Using a CP 443-1 OPC UA as Gateway for MES/ERP

CP 443-1 OPC UA / SIMATIC RF680R / STEP 7 V5.5 / V1.0

https://support.industry.siemens.com/cs/ww/de/view/109743832

Siemens Industry Online Support

Warranty and Liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These Application Examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.
If there are any deviations between the recommendations provided in these Application Examples and other Siemens publications − e.g. Catalogs − the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.
Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act ("Produkthaftungsgesetz"), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract ("wesentliche Vertragspflichten"). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.
Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of the Siemens AG.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.
In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.
Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.
Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit http://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.
To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under http://www.siemens.com/industrialsecurity.

Table of Contents

1 Introduction

1.1 Overview

OPC UA enables an automation system to directly communicate with numerous other systems – from ERP to field level.

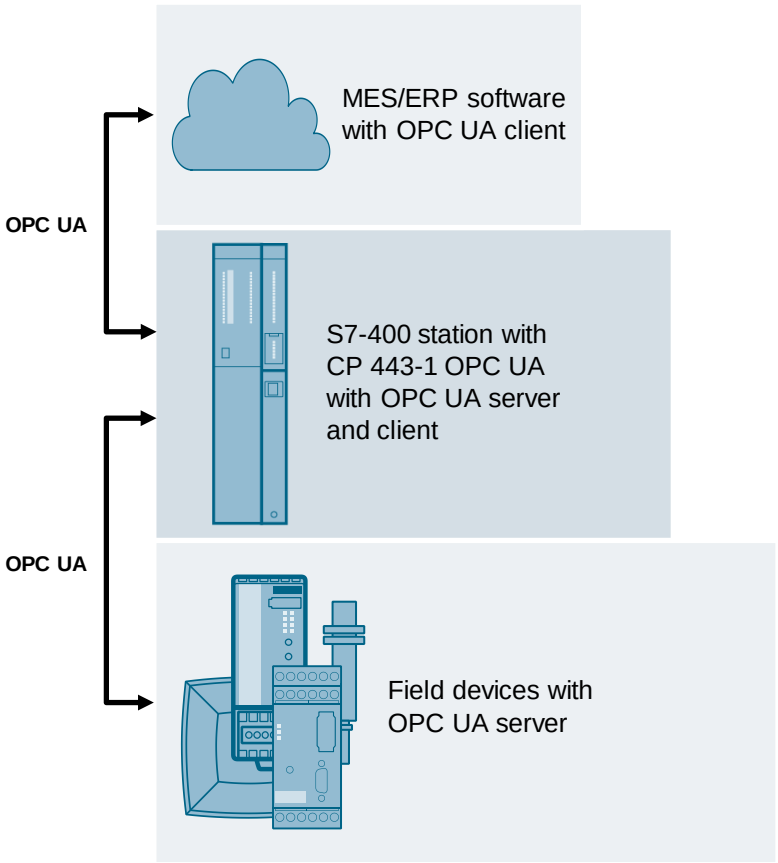
With the introduction of the CP 443-1 OPC UA communication processor for the S7-400 automation system, it is able to function as an OPC UA server or as an OPC UA client.

Security measures like encryption or data signature and additional authentication features make sure the general security principles are maintained for OPC UA.

Content of this application example

In this example, the S7-400 station with CP443-1 OPC UA serves as a gateway between subordinate field devices and the MES/ERP level. The S7 station collects raw or process data from the subordinate field devices, pre-processes (aggregates) them and brings them into a structure which higher-level systems can directly work with.
Only OPC UA is used as a communication protocol. In this application example, the data is continuously backed up and transferred with a signature.

Figure 1-1



Advantages of the application example

This application example offers you the following advantages:

- Expandable STEP 7 project with preconfigured OPC UA server and client for your SIMATIC S7-400.
- Reusable function block which encapsulates the individual OPC UA blocks and thus serves for the simplified control of the basic OPC UA client functionality.

Assumed knowledge

The following basic knowledge is required by the user:

- Basics of programming in FBD and SCL.
- Basics of configuration in SIMATIC Manager
- Basics of OPC UA
- Basics in software security and certificate handling

1.2 Mode of operation

The following figure displays the components and the data flow in application example.

Figure 1-2

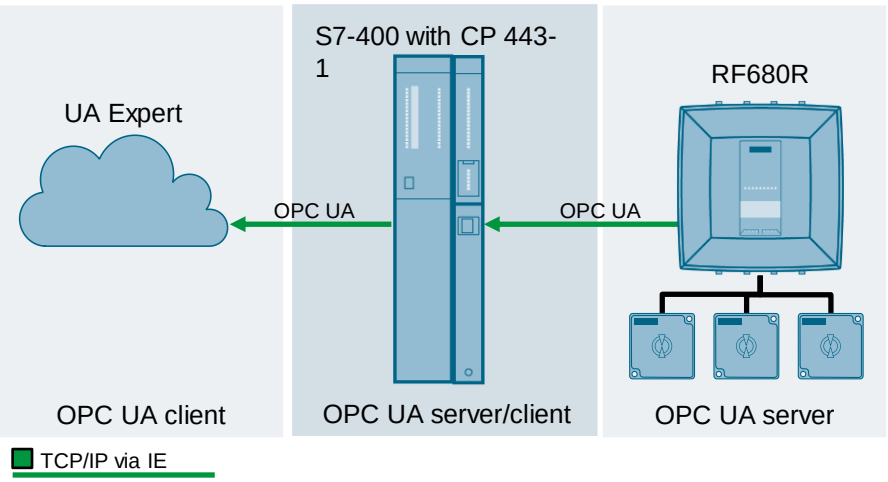


Table 1-1

Component	Functions realized
SIMATIC RF680R	The UHF RFID reader serves as a field device in this application example. The reader is operated with three antennas, with each one automatically reading the ID (EPC ID) of transponders. There is an OPC UA tag in the UA address range of the reader for each antenna/read point, which contains the last read transponder ID.
CP443-1 OPC UA	The CP443-1 OPC UA reads the transponder IDs of the reader as a string value via OPC UA. After the IDs have been evaluated by the S7-400 station, the CP provides the result to the higher-level system via its OPC UA server functionality.
SIMATIC S7-400	The S7-400 station controls the client of the CP443-1 OPC UA and compares the transponder IDs read by the CP. When all three read points provide the same ID, a bit in a user data block is set.

