10
; STATETEXT
; INFOTEXT
; COMMENT
```

3. If required, modify the mapping of the message attributes in the section.
4. Restart Runtime.

The "Mapping Mode 3" of the WinCC OPC UA server with its special message attributes is activated.

See also

OPC UA Alarm & Conditions (Page 762)

9.9.10 Configuration of the WinCC OPC UA server

9.9.10.1 Configuration file of the WinCC OPC UA Server

Introduction

The WinCC OPC UA server is configured using the configuration file "OPCUAServerWinCC.xml".

The configuration file is broken down into multiple sections. This section describes the layout of the configuration file.

The chapter "How to configure the OPC UA server (Page 773)" describes how you configure the WinCC OPC UA server.

Path of the configuration file

Two configuration files "OPCUAServerWinCC.xml" exist for the WinCC OPC UA server:

Configuration file	Storage path
Server-specific configuration file	<WinCC installation path>\opc\UAServer\
Project-specific configuration file	<WinCC project folder>\OPC\UAServer

Editing the configuration file

You require the following authorizations to carry out changes in the configuration files:

Server-specific configuration file	Windows Administrator rights
Project-specific configuration file	The user must be a member of the "SIMATIC HMI" user group.

Note

Same parameters: Priority of the files

Some parameters are contained in both configuration files.

If the parameters do not match, the settings of the project-specific configuration file have a higher priority.

Structure: Section <SecuredApplication>

In this section, the OPC UA application security is set in compliance with OPC UA Specification / Part 6 / § "Security Settings Management".

You can find additional information on the URL under "Security concept of OPC UA (Page 748)".

`<SecuredApplication>`	

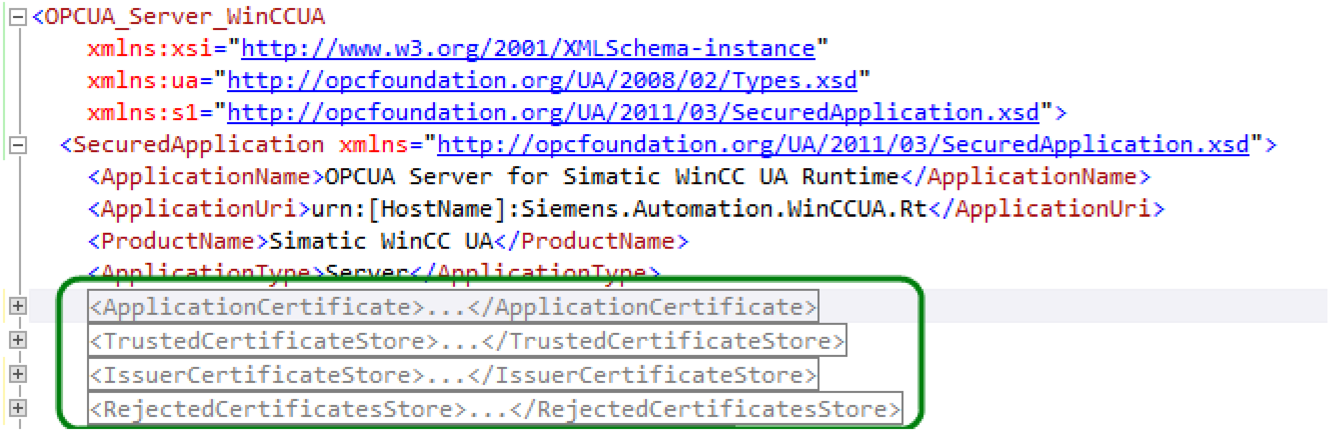
`<BaseAddresses>`
`  <...></...>`
`</BaseAddresses>`
	Configuration of the URL of the WinCC OPC UA server.

`<SecurityProfileUris>`
`  <SecurityProfile>`
`    <...></...>`
`  </SecurityProfile>`
`    ...`
`</SecurityProfileUris>`
	Configuration of the supported security policies
	Use the "none" setting only for test and diagnostics purposes

`<ApplicationCertificate>`
`<TrustedCertificateStore>`
`<TrustedCertificates>`
`<...>`
	Revision of the default certificate configuration according to OPC UA Specification / Part 6.
	(optional)
	These parameters are only contained in the server-specific configuration file.

`</SecuredApplication>`	

Example: OPC UA application security



```
<OPCUA_Server_WinCCUA
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:ua="http://opcfoundation.org/UA/2008/02/Types.xsd"
    xmlns:s1="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
  <SecuredApplication xmlns="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
    <ApplicationName>OPCUA Server for Simatic WinCC UA Runtime</ApplicationName>
    <ApplicationUri>urn:[HostName]:Siemens.Automation.WinCCUA.Rt</ApplicationUri>
    <ProductName>Simatic WinCC UA</ProductName>
    <ApplicationType>Server</ApplicationType>
    <ApplicationCertificate>...</ApplicationCertificate>
    <TrustedCertificateStore>...</TrustedCertificateStore>
    <IssuerCertificateStore>...</IssuerCertificateStore>
    <RejectedCertificatesStore>...</RejectedCertificatesStore>
```

Structure: Section <ServerConfiguration>

Server-specific parameters are set in this section.

For more information about message security modes, refer to "Security concept of OPC UA (Page 748)".

<ServerConfiguration>	

<SecurityPolicies>
  <SecurityPolicy>
  <...></...>
  </SecurityPolicy>
  ...
</SecurityPolicies>
	Configuration of the message security modes.
	Use the "none" setting only for test and diagnostics purposes

<UserTokenPolicies>
  <UserTokenPolicy>
  <...></...>
  </UserTokenPolicy>
  ...
</UserTokenPolicies>
	Configuration of user identification
	Use the "Anonymous" setting only for test and diagnostics purposes

<FastInsert>
  <Users>
    <...></...>
  </Users>
  <Clients>
    <...></...>
  <Clients>
</FastInsert>
	Configuration of the optimized WinCC archive write access

</ServerConfiguration>	

Structure: Section <CertificateDescriptor>

You specify the certificate parameters for the WinCC OPC UA server under the <CertificateDescriptor> heading in the <ServerConfiguration> section.

These parameters are only contained in the server-specific configuration file.

You can find additional information on the instance certificates under "Security concept of OPC UA (Page 748)".

<ServerConfiguration> <CertificateDescriptor>	
<OrganizationUnit>...</...> <Organization>...</...> <Country>...</...>	Descriptive elements The parameters can be changed and have no effect on the function of the applications.
<KeyLength>...</...>	Length of the private key with which the certificate is created The length depends on the signature algorithm. • 1024: Minimum length for secure communication via OPC UA • 2048: Minimum length when Sha256 is used [1)]

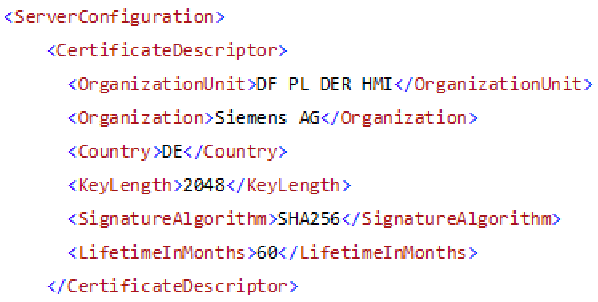
`<SignatureAlgorithm>...</...>`	Signature algorithm used to sign the certificate • Possible values: Sha1, Sha224, Sha256, Sha384, Sha512 • Usual values: Sha1, Sha256 • Default value: Sha256 with key length 2048 [1]
`<LifetimeInMonths>...</...>`	Validity period of the certificate in months After the specified time has expired, the server can no longer be operated with this certificate. • Default value: 60

`</CertificateDescriptor>`
`</ServerConfiguration>`	

[1] To establish a secure connection 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

Example: Parameters for the control of the certificate



```
<ServerConfiguration>
    <CertificateDescriptor>
      <OrganizationUnit>DF PL DER HMI</OrganizationUnit>
      <Organization>Siemens AG</Organization>
      <Country>DE</Country>
      <KeyLength>2048</KeyLength>
      <SignatureAlgorithm>SHA256</SignatureAlgorithm>
      <LifetimeInMonths>60</LifetimeInMonths>
    </CertificateDescriptor>
```

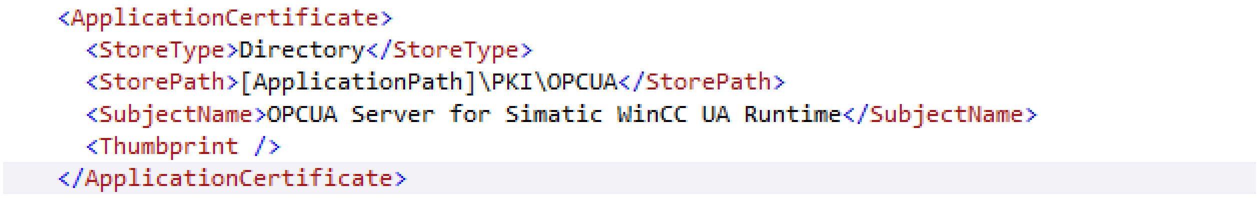
Changing the storage path of the server certificate

If required, the storage location for the certificate of the WinCC OPC UA server can be adapted by the plant administration.

You can change these parameters only in the server-specific configuration file.

Parameter	Value	Meaning
StoreType	Directory	Type of certificate storage. The storage location must be "Directory".
StorePath	[ApplicationPath]\PKI\WINCC-OPC-UA-Server	The certificate and the private key are stored under this folder.

Example: Storage path of the server certificate



```
<ApplicationCertificate>
  <StoreType>Directory</StoreType>
  <StorePath>[ApplicationPath]\PKI\OPCUA</StorePath>
  <SubjectName>OPCUA Server for Simatic WinCC UA Runtime</SubjectName>
  <Thumbprint />
</ApplicationCertificate>
```

Creating new server certificates

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

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

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

9.9.10.2 How to configure the OPC UA server

Requirement

A WinCC project·has been created.

Opening the configuration file

1. Open Windows Explorer.·Navigate to the directory "<WinCC project folder>OPC\UAServer".
2. Open the "OPCUAServerWinCC.xml" configuration file. For more information, refer to "Configuration file of the WinCC OPC UA Server (Page 769)"

Changing the port number of the WinCC OPC UA server

1. If necessary, change the port number 4862 under `<BaseAdresses>`.
   Do not use a port number that is already assigned to another application.
   The parameter [HostName] is the placeholder for the computer name and is determined during runtime.
   Example:
   ```
   <BaseAdresses>
   <ua:String>opc.tcp://[HostName]:5210</ua:String>
   <BaseAdresses>
   ```

Specifying security settings

1. Specify the security settings for communication.·For additional information, refer to "Security concept of OPC UA (Page 748)"
2. Under `<SecurityProfileUris>`, you configure the supported "Security Policies".
   - Enable the setting with "`true`".
   - Disable the setting with "`false`".
     Example:
     ```
     <SecurityProfile>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#None</
     ProfileUri>
        <Enabled>false</Enabled>
     </SecurityProfile>
     ```
3. Under `<SecurityPolicies>`, you configure the associated "Message·security·modes".
   To deactivate a setting, delete the entire entry `<SecurityPolicy>... </Security Policy>`.
   Example:
   ```
   <SecurityPolicy>
      <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#None</
   ProfileUri>
      <MessageSecurityModes>None</MessageSecurityModes>
   </SecurityPolicy>
   ```

Note

Unsecured communication between client and server

Use the "none" setting only for test and diagnostics purposes.

For secure client/server communication·in production mode, you need to use at least the following settings:

- SecurityPolicy:·Basic128Rsa15
  Message·Security·Mode:·Sign

Specifying user identification

1. Specify the user identification for setting up the connection under `<UserTokenPolicies>`. For more information, refer to "Security concept of OPC UA (Page 748)"
   To deactivate a setting, delete the entire entry.
   Example
   ```
   <UserTokenPolicy>
   <TokenType>Anonymous</TokenType>
   </UserTokenPolicy>
   ```

Configuring optimized WinCC archive write access

1. If necessary, configure optimized WinCC archive write access under `<FastInsert>`.
   - Set "`true`" to activate the optimized write access to WinCC archives for all OPC UA clients.
   - Set "`false`" to set optimized WinCC archive write access for specific Windows users or OPC UA clients.
     You specify the Windows users under `<Users>`.
     You specify the OPC UA clients under `<Clients>`. Use the "Common Name" that is entered in the client certificate as `ClientName`.
     Example:
     ```
     <EnabledByDefault>false</EnabledByDefault>
     <Users>
       <User>domain\user1</User>
     </Users>
     <Clients>
       <Client>ClientName1</Client>
     </Clients>
     ```

9.10 Diagnostics

Trace file

All servers offer the possibility to activate the output of diagnostic data for test purposes and for troubleshooting.

The data of a server is written to a trace file.

Setting

You specify the output of diagnostic data in the configuration file of the respective server.

For more information, refer to the SIMATIC Customer Support.

WinCC REST communication 10

10.1 REST interface in WinCC

WinCC REST service (REpresentational State Transfer)

The WinCC REST service is a self-hosted service for monitoring the port. To monitor the configured URL and the selected port, the service uses Microsoft C++ SDK.

Via REST communication, WinCC supports access to Runtime data and configuration data:

- WinCC supports secure REST communication via flexible authentication mechanisms, such as HTTPS.
- The **WinCC REST service** allows external applications to access WinCC data.
  External applications can read and write WinCC configuration data and tag values via the REST interface.
  The following operations are supported:
  - Querying data
  - Editing data

  For access, use the usual HTTP methods and JSON display of resources.
- With the **WinCC REST Connector** you can send Runtime data in the form of tag values and message texts to an external REST interface. You can configure the body of the REST calls variably according to the requirements of the external interface. The security settings allow for different authentication procedures.
- You can send Runtime data to a cloud in the form of tag values with the **WinCC/Cloud Connector**, via the REST protocol. The REST calls use basic authentication and a predefined body structure. More information:
  - "Interfaces > WinCC/Cloud Connector > Data transfer to the cloud via REST (Page 837)"

Restriction: Supported data types

The following data types are not supported when accessing tags:

- Raw data tag
- Text reference

API structure

To transmit WinCC runtime data and configuration data, WinCC uses HTTP requests and HTTP responses.

These requests and responses consist of the following elements:

Header	Metadata for the request
Body	• Data sent to the WinCC API as request body • Data sent by the WinCC API as response body
Service Endpoint	Service endpoint: Base URL with the network address of the WinCC API service

HTTP methods

Method	Description
GET	Reads a resource from the server. Used to read Runtime values and configuration data from the WinCC project.
POST	Reads a resource from the server. Addresses the resources in the Body. Used to send multiple values.
PUT	Write access Addresses a resource via the URL. Used to send an individual WinCC value to the cloud.

URL coding

The URL of the HTTP call has a specified structure. The structure and function of a URL are identified by the following reserved characters within the printout:

