



SMART
SAFETY.

Manual

OPC UA Server[®]

SILworX



All of the HIMA products mentioned in this manual are trademark protected. Unless noted otherwise, this also applies to other manufacturers and their respective products referred to herein.

HIQuad®, HIQuad X®, HIMax®, HIMatrix®, SILworX®, XMR®, HICore® and FlexSILon® are registered trademarks of HIMA Paul Hildebrandt GmbH.

All of the technical specifications and information in this manual were prepared with great care and effective control measures were employed for their compilation. For questions, please contact HIMA directly. HIMA appreciates any suggestion on which information should be included in the manual.

Equipment subject to change without notice. HIMA also reserves the right to modify the written material without prior notice.

All the current manuals can be obtained upon request by sending an e-mail to:
documentation@hima.com.

© Copyright 2020, HIMA Paul Hildebrandt GmbH
All rights reserved.

Kontakt

HIMA Paul Hildebrandt GmbH

Postfach 1261

68777 Brühl

Tel.: +49 6202 709-0

Fax: +49 6202 709-107

E-Mail: info@hima.com

Document designation	Description
HI 801 548D, Rev. 12.00 (2025)	German original document
HI 800 551 E, Rev. 12.00.00 (2027)	English translation of the German original document

Table of Contents

1 Introduction	6
1.1 Structure and Use of This Manual	7
1.1.1 Target Audience and Required Competence	7
2 Safety	8
2.1 Intended Use	8
2.2 Residual Risk	8
2.3 Safety Precautions	8
2.4 Emergency Information	8
2.5 Automation Security for HIMA Systems	9
2.5.1 Product Properties	9
2.5.2 Risk Analysis and Planning	10
2.6 Writing Conventions	10
2.6.1 Safety Notices	11
2.6.2 Operating Tips	11
2.7 Safety Lifecycle Services	12
3 OPC Unified Architecture	13
3.1 System Requirements and List of Components for OPC UA Server	13
4 Concept of the HIMA OPC UA Servers	15
4.1 Redundancy of OPC UA Servers	17
5 Configuring a HIMA OPC UA Server	18
5.1 Configuration in SILworX	18
5.2 Configuring the Namespace	19
5.2.1 OPC UA Object in the Namespace	19
5.2.1.1 Creating OPC UA Objects	19
5.2.1.2 Editing OPC UA Objects	20
5.2.2 OPC UA Types in the Namespace	20
5.2.2.1 Creating OPC UA Types	20
5.2.2.2 Editing OPC UA Types	20
5.2.3 OPC UA Variables in the Namespace	20
5.2.3.1 Creating OPC UA Variables	20

5.2.3.2 Using Drag&Drop to Create OPC UA Variables	21
5.2.3.3 Selecting the Data Type of the OPC UA Variable	21
6 OPC UA Set	22
6.1 OPC UA Set Editor	22
6.1.1 Tab Set Elements	22
6.1.2 The Properties Tab	22
6.2 COM Reference	23
6.2.1 The Properties Tab	23
6.3 Namespace Editor	24
6.3.1 The Objects and Types Tabs	24
6.3.2 The Properties Tab	25
6.4 Type Reference (in Namespace)	27
6.4.1 Editing the Type Reference	27
6.4.2 Editing OPC UA Variables	27
6.4.3 Editing the Global Variable Reference	27
6.4.4 Incorrect Type References	28
6.4.4.1 Invalid Type References	28
6.4.4.2 Circular Statements	28
6.4.4.3 Circular Statements in Type Use	28
6.4.4.4 Circular Statements in the Type Derivative of Types	29
6.5 Certificates	30
6.5.1 To create the certificate editor for a resource	31
6.5.1.1 Load Server Certificate	32
6.5.1.2 Load Client Certificate	32
7 Alarms&Events	34
7.1 Activating the A&E Functionality for a Resource	34
7.2 Data in Event Notifications Provided to an OPC UA Client	35
7.3 A&E Editor	36
7.3.1 Indication in the OPC Client	37
8 Control Panel (Online)	38
8.1 Trace Logging (Online)	39
8.2 OPC UA Server Diagnostics	40

8.2.1 Opening a Session	40
8.2.2 Activating a Session	40
8.2.3 Closing a Session	40
8.2.4 Creating a Subscription	41
8.2.5 Deleting a Subscription	41
9 Code Generation and Reload	42
9.1 Code Generation	42
9.2 Reload	42
9.3 Certificates	42
10 Version Comparison	43
10.1 OPC UA Server	43
10.1.1 OPC UA Section	43
10.1.2 Namespace Section	43
10.1.3 Node Section	43
10.1.4 Variables Section	44
10.1.5 Reference Section	44
10.1.6 EventSource Section	44
10.1.7 Condition Section	44
10.2 OPC UA Certificates	45
10.2.1 Own Section	45
10.2.2 Client Section	45

1 Introduction

This manual describes the configuration of the OPC UA Server in SILworX for controllers of the system families HIMax, HIMatrix and HIQuad X.

The following conditions must be met to safely install and start up the system and to ensure safety during their operation and maintenance:

- Knowledge of regulations.
- Proper technical implementation of the safety instructions detailed in this manual performed by qualified personnel.

HIMA will not be held liable for severe personal injuries, damage to property or the environment caused by any of the following:

- Unqualified personnel working on or with the systems.
- De-activation or bypassing of safety functions.
- Failure to comply with the instructions detailed in this manual.

HIMA develops, manufactures and tests the HIMA systems in compliance with the pertinent safety standards and regulations. The use of the systems is only allowed if the following requirements are met:

- They are only used for the intended applications.
- They are operated under the specified environmental conditions.

1.1 Structure and Use of This Manual

The manual contains the following chapters:

- Introduction and writing conventions
- Safety
- Product Description

Additionally, the following documents must be taken into account:

Name	Content	Document no.
HIMax system manual	Hardware description of the HIMax system	HI 801 001 E
HIMax safety manual	Safety functions of the HIMax system	HI 801 003 E
HIMatrix safety manual	Safety functions of the HIMatrix system	HI 800 023 E
HIMatrix compact system manual	Hardware description of the HIMatrix system	HI 800 141 E
HIMatrix modular system manual	Hardware description of the HIMatrix system	HI 800 191 E
HIQuad X system manual	Hardware description of the HIQuad X system	HI 803 211 E
HIQuad X safety manual	Safety functions of the HIQuad X system	HI 803 209 E
Communication manual	Description of the communication protocols.	HI 801 101 E
Getting started with SILworX	Introduction to SILworX.	HI 801 103 E

The current manuals can be obtained upon request by sending an e-mail to: documentation@hima.com. Registered customers can download the product documentation from the HIMA Extranet.

1.1.1 Target Audience and Required Competence

This document is aimed at the planners, design engineers, programmers and the persons authorized to start up, operate and maintain the automation systems. Specialized knowledge of safety-related automation systems is required.

2 Safety

All safety information, notices and instructions specified in this document must be strictly observed. The HIMA systems may only be used if all guidelines and safety instructions are adhered to.

The HIMA controllers are operated with SELV or PELV. No imminent risk results from the controllers themselves. Use in the Ex zone is only permitted if additional measures are taken.

2.1 Intended Use

To use the HIMA systems, all pertinent requirements must be met, see additionally applicable manuals listed in Chapter 1.1.

2.2 Residual Risk

No imminent risk results from a HIMA system itself.

Residual risk may result from:

- Faults related to engineering.
- Faults in the user program.
- Faults related to the wiring.

2.3 Safety Precautions

Observe all local safety requirements and use the protective equipment required on site.

2.4 Emergency Information

A HIMA system is a part of the safety equipment of an overall system. If the controller fails, the system enters the safe state.

In case of emergency, no action that may prevent the HIMA systems from operating safely is permitted.

2.5 Automation Security for HIMA Systems

HIMA distinguishes between the terms *safety*, which refers to functional safety, and *security*, which refers to the system protection against manipulation.

Industrial controllers (PES) must be protected against IT-specific problem sources, for instance:

- Inadequate protection of IT equipment (e.g., open WLAN, obsolete operating systems).
- Lack of awareness of proper use of the equipment (e.g., USB sticks).
- Direct access to protected areas.
- Attackers inside the company premises.
- Attackers via communication networks inside and outside the company premises.

HIMA safety systems are composed of the following parts to be protected:

- Safety-related automation system.
- PADT.
- Optional X-OPC Server (on a host PC)
- Optional communication connections to external systems.

2.5.1 Product Properties

The HIMA systems with basic settings already fulfils the requirements for automation security.