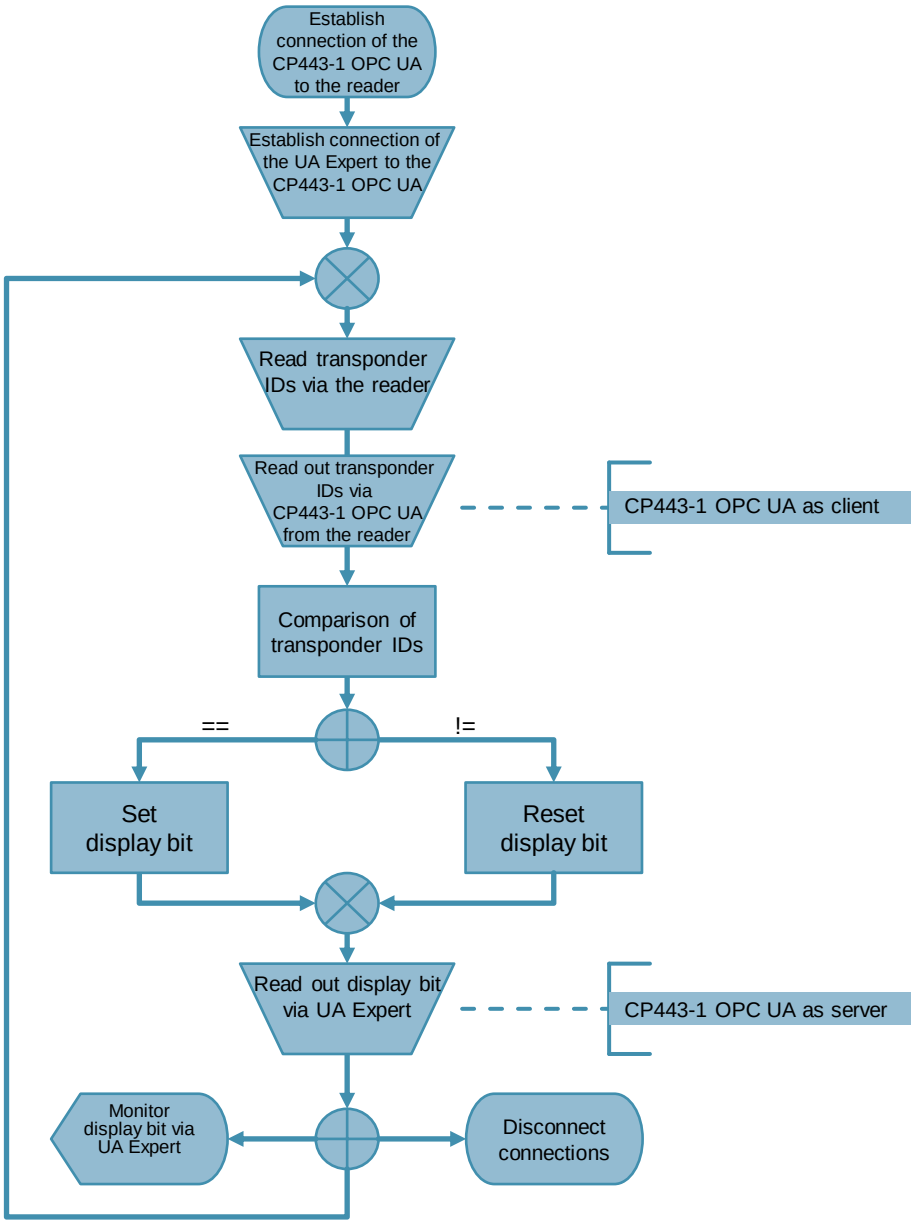
Component	Functions realized
UA Expert	The free OPC UA browser UA Expert is used as higher-level OPC UA client in this example. The result bit of the S7-400 station is provided to the UA Expert via the server functionality of the CP.

The OPC UA server functionality of the CP443-1 OPC UA is only realized via configuring screens in SIMATIC Manager. The client functionality is, in addition to configuration, realized by programming of S7 blocks.

Functional sequence

Once the configuration and programming of OPC UA server and clients is completed, the following functional sequence for the application example results:

Figure 1-3



1.3 Components used

This application example was created and tested with the following hardware and software components:

Table 1-2

Component	Number	Article number	Note
SIMATIC S7-400 CPU 416-3 PN/DP	1	6ES7416-3ER05-0AB0	
CP 443-1 OPC UA	1	6GK7443-1UX00-0XE0	
SIMATIC RF680R ETSI	1	6GT2 811-6AA10-0AA0	With FW V3.0
Antenna RF620A	3	6GT2812-1EA00	
Antenna cable UHF	3	6GT2815-0BN10	
Transponder RF640T	>1	6GT2810-2DC00	As an alternative, you can use any other RF600 ETSI transponder
STEP 7 V5.5	1	6ES7810-4C.10-..	With HF 10 and HSP1104
Security Configuration Tool	1		V4.2 or higher
UA Expert	1		The download link can be found in the appendix (\9\)

This application example consists of the following components:

Table 1-3

Component	File name
STEP 7 V5.5 project	109743832_CP443-1_OPC_UA_V55_CODE_V10.zip
Documentation	109743832_CP443-1_OPC_UA_V55_DOKU_V10_en.pdf

2 Engineering

2.1 Configuring the CP443-1 as OPC UA server and client

The following instructions show how to configure the 3-1 OPC UA as OPC UA server and client. The example project included in this application example has been prepared to match the function in the example.

Prerequisites

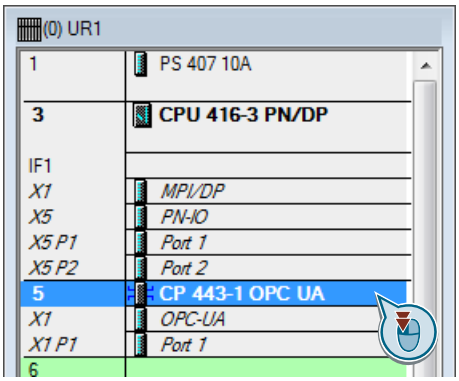
- Install the HF 10 and HSP1104.
- Install the Security Configuration Tool V4.2 or higher.
- Create a STEP 7 project with the SIMATIC manager.
- Configure a SIMATIC S7-400 with firmware >= V5.3.
- Set the CPU as “Master” for time synchronization.
- Set the current time and date at the CPU.
- Configure the CP443-1 OPC UA in the hardware configuration.
- Connect the CP443-1 OPC UA with a subnet.

2.1.1 Enabling/configuring the OPC UA server and client

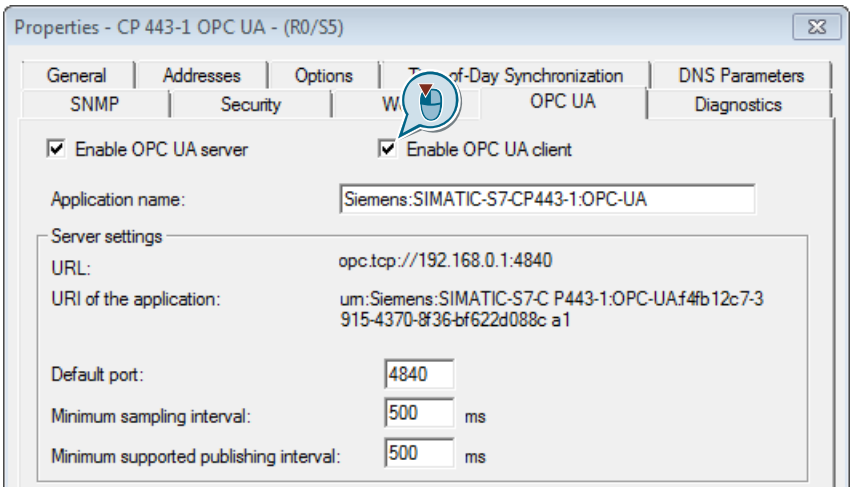
The OPC UA client of the CP 443-1 is deactivated as default, the server is enabled. The following instructions show the steps required to enable the client functionality as well.

Enabling the OPC UA client

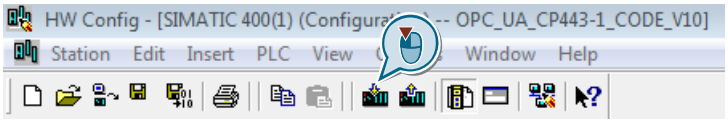
1. Go to the hardware configuration of the SIMATIC Manager in your project.
2. Double-click on the CP 443-1 in your rack.