```
! # $ & ' ( ) * + , / : ; = ? @ [ ]
```

If these reserved characters are not to be given their special meaning within the URL, but are part of a name, for example, encode these characters.

Example

You want to read the Runtime value of the structure tag element "MyTag.[NewElement]".

To achieve this, you have to encode the characters "[" with "%5B" and "]" with "%5D" as follows in the URL:

```
https://<Host>:<Port>/WinCCRestService/tagManagement/Value/
MyTag.%5BNewElement%5D
```

You can find more information and a coding table in Wikipedia: URL encoding (https://en.wikipedia.org/wiki/URL_encoding).

Error code

Response code	Description	Comment
200	OK	The WinCC API has successfully processed the request.
400	Bad Request	Invalid request
401	Unauthorized	No access authorization

Response code	Description	Comment
402	Payment Required	No valid WinCC license found
404	Not Found	Resource not found The URL does not match the resource type of the WinCC API. Correct the format of the URL.
416	RangeNotSatisfiable	Invalid request The URL structure is incorrect. The requested range cannot be fulfilled. The reason is, for example, that the name of the addressed resource is missing when "GET" is called.
423	Locked	Resource is locked. The reason is, for example, that the addressed data is currently being read. Send the request again after a wait time.
500	InternalError	Internal error The server reports an unexpected error.
501	NotImplemented	Invalid request The server does not support the requested function.

See also

Data transfer to the cloud via REST (Page 837)

Overview of the methods (Page 788)

Filter for methods (Page 825)

REST settings in the WinCC Cloud Connector (Page 846)

How to configure the WinCC REST service (Page 780)

10.2 How to configure the WinCC REST service

In the "Computer" editor, you configure the connection to the REST server in the project properties.

To use secure communication via HTTPS, select the REST server and a certificate installed there.

WinCC users and authorizations

Tag Management: Authorizations for tag access

For each tag, you can limit reading or writing of values to specific authorizations in the Tag Management.

In the tag properties in the "REST API" group, select the desired authorization in the following fields:

- Read authorization
- Write authorization

Archive system: Authorization for archive tag access

For each archive tag, you can limit reading of values to specific authorizations in the "Tag Logging" editor.

In the tag properties in the "REST API" group, select the desired authorization in the "Read authorization" field.

User Administrator: Assigning authorizations

For access via WinCC as REST service, you create WinCC users in the User Administrator.

To access a tag, give the WinCC user the authorizations defined in the tag properties.

Procedure

1. Open the "Computer" editor in the WinCC Configuration Studio.
2. Select the "Project" entry in the navigation area.
3. To start the WinCC REST service, activate the "Switch on" option in the "Properties - Project" area under "REST settings".
4. If the application "REST Service" is not yet activated in the startup list of the computer, you will be prompted to activate it.
   Confirm the activation with "Yes".
   If the application is disabled, the REST service is not executed in WinCC Runtime.
5. Select the "Local settings" entry in the navigation area.

6. Specify the connection data under "REST settings" in the "Properties - Local settings" area.
   - **Host name**: Name of the WinCC server
   - **Port**: Port number used for access

   The path to the server is displayed in the "Service URL" field.

   Note

   The settings "Host name" and "Port" are saved in the local Windows registry and must be made on the computer on which the project is running in Runtime.

7. To set up a secure connection, click the "..." button in the "Certificate" field.
   The available certificates are displayed.
8. Select a certificate and confirm with "OK".

   Note

   Use the WinCC Certificate Manager (Windows program group "Siemens Automation" > "WinCC Certificate Manager") to create and install certificates system-wide. The certificates are subsequently available for selection in the "Computer" editor.

See also

REST interface in WinCC (Page 777)

Overview of the methods (Page 788)

10.3 How to configure the WinCC REST Connector

The WinCC REST Connector sends Runtime data via REST calls to any external REST interface. In contrast to the WinCC/Cloud Connector, the body of the call can be configured variably. In addition to basic authentication, other authentication methods are supported.

You configure the REST server and its endpoints in the "REST Connector" editor. For each endpoint you configure a trigger and the structure of the JSON body. The JSON body may contain placeholders for values to be sent.

To authorize yourself at an external REST interface, configure the supported authentication method of the server.

Configuring an external REST server

1. Activate the "REST Service" application in the startup list of the computer.
2. Open the "REST Connector" editor in WinCC Configuration Studio.
3. In the navigation area, right-click the "Server lists" entry and select "New server" in the shortcut menu.
   A REST server is created.
4. In the "Properties - REST server" area configure the connection data of the external REST server:
   - **Address**: IP address or computer name
   - **Port**: Port number to be used for access.
   - **Path**: BaseURL
   - **Security**: Authentication method used

   The "URL" field displays the path to the server.
5. To set up an authenticated connection, select the authentication method of the external server in the "Security" field.
   The respective fields for the authentication settings are activated.
6. Configure the authentication method.

Configuring the authentication method

To configure different authentication requirements of external REST interfaces flexibly, the WinCC REST Connector supports the following authentication methods:

No authentication

The external REST interface is freely available and requires no authentication. No authorization header is added to the REST call.

Basic authentication

The external REST interface supports the authentication with user name and password. The access data are specified on the external REST server.

1. Select the "Basic authentication" option in the "Security" field in the "Properties - REST server" area.
2. Enter the access data in the "User" and "Password" fields.

You can authenticate yourself to the external REST server using "Basic authentication".

Bearer token

The external REST interface supports authentication with a bearer token (bearer authentication). To obtain a bearer token, follow the instructions provided by the REST interface provider.

1. Select the option "Bearer token" in the "Security" field in the "Properties - REST server" area.
2. Copy the token into the "Bearer token" field or click the "..." button and import the token as a file.

You can authenticate yourself with the bearer token at the external REST server.

JSON Web Token

To authenticate yourself with a JSON Web Token (JWT) at the REST interface, you first have to create the token:

1. Select the "JSON Web Token" option in the "Security" field in the "Properties - REST server" area.
2. Click the "..." button in the "JSON Web Token" field.
   The window for configuring and creating the JWT is displayed.
3. Configure the following settings of the JWT in accordance with the specification of the REST interface provider:
   - **Token type**: For the format of the authorization header: <Token type> <Token>
   - **Lifetime**: Validity of the access token in seconds
   - **Payload**: Describes the JWT claims as key/value pairs.
   - **Exported certificate**: Certificate for signing the access token in the .pfx or .p12 file format
   - **Password**: Password of the certificate
4. Click "Create token".
   The JWT is created from the configuration data and displayed in the "JWT access token" field.
5. Click "OK" to apply the created JWT.

You can authenticate with the JWT at the external REST server. The JWT is automatically created again each time Runtime is started and after the lifetime has expired.

Authorization code

Use the "Authorization code" method to request a token via the OAuth2 authentication flow "Authorization Code" at the REST interface. In the process, a web page is shown where you authorize yourself with your user name and password.

For preparation create a new user in the external REST interface. In addition you require an OAuth2 client with client ID, client secret, lifetime of the access token, and return address to transfer the token.

To create a user and an OAuth2 client, follow the instructions provided by the REST interface provider. In addition the specifications on endpoints and scopes of the authentication flow of the REST interface are required.

1. Select the option "Authorization code" in the "Security" field in the "Properties - REST server" area.
2. In the "Authorization code" field, click the "..." button.
   The window for entering the authentication details is displayed.
3. Configure the following settings in accordance with the specification of the REST interface provider:
   - **Endpoint authorization**: Endpoint for the authorization by entering the user name and password in the displayed web page.
   - **Endpoint token**: Endpoint of the REST interface for the authorization. Returns the required access token.
   - **Redirect URL**: Return address which is handed over to the access token. Must contain "localhost".
   - **Scope**: Scopes which the REST service requires for authorization.
   - **Client Key**: Client ID of the OAuth2 client
   - **Client secret**: Client secret of the OAuth2 client
   - **With PKCE:** Activate additional safety mechanism with SHA256 hash function.
   - **Implicit grant**: Returns the token directly from the authorization endpoint without using the token endpoint.
   - **Basic authentication**: Use basic authentication instead of authorization code for the token endpoint.
   - **Bearer authentication**: Access token is handed over in the header instead of in the call.
4. Click "Request token".
   The standard web browser opens the page for authorization.
5. Specify the access data of the new user.
   The authorization endpoint transfers an authorization code and the OAuth2 client credentials to the token endpoint and requests a token.
   The token endpoint returns the access token. The "Access token", "Refresh token", "Token type", and "Valid until" fields display the values of the token.

   Note

   Not every REST service generates a refresh token. Refresh tokens automatically request access tokens without user interaction.

6. Click "OK" to accept the generated token.

You can authenticate yourself with the access token at the external REST server.

New access tokens are only automatically created in Runtime if a refresh token was assigned by the REST service. Without refresh tokens, the access token has to be requested again via the editor after the lifetime has expired.

Client credentials

You can use the "Client credentials" method to request a token via the OAuth2 authentication flow "Client Credentials" in the REST interface.

To prepare, create an OAuth2 client with client ID, client secret, and lifetime of the access token in the external REST interface. To create an OAuth2 client, follow the instructions provided by the REST interface provider. In addition, data on the specifications on endpoint and scopes of the authentication flow of the REST interface are required.

1. Select the option "Client credentials" in the "Security" field in the "Properties - REST server" area.
2. Click the "..." button in the "Client credentials" field.
   The window for entering the client credentials is displayed.
3. Configure the following settings in accordance with the specification of the REST interface provider:
   - **Endpoint authorization**: Endpoint of the REST interface for the authorization. Returns the required access token.
   - **Scope**: Scopes which the REST service requires for authorization.
   - **Client Key**: Client ID of the OAuth2 client
   - **Client secret**: Client secret of the OAuth2 client
4. Click "Request token".
   The authorization endpoint returns the access token. The values of the access token are displayed in the "Access token", "Token type", and "Valid until" fields.
5. Click "OK" to apply the created access token.

You can authenticate yourself via an access token at the external REST server. The access token is automatically recreated in Runtime after the lifetime has expired.

Configuring endpoints of the REST server

After you have configured an external REST server with base, you have to specify the individual endpoints of the REST interface and their message formats.

To create an endpoint, proceed as follows:

1. In the navigation area, right-click a configured server and select "New endpoint" in the shortcut menu.
   An endpoint of the REST server is created.
2. In the "Properties - Endpoint" area, configure the following general settings:
   - **Mode**: Type of HTTP method
   - **Trigger**: Specifies whether the REST calls are triggered by a tag or alarm trigger.
   - **Body**: Body of the REST call
   - **Body type**: Type of body. Structure and placeholder of a JSON body are modelled via the editor. In the "User-defined" setting the body is manually specified via the "Body" field.
   - **Content type**: User data format of the REST call
   - **Status**: Tag into which the status code of the REST server response is written.
   - **Response**: Tag into which the body of the REST server response is written.
3. To specify the trigger of the REST calls, proceed as follows:
   - For "Tag trigger": Specify the tags and the cycle in the "Tag trigger" tab. If a cycle is defined without a tag, the REST call is triggered cyclically in the specified time intervals.
   - For "Alarm trigger": In the "Alarm trigger" tab, specify the filters for alarms which trigger a REST call.
4. To create placeholders for values of the JSON body, open the "Placeholder" tab.
5. Create the required placeholders with the following settings:
   - **Type**: "Value", "Quality", "Time stamp", or "Text list" of tags. With an alarm trigger, the "Message text" is also available for selection.
   - **Tag name**: Tag which is used for the placeholder.
   - **Message block**: Only for "Message text" type: Message block of the alarm trigger from which the message text is taken.
   - **Text list**: Only "Text list" type: Name of a text list with texts for the tag value.
   - **Bit number**: Number of a bit from the text list.
6. To specify the structure of the JSON body, open the "JSON body" tab.
7. Add the individual elements of the JSON body in the desired sequence. In the "Data type" field, specify one of the following types for the elements:
   - Object
   - Array
   - Boolean value
   - Number
   - Character string
   - Null
8. Assign a placeholder or a statistical value to all elements of the file type "Boolean value", "Number", or "Character string" in the "Value" field.

9. Finally, check the result of the JSON body in "Properties - Endpoint" in the "Body" field.
10. Save the settings of the "REST Connector" editor.

Note

When saving a changed configuration the REST Connector is reinitialized in Runtime. Note that the tag triggers trigger and that triggers are not repeated during the initialization phase.

This completes the endpoint configuration. The REST calls of the endpoint are triggered by the trigger.

10.4 Overview of the methods

Tag management methods

Base URL

`https://<Host>:<Port>/WinCCRestService/tagManagement/`

- `Connection/<ConnectionName>`
  Read configuration data of a connection (Page 790)
- `Connections`
  Read configuration data of all connections (Page 790)
- `Group/<GroupName>`
  Read configuration data of a tag group (Page 791)
- `Groups`
  Read configuration data of all tag groups (Page 792)
- `StructureType/<StructureName>`
  Read configuration data of a structure type (Page 793)
- `StructureTypes`
  Read configuration data of all structure types (Page 794)
- `StructureVariable/<StructureTypeName>`
  Read instances of a structure type (Page 796)
- `StructureVariables`
  Read instances of multiple structure types (Page 797)
- `Value/<VariableName>`
  Read Runtime value of a tag (Page 798)
  Write value to a tag (Page 799)
- `Values`
  Read Runtime values of all tags (Page 799)
  Write values to multiple tags (Page 800)
- `variable/<VariableName>`
  Read configuration data of a tag (Page 801)
- `variables`
  Read configuration data of all tags (Page 802)

Archiving system methods

Base URL

`https://<Host>:<Port>/WinCCRestService/tagLogging/`

End points

- `Archives`
  Reading configuration data of all process value archives (Page 804)
- `Archive/`
  - `<ArchiveName>`
    Reading configuration data of a single process value archive (Page 807)
  - `<ArchiveName>/Variable/<VariableName>`
    Reading configuration data of a single process value archive tag (Page 807)
  - `<ArchiveName>/Variables`
    Reading configuration data of all tags of a process value archive (Page 808)
  - `<ArchiveName>/Value/<VariableName>`
    Reading the Runtime value of a tag of a process value archive (Page 810)
  - `<ArchiveName>/Values`
    Reading Runtime values of multiple tags of a single process value archive (Page 812)
    Reading Runtime values of multiple tags from different process value archives (Page 814)
- `Timer/<TimerName>`
  Reading configuration data of a single time of the archive system (Page 817)
- `Timers`
  Reading configuration data of all times of the archive system (Page 817)
- `Variable/<VariableName>`
  Reading configuration data of an archive system tag (Page 821)
- `Variables`
  Reading configuration data of all archive system tags (Page 822)

See also

REST interface in WinCC (Page 777)

Filter for methods (Page 825)

How to configure the WinCC REST service (Page 780)

10.5 Tag management methods

10.5.1 Read configuration data of a connection

Description

Reads the configuration data of a connection that is created below a communication driver.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/Connection/
<ConnectionName>
```