Protective mechanisms for preventing unintentional or unapproved modifications to the safety system are integrated into the controllers and the programming tool:

- Each change to the user program or controller configuration results in a new configuration CRC.
- Online changes of the safety parameters can be deactivated in the controller. Therefore, changes to the safety parameters are only possible by performing a download or reload.
- The user can set up a user management scheme to increase security. This scheme is used to specify the user groups, user accounts, access permissions for PADT and controllers (PES) for each project. In the user management scheme, the user can define if an authorization is required to open the project and log in to a controller.
- The data of a controller can only be accessed if the user projects loaded in the PADT and controller are the same. The CRCs must be identical (archive maintenance!).
- A physical connection between PADT and controller (PES) is not required during operation and must be interrupted for security reasons. The PADT can be reconnected to the controller for diagnostic and maintenance purposes.

The requirements of the safety and security standards must be complied with. The operator is responsible for authorizing personnel and implementing the required protective actions.

WARNING

Physical injury possible due to unauthorized manipulation of the controllers!

Protect the controllers against unauthorized access:

- Change the default settings for login and password.
- Supervise access to controllers and PADTs!
- For further protection measures, refer to the automation security manual (HI 801 373 E).

2.5.2 Risk Analysis and Planning

Security is a process, not a product. Maintained network maps, for instance, help to ensure that secure networks are permanently separated from public networks. It is recommended to only have one well-defined connection, e.g., via a firewall or a DMZ (demilitarized zone).

Careful planning should identify the necessary measures. The required measures are to be implemented after the risk analysis is completed, and may include:

- Assignment of access permissions for user groups and user accounts according to the intended tasks.
- Use of passwords in accordance with the security requirements.

A periodical review of the security measures is necessary, e.g., every year.



The operator is responsible for implementing the necessary measures in a way suitable for the plant!

Refer to the HIMA automation security manual (HI 801 373 E) for more details.

2.6 Writing Conventions

To improve readability and comprehensibility, the following writing conventions are used in this document:

Format	Description
Bold	To highlight important parts. Names of buttons, menu functions and tabs that can be clicked and used in the programming tool.
<i>Italics</i>	Parameters, system variables and references.
<code>Courier</code>	Literal user inputs.
RUN	Operating states are designated by capitals.
Chapter 1.2.3	Cross-references are hyperlinks even if they are not specially marked. In the electronic document (PDF): When the mouse pointer hovers over a hyperlink, it changes its shape. Click the hyperlink to jump to the corresponding position.

Safety notices and operating tips are specially marked.

2.6.1 Safety Notices

Safety notices must be strictly observed to ensure the lowest possible risk.



The safety notices are represented as follows:

- Signal word: warning, caution, notice.
- Type and source of risk.
- Consequences arising from non-observance.
- Risk prevention.


The signal words have the following meanings:

- Warning indicates hazardous situations which, if not avoided, could result in death or serious injury.
- Caution indicates hazardous situation which, if not avoided, could result in minor or moderate injury.
- Notice indicates a hazardous situation which, if not avoided, could result in property damage.

SIGNAL WORD


-  **Type and source of risk!**
-  **Consequences arising from non-observance**
- Risk prevention.**

NOTICE

-  **Type and source of damage!**
- Damage prevention.**

2.6.2 Operating Tips

Additional information is structured as presented in the following example:

-  The text giving additional information is located here.

2.7 Safety Lifecycle Services

HIMA provides support throughout all the phases of a plant's safety lifecycle, from planning and engineering through commissioning to maintenance of safety and security.

HIMA's technical support experts are available for providing information and answering questions about our products, functional safety and automation security.

To achieve the qualification required by the safety standards, HIMA offers product or customer-specific seminars at HIMA's training center or on site at the customer's premises. The current seminar program for functional safety, automation security and HIMA products can be found on HIMA's website.

Safety Lifecycle Services:	
Onsite+ / On-Site Engineering	In close cooperation with the customer, HIMA performs changes or extensions on site.
Startup+ / Preventive Maintenance	HIMA is responsible for planning and executing preventive maintenance measures. Maintenance actions are carried out in accordance with the manufacturer's specifications and are documented for the customer.
Lifecycle+ / Lifecycle Management	As part of its lifecycle management processes, HIMA analyzes the current status of all installed systems and develops specific recommendations for maintenance, upgrading and migration.
Hotline+ / 24 h hotline	HIMA's safety engineers are available by telephone around the clock to help solve problems.
Standby+ / 24 h call-out service	Faults that cannot be resolved over the phone are processed by HIMA's specialists within the time frame specified in the contract.
Logistic+ / 24 h spare parts service	HIMA maintains an inventory of necessary spare parts and guarantees quick, long-term availability.

Contact details:	
Safety Lifecycle Services	https://www.hima.com/en/about-hima/contacts-worldwide/
Technical support	https://www.hima.com/en/products-services/support/
Seminar program	https://www.hima.com/en/products-services/seminars/

3 OPC Unified Architecture

OPC UA stands for *Open Platform Communications Unified Architecture* and is specified for communication in industrial automation and connectivity between products from different manufacturers.

3.1 System Requirements and List of Components for OPC UA Server

Element	Description
Programming Tool	SILworX V12 and higher
Controller and operating system	<ul style="list-style-type: none"> • HIMax CPU BS \geq V13, COM BS \geq V13 • HIMatrix CPU BS \geq V17, COM BS \geq V22 • HIQuad X CPU BS \geq V13, COM BS \geq V13
License	One activation license is required for each COM module on which the OPC UA protocol is to run.
Number of OPC UA Servers	<p>The number of OPC UA Servers on a controller depends on the number of communication modules. One OPC UA Server can be operated on each communication module:</p> <ul style="list-style-type: none"> • HIMax: 20 (with 20 COM modules) • HIMatrix: 1 (1 COM module permanently installed) • HIQuad H51X: 10 (with 10 COM modules) • HIQuad H41X: 2 (with 2 COM modules)
Number of OPC UA Clients	The OPC UA Server can run up to 4 OPC UA Client sessions in parallel.
Safety-related	The OPC UA Server enables HIMA controllers to exchange process data with third-party systems that have OPC UA Client functionality. The OPC UA Server must not be used for safety-related communication.
Automation security.	<p>Automation security is a central issue of the OPC UA specifications.</p> <p>The OPC UA Server supports the following security profiles:</p> <ul style="list-style-type: none"> • SecurityPolicy - None. • SecurityPolicy [B] • Basic256Sha256 <p>The OPC UA Server offers different endpoints for OPC UA connections depending on the configuration (server certificate and <i>Online allow encrypted connections</i> switch):</p> <ul style="list-style-type: none"> • None - None (endpoint for unencrypted OPC UA connections) Condition: The <i>Online allow encrypted connections</i> switch is disabled. • Basic256Sha256 - Sign (endpoint for OPC UA connections with signature verification) Condition: The <i>Online allow encrypted connections</i> switch is disabled and a configured server certificate. • Basic256Sha256 - Sign & Encrypt (endpoint for encrypted OPC UA connections with signature verification). Condition: A configured server certificate.
Interfaces	Ethernet interfaces of the communication module.
Number of namespaces	1 per OPC UA Server.