3. Go to the “OPC UA” task card and check the “Enable OPC UA client” check box.

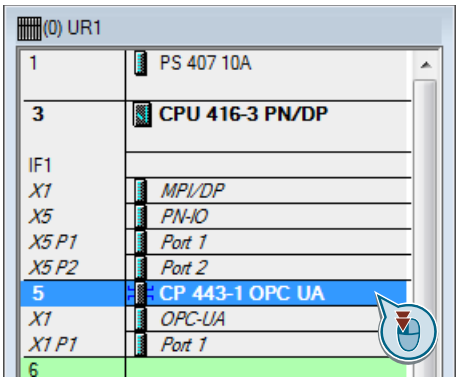


4. Confirm with “OK”.
5. Load the hardware configuration to your CPU via “Download to module”.

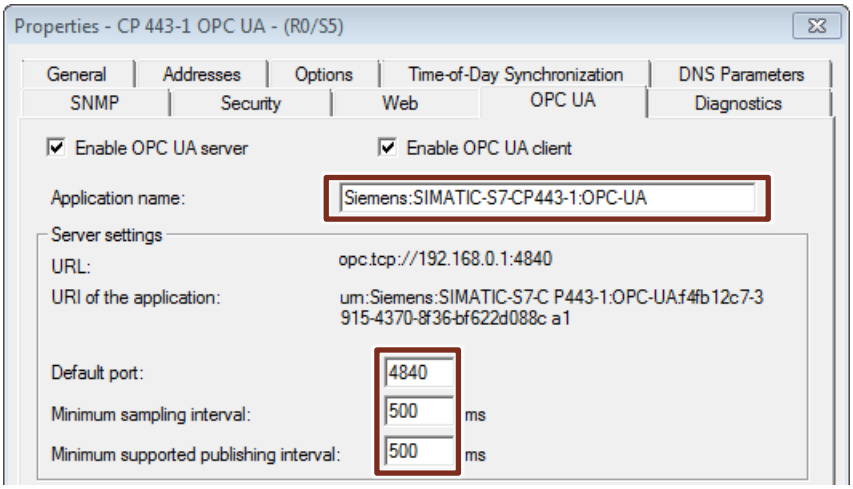


Configuring the basic settings for the OPC UA server

1. Go to the hardware configuration of the SIMATIC Manager in your project.
2. Double-click on the CP 443-1 in OPC UA your rack.

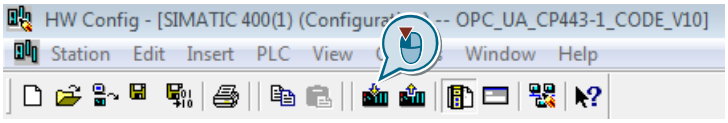


3. Go to the “OPC UA” task card.



- Assign an application name for your server in the “Application name” field. This name is displayed for the OPC UA clients.
- Enter the port on which the server is to accept connections in the “Default port” field.
- Enter the smallest possible desired sampling interval of the OPC UA server (100 to 65535 ms) in the “Minimum sampling interval” field. This value determines how often the OPC UA server of the CP443-1 OPC UA compares its data contents with the S7 controller. Decreasing this value increases the CPU load. Therefore, adjust this value to your conditions.
- Enter the smallest possible publishing interval for subscriptions to the server (100 to 65535 ms) in the “Minimum supported publishing interval” field. This value determines how often value changes in the data contents of the OPC UA server are published to the clients. Decreasing this value increases the load in your network. Therefore, adjust this value to your conditions.

4. Then click “OK” to confirm.
5. Load the hardware configuration to your CPU via “Download to module”.

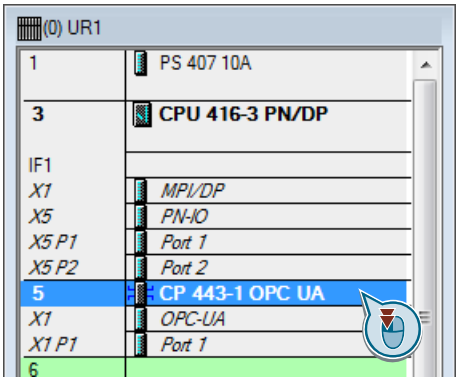


Note The OPC UA client and server functionality is now activated. The client functions can be controlled via the “UA_xyz” function blocks from the current SIMATIC_NET_CP block library.

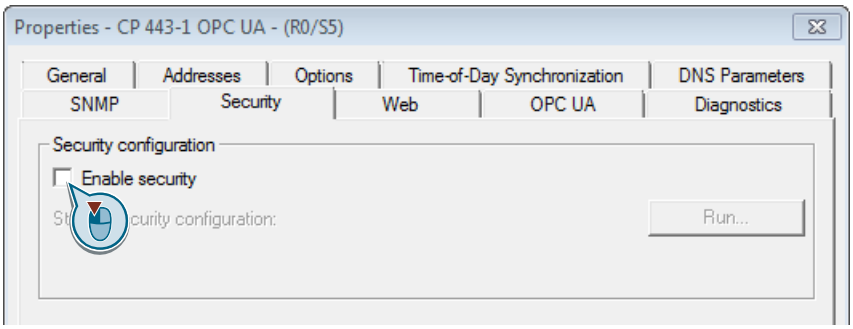
2.1.2 Enabling security for the OPC UA server and client

To configure the security functions of OPC UA, Security must be enabled in your project. Proceed as follows:

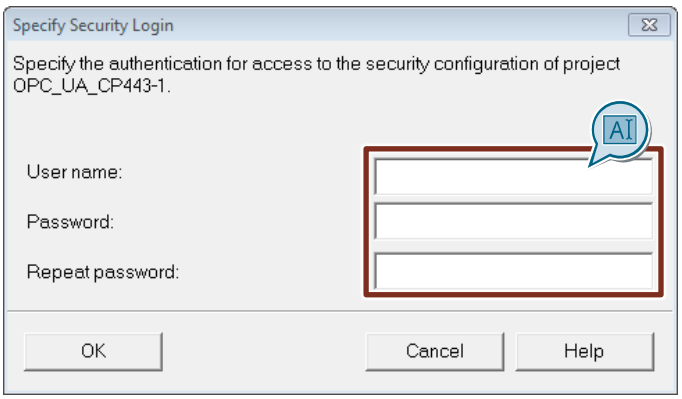
6. Go to the hardware configuration of the SIMATIC Manager in your project.
7. Double-click on the CP 443-1 in your rack.



8. Go to the “Security” task card and check the “Enable security” check box.



9. Once you enable the check box, you need to assign a user name and password in the dialog. Fill in the dialog and confirm with “OK”.



Note

You have to authenticate yourself with the user ID created before for any other security-relevant settings in your project or to load the hardware configuration. If you forget your login data, you have to create a new project.

The created user will automatically be activated for user login on the OPC UA server of the CP 443-1 OPC UA. For this reason, do not forward these login data to anybody else.

2.1.3 Configuring security for the OPC UA server and client

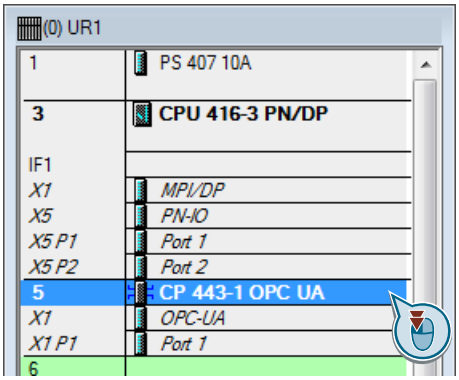
The OPC UA server offers you various security settings for authorization and authentication. For the OPC UA client, you only specify settings in the certificate validation.

The following settings can be made:

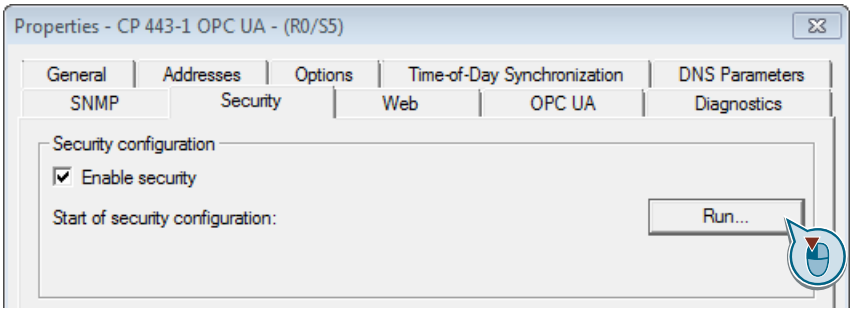
- Selection of the security profile for the server end points
- Selection of the security method of the server end points
- Specify authorization for anonymous access to the server
- Configure certificate validation (server and client)

The following instruction shows where and how to make these settings for your OPC UA server and client:

1. Go to the hardware configuration of the SIMATIC Manager in your project.
2. Double-click on the CP 443-1 in your rack.