Body

```
None
```

Return value: Example

```
{
"connectionName":"NewConnection_Name",
"channelUnit":"OMS+",
"channelName":"SIMATIC S7-1200, S7-1500 Channel",
"lastChange":"2020-01-30 15:08:31.000"
}
```

10.5.2 Read configuration data of all connections

Description

Reads the configuration data of all connections that are created below the communication drivers in Tag Management.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/Connections
```

Body

```
None
```

Return value: Example

```
{"connections":[
{"connectionName":"Internal Tag",
"channelUnit":"Internal Tag",
"channelName":"Internal Tag",
"lastChange":"2020-01-30 15:01:30.297"},
{"connectionName":"NewConnection_Name",
"channelUnit":"OMS+",
"channelName":"SIMATIC S7-1200, S7-1500 Channel",
"lastChange":"2020-01-30 15:08:31.000"}
]}
```

Alternative addressing

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/Connections?
itemLimit=20
```

Return value: Example

```
{"continuationPoint":2,
"moreAvailable":false,
"connections":[
{"connectionName":"Internal Tag",
"channelUnit":"Internal Tag",
"channelName":"Internal Tag",
"lastChange":"2020-01-30 15:01:30.297"},
{"connectionName":"NewConnection_Name",
"channelUnit":"OMS+",
"channelName":"SIMATIC S7-1200, S7-1500 Channel",
"lastChange":"2020-01-30 15:08:31.000"}
]}
```

10.5.3 Read configuration data of a tag group

Description

Reads the configuration data of a tag group that is created below a communication driver or as internal group.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/Group/
<GroupName>
```

Body

```
None
```

Return value: Example

```
{
"groupName":"ProcessHistorian",
"connectionName":"Internal Tag",
"lastChange":"2020-06-08 14:13:14.467"
}
```

10.5.4 Read configuration data of all tag groups

Description

Reads the configuration data of the tag groups that are created below the communication drivers or as internal group.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/Groups
```

Body

```
None
```

Return value: Example

```
{"groups":[
{"groupName":"Script",
"connectionName":"Internal Tag",
"lastChange":"2020-06-08 14:13:14.463"},
{"groupName":"TagLoggingRt",
"connectionName":"Internal Tag",
"lastChange":"2020-06-08 14:13:14.467"},
{"groupName":"ProcessHistorian",
"connectionName":"Internal Tag",
"lastChange":"2020-06-08 14:13:14.467"},
{"groupName":"Performance",
"connectionName":"Internal Tag",
"lastChange":"2020-06-08 14:13:14.470"},
{"groupName":"Connection1_Group_1",
"connectionName":"NewConnection_1",
"lastChange":"2020-06-08 15:23:57"},
{"groupName":"Connection2_Group_1",
"connectionName":"NewConnection_2",
"lastChange":"2020-06-08 15:24:39"
}]}
```

Alternative addressing

URL

`https://<Host>:<Port>/WinCCRestService/tagManagement/Connections?itemLimit=2&continuationPoint=0`

Return value: Example

```
{"continuationPoint":3,"moreAvailable":true,
"groups":[
{
"groupName":"Script",
"connectionName":"Internal Tag",
"lastChange":"2020-06-08 14:13:14.463"},
{"groupName":"TagLoggingRt",
"connectionName":"Internal Tag",
"lastChange":"2020-06-08 14:13:14.467"
}]}
```

10.5.5 Read configuration data of a structure type

Description

Reads the configuration data of a structure type that is created under "Structure tags".

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/StructureType/
<StructureName>
```

Body

```
None
```

Return value: Example

```
{"typeName":"NewStructure_1",
"typeMembers":[
{"memberName":"NewElement_1",
"dataType":2,
"lastChange":"2020-03-19 08:46:06.000"},
{"memberName":"NewElement_2",
"dataType":2,
"lastChange":"2020-03-19 08:46:07.000"},
{"memberName":"NewElement_3",
"dataType":2,
"lastChange":"2020-03-19 08:46:08.000"},
{"memberName":"NewElement_4",
"dataType":2,
"lastChange":"2020-03-19 08:46:09.000"}
]}
```

10.5.6 Read configuration data of all structure types

Description

Reads the configuration data of all structure types that are created under "Structure tags".

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/StructureTypes
```

Body

```
None
```

Return value: Example

```
{"structures":
[
{"typeName":"NewStructure_1","typeMembers":[
{"memberName":"NewElement_1",
"dataType":2,
"lastChange":"2020-03-19 08:46:06.000"},
{"memberName":"NewElement_2",
"dataType":2,
"lastChange":"2020-03-19 08:46:07.000"},
{"memberName":"NewElement_3",
"dataType":2,
"lastChange":"2020-03-19 08:46:08.000"},
{"memberName":"NewElement_4",
"dataType":2,
"lastChange":"2020-03-19 08:46:09.000"}]},
{"typeName":"NewStructure_2","typeMembers":[
{"memberName":"NewElement_1",
"dataType":2,
"lastChange":"2020-03-19 08:46:12.000"},
{"memberName":"NewElement_2",
"dataType":2,
"lastChange":"2020-03-19 08:46:13.000"},
{"memberName":"NewElement_3",
"dataType":2,
"lastChange":"2020-03-19 08:46:14.000"}]},
{"typeName":"NewStructure_3","typeMembers":[
{"memberName":"NewElement_1",
"dataType":2,
"lastChange":"2020-03-19 08:46:18.000"},
{"memberName":"NewElement_2",
"dataType":2,
"lastChange":"2020-03-19 08:46:20.000"},
{"memberName":"NewElement_3",
"dataType":2,
"lastChange":"2020-03-19 08:46:21.000"}
]}
]}
```

Alternative addressing

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/StructureTypes?
itemLimit=2&continuationPoint=0
```

Return value: Example

```
{"continuationPoint":1026,
"moreAvailable":true,
"structures":[
{"typeName":"NewStructure_1",
"typeMembers":[
{"memberName":"NewElement_1",
"dataType":2,
"lastChange":"2020-03-19 08:46:06.000"},
{"memberName":"NewElement_2",
"dataType":2,
"lastChange":"2020-03-19 08:46:07.000"},
{"memberName":"NewElement_3",
"dataType":2,
"lastChange":"2020-03-19 08:46:08.000"},
{"memberName":"NewElement_4",
"dataType":2,
"lastChange":"2020-03-19 08:46:09.000"}]},
{"typeName":"NewStructure_2","typeMembers":[
{"memberName":"NewElement_1",
"dataType":2,
"lastChange":"2020-03-19 08:46:12.000"},
{"memberName":"NewElement_2",
"dataType":2,
"lastChange":"2020-03-19 08:46:13.000"},
{"memberName":"NewElement_3",
"dataType":2,
"lastChange":"2020-03-19 08:46:14.000"}
]}
]}
```

10.5.7 Read instances of a structure type

Description

Reads the structure tag elements of a structure type that is created under "Structure tags".

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/
StructureVariable/<StructureTypeName>
```

Body

```
None
```

Return value: Example

```
{"structureVariables":
[
{"variableName":"StructureInstance1_1",
"typeName":"NewStructure_1",
"connectionName":"Internal Tag",
"comment":"",
"lastChange":"2020-03-19 17:38:15.000"},
{"variableName":"StructureInstance1_2",
"typeName":"NewStructure_1",
"connectionName":"Internal Tag",
"comment":"",
"lastChange":"2020-03-19 17:38:20.000"},
{"variableName":"StructureInstance1_3",
"typeName":"NewStructure_1",
"connectionName":"Internal Tag",
"comment":"",
"lastChange":"2020-03-19 17:38:20.000"}
]
```

10.5.8 Read instances of multiple structure types

Description

Reads the structure tag elements of multiple structure types that are created under "Structure tags".

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/
StructureVariables
```

Body

```
{
    "typeNames" : [ "NewStructure_1", "NewStructure_2" ]
}
```

Return value: Example

```
{"structureVariables":
[
{"variableName":"StructureInstance1_1",
"typeName":"NewStructure_1",
"connectionName":"Internal Tag",
"comment":"",
"lastChange":"2020-03-19 17:38:15.000"},
{"variableName":"StructureInstance1_2",
"typeName":"NewStructure_1",
"connectionName":"Internal Tag",
"comment":"",
"lastChange":"2020-03-19 17:38:20.000"},
{"variableName":"StructureInstance1_3",
"typeName":"NewStructure_1",
"connectionName":"Internal Tag",
"comment":"",
"lastChange":"2020-03-19 17:38:20.000"},
{"variableName":"StructureInstance2_1",
"typeName":"NewStructure_2",
"connectionName":"Internal Tag",
"comment":"",
"lastChange":"2020-03-19 17:38:32.000"},
{"variableName":"StructureInstance2_2",
"typeName":"NewStructure_2",
"connectionName":"Internal Tag",
"comment":"",
"lastChange":"2020-03-19 17:38:36.000"},
{"variableName":"StructureInstance2_3",
"typeName":"NewStructure_2",
"connectionName":"Internal Tag",
"comment":"",
"lastChange":"2020-03-19 17:38:36.000"}
]
```

10.5.9 Read Runtime value of a tag

Description

Reads the tag value of a process tag or an internal tag in Runtime.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/Value/
<VariableName>
```

Body

```
None
```

Return value: Example

```
{"variableName":"Tag2",
"dataType":2,
"value":"0",
"timestamp":"2020-03-20T09:12:38.5830Z",
"qualitycode":"0x4C - uncertain - initial value",
"errorcode":0}
```

10.5.10 Write value to a tag

Description

Writes a value synchronously to a tag.

Call

```
PUT
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/Value/
<VariableName>
```

Body

```
{"value":"text value"}
```

Return value: Example

```
{"variableName":"Tag11",
"errorcode":0}
```

10.5.11 Read Runtime values of all tags

Description

Reads the tag values of all registered process tags and internal tags in Runtime.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/Values
```

Body

```
{
    "variableNames" : [ "Tag name 1", "Tag name 2", "Tag name 3",
"Tag name 4" ]
}
```

Return value: Example

```
[
{"variableName":"Tag1",
"dataType":1,
"value":"false",
"timestamp":"2020-03-20T09:12:38.5830Z",
"qualitycode":"0x4C - uncertain - initial value",
"errorcode":0},
{"variableName":"Tag2",
"dataType":2,
"value":"0",
"timestamp":"2020-03-20T09:12:38.5830Z",
"qualitycode":"0x4C - uncertain - initial value",
"errorcode":0},
{"variableName":"Tag10",
"dataType":5,
"value":"This variable contains text values",
"timestamp":"2020-03-20T09:12:38.5830Z",
"qualitycode":"0x4C - uncertain - initial value",
"errorcode":0},
{"variableName":"Tag",
"error":"Not Found"}
]
```

10.5.12 Write values to multiple tags

Description

Writes values synchronously to the listed tags.

Specify the value to be written for each tag.

Call

```
PUT
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/Values
```

Body: Example

```
[
{"variableName":"Tag name 1","value":true},
{"variableName":"Tag name 2","value":6},
{"variableName":"Tag name 3","value":8},
{"variableName":"Tag name 4","value":10}
]
```

Return value: Example

```
[
{"variableName":"Tag name 1","errorcode":0},
{"variableName":"Tag name 2","errorcode":0},
{"variableName":"Tag name 3","error":"Not Found"},
{"variableName":"Tag name 4","errorcode":0}
]
```

10.5.13 Read configuration data of a tag

Description

Reads the configuration data of a process tag or an internal tag.

Call

```
GET / POST
```

Url

```
https://<Host>:<Port>/WinCCRestService/tagManagement/variable/
<VariableName>
```

Body

```
None
```

Return value: Example

```
{"variableName":"Tag2",
"dataType":2,
"comment":"comment of Tag2",
"canRead":true,
"canWrite":true,
"lastChange":"2020-03-03 13:48:29.000"}
```

10.5.14 Read configuration data of all tags

Description

Reads the configuration data of all registered process tags and internal tags.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/variables
```

Body

```
None
```

Return value: Example

```
{"variables":
[
{"variableName":"@SCRIPT_COUNT_TAGS",
"dataType":2,
"comment":"",
"canRead":true,
"canWrite":true,
"lastChange":"2020-03-20 09:12:37.000"},
{"variableName":"@SCRIPT_COUNT_REQUESTS_IN_QUEUES",
"dataType":2,
"comment":"",
"canRead":true,
"canWrite":true,
"lastChange":"2020-03-20 09:12:37.000"},
…..]
```

Alternative addressing

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/variables?
itemLimit=2&continuationPoint=0
```

Return value: Example

```
{"continuationPoint":3,
"moreAvailable":true,
"variables":[
{"variableName":"@SCRIPT_COUNT_TAGS",
"dataType":2,
"comment":"",
"canRead":true,
"canWrite":true,
"lastChange":"2020-03-20 09:12:37.000"},
{"variableName":"@SCRIPT_COUNT_REQUESTS_IN_QUEUES",
"dataType":2,
"comment":"",
"canRead":true,
"canWrite":true,
"lastChange":"2020-03-20 09:12:37.000"}
]}
```

10.6 Archiving system methods

10.6.1 Reading configuration data of all process value archives

Description

Reads the configuration data of all process value archives.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagLogging/Archives
```

Body

```
None
```

Return value: Example

```
{
  "archives": [
    {
      "archiveName": "Archive1",
      "archiveType": "process value archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:10:31.000"
    },
    {
      "archiveName": "Archive2",
      "archiveType": "process value archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:10:39.000"
    },
    {
      "archiveName": "Archive3",
      "archiveType": "process value archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:10:44.000"
    },
    {
      "archiveName": "Archive4",
      "archiveType": "process value archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:11:57.000"
    },
    {
      "archiveName": "CompArchive1",
      "archiveType": "compressed archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:10:50.000"
    },
    {
      "archiveName": "CompArchive2",
      "archiveType": "compressed archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:11:03.000"
    },
    {
      "archiveName": "CompArchive3",
      "archiveType": "compressed archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:11:03.000"
    },

      "archiveName": "CompArchive4",
      "archiveType": "compressed archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:12:20.000"
    }
  ]
}
```

Alternative addressing

URL

```
https://<Host>:<Port>/WinCCRestService/taglogging/Archives?
itemLimit=3&continuationPoint=2
```

Return value: Example

```
{
  "continuationPoint": 5,
  "moreAvailable": true,
  "archives": [
    {
      "archiveName": "Archive3",
      "archiveType": "process value archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:10:44.000"
    },
    {
      "archiveName": "Archive4",
      "archiveType": "process value archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:11:57.000"
    },
    {
      "archiveName": "CompArchive1",
      "archiveType": "compressed archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:10:50.000"
    }
  ]
}
```

10.6.2 Reading configuration data of a single process value archive

Description

Reads the configuration data of a single process value archive.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagLogging/Archive/
<ArchiveName>
```

Body