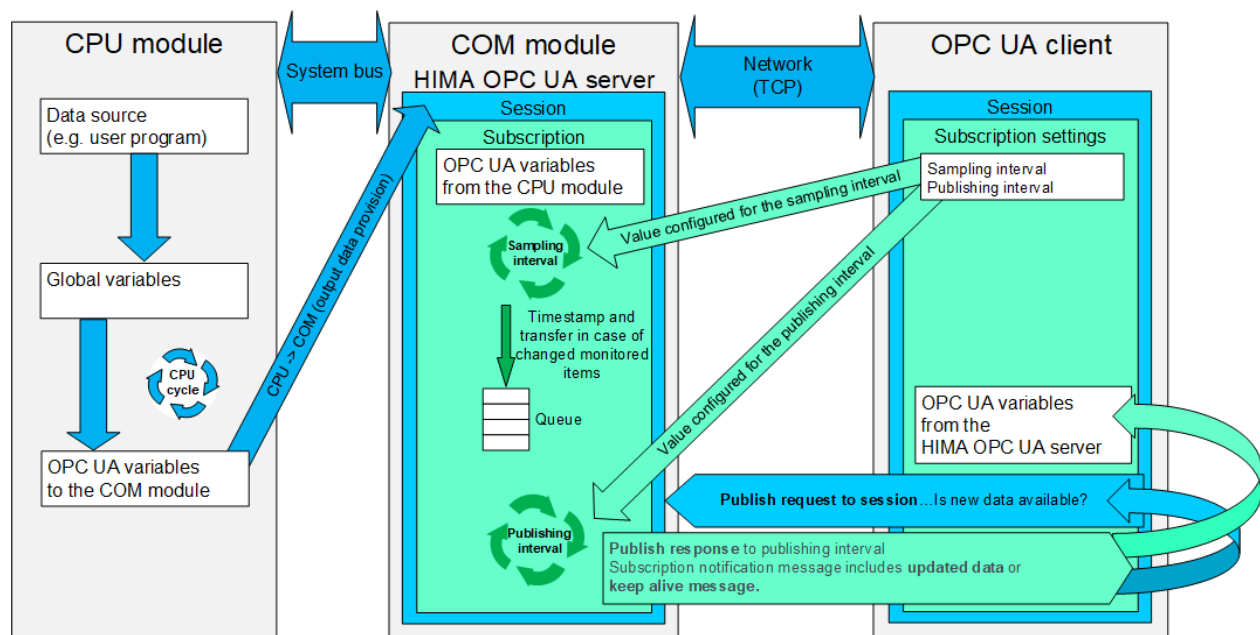
Element	Description
Number of Nodes	<ul style="list-style-type: none">• Variables: Max. 10000• Objects: Max. 4000• Types: Max. 1000
Number of references	45 000
Reload changes	Up to 1000 new nodes and up to 1000 attribute changes to existing nodes in the reload configuration.
Event configuration	The event configuration can be activated for a maximum of 4 OPC UA and X-OPC Servers.
TCP Port	4840

4 Concept of the HIMA OPC UA Servers




The HIMA OPC UA Server runs in a communication module, directly on the controller. An OPC UA client connects directly to the communication module.

The OPC UA client can subscribe to variables and let the OPC UA Server monitor the variables in these subscriptions. The OPC UA client is only notified when these variables change. This mechanism reduces the amount of variables to be transferred and results in a significant reduction of the required bandwidth.

The following figure shows the process from the data source in the CPU module to the reception of the OPC UA variables in the subscriptions.



Element	Description
CPU module	Processor module on which the input/output data and the user program are processed.
COM module	Communication module on which the HIMA OPC UA Server is running.
OPC UA client	The OPC UA client can access the process data received by the OPC UA Server, providing it, e.g., to a control system. The two parameters Sampling Interval and Publishing Interval are used to configure the access to the subscriptions in the OPC UA client.
System bus	The system bus connects the CPU and COM module to one another.
Network (TCP)	The Ethernet network connects the COM module and the OPC UA client to one another.
Data source	Process data of the controller (e.g., from the user program or from hardware inputs).
Global variable	Global variables can be created at different levels of the SILworX project tree and apply to all subordinate levels within this scope. The global variables are the connection element between the OPC UA Server and the data sources of the controller.
OPC UA variable	An OPC UA variable has any IEC data type supported by SILworX. Additionally, access permissions (read, write) can be configured for each OPC UA variable.

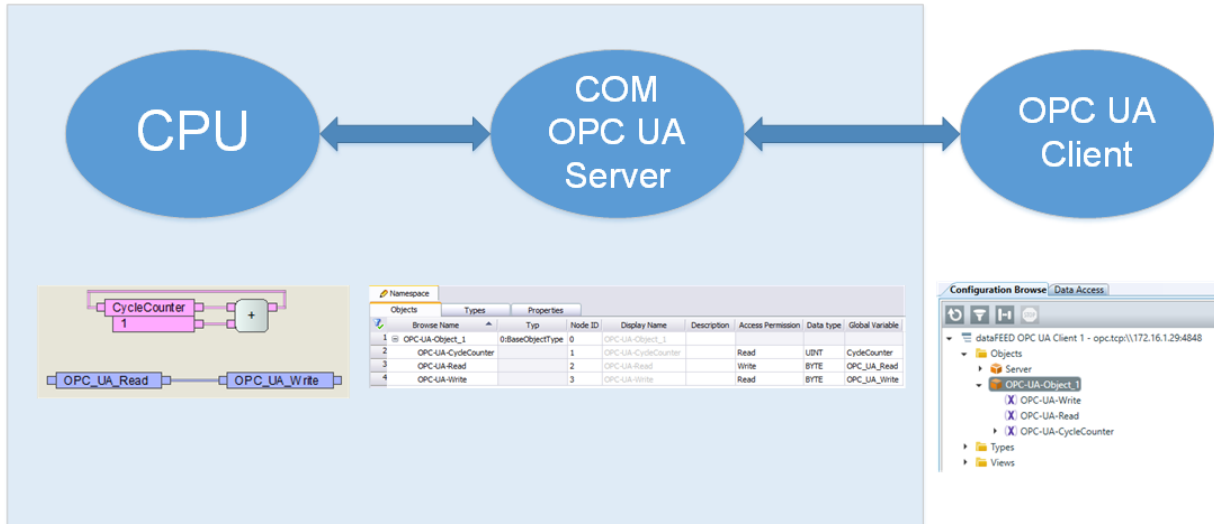
Element	Description
Sessions	The session is the connection between the OPC UA client and the OPC UA Server. Max. 4 OPC UA client sessions for each OPC UA Server.
Subscription	<p>An OPC UA client subscribe to a selection of monitored items, so-called subscriptions. The Server notifies the client as soon as the monitored items change.</p> <p> Multiple subscriptions allow different settings and optimized load balancing.</p> <p>Max. 20 subscriptions for each OPC UA Server. Max. 10 subscriptions for each session.</p>
Monitored items	A monitored item is a variable that is monitored within a subscription.
Queue	<p>Only changed monitored item values are transferred to the queue. Each monitored element is assigned a queue</p> <p>Users can set the queue size for each monitored item.</p> <p>Range of values: 1...2</p>
CPU cycle	<p>In a simplified overview, a processor module cycle (CPU cycle) runs through the following phases:</p> <ol style="list-style-type: none"> 1. Processing of the input data. 2. Processing of the user program. 3. Provision of the output data. <p>For further details, refer to the corresponding system manual.</p>
Sampling interval	<p>Users can set the sampling interval for the monitored items in the OPC UA client. All of the monitored items within a subscription have the same sampling interval. If a subscription already includes a monitored item, each monitored item added to the existing set will receive the sampling interval applying to the subscription.</p> <p>The use of one sampling interval for each monitored item, as defined in the OPC Unified Architecture Specification Part 4: Services Specification, would overload the COM in terms of computing time if large amounts of process data were to be calculated.</p> <p>Range of values: 10.0...10000.0 ms</p> <p> The sampling interval has a significant effect on the COM load. The shorter the sampling interval, the larger the COM load.</p>
Publishing interval	<p>The cyclic rate at which the Server sends notification messages to clients. Users can set the publishing interval for a subscription.</p> <p>The publishing interval serves as a 'brake' to prevent data from being sent to the client too often.</p> <p>Range of values: 10.0 ... 10000.0 ms</p> <p> The shorter the publishing interval, the faster the client-server communication and the larger the client load.</p>
Keep alive	<p>Users can set the <i>Max Keep Alive Count</i> for a subscription.</p> <p>If a queue contains no new data and therefore no data transfer from the Server is required, the Server triggers an empty <i>publish response telegram</i> after the <i>keep alive timeout</i>.</p>

4.1 Redundancy of OPC UA Servers

A redundant OPC UA Server connection can be implemented by configuring a second OPC UA Server. Redundancy of the OPC UA variables must be implemented in the redundant OPC UA clients.

5 Configuring a HIMA OPC UA Server

In this example a connection is created between a HIMA OPC UA Server and an OPC UA Client. This is illustrated in the following overview:



5.1 Configuration in SILworX

Creating an OPC UA Server in SILworX.

To create a new **OPC UA Server Set** in SILworX

- In the structure tree, open the **Resource** in which the OPC UA Server is to be created.
- Right-click **Protocols** and select **New, OPC UA Server Set**.
A new *OPC UA Server Set* object is created.

To open the **OPC UA Server Set Editor** in SILworX

- Right-click **OPC UA Server Set** and select **Edit** from the context menu.
- The "OPC UA Set" auf Seite 22 contains the **Set Elements** and **Properties** tabs.

To select the **COM Module**.

- Select the **Set Elements** tab.
- Right-click **COM Reference** and select **Edit** to open the *COM Reference Editor*.
- In the *COM Reference Editor*, select the **COM Module** on which the OPC UA Server should run.