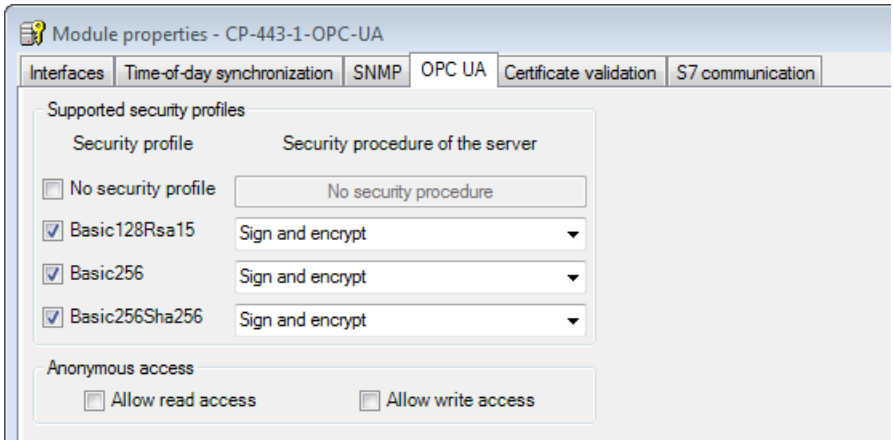


3. Go to the “Security” task card and click “Run...”.



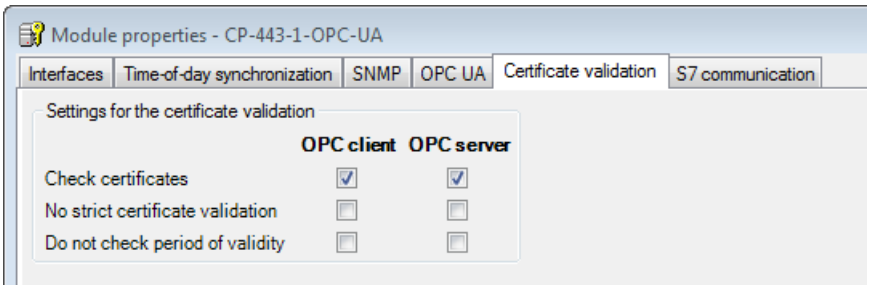
4. Enter the access data you have assigned when enabling security (chapter 2.1.2 Enabling security for the OPC UA server and client, step 4) in the subsequent dialog. Confirm with “OK”.

5. After successful login, the dialog “Module properties” appears. Go to the “OPC UA” task card.



Enable the endpoints you require and decide whether to transfer the frames only with a signature or also encrypted. Additionally, you can grant read and write rights for the anonymous access.
For further information, see 3.1.3 OPC UA Security.

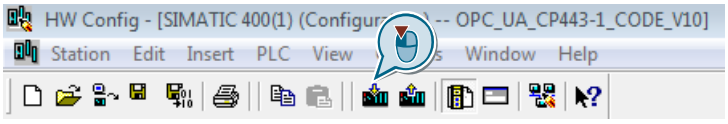
6. Go to the “Certificate validation” task card.



Determine with the “Check certificate” check box, whether the OPC UA server should check the client certificates.
The check box “No strict certificate validation” determines if the server should match the client IP address with the certificate IP address. And the purpose (OPC UA server/client) of the certificate is compared with the one of the client.
The check box “Do not check period of validity” determines whether the server should check the period of validity of the certificate.

Make these settings from the point of view of the server and the client of the CP443-1 OPC UA.

7. Then click “OK” to confirm.
8. Load the hardware configuration to your CPU via “Download to module”.

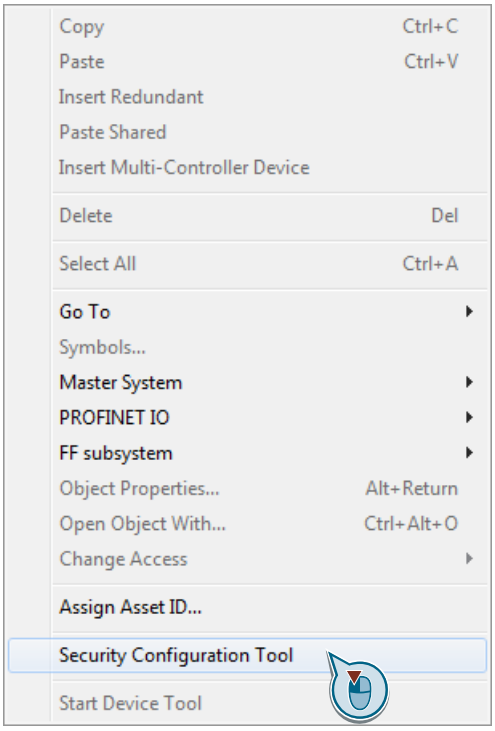


Note In order to establish a connection to the field device SIMATIC RF680R, you have to disable the certificate check as OPC client in the CP443-1 OPC UA.

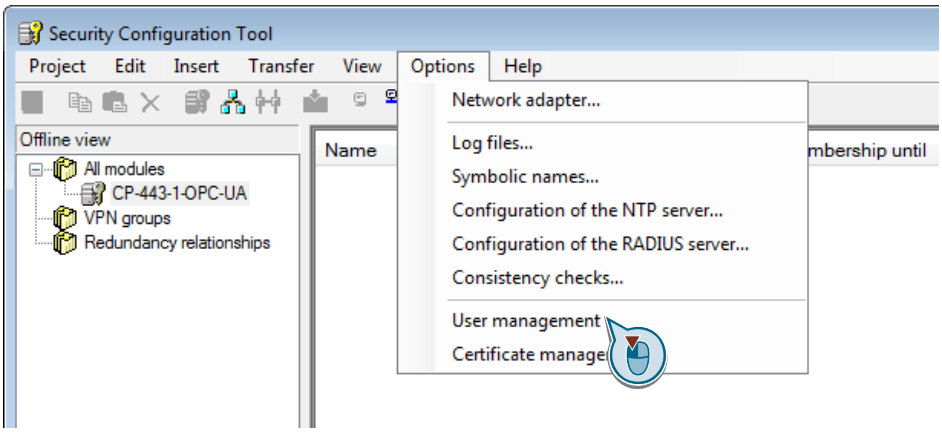
2.1.4 Creating user for the OPC UA server

If you have not granted read or write rights for the anonymous access to the OPC UA server of the CP443-1 OPC UA or if you want to create an additional user for the server, proceed as follows:

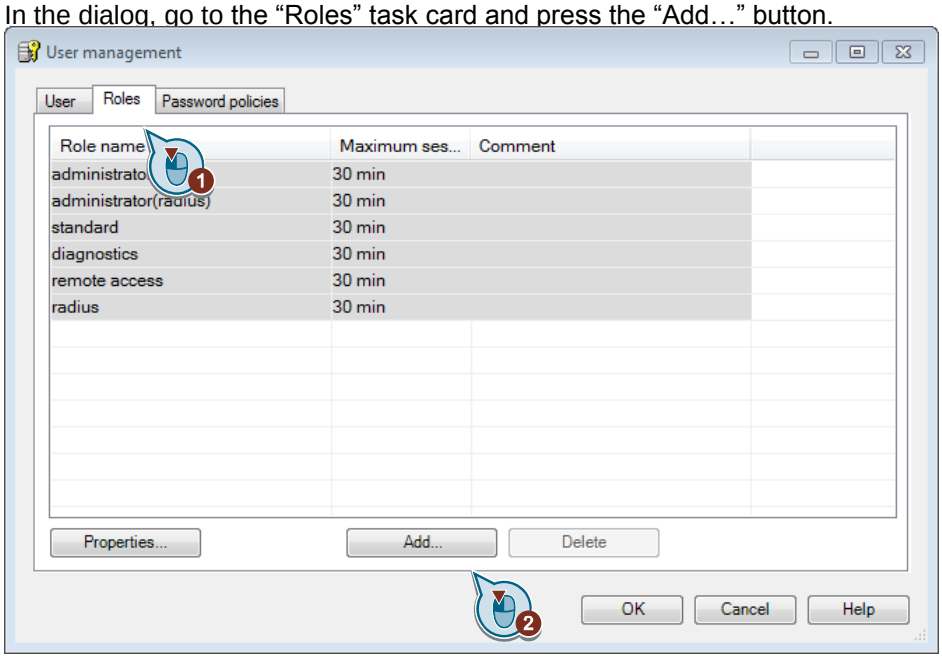
1. Go to the hardware configuration of the SIMATIC Manager in your project.
2. In the main menu of the hardware configuration, go to “Edit” > “Security Configuration Tool”.