```
None
```

Return value: Example

```
{
  "archiveName": "Archive2",
  "archiveType": "process value archive",
  "comment": "",
  "disabled": false,
  "manualInput": true,
  "lastChange": "2022-12-12 07:10:39.000"
}
```

10.6.3 Reading configuration data of a single process value archive tag

Description

Reads the configuration data of a single tag of a process value archive.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagLogging/Archive/
<ArchiveName>/Variable/<VariableName>
```

Body

```
None
```

Return value: Example

```
{
  "variableName": "Tag_14",
  "unit": "ha",
  "archiveName": "Archive3",
  "dataType": "analog",
  "comment": "",
  "processVariableName": "Tag_14",
  "disabled": false,
  "manualInput": true,
  "relevantLongTerm": true,
  "acquisitionCycle": "500 ms",
  "acquisitionType": "cyclicContinuous",
  "archivingCycle": "500 ms",
  "archivingFactor": 1,
  "canRead": false,
  "canWrite": false,
  "lastChange": "2022-12-12 09:45:48.000"
}
```

10.6.4 Reading configuration data of all tags of a process value archive

Description

Reads the configuration data of all tags of a process value archive.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagLogging/Archive/
<ArchiveName>/Variables
```

Body

```
None
```

Return value: Example

```
{
  "variables": [
    {
      "variableName": "Tag_11",
      "unit": "Pound",
      "archiveName": "Archive3",
      "dataType": "analog",
      "comment": "",
      "processVariableName": "Tag_11",
      "disabled": false,
      "manualInput": true,
      "relevantLongTerm": true,
      "acquisitionCycle": "500 ms",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle": "500 ms",
      "archivingFactor": 1,
      "canRead": false,
      "canWrite": false,
      "lastChange": "2022-12-12 09:45:33.000"
    },
    {
      "variableName": "Tag_15",
      "unit": "lbs",
      "archiveName": "Archive3",
      "dataType": "analog",
      "comment": "",
      "processVariableName": "Tag_15",
      "disabled": false,
      "manualInput": true,
      "relevantLongTerm": true,
      "acquisitionCycle": "500 ms",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle": "500 ms",
      "archivingFactor": 1,
      "canRead": false,
      "canWrite": false,
      "lastChange": "2022-12-12 09:45:56.000"
    }
  ]
}
```

Alternative addressing

URL

```
https://<Host>:<Port>/WinCCRestService/taglogging/Archive/
<ArchiveName>/Variables?itemLimit=2&continuationPoint=12
```

Return value: Example

```
{
  "continuationPoint": 14,
  "moreAvailable": true,
  "variables": [
    {
      "variableName": "Tag_13",
      "unit": "ar",
      "archiveName": "Archive3",
      "dataType": "analog",
      "comment": "",
      "processVariableName": "Tag_13",
      "disabled": false,
      "manualInput": true,
      "relevantLongTerm": true,
      "acquisitionCycle": "500 ms",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle": "500 ms",
      "archivingFactor": 1,
      "canRead": false,
      "canWrite": false,
      "lastChange": "2022-12-12 09:45:44.000"
    },
    {
      "variableName": "Tag_14",
      "unit": "ha",
      "archiveName": "Archive3",
      "dataType": "analog",
      "comment": "",
      "processVariableName": "Tag_14",
      "disabled": false,
      "manualInput": true,
      "relevantLongTerm": true,
      "acquisitionCycle": "500 ms",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle": "500 ms",
      "archivingFactor": 1,
      "canRead": false,
      "canWrite": false,
      "lastChange": "2022-12-12 09:45:48.000"
    }
  ]
}
```

10.6.5 Reading the Runtime value of a tag of a process value archive

Description

Reads the value of a process value archive tag in Runtime.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/TagLogging/Archive/
<ArchiveName>/Value/<VariableName>
```

Body

```
None
```

Return value: Example

```
{
  "archive": "Archive3",
  "variableName": "Tag_13",
  "unit": "ar",
  "errorcode": 0,
  "values": [
    {
      "value": "79",
      "timestamp": "2022-12-12T13:57:18.5490Z",
      "qualitycode": 128
    },
    {
      "value": "68",
      "timestamp": "2022-12-12T13:57:19.490Z",
      "qualitycode": 128
    },
    {
      "value": "68",
      "timestamp": "2022-12-12T13:57:19.5490Z",
      "qualitycode": 128
    },
    {
      "value": "56",
      "timestamp": "2022-12-12T13:57:20.490Z",
      "qualitycode": 128
    },
    {
      "value": "56",
      "timestamp": "2022-12-12T13:57:20.5490Z",
      "qualitycode": 128
    }
  ]
}
```

10.6.6 Reading Runtime values of multiple tags of a single process value archive

Description

Reads the values of multiple tags of a single process value archive in Runtime.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagLogging/Archive/
<ArchiveName>/Values
```

Body

```
{
  "variableNames" : [ "Tag1", "Tag2", "Tag10", "Tag" ]
}
```

Return value: Example

```
[
  {
    "archive": "Archive3",
    "variableName": "Tag_11",
    "unit": "Pound",
    "errorcode": 0,
    "values": [
      {
        "value": "44",
        "timestamp": "2022-12-12T14:01:56.490Z",
        "qualitycode": 128
      },
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:56.5490Z",
        "qualitycode": 128
      },
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:57.490Z",
        "qualitycode": 128
      }
    ]
  },
  {
    "archiveName": "Archive3",
    "variableName": "Tag_12",
    "error": "Forbidden"
  },
  {
    "archive": "Archive3",
    "variableName": "Tag_13",
    "unit": "ar",
    "errorcode": 0,
    "values": [
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:56.5490Z",
        "qualitycode": 128
      },
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:57.490Z",
        "qualitycode": 128
      },
      {
        "value": "21",
        "timestamp": "2022-12-12T14:01:57.5490Z",
        "qualitycode": 128
      }
    ]
  }
]
```

10.6.7 Reading Runtime values of multiple tags from different process value archives

Description

Reads the values of multiple tags from different process value archives in Runtime.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagLogging/Archive/
<ArchiveName>/Values
```

Body

```
{
    "archives": [
      {
        "name": "Archive1",
        "variables": [
          {
            "name": "Variable1",
            "timeFrom": "2022-11-25T17:19:18.8980Z",
            "timeTo": "2022-11-25T17:20:18.8980Z",
            "maxValues": 100
          },
          {
            "name": "Variable2",
            "timeFrom": "2022-11-25T17:19:18.8980Z",
            "range": 10,
            "maxValues": 50
          }
        ]
      },
      {
        "name": "Archive2",
        "variables": [
          {
            "name": "Variable1",
            "timeTo": "2022-11-25T17:20:38.8980Z",
            "maxValues": 300
          },
          {
            "name": "Variable2",
            "range": 60,
            "maxValues": 400
          }
        ]
      }
    ]
  }
}
```

Return value: Example

```
[
  {
    "archive": "Archive1",
    "variableName": "Variable1",
    "unit": "Pound",
    "errorcode": 0,
    "values": [
      {
        "value": "44",
        "timestamp": "2022-12-12T14:01:56.490Z",
        "qualitycode": 128
      },
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:56.5490Z",
        "qualitycode": 128
      },
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:57.490Z",
        "qualitycode": 128
      }
    ]
  },
  {
    "archiveName": "Archive1",
    "variableName": "Variable2",
    "error": "Forbidden"
  },
  {
    "archive": "Archive2",
    "variableName": "Variable1",
    "unit": "ar",
    "errorcode": 0,
    "values": [
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:56.5490Z",
        "qualitycode": 128
      },
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:57.490Z",
        "qualitycode": 128
      },
      {
        "value": "21",
        "timestamp": "2022-12-12T14:01:57.5490Z",
        "qualitycode": 128
      }
    ]
  },
  {
    "archiveName": "Archive2",
    "variableName": "Variable2",
    "error": "Forbidden"
  }
]
```

10.6.8 Reading configuration data of a single time of the archive system

Description

Reads the configuration data of a single time of the archive system.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagLogging/Timer/<TimerName>
```

Body

```
None
```

Return value: Example

```
{
  "timerName": "3 seconds",
  "type": "cycleTime",
  "base": "1 second",
  "factor": "3",
  "lastChange": "2022-12-12 07:50:45.000"
}
```

10.6.9 Reading configuration data of all times of the archive system

Description

Reads the configuration data of all times of the archive system.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagLogging/Timers
```

Body

```
None
```

Return value: Example

```
{
  "timers": [
    {
      "timerName": "1 day",
      "type": "cycleTime",
      "base": "1 day",
      "factor": 1,
      "lastChange": "2022-12-12 07:08:42.480"
    },
    {
      "timerName": "1 hour",
      "type": "cycleTime",
      "base": "1 hour",
      "factor": 1,
      "lastChange": "2022-12-12 07:08:42.480"
    },
    {
      "timerName": "1 minute",
      "type": "cycleTime",
      "base": "1 minute",
      "factor": 1,
      "lastChange": "2022-12-12 07:08:42.477"
    },
    {
      "timerName": "1 second",
      "type": "cycleTime",
      "base": "1 second",
      "factor": 1,
      "lastChange": "2022-12-12 07:08:42.473"
    },
    {
      "timerName": "3 seconds",
      "type": "cycleTime",
      "base": "1 second",
      "factor": "3",
      "lastChange": "2022-12-12 07:50:45.000"
    },
    {
      "timerName": "500 ms",
      "type": "cycleTime",
      "base": "500 ms",
      "factor": 1,
      "lastChange": "2022-12-12 07:08:42.470"
    },
    {
      "timerName": "TimeSeries Daily",
      "type": "timeSeries",
      "base": "daily",
      "factor": 1,
      "lastChange": "2022-12-12 07:51:14.000"
    },
    {
      "timerName": "TimeSeries Monthly",
      "type": "timeSeries",
      "base": "monthly",
      "factor": 1,
      lastChange": "2022-12-12 07:51:38.000"
    },
    {
      timerName": "TimeSeries Weekly",
      "type": "timeSeries",
      "base": "weekly",
      "factor": 1,
      "lastChange": "2022-12-12 07:51:26.000"
    },
    {
      "timerName": "TimeSeries Yearly",
      "type": "timeSeries",
      "base": "yearly",
      "factor": 1,
      "lastChange": "2022-12-12 07:51:55.000"
    }
  ]
}
```

Alternative addressing

URL

`https://<Host>:<Port>/WinCCRestService/taglogging/Timers?itemLimit=2&continuationPoint=0`

Return value: Example

```
{
  "continuationPoint": 2,
  "moreAvailable": true,
  "timers": [
    {
      "timerName": "1 day",
      "type": "cycleTime",
      "base": "1 day",
      "factor": 1,
      "lastChange": "2022-12-12 07:08:42.480"
    },
    {
      "timerName": "1 hour",
      "type": "cycleTime",
      "base": "1 hour",
      "factor": 1,
      "lastChange": "2022-12-12 07:08:42.480"
    }
  ]
}
```

10.6.10 Reading configuration data of an archive system tag

Description

Reads the configuration data of a single tag of the archive system.

The names of the tags are only unique within individual archives. Multiple tags can be returned.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagLogging/Variable/
<VariableName>
```

Body

```
None
```

Return value: Example

```
{
  "variables": [
    {
      "variableName": "Tag_4",
      "unit": "l",
      "archiveName": "Archive1",
      "dataType": "analog",
      "comment": "",
      "processVariableName": "Tag_4",
      "disabled": false,
      "manualInput": true,
      "relevantLongTerm": true,
      "acquisitionCycle": "500 ms",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle": "500 ms",
      "archivingFactor": 1,
      "canRead": false,
      "canWrite": false,
      "lastChange": "2022-12-12 09:44:14.000"
    },
    {
      "variableName": "Tag_4",
      "unit": "",
      "archiveName": "CompArchive1",
      "dataType": "compress",
      "comment": "",
      "processVariableName": "",
      "disabled": false,
      "manualInput": true,
      "relevantLongTerm": true,
      "acquisitionCycle": "",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle": "",
      "archivingFactor": 1,
      "canRead": false,
      "canWrite": false,
      "lastChange": "2022-12-12 07:12:32.000"
    }
  ]
}
```

10.6.11 Reading configuration data of all archive system tags

Description

Reads the configuration data of all tags of the archive system.

Call

```
GET / POST
```

URL

```
https://<Host>:<Port>/WinCCRestService/tagLogging/Variables
```

Body

```
None
```

Return value: Example with paging

URL

```
https://<Host>:<Port>/WinCCRestService/taglogging/Variables?
itemLimit=2&continuationPoint=12
```