Only one OPC UA Server can be created per COM module.

5.2 Configuring the Namespace

Select **Edit** from the context menu of the namespace to open the Namespace Editor. The dialog box contains the following tabs:

- Objects
- Types
- Properties

To open the **Namespace Editor** in the OPC UA Server Set

- Right-click **Namespace** and select **Edit** to open the Namespace Editor.

Namespace								
Objects		Types		Properties				
	Browse Name	Type	Node ID	Display Name	Description	Access Permission	Data type	Global Variable
1	OPC-UA-Object_1	0:BaseObjectType	0	OPC-UA-Object_1				
2	OPC-UA-CycleCounter		1	OPC-UA-CycleCounter		Read	UINT	CycleCounter
3	OPC-UA-Read		2	OPC-UA-Read		Write	BYTE	OPC-UA-Read
4	OPC-UA-Write		3	OPC-UA-Write		Read	BYTE	OPC-UA-Write

5.2.1 OPC UA Object in the Namespace

The element OPC UA Object has a structuring function in the address space.

For further information, refer to the *OPC Unified Architecture Specification, Part 3: Address Space Model, Chapter 5.5*.

5.2.1.1 Creating OPC UA Objects

To add an **OPC UA Object** at the top level

- Select the **Objects** tab in the Namespace Editor.
- Right-click a free space in the workspace of the Namespace Editor and select **Create Root Element** from the context menu.
The dialog box for selecting objects appears.
- In the dialog bot, select **OPC UA Object** to create an OPC UA object.
- Enter the type name directly in the **Type** table cell or click the ... button and select the required type from the list.

If required, assign a name and click **OK** to confirm the action.

To add an **OPC UA Object** as a child node of an existing OPC UA object

- Right-click an existing **OPC UA Object** in the workspace and select **New** from the context menu.
The dialog box for selecting objects appears.
- In the dialog box, select **OPC UA Object** to create an OPC UA variable.

If required, assign a name and click **OK** to confirm the action.

5.2.1.2 Editing OPC UA Objects

OPC UA objects including all child elements can be deleted, copied or cut and then inserted again at possible insertion positions (see above).



Double-click the respective object to select or change the browse name of the OPC UA objects.

5.2.2 OPC UA Types in the Namespace

The **OPC UA Type** element allows preconfigured structures to be used when modeling OPC UA objects and OPC UA variables.

For further information, refer to the *OPC Unified Architecture Specification, Part 3: Address Space Model, Chapter 4.5*.

5.2.2.1 Creating OPC UA Types

To add **OPC UA types** at the top level within the **Types** tab

➤ Right-click a free space in the workspace of the Namespace Editor and select **Create Root Element** from the context menu.
The dialog box for selecting objects appears.

➤ In the dialog box, select **OPC UA Type** to create an OPC UA type.

If required, assign a name and click **OK** to confirm the action.

5.2.2.2 Editing OPC UA Types

OPC UA types including all child elements can be deleted, copied or cut and then inserted again at possible insertion positions (see above).

5.2.3 OPC UA Variables in the Namespace

The OPC UA variable element represents concrete values in the information model: Properties or Data.

For further information, refer to the *OPC Unified Architecture Specification, Part 3: Address Space Model, Chapter 5.6*.

In addition to the properties available for each node, a variable also has other editable properties:

5.2.3.1 Creating OPC UA Variables

To add an **OPC UA Variable** at the top level within the **Objects** tab

➤ Select the **Objects** tab in the Namespace Editor.

➤ Right-click a free space in the workspace of the Namespace Editor and select **Create Root Element** from the context menu.
The dialog box for selecting objects appears.

➤ In the dialog box, select **OPC UA Variable** to create an OPC UA variable.

If required, assign a name and click **OK** to confirm the action.

To add an **OPC UA Variable** as a child node of an existing OPC UA object

- Right-click an existing **OPC UA Object** in the workspace and select **New** from the context menu.
The dialog box for selecting objects appears.
- In the dialog box, select **OPC UA Variable** to create an OPC UA variable.
If required, assign a name and click **OK** to confirm the action.

5.2.3.2 Using Drag&Drop to Create OPC UA Variables

To add an **OPC UA Variable** as a child node of an existing OPC UA object or OPC UA type by Drag&Drop

- Select one or more **global variables** in the Object Panel and drag them onto an existing **OPC UA Object** or **OPC UA Type**.
An **OPC UA variable** is created for each global variable and is assigned the name, description and data type of the global variable. A reference of the **OPC UA variable** to the global variable is created in the **Objects** tab.

5.2.3.3 Selecting the Data Type of the OPC UA Variable

To select the data type of the **OPC UA Variable**

- Double-click the respective **Data Type** field and select the required data type from the drop-down list.

To connect the **OPC UA Variables** to global variables

- Select a **Global Variable** in the Object Panel and drag it onto the Global Variable column to connect it to the required OPC UA variable.

To select the access permissions of the **OPC UA Variable**

- Double-click the respective **Access Permission** field and select the required access permission from the drop-down list.

See also [Certificate](#)

6 OPC UA Set

Users can create any number of OPC UA Server Sets in the *Protocols* folder. However, the number of OPC UA Server Sets that can actually be used, is limited by the number of COM modules within the controller.

6.1 OPC UA Set Editor

Select **Edit** from the context menu of the OPC UA Server Set to open the editor for the OPC UA Server Set. The dialog box contains the following tabs:

6.1.1 Tab Set Elements

The *Set Elements* tab of an OPC UA Server Set lists the elements that are displayed below the Set in the project tree. It is used to create and delete elements or edit their names.

[COM Reference](#)

[Namespace](#)

The object "Certificates" auf Seite 30 can also be created by the users.

6.1.2 The Properties Tab

The *Properties* tab contains the following tabs:

Element	Description
Type	OPC UA Server Set.
Name	Any unique name for an OPC UA Server Set. Type: String Length: 1...120 ASCII characters
Max. μ P Budget in [%]	Maximum μ P budget of the COM module that can be used when processing the protocol. Range of values: 0...100% Default value: 50%
Warning if μ P budget exceeded, in [%]	Warning threshold for the μ P budget of the COM module. If this threshold is exceeded, the OPC UA Server must report a communication warning and display it in the online service. The value must be lower than the value of Maximum μ P Budget. Range of values: 1...100% Default value: deactivated

Element	Description
Enable Events	Activates alarms and events of a resource for this OPC UA Set.
	Activated: A&E functionality is activated.
	Deactivated: A&E functionality is deactivated.
	Default value: Activated
Only allow encrypted connections	The OPC UA Server can offer different endpoints for connections, see Automation Security . This switch is used to configure the quantity of endpoints offered.
	Activated: The OPC UA Server restricts the selection of endpoints offered to Sign & Encrypt. If no certificates are available when the switch is activated, SILworX reports an error.
	Deactivated: The OPC UA Server does not restrict the selection of endpoints offered. If no certificates are available when the switch is deactivated, SILworX issues a warning and points out the possibility of limiting connections to <i>Encrypted Only</i> . If no certificates are available, SILworX reports that unencrypted connections are allowed.
	Default value: Activated

6.2 COM Reference

By default, the OPC UA Server Set contains the *COM Reference* object for configuring the COM module on which the OPC UA Server is to run. A COM module must be selected for successful code generation.

6.2.1 The Properties Tab

Select **Edit** from the COM reference context menu to open the COM reference editor. This contains the *Properties* tab with the following parameters.

Element	Description
Type	COM Reference
Name	Any unique name for a COM Reference. Type: String Length: 1...120 ASCII characters
Module	The COM module on which the OPC UA server is running. The appropriate module can be selected from the drop-down list of available COM modules.

6.3 Namespace Editor

In the namespace, users are able to structure the data to be transferred via OPC UA in accordance with their requirements. The OPC UA address space consists of all data modeled by the user.

Namespace 0 is specified by the OPC Foundation and contains the basic definitions for OPC UA models. The number of elements that can be created directly below this namespace 0 is limited to 80. This includes:

- All OPC UA types created by the user and derived from namespace 0 types.
- All OPC UA objects and variables created by the user at the root level because they refer to namespace 0 objects.

For further information, refer to the *OPC Unified Architecture Specification, Part 3: Address Space Model*.

The namespace must contain at least one element and may contain a maximum of 10000 OPC UA variables, 4000 OPC UA objects and 1000 OPC UA types. Underrunning or overrunning the supported limit values is reported in the logbook as an error.

The nodes of the namespace are:

- OPC UA Objects
- OPC UA Types
- OPC UA Variables

Types to be used for modeling the objects can be created and edited in the Types tab.