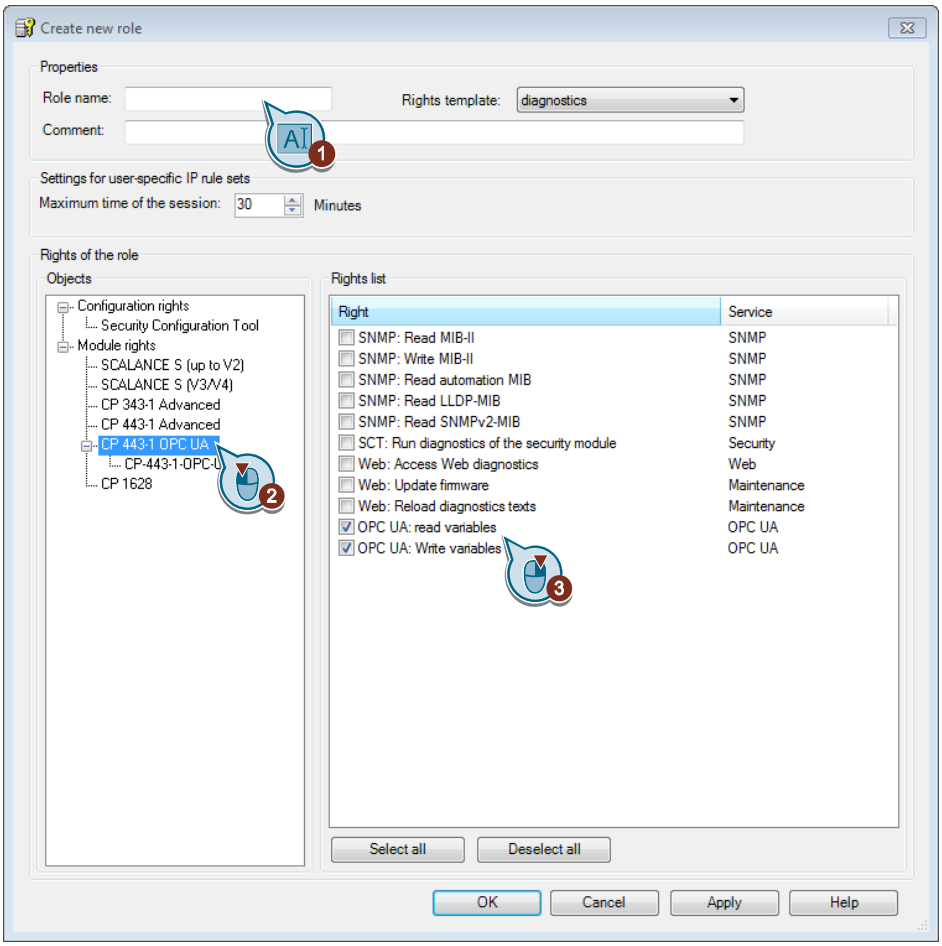
3. Click on “Options” > “User management”.



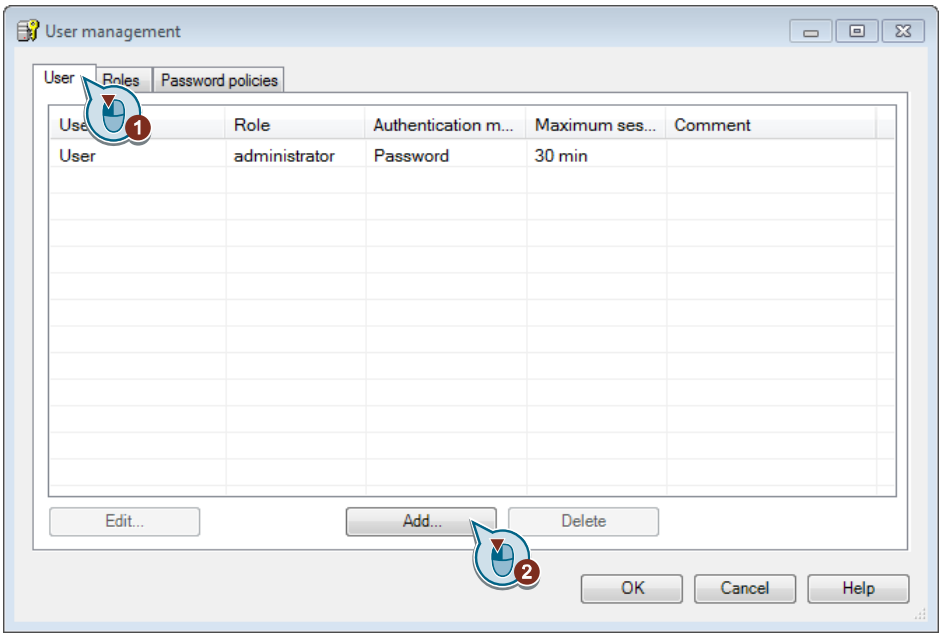
4. In the dialog, go to the “Roles” task card and press the “Add…” button.



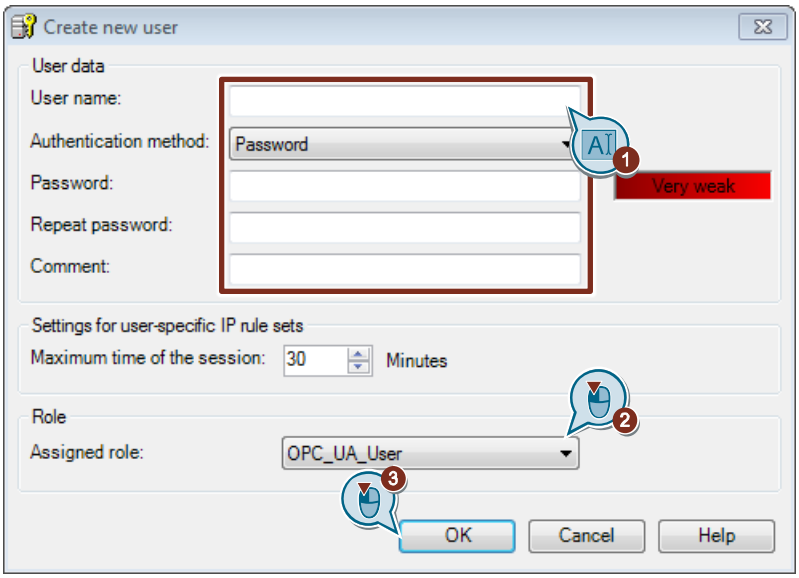
5. Enter a name for the user group in the list in the “Role name” field in the dialog. In the “Objects” list, select CP443-1 OPC UA and assign read or write rights to the user group in the “Rights list” for OPC UA. Then click “OK” to confirm.