```
{
  "continuationPoint": 14,
  "moreAvailable": true,
  "variables": [
    {
      "variableName": "Tag_13",
      "unit": "ar",
      "archiveName": "Archive3",
      "dataType": "analog",
      "comment": "",
      "processVariableName": "Tag_13",
      "disabled": false,
      "manualInput": true,
      "relevantLongTerm": true,
      "acquisitionCycle": "500 ms",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle": "500 ms",
      "archivingFactor": 1,
      "canRead": false,
      "canWrite": false,
      "lastChange": "2022-12-12 09:45:44.000"
    },
    {
      "variableName": "Tag_14",
      "unit": "ha",
      "archiveName": "Archive3",
      "dataType": "analog",
      "comment": "",
      "processVariableName": "Tag_14",
      "disabled": false,
      "manualInput": true,
      "relevantLongTerm": true,
      "acquisitionCycle": "500 ms",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle": "500 ms",
      "archivingFactor": 1,
      "canRead": false,
      "canWrite": false,
      "lastChange": "2022-12-12 09:45:48.000"
    }
  ]
}
```

10.7 Filter for methods

You can use filters to delimit requests.

This will reduce the amount of transferred data.

Inserting filters

The filter is transmitted as part of the URL:

- <URL>?<Filter criterion>

Example:

- https://<URL>/WinCCRestService/tagManagement/variables?canWrite

The filter strings are not case-sensitive.

Link

You can use multiple various filters.

For the logic operation, use the operand "&", for example:

- <URL>?<Filter criterion_1>&<Filter criterion_2>&<Filter criterion_3>

The more filters you link, the more you restrict the search result. OR logic operations are not possible.

You can only use each filter parameter once in a request.

Example of an invalid filter combination:

- variables?variableName=mot*&variableName=!motor

Filter criteria: Parameters

Parameter	Object	Description
canRead	Tags	Returns the names of the tags to which the logged on user has read access.
canWrite	Tags	Returns the names of the tags to which the logged on user has write access.
variableName	Tags	Returns the names of all tags that contain the searched string. You can use the placeholders "?" and "*".
connection	Tags: Connection data	Provides information on the connection under which a tag, structure tag or tag group is created.
structureType	Structure tags	Returns the names of all structure type elements for the structure type.
structureVariable	Structure tags	Returns the names of all structure tag elements for the structure type.
group	Tag group	Returns the names of all tags in the tag group.
channel	Communication channel	Returns the names of all connections under the channel.

Parameter	Object	Description
archiveName	Process value archives	Returns all process value archives that contain the searched string. You can use the placeholders "?" and "*".
begin	Process value archives	Start of the time range in which the tag values from a process value archive are returned. Date format: YYYY-MM-DD hh:mm:ss:ms Example: 2015-12-31 16:30:00:000
end	Process value archives	End of the time range in which tag values from a process value archive are returned. Date format: YYYY-MM-DD hh:mm:ss:ms Example: 2015-12-31 16:30:00:000
range	Process value archives	Time range in seconds in which tag values from a process value archive are returned. If used without linking to the "begin" or "end" parameters, the last seconds of a process value archive are set as the time span.
maxValues	Process value archives	Maximum number of returned tag values. Max. 1 000 tag values, unless otherwise specified.
changed_After	Configuration data	Returns the configuration data that were changed after the specified date. Date format: YYYY-MM-DD hh:mm:ss:ms Example: 2015-12-31 16:30:00:000
itemLimit	Configuration data	Limits the number of elements returned. This improves the performance of large WinCC projects.
continuationPoint	Configuration data	Continues a request that was limited via "itemLimit". If the number of elements found is greater than the value set via "itemLimit", the output is terminated when the maximum value is reached. In additional requests you can continue the output until all elements found have been returned.
* ?	Wildcards	You can only use the wildcards when filtering for tag names (variableName). • ?: Any character in the string • *: Any number of characters at the beginning or at the end of the string
!	Operand: Negation (NOT)	You can only use a single NOT filter per request.
&	Operand: Logic operation (AND)	You can use several AND logic operations for each request.

Example: Wildcards

Request for tags whose name contains "Motor":

- /variables?variableName=*motor*

Request for tags whose name begins with "parfum" or "perfum":

- /variables?variableName=p?rfum*

Example: channel

Request for the connection names under the channel "SIMATIC S7-1200, S7-1500 Channel":

- /connections?channel=SIMATIC S7-1200, S7-1500 Channel

Quote the name of a communication channel as it is displayed in the WinCC Tag Management.

Example: itemLimit / continuationPoint

The request should return all tags to which the logged-on user has write access.

You limit the request with "itemLimit" to a maximum of 90 tags, although a total of 250 tags are found.

- /variables?canWrite&itemLimit=90

The response body contains the first 90 tags.

To query the next 90 tags, repeat the request and add "continuationPoint":

- /variables?canWrite&itemLimit=90&continuationPoint

The response body contains the next 90 tags.

Repeat this request again to query the last 70 tags.

Example: ! (NOT)

changed_After: Before the specified time

To request all connections that have been created up to a certain point in time, select a negation of "changed_After":

- /connections?changed_After!=2015-11-26 10:30:02.000

connection: Excluding connection names

If you want to request the connection data of the following tags:

- The tag name contains the string "_S7_"
- The tags are not created under the connection "S7_1".

Filters:

- /variables?Connection!=S7_1&variableName=*_S7_*

Example: begin / end / range

To define a time range in which the tag values are returned from the process value archive, use the parameters "begin", "end", and "range".

If you want to request tag values between a start and end time:

- /Values?begin=2015-11-26 10:30:02.000&end=2015-12-01 10:30:02.000

If you want to request tag values in the hour after a start time:

- /Values?begin=2015-11-26 10:30:02.000&range=3600

If you want to request tag values in the hour before an end time:

- /Values?end=2015-12-01 10:30:02.000&range=3600

See also

REST interface in WinCC (Page 777)

Overview of the methods (Page 788)

WinCC/Cloud Connector 11

11.1 WinCC/Cloud Connector

With the WinCC/Cloud Connector, you can automatically transfer tags from the WinCC station to a cloud.

You can use the data stored in the cloud for further analysis or output the tag values, for example via dashboards.

Cloud provider

WinCC/Cloud Connector supports the following providers:

- Siemens MindSphere - MindConnect IoT Extension
- Siemens MindSphere - MindConnect EU1
- Amazon: AWS
- Microsoft: Azure
- Siemens MindSphere on Alibaba Cloud (PR China)
- Generic MQTT
- REST protocol

How the Cloud Connector works

The "Cloud Connector" uses the "MQTT" or "REST" protocols for sending tag values.

To increase communication security, use an encrypted connection with certificate handshake.

You can only send data with the Cloud Connector. Data cannot be received in WinCC.

Message Queue Telemetry Transport (MQTT)

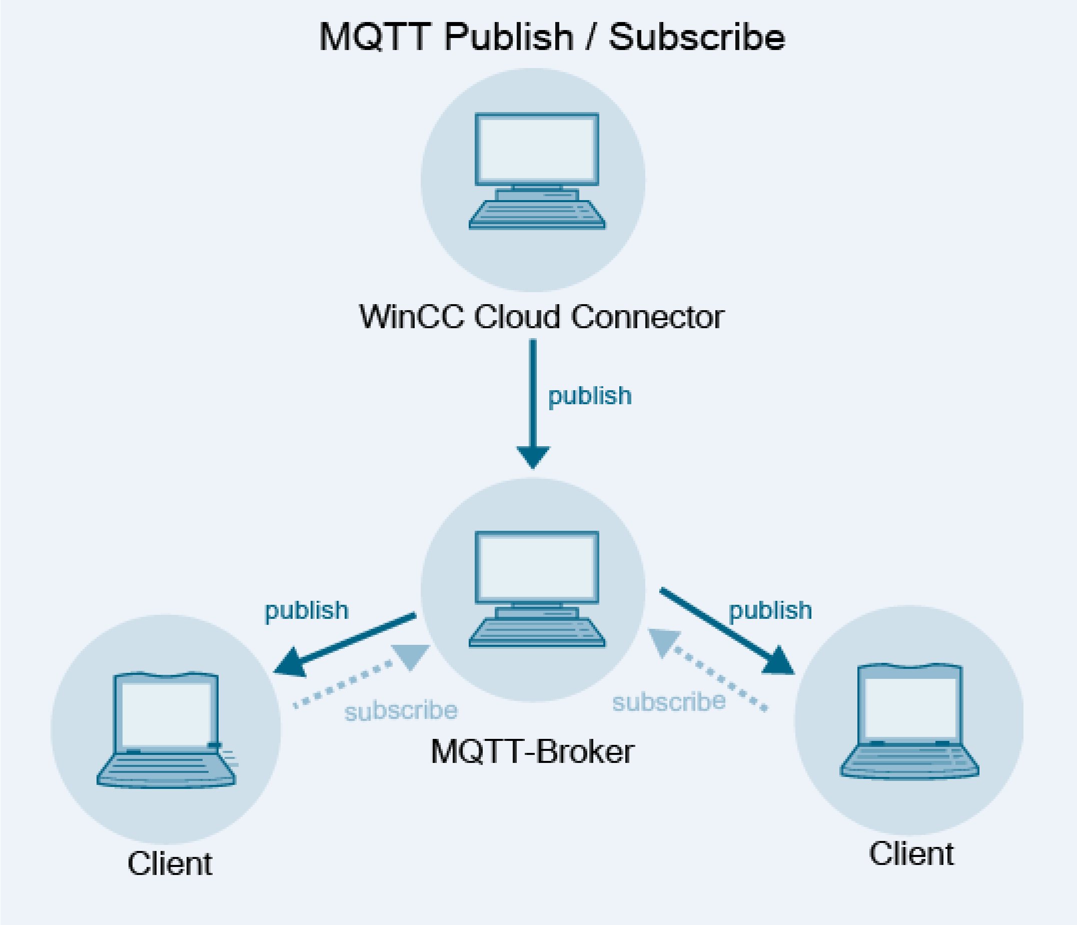
A central server, the MQTT broker, is used for data transfer via the MQTT protocol.

The data exchange between sending and receiving devices takes place exclusively via the MQTT broker.

You can find additional information on the MQTT protocol on the Internet:

- https://mqtt.org/ (https://mqtt.org/)

Communication via MQTT



Representational State Transfer (REST)

The REST protocol uses HTTP or HTTPS for communication in distributed client-server systems.

Data exchange via REST

Similar to MQTT, data is sent to a central server.

You can address any provider. Authentication takes place using the user name and password at the provider.

WinCC as REST service

WinCC as a REST service allows external applications to access WinCC data.

Additional information on access to the REST interface of WinCC via HTTP methods:

- "Interfaces > REST interface (Page 777)"

See also

REST settings in the WinCC Cloud Connector (Page 846)

REST interface in WinCC (Page 777)

MQTT protocol (https://mqtt.org/)

11.2 Licensing Cloud Connector

Licenses

You need a separate license for the WinCC/Cloud Connector:

- SIMATIC WinCC Cloud Connect

Without the license you can transfer a maximum of 5 tags for test purposes.

11.3 Data transfer to the cloud via MQTT

Cloud provider

Data is written using the MQTT protocol via the Cloud Connector.

The following cloud providers are currently supported:

- Siemens MindSphere - MindConnect IoT Extension (MQTT)
- Siemens MindSphere - MindConnect EU1 (MQTT)
- Amazon Web Services (MQTT)
- Microsoft Azure (MQTT)
- Siemens MindSphere on Alibaba Cloud (MQTT):

Service CCCloudConnect

The Windows service CCCloudConnect is used to establish a connection between the WinCC project and the cloud system.

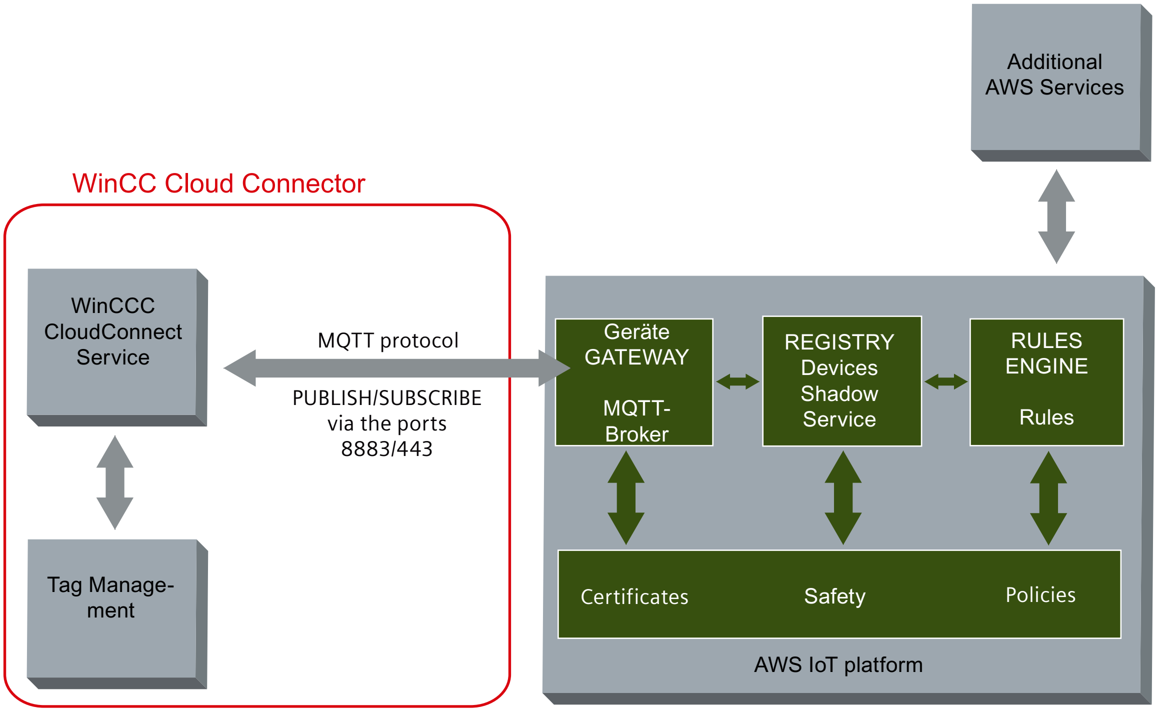
The CCCloudConnect service is an MQTT client that connects to the MQTT broker of the cloud to send data over the standard ports 8883 or 443.

In WinCC, the CCCloudConnect service logs the value changes of the WinCC tags. The values are written to the cloud.

If CCCloudConnect receives a change in value from the tag management, the service creates a message. The service transmits this message to the MQTT broker.

Example: WinCC/Cloud Connector and AWS-MQTT

The following graphic shows the data transfer between the Cloud Connector and the AWS platform:



MQTT topics

Naming convention

A separate MQTT topic is created for each tag that is sent to the MQTT broker.