6.3.1 The Objects and Types Tabs

The nodes in the tabs **Objects** and **Types** have the following common parameters:

Element	Description
Browse Name	<p>Maximum length: 511 bytes UTF-8 characters.</p> <p>Restrictions:</p> <ul style="list-style-type: none">• The browse name must not contain any dots (.), otherwise table imports and exports are not possible.• The browse name must not begin with an asterisk (*).• The browse name must not begin with a number followed by a colon (:). <p>Browse name qualification:</p> <p>The browse name of an OPC UA node is assigned ("qualified") to a namespace by means of a namespace index, e.g., '2:MyVariable'.</p> <p>A different namespace index is stored and displayed as entered. This is reported as an error in the logbook.</p>
Type	<p>The reference text displayed in the Type column matches the browse name of the referenced type. Consequently, it can also be qualified by a namespace (see Browse Name qualification). Currently, only the index of the namespace in which the element is located, is accepted; other namespaces are reported as errors during the validation process. An exception is the type "0:BaseObjectType", as this base type in namespace 0 is always available.</p> <ul style="list-style-type: none">• Reference to an OPC UA type where the reference text matches the browse name of the referenced type node.• OPC UA objects thus become instances of the type.• OPC UA types thus become derivatives of this type.

Element	Description	
Mandatory	The <i>Mandatory</i> property corresponds to the Mandatory modeling rule in the OPC UA address space and is only available in the Types tab for the elements (OPC UA objects and variables) of a type.	
	Activated	This child element must exist in an OPC UA object of this type.
	Deactivated	This child element is not mandatory.
Default setting: Activated		
Node ID	The Node ID is the unique numerical identifier of an element in the address space. The Node ID is assigned automatically and can be changed manually afterwards. Range of values: 0 ... 1073741823	
Display Name	The display name is the name used to display the element in the OPC UA Client. SILworX automatically adopts the browse name for the display name. This can be changed manually afterwards. If users assign the default display name, the display name matches the browse name. This has the advantage that no additional memory in the configuration file is required for the display name. In the editor, the default display name is shown in gray letters. Maximum length: 511 bytes UTF-8 characters.	
Description	General description or comment of the node. Maximum length: 4095 bytes UTF-8 characters.	
Access Permission	The access permission determines whether the current value of an OPC UA variable is readable or writable. Double-click the Access Permission field of the OPC UA variable and select the required access permission from the drop-down list.	
	Read	An OPC UA Client can read the value of the variable.
	Write	An OPC UA Client can change the value of the variable.
	Read and Write	An OPC UA Client can read and change the value of the variable. When an OPC UA Client writes a value, it is initially transferred to the CPU module. The value for reading is always the value resulting from the CPU module.
Default setting: Read		
Data type	Specification of any elementary data type. A new OPC UA variable is always created with the BOOL data type. Double-click the Data Type field of the OPC UA variable and select the required data type from the drop-down list. Default setting: BOOL	
Global Variable	References to a global variable whose data type must match the data type of the OPC UA variable. The reference text corresponds to the name of the global variable. For an OPC UA variable which references a constant global variable, the access permission <i>Read</i> is possible. Only applies to the Objects tab.	

Use the context menu functions **Save Table Content as CSV** or **Import Table Content from CSV**. Refer to the topic *CSV Import and Export* in the SILworX online help for further details.

6.3.2 The Properties Tab

The **Properties** tab of a namespace contains the following parameters:

Element	Description
Type	Namespace, not changeable.
Name	Any unique name for a namespace type: String. This name only serves to identify the namespace within SILworX and is not used in the OPC UA address space. Type: ASCII string Length: 1...120 characters
Namespace URI	Arbitrary, unique and non-empty name for a namespace URI (Uniform Resource Identifier). The namespace URI is used in the OPC UA address space to identify a namespace. Type: ASCII string Namespace length: 1...255 characters
Namespace Index	Unique, numeric value to identify the namespace. Range of values: 2...9 Default value: 2

6.4 Type Reference (in Namespace)

When an OPC UA type is referenced, SILworX automatically creates all child elements of the type in the element containing the reference. Any existing child elements remain unchanged.

If an element already has child elements whose browse names are identical to the name of child elements of the type, the type is not assigned. An error message in the logbook lists the elements that are not unique.

The verification process checks if all mandatory child elements of the assigned type exist for OPC UA objects and types. Additionally, the values of *Display Name*, *Type*, *Data Type* and *Access Permission* of all child elements, i.e., including the optional elements, must match the values of the corresponding elements in the type.



The context menu for the type selection dialog box offers the export of the table to a csv file.

6.4.1 Editing the Type Reference

Edit the **Type Reference** in the Namespace Editor for **OPC UA objects** and **OPC UA types**.

To edit the type reference of an element to an OPC UA type

- Right-click **Namespace** and select **Edit** to open the Namespace Editor.
- In the Namespace Editor, select the **Objects** or **Types** tab.
- Enter the type name directly in the **Type** table cell or click the ... button and select the required type from the list.

6.4.2 Editing OPC UA Variables

OPC UA variables can be deleted, copied or cut and then inserted again at possible insertion positions (see above).

6.4.3 Editing the Global Variable Reference

To edit the **Reference** of an OPC UA variable to a global variable

- Select the **Objects** tab in the Namespace Editor.
- Select one or more **global variables** in the Object Panel and drag them onto the **Global Variable** column of the OPC UA variable.

SILworX automatically creates assignments of IEC data types to OPC UA data types according to the following table:

IEC data types	OPC UA data types	Range of values
BOOL	Boolean	TRUE or FALSE
BYTE	BYTE	0...255
WORD	UInt16	0...65535

IEC data types	OPC UA data types	Range of values
DWord	UInt32	0...4294967295
LWORD	UInt64	0...18446744073709551615
SINT	SByte	-128...127
INT	Int16	-32768...32767
DINT	Int32	-2147483648...2147483647
LINT	Int64	-9223372036854775808...9223372036854775807
USINT	byte	0...255
UINT	UInt16	0...65535
UDINT	UInt32	0...4294967295
ULINT	UInt64	0...18446744073709551615
REAL	Float	Floating point number (IEEE 754-1985 single precision)
LREAL	Double	Floating point number (IEEE 754-1985 double precision)
TIME	UInt64	Integer from 0...18446744073709551615

For further information, refer to the *OPC Unified Architecture Specification, Part 3: Address Space Model, Chapter 8*.

6.4.4 Incorrect Type References

6.4.4.1 Invalid Type References

Invalid OPC UA type references can result from the following actions and are reported as errors during the verification process:

- Deletion or editing of types that are already in use.
- CSV import of unknown type names.

6.4.4.2 Circular Statements

Mutual or self-referencing is not possible in a hierarchical system, as this would result in an endless chain of mutual dependencies. In a case like this, SILworX cannot correctly model the object elements.




Consequently, the Types tab of the Type Selection dialog box does not include any types that would lead to mutual or self-referencing.

If such types have nevertheless been created and used unintentionally, e.g., through table import or by copying or moving elements, the incorrect circular statements must first be removed from the types before incorrect type assignments in the uses can be corrected.







6.4.4.3 Circular Statements in Type Use

If OPC UA objects in the **Types** tab reference types that were created by the user, the following states are not allowed and are reported as errors during the verification process:

- Self-referencing of a type. Nowhere in its elements may a type contain an object that refers to this same type.
Example: T1 and T2 are not valid for T1.O1.

Objects	Types	Properties
	Browse-Name	Typ
1 	T1	0:BaseObjectType
2	O1	0:BaseObjectType
3 	T2	T1
4	O1	0:BaseObjectType


- Mutual referencing of two types. Nowhere in its elements may a type contain an object that refers to a type containing elements of the first type.
Example: T1, T2 and T3 are not valid for T1.O1.

Objects	Types	Properties
	Browse-Name	Typ
1 	T1	0:BaseObjectType
2	O1	0:BaseObjectType
3 	T2	0:BaseObjectType
4 	O1	T1
5	O1	0:BaseObjectType
6 	T3	T2
7 	O1	T1
8	O1	0:BaseObjectType

6.4.4.4 Circular Statements in the Type Derivative of Types

If OPC UA types in the **Types** tab are derived from other OPC UA types, the following states are not allowed and are reported as errors during the verification process:

- Self-derivation of a type.
A type must not be derived from itself.
Example: T1, T2 and T3 are not valid for T1.