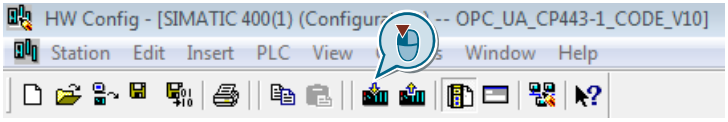
6. Go to the “User” task card and click “Add…”.



7. In the following dialog, enter a user name under “User name” and a password under “Password” and repeat the password under “Repeat password”. Select the previously created user group in the “Assigned role” drop-down menu. Then click “OK” to confirm.



8. Load the hardware configuration to your CPU via “Download to module”.

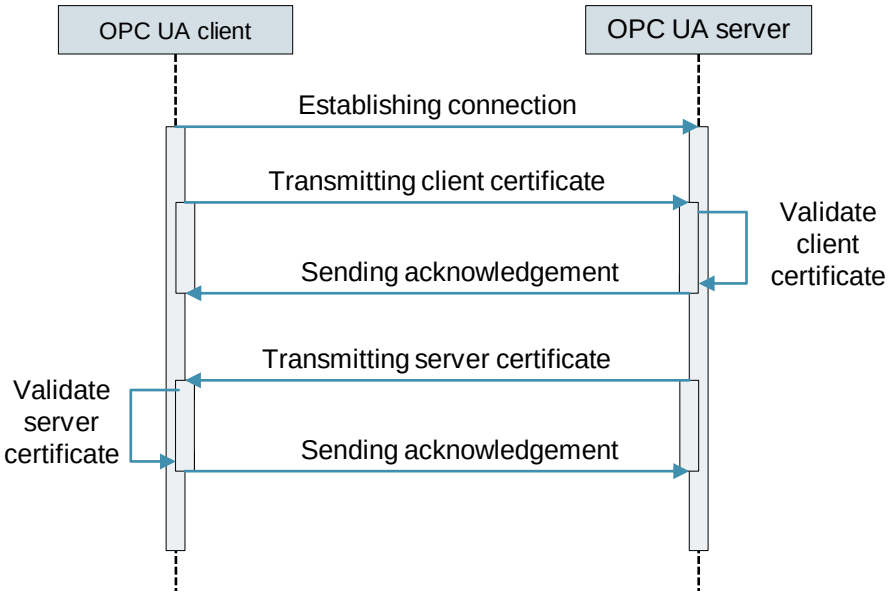


Note OPC UA clients can be authenticated on the server of the CP443-1 OPC UA via the user ID created before.

2.1.5 Managing certificates

The following sequence diagram shows in simplified form how certificate handling is managed when establishing a connection from an OPC UA client to a server:

Figure 2-1



When a connection is established, the server is provided with the client certificate and the client is provided with the server certificate. Client and server then have to validate the certificates and send an acknowledgment to the connection partner.

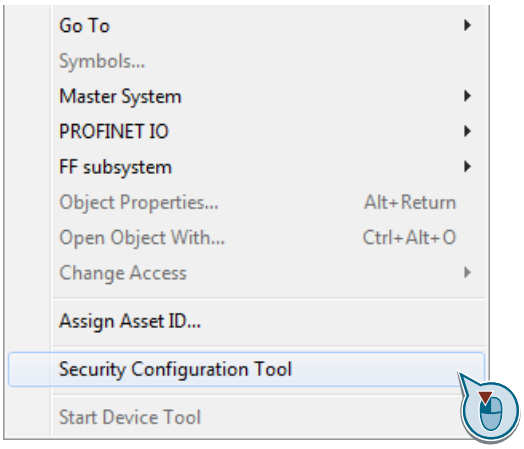
The procedure for the validation of the certificates of the CP443-1 OPC UA is the same for server and client. The CP443-1 OPC UA receives the certificate of the connection partner and validates it depending on the settings made (see chapter 2.1.3).

If the certificate check in the CP is enabled, the CP443-1 OPC UA must know the certificate of the connection partner before the connection is established. This is required because the CP does not offer a mechanism to have a certificate accepted by an administrator afterwards. The CP compares the certificate received when a connection is established with a certificate stored beforehand. If both certificates are identical, the validation is successful.

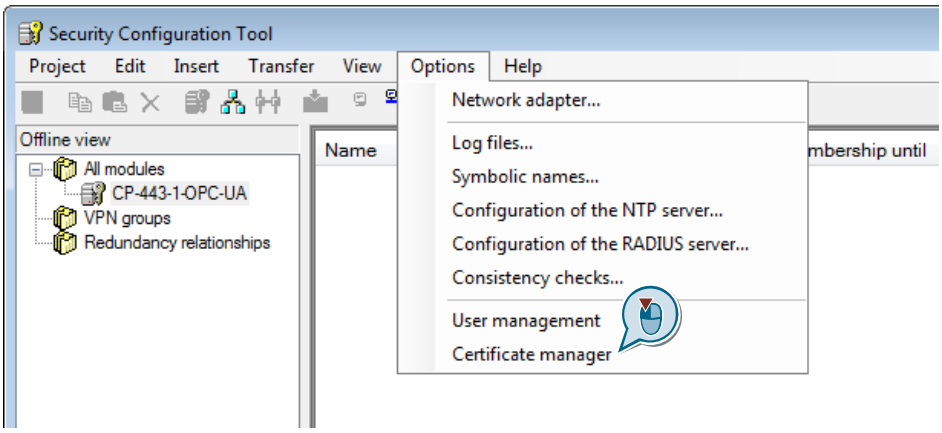
Use the Security Configuration Tool to make the certificates of the connection partners available to the CP443-1 OPC UA beforehand and/or export the CP certificate to make it available for connection partners with similar validation mechanisms.

The following instruction explains how to fulfill these tasks with the certificate manager in the SCT:

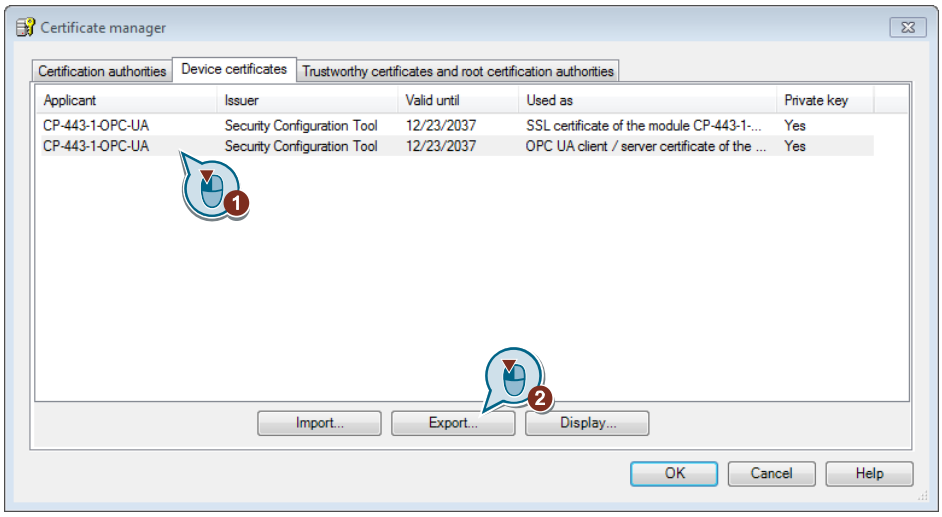
1. Go to the hardware configuration of the SIMATIC Manager in your project.
2. In the main menu of the hardware configuration, go to “Edit” > “Security Configuration Tool”.