Every client that wants to receive this topic from the broker must know the topic.

The naming convention for WinCC tags is:

- <Station name>/<WinCC project name>/<tag name>

MQTT clients that want to receive these values must subscribe to the MQTT topics with the appropriate path.

You can change the default station name "WinCC" in the Cloud Connector settings.

MQTT topics in MindSphere

If you select the "MindSphere - MindConnect IoT Extension" or "MindSphere - MindConnect EU1" provider, the tag name from WinCC is applied.

If you select the "MindSphere - MindConnect EU1" provider, the tag comment from WinCC is also applied as a unit.

As station name, the device name from MindSphere is used.

Example

A WinCC project with the name "MyWinCCProject" has been created.

Two tags with the names "MyTag1" and "MyTag2" have been activated for the cloud in the WinCC project.

CCCloudConnect sends the following MQTT topics for these tags:

- WinCC/MyWinCCProject/MyTag1
- WinCC/MyWinCCProject/MyTag2

MindSphere example

A device with the name "WinCCStation1" has been created.

The "MyTag1" and "MyTag2" tags are displayed after the first transmission on the device "WinCCStation1".

Queue

During data transfer via the Cloud Connector, messages are sent according to the queue principle.

The message added first is also sent first.

Data types

Most data types are permissible for the transfer.

Exceptions

- Structured data types are not supported, for example, STRUCT or ARRAY.
- Further limitations depend on the respective cloud provider.

Tag type "Date/Time"

The format for the "Date/Time" tag type depends on the cloud used.

Tag types in MindSphere

In MindSphere, the following data types are not supported for WinCC tags:

- Text tag, 8-bit font
- Text tag, 16-bit font
- Text reference
- Date/time

Time stamp

The time stamp is generated by the WinCC station and sent to the cloud.

The cloud providers use the Coordinated Universal Time (UTC) for time stamps.

Data buffer for connection interruption

To withstand short connection interruptions to the cloud without loss of data, a data buffer is created for communication via MQTT.

The tag values to be transmitted are stored temporarily in this data buffer.

- Standard size of the data buffer: 2 000 000 Tag values
  More tag values may be possible depending on the system configuration and the quality of the cloud connection.

As soon as the connection is restored after a connection interruption, the tag values from the data buffer are saved to the cloud.

Disable WinCC Runtime

Disabling WinCC Runtime also closes the connection to the cloud.

The last message sent to the cloud is stored in the diagnostics file "CCCloundConnect.log".

See also

MQTT settings in the WinCC Cloud Connector (Page 840)

Diagnostics of the cloud connection (Page 861)

Settings in WinCC tag management (Page 838)

11.4 Data transfer to the cloud via REST

Data is written using the REST protocol via the Cloud Connector.

Representational State Transfer (REST)

In contrast to communication via the CCCloudConnect Windows service, the connection is established via HTTP or HTTPS.

You can read tag values and write them to the cloud both on changes and cyclically.

The values are transferred as HTTP requests and HTTP responses.

Data exchange via REST

WinCC sends values to a cloud via REST instead of MQTT.

Similar to MQTT, data is sent to a central server.

You can address any provider.

Authentication takes place using the user name and password at the provider.

Queue

During data transfer via the Cloud Connector, messages are sent according to the queue principle.

The message added first is also sent first.

Data types

Most data types are permissible for the transfer.

Exceptions

- Structured data types are not supported, for example, STRUCT or ARRAY.
- Further limitations depend on the respective cloud provider.

Time stamp

The time stamp is generated by the WinCC station and sent to the cloud.

The cloud providers use the Coordinated Universal Time (UTC) for time stamps.

Disable WinCC Runtime

Disabling WinCC Runtime also closes the connection to the cloud.

The last message sent to the cloud is stored in the diagnostics file "CCCloundConnect.log".

See also

REST interface in WinCC (Page 777)

11.5 Settings in WinCC tag management

In the WinCC Tag Management, you specify which tags are to be transferred.

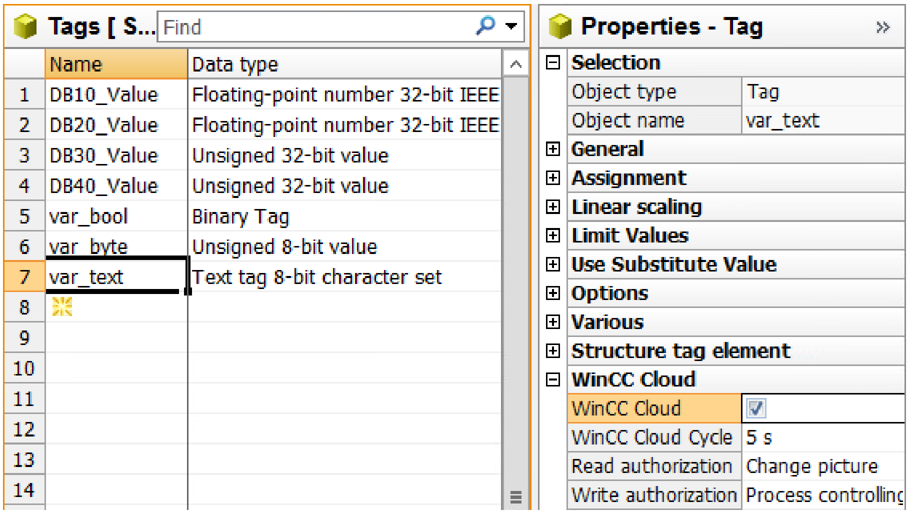
The cloud providers each support different tag types. For information on limitations, see "Data transfer to the cloud via MQTT (Page 833)".

Enabling cloud transfer

To send the values of a tag to the cloud, activate the tag property "WinCC Cloud" in the Tag Management.

Select the desired acquisition cycle.

The assignment of authorizations is only relevant if you are using WinCC as REST service.



Tags [S... Find

	Name	Data type
1	DB10_Value	Floating-point number 32-bit IEEE
2	DB20_Value	Floating-point number 32-bit IEEE
3	DB30_Value	Unsigned 32-bit value
4	DB40_Value	Unsigned 32-bit value
5	var_bool	Binary Tag
6	var_byte	Unsigned 8-bit value
7	var_text	Text tag 8-bit character set
8		
9		
10		
11		
12		
13		
14		

Properties - Tag

⊟ **Selection**	
Object type	Tag
Object name	var_text
⊞ **General**	
⊞ **Assignment**	
⊞ **Linear scaling**	
⊞ **Limit Values**	
⊞ **Use Substitute Value**	
⊞ **Options**	
⊞ **Various**	
⊞ **Structure tag element**	
⊟ **WinCC Cloud**	
WinCC Cloud	☑
WinCC Cloud Cycle	5 s
Read authorization	Change picture
Write authorization	Process controlling

Setting the cycle time

The "WinCC Cloud Cycle" can be set individually for each tag that is to be transferred to the cloud.

The configuration corresponds to the cycle time setting in WinCC tag logging.

- If you do not make any settings for the cycle time, a cycle of one minute is used as the default setting.
- The shortest possible cycle time is one second.
- Select either "On value change" or a fixed value from the list of tag cycles for the cycle time:
  - 1 / 2 / 5 / 10 seconds
  - 1 / 5 / 10 minutes
  - 1 hour

Note

ODK / VBA

The cycle time cannot be changed via VBA or ODK.

See also

Data transfer to the cloud via MQTT (Page 833)

MQTT settings in the WinCC Cloud Connector (Page 840)

Diagnostics of the cloud connection (Page 861)

How to configure the MindSphere connection via the IOT Extension (Page 852)

How to configure a cloud connection via REST (Page 859)

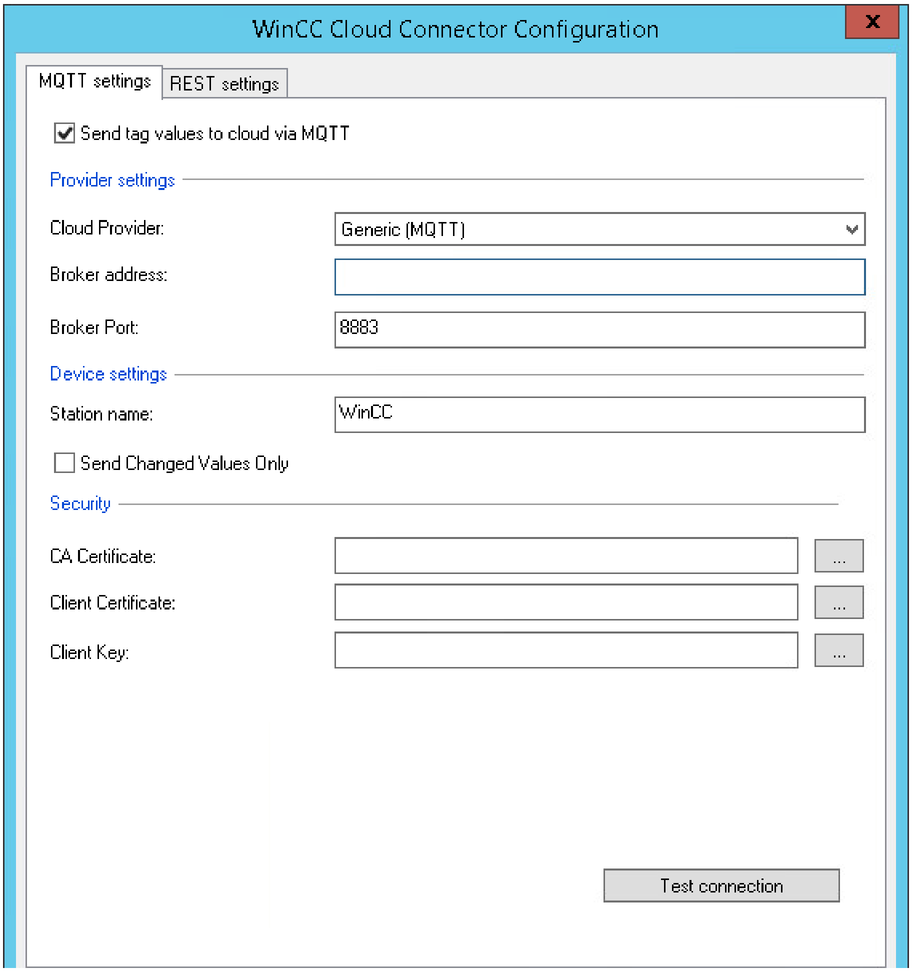
How to configure a cloud connection via MQTT (Page 848)

11.6 MQTT settings in the WinCC Cloud Connector

Introduction

In the "WinCC Cloud Connector Configuration" dialog box, you configure the URL and access settings of the cloud that is used.

To use the MQTT protocol, select the option "Send tag values to cloud via MQTT" in the "MQTT Settings" tab.



WinCC Cloud Connector Configuration

☑ Send tag values to cloud via MQTT

Provider settings

Cloud Provider: Generic (MQTT)

Broker address:

Broker Port: 8883

Device settings

Station name: WinCC

□ Send Changed Values Only

Security

CA Certificate:

Client Certificate:

Client Key:

Cloud provider

Select "Generic MQTT" or a provider.

The following providers are supported:

- MindSphere (MindConnect IoT Extension)
- MindSphere (MindConnect EU1)
- Amazon Web Services (MQTT)

- Microsoft Azure (MQTT)
- Siemens MindSphere on Alibaba Cloud (MQTT):

Application example

You can find a detailed example of how to configure the data connection for the various cloud providers on the Internet:

- Application example 109760955: "WinCC data connection in the cloud (https://support.industry.siemens.com/cs/ww/en/view/109760955)"

Broker address

End point of the MQTT cloud.

The MQTT broker address is made available by the cloud provider.

Broker port

The standard ports are supported:

- 8883
- 443

Station name

Assign a unique name for your client.

The name of the client is used for the path of the MQTT topic during tag transfer.

Station name in MindSphere

When you use the provider "MindSphere (MindConnect IoT Extension)" or "MindSphere (MindConnect EU1)", the station name is used as the device name.

When registering the device in MindSphere, the name is specified as the device ID.

Only send changed values

If you select this option, only the changed data is sent to the cloud.

This may reduce the load during data transmission.

CA certificate

You can obtain the CA certificate from the cloud provider. "CA" stands for "Certificate Authority".

You save the certificate locally on the WinCC station.

WinCC default path:

- \Program Files (x86)\Siemens\WinCC\CloudConnector\Certificate

Alibaba certificates

To access the Alibaba cloud, use the provider's own certificate.

AWS certificates

Amazon Web Services use certificates generated by AWS IoT or certificates with CA certification for device identification.

AWS IoT certificates are signed by the following CA certificates:

- RSA 2048 bit key: VeriSign Class 3 Public Primary G5 root CA certificate
- RSA 2048 bit key: Amazon Root CA 1
- RSA 4096 bit key: Amazon Root CA 2
- ECC 256 bit key: Amazon Root CA 3
- ECC 384 bit key: Amazon Root CA 4

To validate your devices with the AWS IoT server certificate, AWS recommends to install all five certificates on the WinCC stations.

Azure certificate

The Microsoft Azure cloud uses temporary certificates.

MindSphere certificates

A CA certificate for MindSphere is installed during the WinCC installation.

When you select the provider "MindSphere (MindConnect IoT Extension)" or "MindSphere (MindConnect EU1)", then instead of the "Security" area, the "MindSphere" area appears and the MindSphere certificate is displayed.

Client certificates

You store the certificates that you receive from your cloud provider locally on the WinCC station.

The certificates have the following file extensions:

- .cer
- .crt
- .pem

In the input field, enter the path to the certificates:

- \Program Files (x86)\Siemens\WinCC\CloudConnector\Certificate

MindSphere client certificates