Objects	Types	Properties
	Browse-Name	Typ
1	T1	0:BaseObjectType
2	T2	T1
3	T3	T2

- Mutual derivation of two types.
A type must not be derived from a type or its derivatives for which it already serves as the basic type.

6.5 Certificates

The HIMA OPC UA Server uses the server certificate to authenticate itself to clients that want to connect to it. In turn, the clients must be authenticated by the server and for this purpose must have a client certificate which is loaded in the server's certificate management section. Only under these conditions can encrypted connections be established.

The HIMA OPC UA Server supports the following security profiles:

- SecurityPolicy - None.
- SecurityPolicy [B] - Basic256Sha256.

If the SecurityPolicy [B] - Basic256Sha256 security profile is to be used, a configured server certificate and a configured client certificate are required.

The certificates required for encrypted and signed transport of OPC UA data can be managed in the OPC UA Server Set. This includes a server certificate and at least one client certificate. If client certificates are loaded, but no server certificate, SILworX issues an error.



The required server certificate key file can be generated by the responsible system administrator.

The certificate files must comply with the ITU-T standard X.509v3. The certificate files must be encoded in accordance with the Distinguished Encoding Rules (DER) defined by the ITU-T Recommendation X.690.

Server Certificate

A server certificate is identified by a name and consists of a certificate file and a key file. Both files must be provided by the user and loaded into the project. There can only be one server certificate per OPC UA Server.

Select **Load Server Certificate** from the context menu of the Server Certificate tab to load a server certificate into the SILworX project. In the dialog box, select **Server Certificate**, assign a name, if required, and confirm these actions. In the subsequent dialog box **Load Server Certificate**, select the certificate file (with the file extension **.der**) and the key file (with the file extension **.pem**).

No changes to the certificate file or key file can be made in SILworX. If necessary, delete the certificate element from the editor and load the certificate into the project again, as described above.

The server certificate key file contains the private key, which is the counterpart of the public key from the server certificate file.

The key file must be encoded in accordance with the PEM format.

This is a Base64 encoded variant of the key, enclosed by a header and footer, which describes the key format.

Server Certificate File

The server certificate file must contain the following fields:

Element	Description
Version	The version must be V3.
Serial	Unique serial number of the certificate determined by the issuer.

Element	Description
Number	
Signature algorithm	The signature algorithm used for the fingerprint.
Fingerprint	Signature of the certificate.
Issuer	Name of the issuer of the certificate, as described in RFC 3280.
Duration of validity	Specification of the validity period, i.e., when the certificate becomes valid and when it expires.
Subject	Name of the certificate subject, as specified in RFC 3280.
Alternative subject name	The alternative subject name should contain two entries: <ol style="list-style-type: none"> 1. A Uniform Resource Identifier (URI), which should correspond to the URN of the OPC UA Server. 2. The IP address of the COM module on which the OPC UA Server is configured
Public key	The public key includes the public key itself and the encipherment algorithm used.
Key usage	The key usage of the certificate should include the following: <ul style="list-style-type: none"> • Digital signature • Non-repudiation • Key encipherment • Data encipherment
Extended key usage	The extended key usage should include the server authentication (1.3.6.1.5.5.7.3.1).
Authority Key Identifier	The authority key identifier provides information about the key that was used to sign the certificate. This is necessary for certificates issued by a certificate authority and should also be specified for self-signed certificates.

Client Certificates

A client certificate is identified by a name and consists of a certificate file. The file must be provided by the user and loaded into the project. Up to 8 client certificates can be handled per OPC UA Server.

Select **Load Client Certificate** from the context menu of the Client Certificate tab to load a client certificate into the SILworX project. In the dialog box, select **Client Certificate**, assign a name, if required, and confirm these actions. In the subsequent dialog box **Load Client Certificate**, select the certificate file (with the file extension **.der**).

No changes to the certificate file or key file can be made in SILworX. If necessary, delete the certificate element from the editor and load the certificate into the project again, as described above.

6.5.1 To create the certificate editor for a resource

To create the certificate editor for a resource

- In the structure tree, select **Resource, Protocols, OPC UA Server Set**.
- Right-click the **OPC UA Server Set** in the workspace and select **New** from the context menu.
The dialog box for selecting objects appears.
- In the dialog box, select **Certificates** to create a certificate object.

The new *Certificates* object is added. It contains the tabs *Server Certificates* and *Client Certificates*.

To create the certificate editor for a resource

- Right-click **Certificates** and select **Edit** to open the Certificate Editor.

When selecting files, SILworX checks whether:

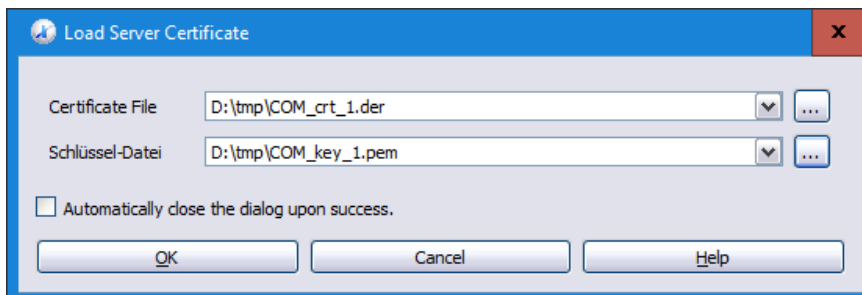
- The files are available.
- The file size is not less than 1 byte (minimum file size).
- A certificate file does not exceed 10 KiB and a key file does not exceed 4 KiB (respective maximum file size).

The file selection can only be completed once all conditions have been met.

6.5.1.1 Load Server Certificate

To load the server certificate into the client certificate editor

- Select the **Server Certificate** tab.
- Right-click a free space in the workspace of the server certificate editor and select **Load Server Certificate** from the context menu.
The dialog box for selecting objects appears.
- In the dialog box, select **Server Certificate**; the **Load Server Certificate** dialog box appears.

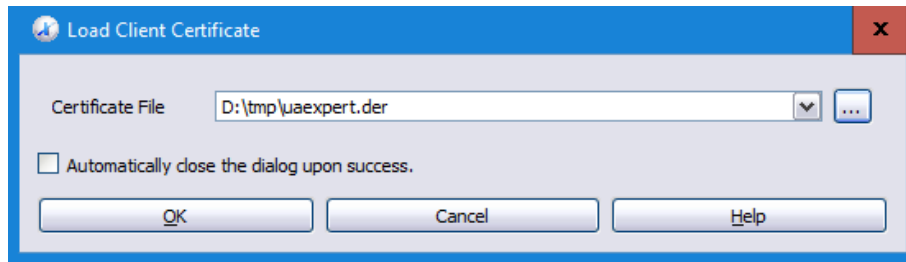


- Navigate to the location where the **certificate file** and the **key file** of the HIMA OPC UA Server are stored, select them and click **OK** to confirm the action.

6.5.1.2 Load Client Certificate

To load the client certificate into the client certificate editor

- Select the **Client Certificate** tab.
- Right-click a free space in the workspace of the client certificate editor and select **Load Client Certificate** from the context menu.
The dialog box for selecting objects appears.
- In the dialog, select **Client Certificate**; the **Load Client Certificate** dialog box appears.



- Navigate to the location where the **certificate file** of the OPC UA Client is stored, select it and click **OK** to confirm the action.



Observe the period of validity of the OPC UA certificates!

It might not be possible to execute a valid certificate if the current date and time are not set in the OPC UA Server or OPC UA Client.

7 Alarms&Events

If events are subscribed to in an OPC UA Client, the OPC UA Server starts to fetch the event data from the system and delivers them to the OPC UA Client.

The system supports a maximum of 5000 events per resource. If this quantity is exceeded, SILworX reports an error and code generation is not possible.

Events can be configured in the "A&E Editor" auf Seite 36. The message texts and priorities of the event definitions are written to the OPC UA configuration file.

7.1 Activating the A&E Functionality for a Resource

To activate the A&E functionality for a resource

- Right-click on **OPC UA Server Set** and select **Edit**.
- Select the **Properties** tab and activate **Activate Events**.

Events in a resource can be activated for a maximum of 4 OPC UA Servers, without using the X-OPC Server or safeEDR Set with activated A&E functionality.

The following systems allow the maximum number of configured accesses to the events:

- HIMax: 4
- HIQuad X: 4
- HIMatrix: 1

The OPC UA Server supports *system events* that occur when the conditions predefined by the system are met.

The following *system events* can be provided to an OPC UA Client:

- System started.
- System stopped.
- Reload activated.
- Communication between system CPU and SOE IO module established.
- Communication between system CPU and SOE IO module lost.
- Events re-initializing.
- No free memory spaces available for event entries in the event buffer.