3. Click on “Options” > “Certificate manager”.

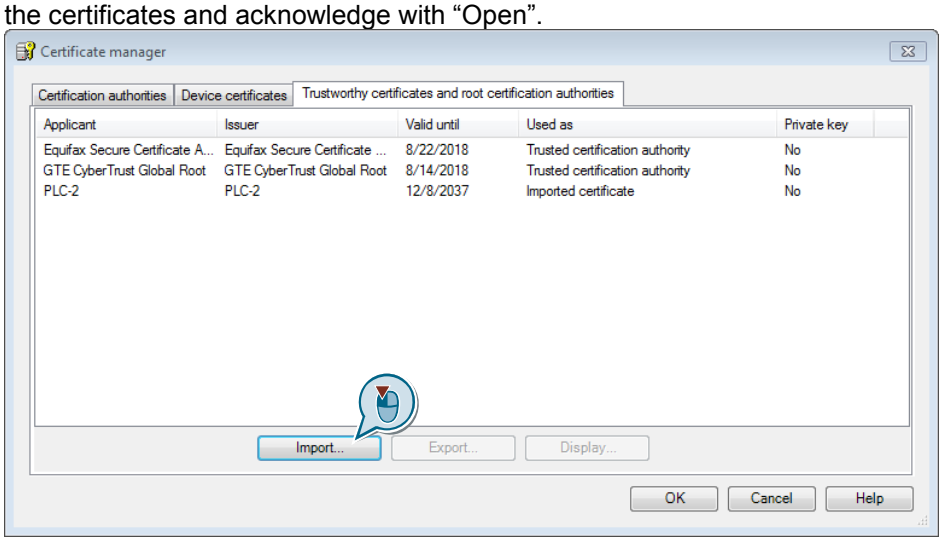


4. Go to the “Device certificates” task card to export the CP443-1 OPC UA certificate for further use.
5. Select the “CP-443-1-OPC-UA” certificate that is described as “OPC UA client / server certificate…” in the “Used as” column. Click “Export” and select a storage location for the certificate. Then click “Save” to confirm.
   Both the OPC UA server and the client of the CP443-1 OPC UA use this certificate file for authentication.

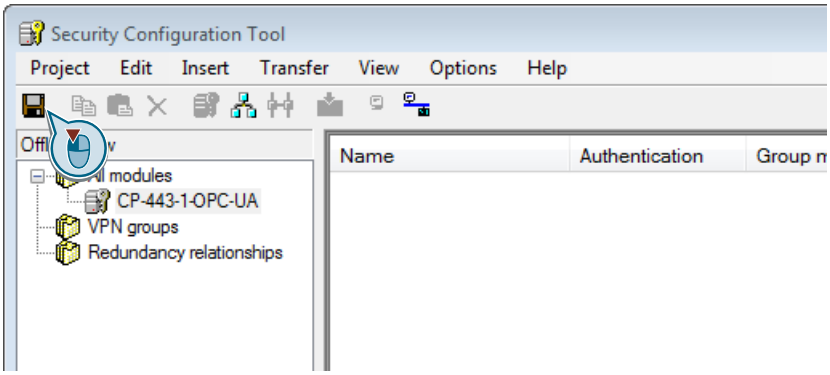


6. Go to the “Trustworthy certificates and root certification authorities” task card to import and thus accept certificates from other OPC UA servers and clients.
7. Click on the “Import...” button and, in the following dialog, navigate to the storage location of the certificates you want to accept on the CP443-1. Select

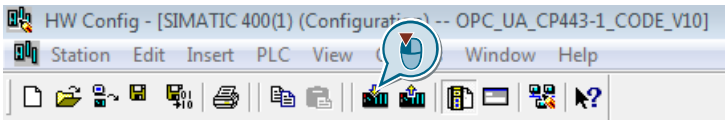
the certificates and acknowledge with “Open”.



8. Confirm with “OK”.
9. Save the settings made via the “Save project” button.



10. Close SCT.
11. Load the hardware configuration to your CPU via “Download to module”.

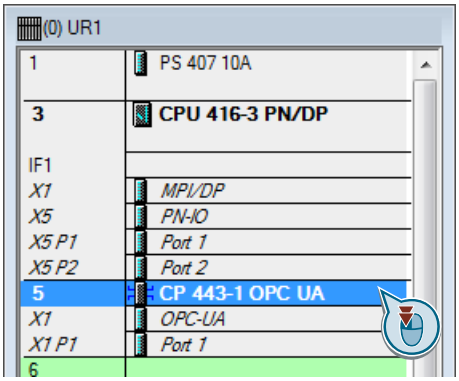


Note Repeat this procedure when the certificate of one of the connection partners changes.

2.1.6 Enabling tags for the OPC UA server

Once you have enabled the OPC UA server of the CP443-1 OPC UA, you can determine which PLC symbols to display via OPC UA. In addition, you can set if a symbol can only be read or also be written in. Proceed as follows:

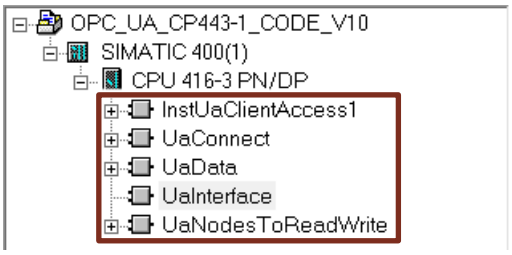
1. Go to the hardware configuration of the SIMATIC Manager in your project.
2. Double-click on the CP 443-1 in your rack.



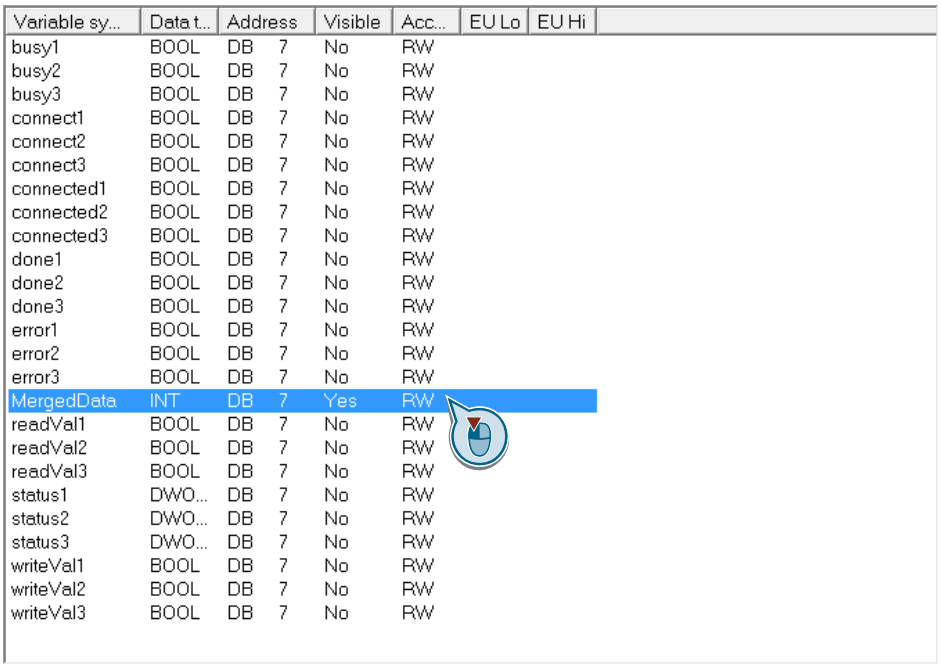
3. Go to the “OPC UA” task card.
4. In the “Use Symbols” area, you can decide whether to enable all PLC symbols for OPC UA or if you want to decide manually which symbols to display.
   - Check the “All symbols” check box to enable all symbols for OPC UA.
   - Check the “Configured symbols” check box to enable specific symbols for OPC UA.

5. When you have enabled the “Configure symbols” check box, you can use the “Configure” button to go to the “Configure symbols” settings mask.
   When you have enabled the “All symbols” check box, proceed with step 9.
6. The “Configure symbols” configuration mask enables you to browse through the different DBs of you PLC using a tree view. Select a DB to lock or enable its symbols for OPC UA.