When you select the provider "MindSphere (MindConnect IoT Extension)" or "MindSphere (MindConnect EU1)", the "MindSphere" area is displayed instead of the "Security" area.

The client certificate is only configured for the provider 'MindSphere (MindConnect EU1)".

Client key

You obtain the client key from your cloud provider. You store the key locally on your WinCC station.

Client keys are also called "Client/Device keys" and have the file extension ".key".

In the input field, enter the path to the client key:

- \Program Files (x86)\Siemens\WinCC\CloudConnector\Private

MindSphere client certificates

When you select the provider "MindSphere (MindConnect IoT Extension)" or "MindSphere (MindConnect EU)", the "MindSphere" area is displayed instead of the "Security" area.

The client key is only configured for the provider 'MindSphere (MindConnect EU1)".

Register / Unregister

The buttons are only displayed when you select the provider "MindSphere (MindConnect IoT Extension)".

The "Register" button starts the device registration at MindSphere.

After successful registration, the button is grayed out and the "Unregister" button is activated instead.

To change the settings of the WinCC station, click "Unregister". The device remains created in MindSphere. Existing data is retained.

User name / Password

The fields are only shown when you select the provider "MindSphere (MindConnect IoT Extension)".

User name and password are specified during registration of MindSphere and displayed in the Cloud Connector.

When you change the password in MindSphere, you must apply the new password in the Cloud Connector.

Tenant-ID

The field is only displayed when you select the provider "MindSphere (MindConnect EU1)".

Enter the MindSphere Tenant-ID in the input field.

Asset Model/Asset Instance

The fields are only shown when you select the provider "MindSphere (MindConnect EU1)".

Asset Model and Asset Instance are saved as a JSON file.

In the input field, enter the path to Asset Model and Asset Instance.

If you do not yet have any Asset Model or any Asset Instance available, you can generate it with the "Export configuration" button from the WinCC tag configuration. Alternatively, it is also possible to create it manually with a text editor.

Load configuration

The button is only displayed when you select the provider "MindSphere (MindConnect EU1)".

You can use the "Load configuration" button to load the configuration of Asset Model and Asset Instance to MindSphere.

Export configuration

The button is only displayed when you select the provider "MindSphere (MindConnect EU1)".

You can use the "Export configuration" button to generate Asset Model and Asset Instance from the WinCC tag configuration.

Enter the following MindSphere-specific values in the dialog:

- Aspect Type Name
- Aspect Type Description
- Aspect Name
- Asset Type Name
- Asset Type Description
- Asset Name
- Asset Description
- File for Asset Model: File name and path for the Asset Model
- File for Asset Instance: File name and path for the Asset Instance

Upon confirmation of your inputs, the Asset Model and Asset Instance will be generated and automatically entered under "Asset Model" and "Asset Instance".

If Asset Model or Asset Instance are already present, you must additionally confirm overwriting of the data.

Checking the connection

You can use the "Check connection" button to reset the connection settings.

See also

Data transfer to the cloud via MQTT (Page 833)

How to configure a cloud connection via MQTT (Page 848)

Diagnostics of the cloud connection (Page 861)

Settings in WinCC tag management (Page 838)

How to configure the MindSphere connection via the IOT Extension (Page 852)

How to configure a cloud connection via REST (Page 859)

REST settings in the WinCC Cloud Connector (Page 846)

How to configure the MindSphere connection via the EU1 broker (Page 855)

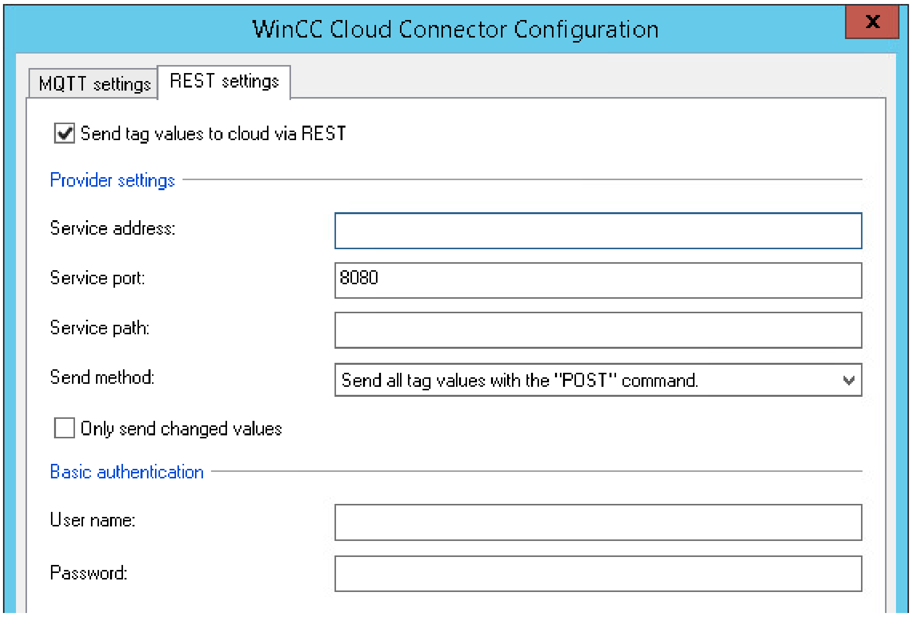
Application example: WinCC data connection to the cloud (https://support.industry.siemens.com/cs/ww/en/view/109760955)

11.7 REST settings in the WinCC Cloud Connector

Introduction

In the "WinCC Cloud Connector Configuration" dialog box, you configure the URL and access settings of the cloud that is used.

To use the REST protocol, select the option "Send tag values to cloud via REST" in the "REST Settings" tab.



☑ Send tag values to cloud via REST

Provider settings

Service address:

Service port: 8080

Service path:

Send method: Send all tag values with the "POST" command.

□ Only send changed values

Basic authentication

User name:

Password:

Provider settings

Enter the connection data of the provider:

Setting	Description
Service address	HTTP address of the provider
Service port	Port number used for access HTTP port "8080" is set by default.
Service path	Path to server directory
Send method	Transmission method: • PUT: Sends all values of a tag • POST: Sends the tag values of all tags

Only send changed values

If you select this option, only the changed data is sent to the cloud.

This may reduce the load during data transmission.

Basic authentication

You configure the user name and password for access protection on the REST server.

Enter the access data for the REST server in the "User name" and "Password" fields.

See also

MQTT settings in the WinCC Cloud Connector (Page 840)

WinCC/Cloud Connector (Page 829)

How to configure a cloud connection via REST (Page 859)

REST interface in WinCC (Page 777)

11.8 How to configure a cloud connection via MQTT

Introduction

You specify the URL and access settings of the cloud used in the "WinCC Cloud Connector Configuration" dialog box.

Application example

You can find a detailed example of how to configure the data connection for the various cloud providers on the Internet:

- Application example 109760955: "WinCC data connection in the cloud (https://support.industry.siemens.com/cs/ww/en/view/109760955)"

Alternative procedure

If you select "MindSphere (MindConnect IoT Extension)" or ""MindSphere (MindConnect EU1)" as Cloud Provider, follow the respective instructions at:

- How to configure the MindSphere connection via the IOT Extension (Page 852)
- How to configure the MindSphere connection via the EU1 broker (Page 855)

To configure a cloud connection via REST, follow the instructions below

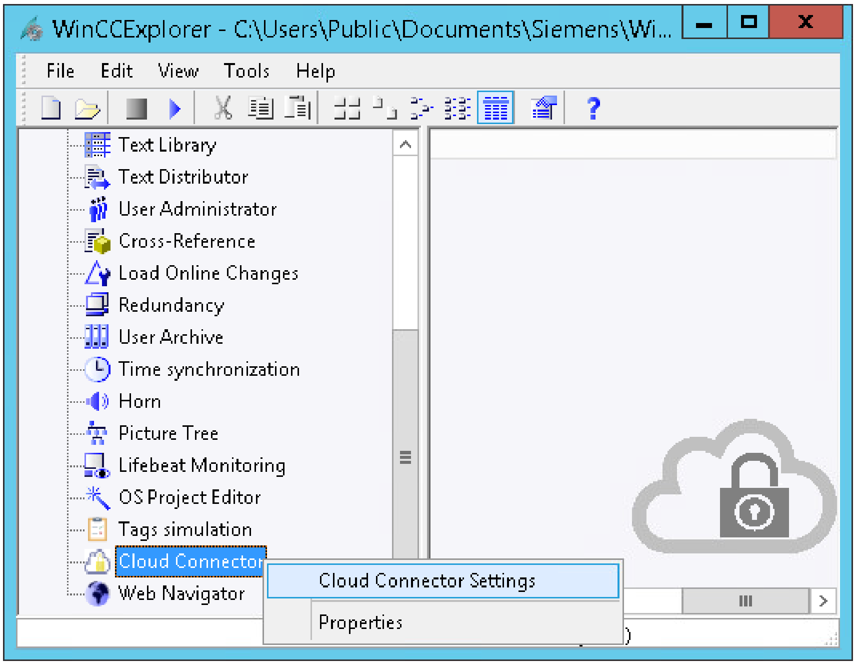
- How to configure a cloud connection via REST (Page 859)

Requirement

- The "WinCC Cloud" option is enabled in WinCC Tag Management for the tags to be transferred.

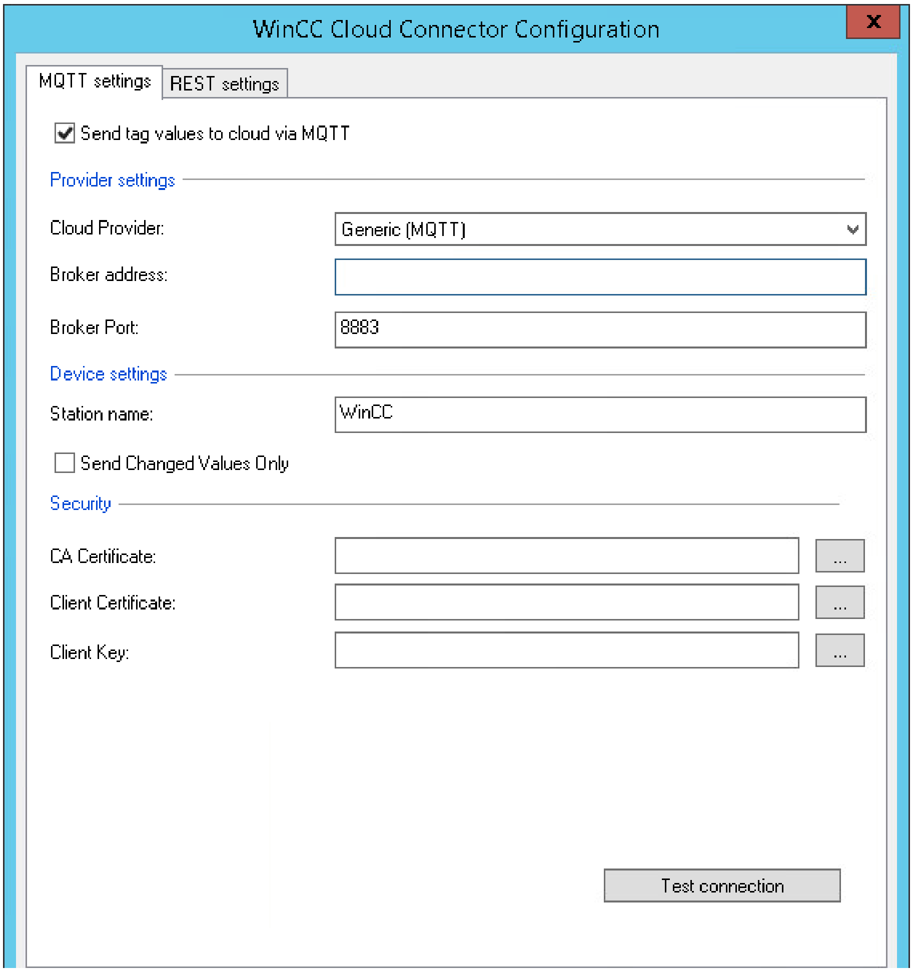
Procedure

1. Open the WinCC Explorer.
2. In the shortcut menu of "Cloud Connector", select the entry "Cloud Connector Settings".



The "WinCC Cloud Connector Configuration" dialog box opens.

3. In the "MQTT Settings" tab, select the "Send tag values to cloud via MQTT" option.



WinCC Cloud Connector Configuration

☑ Send tag values to cloud via MQTT

Provider settings

Cloud Provider: Generic (MQTT)

Broker address:

Broker Port: 8883

Device settings

Station name: WinCC

□ Send Changed Values Only

Security

CA Certificate:

Client Certificate:

Client Key:

4. Specify the connection data:
   - Cloud Provider
   - Broker address
   - Broker port
   - Station name

   To reduce the load during data transfer, select the option "Only send changed values".
5. Select the certificates and the client key.
6. To test the connection settings, click on the "Test connection" button.
7. Close the dialog box with "OK".
8. If the application "Cloud Connector" is not yet activated in the startup list of the computer, you will be prompted to activate it.
   Confirm the activation with "Yes".
   If the application is disabled, the Cloud Connector service is not executed in WinCC Runtime.

See also

MQTT settings in the WinCC Cloud Connector (Page 840)

Data transfer to the cloud via MQTT (Page 833)

How to configure the MindSphere connection via the IOT Extension (Page 852)

How to configure a cloud connection via REST (Page 859)

Settings in WinCC tag management (Page 838)

How to configure the MindSphere connection via the EU1 broker (Page 855)

Application example: WinCC data connection to the cloud (https://support.industry.siemens.com/cs/ww/en/view/109760955)

11.9 How to configure the MindSphere connection via the IOT Extension

Introduction

You specify the URL and access settings of the cloud used in the "WinCC Cloud Connector Configuration" dialog box.

When connecting with MindSphere, the CA certificate installed with WinCC is used.

For additional authentication, MindSphere creates a user name and a password.

Application example

You can find a detailed example of how to configure the data connection for the various cloud providers on the Internet:

- Application example 109760955: "WinCC data connection in the cloud (https://support.industry.siemens.com/cs/ww/en/view/109760955)"

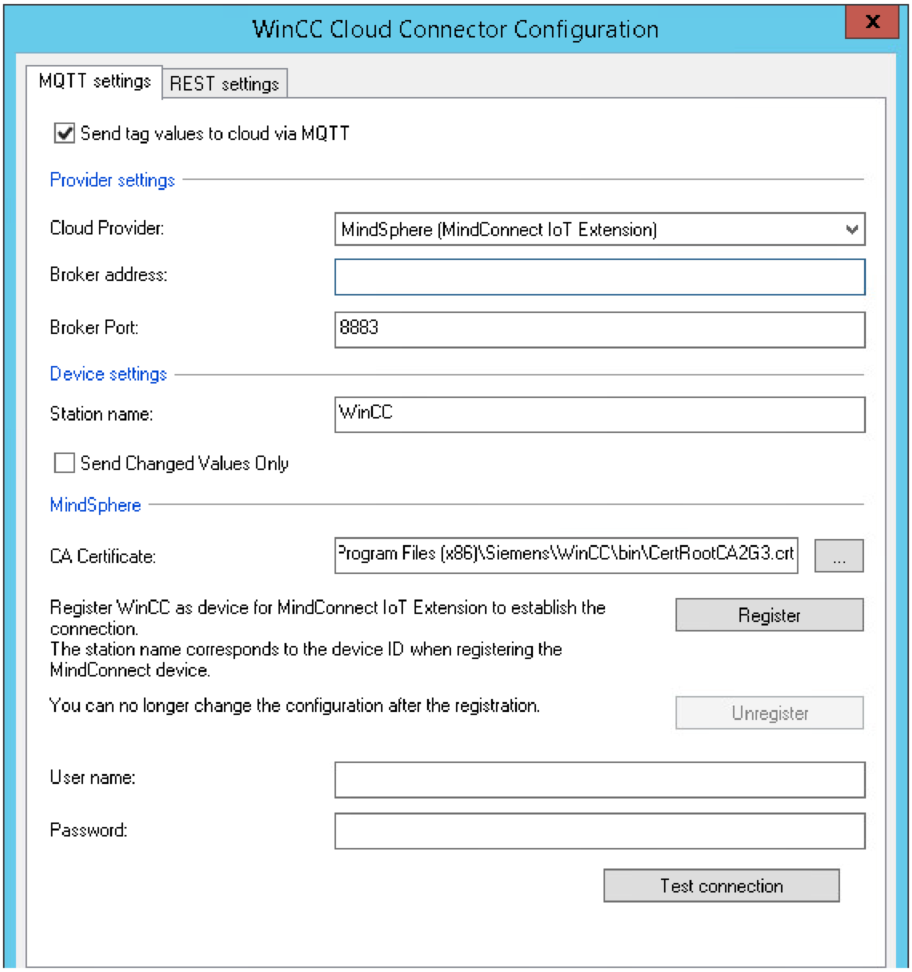
Requirement

- The "WinCC Cloud" option is enabled in WinCC Tag Management for the tags to be transferred.

Procedure

1. Open the WinCC Explorer.
2. In the shortcut menu of "Cloud Connector", select the entry "Cloud Connector Settings".
   The "WinCC Cloud Connector Configuration" dialog box opens.

3. In the "MQTT Settings" tab, select the "Send tag values to cloud via MQTT" option.



WinCC Cloud Connector Configuration

☑ Send tag values to cloud via MQTT

Provider settings

Cloud Provider: MindSphere (MindConnect IoT Extension)

Broker address:

Broker Port: 8883

Device settings

Station name: WinCC

☐ Send Changed Values Only

MindSphere

CA Certificate: Program Files (x86)\Siemens\WinCC\bin\CertRootCA2G3.crt

Register WinCC as device for MindConnect IoT Extension to establish the connection.
The station name corresponds to the device ID when registering the MindConnect device.

You can no longer change the configuration after the registration.

User name:

Password:

4. Specify the connection data:
   - Cloud Provider: "MindSphere (MindConnect IoT Extension)"
   - Broker address
   - Station name