7.2 Data in Event Notifications Provided to an OPC UA Client

For each event definition, SILworX creates an entry (section: EventSource) in the configuration file with the following attributes:

Element	Description
Event ID	The event ID is unique and identifies an event notification. Length: 12 byte array. Not changeable by the user.
Event Type	Corresponds to the <i>BaseEventType</i> for an event that is not a <i>system event</i> , or <i>SystemEventType</i> for a <i>system event</i> .
Source Name	Corresponds to the name of the event definition in the A&E Editor.
Source Node	Corresponds to the Node ID of the first OPC UA variable with a global variable reference to the event definition in the A&E Editor. If no OPC UA variable has a global variable reference to the event definition, the <i>SourceNode</i> corresponds to the <i>server node</i> . If SILworX finds other OPC UA variables with the same global variable reference during code generation, these are ignored and do not reference the event definition.
Severity	Corresponds to the priority from the event definition in the A&E Editor.
Message	Corresponds to the text from the event definition in the A&E Editor.
Time	Corresponds to the timestamp set when the event occurred.
ReceiveTime	Corresponds to the timestamp set when the event was received on the COM.

7.3 A&E Editor

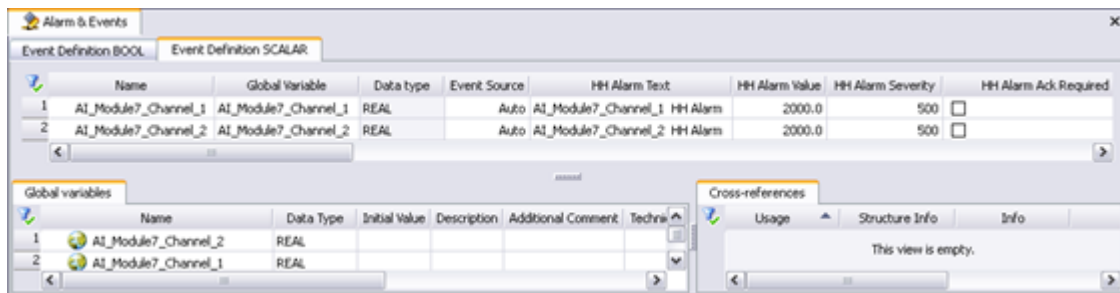
Alarms and events are configured in the A&E Editor of the resource. The events created in the A&E Editor are automatically transmitted via the configured **safeethernet** connection.

To create the A&E Editor for a resource

- In the structure tree, select **Configuration, Resource**.
- Right-click **Resource** and select **New, Alarms&Events** from the context menu.
A new A&E Editor is created. It contains the event definitions and properties.

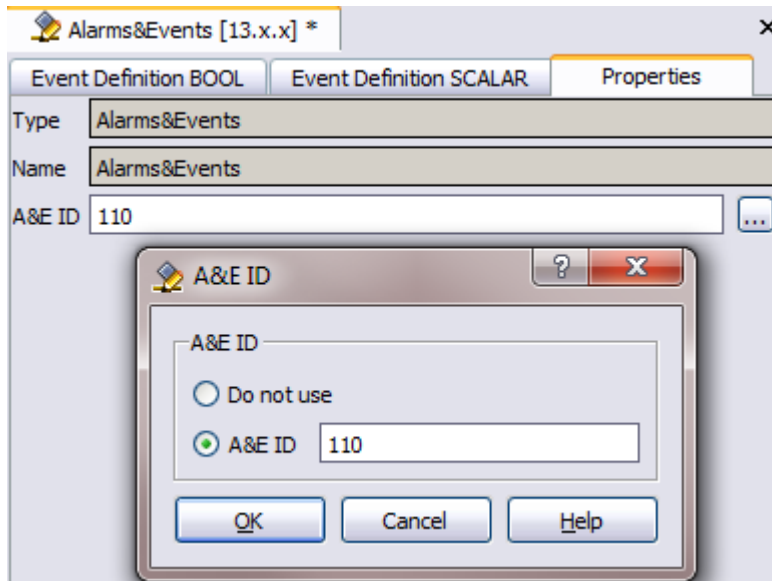
To drag global variables onto the A&E Editor

- Right-click **Alarms&Events** and select **Edit**.
- Select the **Event Definition BOOL** tab for Boolean events.
- Select the **Event Definition SCALAR** tab for scalar events.
- Click **Global Variable** from the Object Panel and drag it onto a free space within the workspace of the A&E Editor.



To create the A&E ID used to generate the unique cookies for the resource

- Select the **Properties** tab of the A&E Editor and click the ... button next to the **A&E ID** field.
The **A&E ID** dialog box appears.
- Select **A&E ID** and enter a unique **A&E ID**.



7.3.1 Indication in the OPC Client

The name of the OPC UA Server that is displayed in the OPC Client is composed of:

HIMA (manufacturer).service name DA (Data Access).

Connect to the OPC Server. The configured DA data must now be transferred to the OPC Client.

Connect to the X-OPC Server. The configured alarms and events must now be transferred to the OPC Client.

8 Control Panel (Online)

The Control Panel can be used to verify and control the settings for the OPC UA. It also displays details on the OPC UA Server's current state.

To open the Control Panel for monitoring the OPC UA Server

- In the structure tree, right-click **Hardware** and select **Online** from the context menu.
- In the **System Login** window, enter the access data to open the online view for the hardware.

To open the diagnostics of the COM module for diagnostics of the OPC UA Server

- Right-click **COM Module** and select **Diagnostics** from the context menu to open the diagnostic view of the COM module.
- In the diagnostic view, select **All Entries** to display the diagnostics of the OPC UA Server.
[OPC UA Server Diagnostics](#).

To open the current status view of the OPC UA Server

- Double-click **COM Module** and select the OPC UA Server in the structure tree. The configured and the current μ P budgets of the OPC UA Server are displayed here.

The Control Panel displays the following online information:

Element	Description
Configured μ P Budget	The maximum μ P budget of the COM module from the configuration of the OPC UA Server Set.
Current μ P Budget	Current μ P budget of the COM module for the processing of the OPC UA Server Set.
Configured Warning if μ P Budget Exceeded in [%]	Configured warning threshold for the μ P budget of the COM module. If this threshold is exceeded, the OPC UA server must report a communication warning.
Configured Warning for μ P Budget Exceeded	The OPC UA Server has reported a communication warning stating that the configured μ P budget has been exceeded.

8.1 Trace Logging (Online)

The *Trace Logging* function provides analysis data if interference occurs while the OPC UA protocol is running. As soon as a trace logging message is generated by the OPC UA Server, it is immediately forwarded to the Syslog client. A Syslog client must be installed (e.g., on the SILworX PC) to receive and log the trace logging messages.

During normal operation, trace logging must remain deactivated as it increases the computing time and communication load.



The user must only activate trace logging when requested to do so by a HIMA employee in order to create a log file that can then be analyzed by HIMA.

To open the dialog box for editing the trace logging parameters in the online Control Panel of the COM module

➤ Right-click **OPC UA Server** and select **Trace Logging** from the context menu.

The dialog box displays the trace logging values that are currently set. If these values are not available, e.g., due to interference, default values are displayed. The trace logging dialog box can be closed with or without accepting the changes.

Parameter	Description
Name	OPC UA Server, not changeable.
Level	The <i>Level</i> parameter determines the levels of the trace outputs. The required value is determined by a HIMA employee upon request. Default value: 0
Facility	The <i>Facility</i> parameter determines the type of trace output. The required value is determined by a HIMA employee upon request. Default value: 0
Destination IP Address	IP address of the Syslog client to which the trace logging messages are sent. Default value: 127.0.0.1 (switched off).
Destination Port	IP port of the Syslog client to which the trace logging messages are sent. Default value: 514

8.2 OPC UA Server Diagnostics

The OPC UA Server generates the following entries in the short-term diagnostics of the COM module on which the OPC UA server is running. The user can read out the short time diagnostics from the diagnostic view of the COM module. Refer to Chapter [Control Panel \(Online\)](#) for details.

8.2.1 Opening a Session

The following diagnostic entry is generated when the connection to an OPC UA Client is being established:

*An OPC UA session (session ID = **sID**, channel ID = **chID**) was created for the OPC UA Client (IP address = <client IPv4 address>). Number of OPC UA sessions: **N** of **Nmax***

All communication between the OPC UA Client and the OPC UA Server occurs within one session. The Server handles the sessions to the Clients. Data is transmitted within the OPC UA communication stack via a secure channel (SecureChannel). The created session is only ready for data transmission after it has been activated.