   The default port 8883 cannot be changed.
   To reduce the load during data transfer, select the option "Only send changed values".
5. Open the MindSphere configuration in the browser.
6. Switch to the editor of the "MindConnect IoT Extension".
   Select the device registration under "Devices".
7. To create a new device, enter the station name.
   The station name in the Cloud Connector and the device name in the editor "MindConnect IoT Extension" must match.
   - The WinCC station is created as device.
   - The status "Waiting for connection" is displayed.

8. Go to the "WinCC Cloud Connector configuration" dialog box and click on the "Register" button.
9. Go back to the editor "MindConnect IoT Extension" in the browser window.
   To complete the registration, click on the "Accept".
   – If the registration has been accepted, the "Register" button is grayed out in the Cloud Connector.
   – The "Unregister" button is activated.
   – User name and password are set by the editor "MindConnect IoT Extension" and displayed in the Cloud Connector.
     When you change the password in the editor "MindConnect IoT Extension", you must apply the new password in the Cloud Connector.
10. To test the connection settings, click on the "Test connection" button in the Cloud Connector.
11. Click "OK" to close the dialog.
12. If the application "Cloud Connector" is not yet activated in the startup list of the computer, you will be prompted to activate it.
    Confirm the activation with "Yes".
    If the application is disabled, the Cloud Connector service is not executed in WinCC Runtime.

See also

MQTT settings in the WinCC Cloud Connector (Page 840)

Settings in WinCC tag management (Page 838)

How to configure a cloud connection via MQTT (Page 848)

How to configure a cloud connection via REST (Page 859)

Application example: WinCC data connection to the cloud (https://support.industry.siemens.com/cs/ww/en/view/109760955)

11.10 How to configure the MindSphere connection via the EU1 broker

Introduction

You specify the URL and access settings of the cloud used in the "WinCC Cloud Connector Configuration" dialog box.

When connecting with MindSphere, the CA certificate installed with WinCC is used.

The connection with MindSphere via the EU1 broker is an alternative to the connection via the IOT Extension with a small delay.

Application example

You can find a detailed example of how to configure the data connection for the various cloud providers on the Internet:

- Application example 109760955: "WinCC data connection in the cloud (https://support.industry.siemens.com/cs/ww/en/view/109760955)"

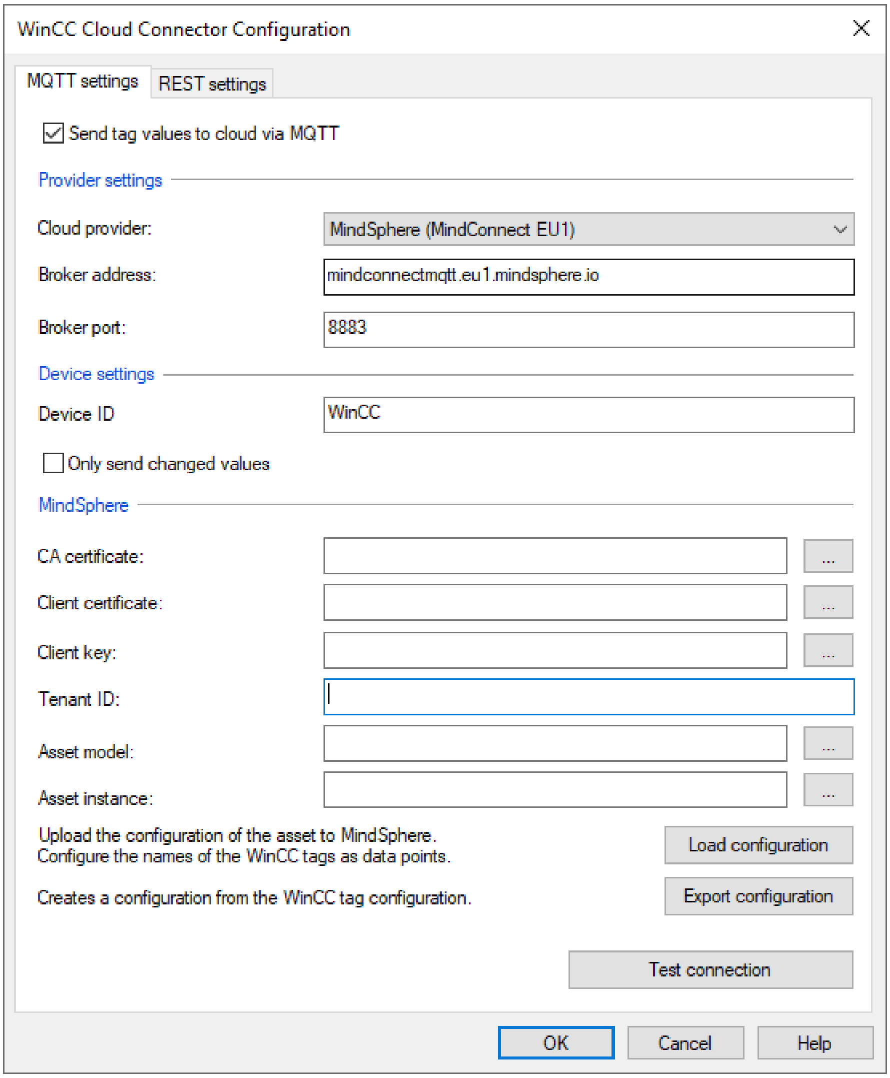
Requirement

- The "WinCC Cloud" option is enabled in WinCC Tag Management for the tags to be transferred.
- The device ID is registered in MindSphere.

Procedure

1. Open the WinCC Explorer.
2. In the shortcut menu of "Cloud Connector", select the entry "Cloud Connector Settings".
   The "WinCC Cloud Connector Configuration" dialog box opens.

3. In the "MQTT Settings" tab, select the "Send tag values to cloud via MQTT" option.



WinCC Cloud Connector Configuration

MQTT settings | REST settings

☑ Send tag values to cloud via MQTT

Provider settings

Cloud provider: MindSphere (MindConnect EU1)

Broker address: mindconnectmqtt.eu1.mindsphere.io

Broker port: 8883

Device settings

Device ID WinCC

☐ Only send changed values

MindSphere

CA certificate:

Client certificate:

Client key:

Tenant ID:

Asset model:

Asset instance:

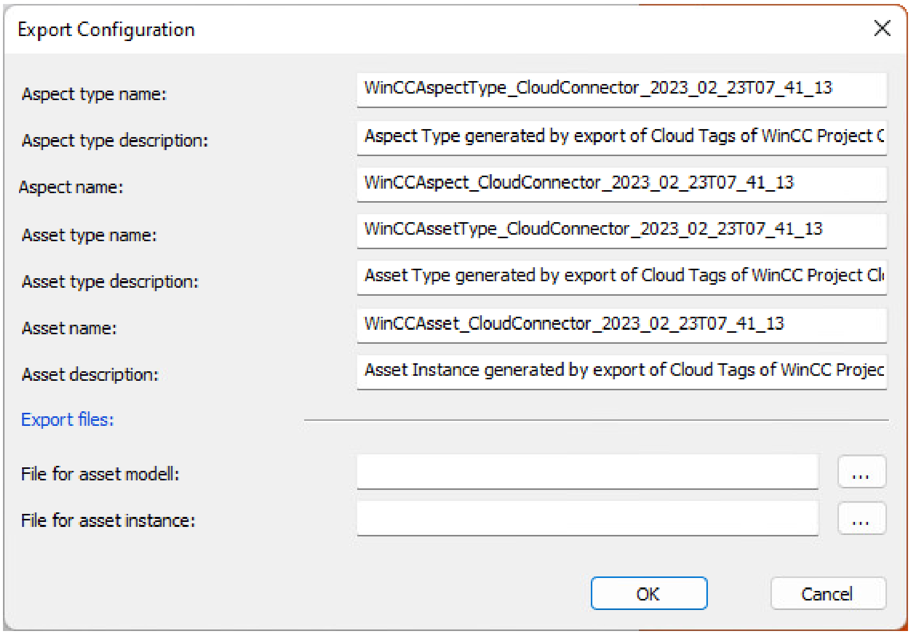
Upload the configuration of the asset to MindSphere.
Configure the names of the WinCC tags as data points.

Creates a configuration from the WinCC tag configuration.

4. Specify the provider and device settings:
   - Cloud Provider: "MindSphere (MindConnect EU1)"
   - Device ID

   The broker address is entered automatically. The default port 8883 cannot be changed.
   To reduce the load during data transfer, select the option "Only send changed values".
5. Select the certificates and the client key.
6. Enter the Tenant-ID.
7. If you already have Asset Model and Asset Instance available, select them and continue with Step 12.

8. Click the "Export configuration" button to generate Asset Model and Asset Instance from the WinCC tag configuration.



Export Configuration

Aspect type name: WinCCAspectType_CloudConnector_2023_02_23T07_41_13

Aspect type description: Aspect Type generated by export of Cloud Tags of WinCC Project C

Aspect name: WinCCAspect_CloudConnector_2023_02_23T07_41_13

Asset type name: WinCCAssetType_CloudConnector_2023_02_23T07_41_13

Asset type description: Asset Type generated by export of Cloud Tags of WinCC Project Cl

Asset name: WinCCAsset_CloudConnector_2023_02_23T07_41_13

Asset description: Asset Instance generated by export of Cloud Tags of WinCC Projec

Export files:

File for asset modell:

File for asset instance:

9. Enter your MindSphere-specific values:
   - Aspect Type Name
   - Aspect Type Description
   - Aspect Name
   - Asset Type Name
   - Asset Type Description
   - Asset Name
   - Asset Description
10. Select the file name and path for the Asset Model and Asset Instance.
11. Confirm your inputs with the "OK" button.
    Asset Model and Asset Instance are automatically entered in the input fields of the MQTT settings.
    If an Asset Model or an Asset Instance has already been generated previously, confirm overwriting of the data.
12. To test the connection settings, click on the "Test connection" button.
13. Load the configuration of the Assets with the "Load configuration" button to MindSphere.
14. Click "OK" to close the dialog.
15. If the application "Cloud Connector" is not yet activated in the startup list of the computer, you will be prompted to activate it.
    Confirm the activation with "Yes".
    If the application is disabled, the Cloud Connector service is not executed in WinCC Runtime.

See also

Application example: WinCC data connection in the cloud (https://support.industry.siemens.com/cs/ww/en/view/109760955)

11.11 How to configure a cloud connection via REST

Introduction

You can send data to an HTTP provider using the REST protocol.

You specify the URL and access settings of the cloud used in the "WinCC Cloud Connector Configuration" dialog box.

Alternative procedure

To configure a cloud connection via MQTT, follow the instructions below:

- How to configure a cloud connection via MQTT (Page 848)

If you select "MindSphere (MindConnect IoT Extension)" or ""MindSphere - MindConnect EU1" as Cloud Provider, follow the relevant instructions at:

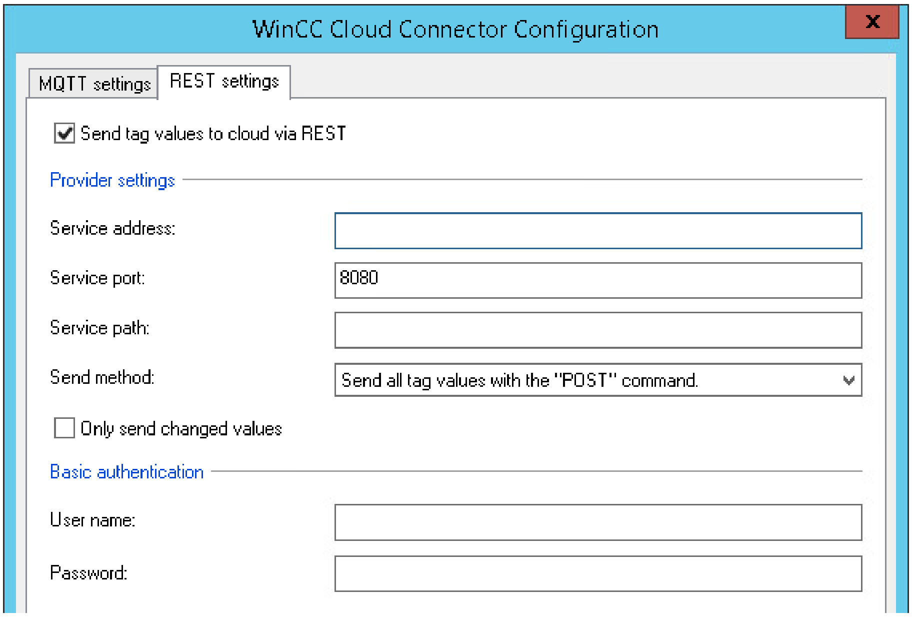
- How to configure the MindSphere connection via the IOT Extension (Page 852)
- How to configure the MindSphere connection via the EU1 broker (Page 855)

Requirement

- The "WinCC Cloud" option is enabled in WinCC Tag Management for the tags to be transferred.

Procedure

1. Open the WinCC Explorer.
2. In the shortcut menu of "Cloud Connector", select the entry "Cloud Connector Settings". The "WinCC Cloud Connector Configuration" dialog box opens.
3. In the "REST settings" tab, select the "Send tag values to cloud via REST" option.



WinCC Cloud Connector Configuration

☑ Send tag values to cloud via REST

Provider settings

Service address:

Service port: 8080

Service path:

Send method: Send all tag values with the "POST" command.

☐ Only send changed values

Basic authentication

User name:

Password:

4. Specify the connection data:
   - Service address
   - Service port
   - Service path
   - Send method

   To reduce the load during data transfer, select the option "Only send changed values".
5. Enter the access data for the REST server in the "User name" and "Password" fields.
   If you change the password on the REST server, you must apply the new password in the Cloud Connector.
6. Close the dialog box with "OK".
7. If the application "Cloud Connector" is not yet activated in the startup list of the computer, you will be prompted to activate it.
   Confirm the activation with "Yes".
   If the application is disabled, the Cloud Connector service is not executed in WinCC Runtime.

See also

How to configure a cloud connection via MQTT (Page 848)

MQTT settings in the WinCC Cloud Connector (Page 840)

Settings in WinCC tag management (Page 838)

How to configure the MindSphere connection via the IOT Extension (Page 852)

REST settings in the WinCC Cloud Connector (Page 846)

How to configure the MindSphere connection via the EU1 broker (Page 855)

11.12 Diagnostics of the cloud connection

Introduction

WinCC supports you in the diagnostics with performance tags and the output of messages in log files.

Performance tags for connection monitoring

When you create a WinCC project, the system tags are created in the internal "Performance" tag group. Additional information:

- "Working with WinCC > Working with Projects > Making Settings for Runtime > System diagnostics with performance tags"

You can select the following system tags for performance analysis:

System tag	Description
@PRF_CLDCN_RESET	The reset tag resets the values of the following performance tags: • @PRF_CLDCN_TAG_FAILED_WRITES_TOTAL • @PRF_CLDCN_TAG_WRITES_TOTAL
@PRF_CLDCN_TAG_FAILED_WRITES_TOTAL	Number of transmitted tags that were not acknowledged by the cloud
@PRF_CLDCN_TAG_WRITES_PER_SECOND	Number of transferred tags per second
@PRF_CLDCN_TAG_WRITES_TOTAL	Total number of tags transferred over a connection

Diagnostics file

The "CCCloudConnect.log" file is created in the WinCC installation path in the "diagnose" folder.

Eclipse Mosquitto

Eclipse Mosquitto error codes are designated with "MOSQ" in the diagnostics file.

You can find additional information in the "libmosquitto API" documentation.

- https://mosquitto.org/ (https://mosquitto.org/)

See also

Data transfer to the cloud via MQTT (Page 833)

MQTT settings in the WinCC Cloud Connector (Page 840)

Settings in WinCC tag management (Page 838)

Internet: https://mosquitto.org/ (https://mosquitto.org/)

Index