Parameter	Description
sID	Unique identification assigned by the OPC UA Server for the session.
chID	Unique identification assigned by the OPC UA Server for the channel.
N	Number of sessions already created on the OPC UA Server, including this session.
Nmax	Maximum number of sessions that can be simultaneously opened by the OPC UA Server.

8.2.2 Activating a Session

The following diagnostic entry is generated when the OPC UA Server requests the activation of a session.

*The OPC UA session with ID = **sID** and with session timeout = **TO** ms was activated on the channel with ID = **chID**.*

If there is no activity from the OPC UA Client after the set timeout window has expired, the session is closed by the Server. The timeout value requested by the OPC UA Client must neither underrun the minimum nor overrun the maximum value specified by the Server, the requested timeout value is otherwise adjusted according to the limit values (min, max) specified by the Server.

Parameter	Description
sID	Unique identification assigned by the OPC UA Server for the session.
chID	Unique identification assigned by the OPC UA Server for the channel.
TO	Session timeout requested by the OPC UA Client (in ms).

8.2.3 Closing a Session

The following diagnostic entry is generated when a session to the OPC UA Client is closed.

*The OPC UA session with ID = **sID** on the channel with ID = **chID** was closed.*

Parameter	Description
sID	Unique identification assigned by the OPC UA Server for the session.
chID	Unique identification assigned by the OPC UA Server for the channel.

8.2.4 Creating a Subscription

The following diagnostic entry is generated when a subscription is created. The total number of subscriptions still available (**N**) does not yet include the creation of this subscription.

*A request from the OPC UA Client (OPC UA Server session ID = **sID**) to create a subscription with the following parameters was received*

- *Publish interval = **Tp** ms*
- *Maximum number of notifications per Publish Response = **Nmax***
- *Life Time Counter = **Cl***
- *Keep Alive Counter = **Cka***

N subscriptions can be currently created.

For a detailed explanation of the parameters **Tp**, **Nmax**, **Cl** and **Cka**, refer to the *OPC UA Unified Architecture Specification, Part 4, Chapter 5.3.12*.

Parameter	Description
sID	Unique identification assigned by the OPC UA Server for the session.
Tp	Cyclic time interval in ms for the transmission of notifications.
Nmax	Maximum number of notifications in one response.
Cl	Lifetime counter.
Cka	Keep Alive counter.
N	Number of subscriptions that can still be created on the OPC UA Server.

8.2.5 Deleting a Subscription

The following diagnostic entry is generated when a subscription is deleted.


*A request from the OPC UA Client (OPC UA Server session ID = **sID**) to delete **N** subscriptions was received. **Nc** out of **Nmax** possible subscriptions currently exist on the OPC UA Server.*

The number of existing subscriptions (parameter **Nc**) does not yet include the deletion of this subscription.

Parameter	Description
sID	Unique identification assigned by the OPC UA Server for the session.
N	Number of subscriptions to be deleted.
Nc	Current number of subscriptions that exist on the OPC UA Server.
Nmax	Maximum number of subscriptions that can be created on the OPC UA Server.

9 Code Generation and Reload

While a download can only be performed if the programmable electronic system (PES) is in the STOP state, the system need not be stopped to perform a reload.

 A reload is an intervention in a running, safety-related system. It can only be performed if the **Reload Allowed** CPU switch was previously activated in the Resource Properties. If *Reload Deactivation* is used in the user program, the system variable must also be set to FALSE.

9.1 Code Generation

During code generation, SILworX generates an *opcuaserver.config* configuration file for each referenced COM module. The configuration file is assigned the display name *OPC UA Server* and the minimum SILworX version V12.

The system supports a maximum of 15000 nodes per configuration. If this quantity is exceeded, SILworX issues an error and code generation is not possible.

9.2 Reload

The integrated OPC UA Server on the COM module always provides consistent system data, even during a reload. Longer processing times in the data traffic may therefore occur during the system upgrade.

The system supports reload code generation after the following changes in the information model:

- Adding a maximum of 1000 nodes in one operation if they have a higher node ID than all nodes previously present in the information model.
- Changing a maximum of 1000 node descriptions in one operation.

During a reload code generation, if SILworX finds unsupported changes in the information model, no reload can be performed for the COM module. SILworX issues a reload warning stating that these changes can only be loaded by performing a cold reload.

9.3 Certificates

If certificates have been configured, SILworX generates an *opcucertificates.config* configuration file for each referenced COM module during code generation; this configuration file is assign the display name *OPC UA Certificates* and the minimum SILworX version V12.

The system does not support changes in the certificate configuration through reload code generation. Changes can only be loaded to the controller by performing a cold reload.

Further information about "Certificates" auf Seite 30.

10 Version Comparison

The version comparison is based on the project checksums (CRCs) created by the code generator. For further information, refer to the SILworX version comparison manual (HI 801 285 E).

During a version comparison, different resource configurations are compared to one another and the differences between the individual configuration files are detected. The result of the version comparison has SIL 3 quality and is based on the configuration files describing the executable code.

10.1 OPC UA Server

For the *opcuaserver.config* configuration file, SILworX provides a version comparison with the description *OPC UA Server*.

10.1.1 OPC UA Section

The version comparison reports changes for the following OPC UA Server parameters:

- Standard Protocol ID
- Number of Namespaces
- Number of Nodes
- Number of References

These messages are listed in the detail view of the version comparison, at the beginning of the table.

10.1.2 Namespace Section

The version comparison reports changes for the following namespace parameters:

- Namespace URI
- Number of Objects
- Number of Variables
- Number of Types
- Number of References
- Number of Strings

These namespace messages appear in the detail view of the version comparison, after the entries for the OPC UA section. They are identified and sorted by their namespace index.

10.1.3 Node Section

The version comparison reports removed or added nodes as well as changes for the following general node parameters:

- Node Class
- Browse Name
- Display Name
- Description

These node messages appear in the detail view of the version comparison, after the entries for the Namespace section. They are identified and sorted by their node ID.

10.1.4 Variables Section

The version comparison reports changes for the following variable parameters:

- Data View Identifier
- Access Permission
- Data Type

These messages appear in the detail view of the version comparison, below the node representing the variable.

10.1.5 Reference Section

The version comparison reports removed or added references.

This is shown as a string which is formed from the node IDs of the source, destination and type of the reference.

Example: 0:85-2:10-00:47

- 0:85 - corresponds to the object folder in namespace 0 of the OPC Foundation
- 2:10 - corresponds to a node with ID 10 in the user's namespace 2
- 00:47 - corresponds to the relationship HasComponent in namespace 0 of the OPC Foundation

These messages appear in the detail view of the version comparison, below the node that is the source of the reference.

10.1.6 EventSource Section

The version comparison reports removed or added events as well as changes for the following general event parameters:

- Event Name
- Event ID
- Event Type
- Node ID of Global Variables

These event messages appear in the detail view of the version comparison, after the entries for the Node section. They are identified and sorted by the event name.

10.1.7 Condition Section

The version comparison reports changes for the following event parameters:

- Texts for alarm (also HH-, H-, L-, LL- and return to normal state texts)
- Alarm priority (also HH, H, L, LL and return to normal state priorities)

These messages appear in the detail view of the version comparison, below the event definition that contains the condition.

10.2 OPC UA Certificates

For the *opcuaCertificates.config* configuration file, SILworX provides a version comparison with the description *OPC UA Certificates*.

The version comparison reports changes for the following parameters of the client certificates:

- Only allow encrypted connections
- Number of Client Certificates

These messages are listed in the detail view of the version comparison, at the beginning of the table.

10.2.1 Own Section

The version comparison reports changes for the following parameters of the server certificate:

- Certificate File
- Key File

These messages are listed in the detail view of the comparison, after the entries for the Certificates section. The changed values are displayed as SHA1 hash strings. The actual content of the certificate or key is not displayed.

10.2.2 Client Section

The version comparison reports changes for the following parameters of the client certificates:

- Certificate File

This client certificate message appears in the detailed view of the version comparison, after the entries for the Server Certificate section. They are identified and sorted according to the order in the *Index* configuration file. The changed values are displayed as SHA1 hash string. The actual content of the certificate is not displayed.

HI 801 551 E (2027)

For further information, please contact:

HIMA Paul Hildebrandt GmbH

Albert-Bassermann-Str. 28
68782 Brühl, Germany

Phone: +49 6202 709-0

E-mail: info@hima.com

 www.hima.com/en/