Figure 2-2



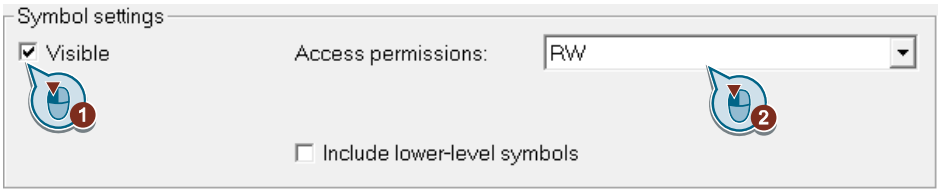
7. After selecting a DB, you can find the symbols contained in it on the right side of the input mask. Select a symbol you want to lock or enable for OPC UA.



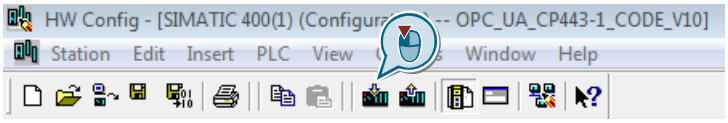
Variable sy...	Data t...	Address	Visible	Acc...	EU Lo	EU Hi
busy1	BOOL	DB 7	No	RW		
busy2	BOOL	DB 7	No	RW		
busy3	BOOL	DB 7	No	RW		
connect1	BOOL	DB 7	No	RW		
connect2	BOOL	DB 7	No	RW		
connect3	BOOL	DB 7	No	RW		
connected1	BOOL	DB 7	No	RW		
connected2	BOOL	DB 7	No	RW		
connected3	BOOL	DB 7	No	RW		
done1	BOOL	DB 7	No	RW		
done2	BOOL	DB 7	No	RW		
done3	BOOL	DB 7	No	RW		
error1	BOOL	DB 7	No	RW		
error2	BOOL	DB 7	No	RW		
error3	BOOL	DB 7	No	RW		
MergedData	INT	DB 7	Yes	RW		
readVal1	BOOL	DB 7	No	RW		
readVal2	BOOL	DB 7	No	RW		
readVal3	BOOL	DB 7	No	RW		
status1	DWO...	DB 7	No	RW		
status2	DWO...	DB 7	No	RW		
status3	DWO...	DB 7	No	RW		
writeVal1	BOOL	DB 7	No	RW		
writeVal2	BOOL	DB 7	No	RW		
writeVal3	BOOL	DB 7	No	RW		

8. After selecting a symbol, you can adapt the OPC UA settings for this symbol in the “Symbol settings” area.
   - Check the “Visible” check box to enable the symbols for OPC UA.
   - In the “Access permissions” drop-down menu, you can specify whether the symbol can only be read "R" or also be written in “RW”.

     The “Include lower-level symbols” check box defines whether the made settings should also be valid for the lower-level tags. This is only relevant for arrays, structures and UDTs related to the read and write rights.



9. Confirm with “OK”.
10. Load the hardware configuration to your CPU via “Download to module”.



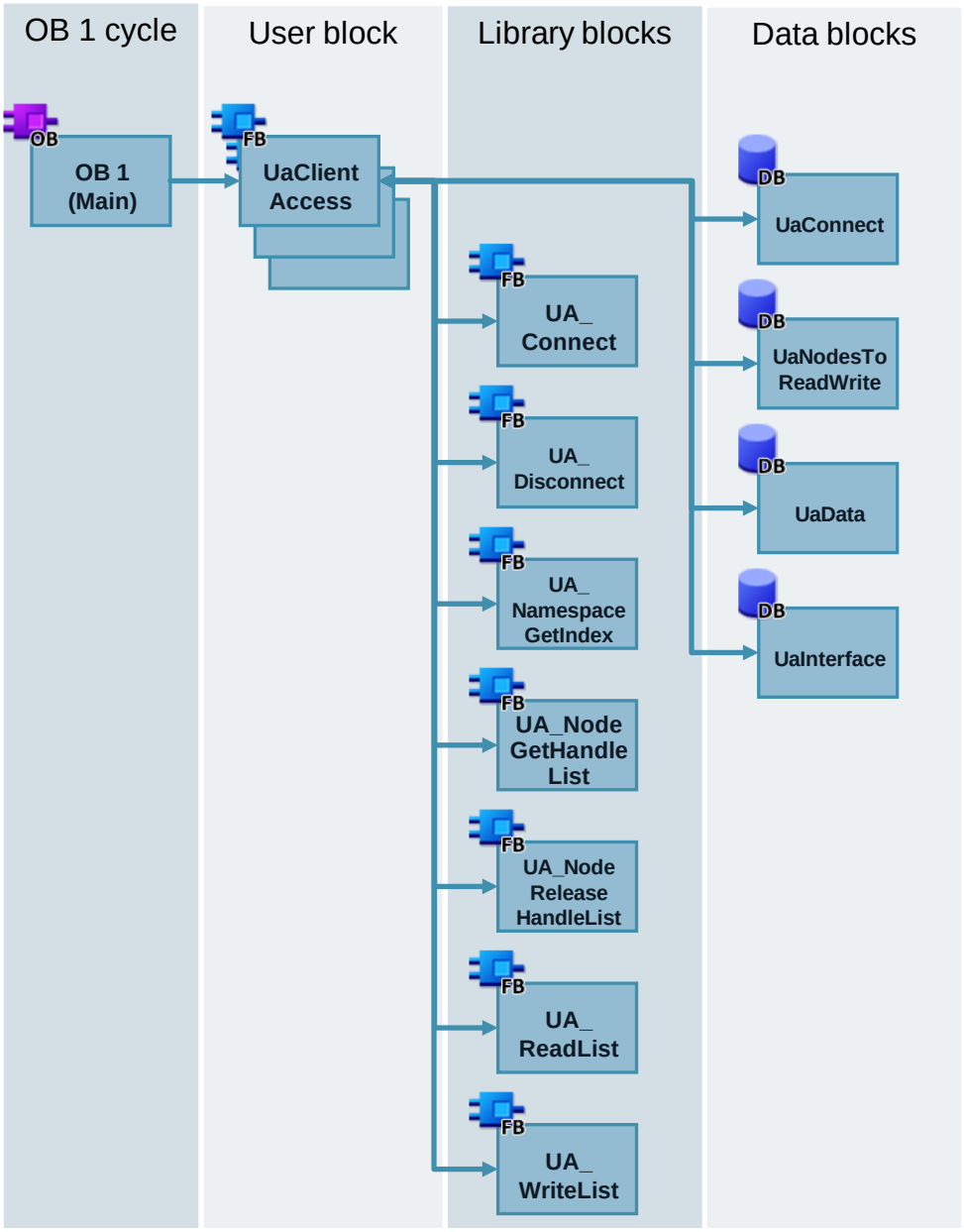
2.2 S7 user program for the OPC UA client functionality

After the OPC UA client functionality of the CP443-1 OPC UA has been enabled in the hardware configuration, the OPC UA client can be controlled using the blocks of the “SIMATIC_NET_CP” block library. This application example summarizes the individual OPC UA block in the “UaClientAccess” function block. The FB is therefore globally relevant for controlling the client functionality.

2.2.1 Program overview

The following figure shows the call hierarchy of the S7 user program of this application example:

Figure 2-3



Explanation of the blocks

The table below explains the function and data blocks called up in the user program:

Table 2-1

Name	Type	Description
Main	OB	Cyclic user OB, calling the user block “UaClientAccess”. The evaluation of the read OPC UA data will be performed in Main.
UaClientAccess	FB	User block which summarizes the individual OPC UA library blocks and simplifies their operation. You can use this block for your own project to read or write any OPC UA data.
UA_Connect	FB	Library block to connect with an OPC UA server.
UA_Disconnect	FB	Library block to disconnect from an OPC UA server.
UA_NamespaceGetIndex	FB	Library block to determine the namespace index of the namespace URI of the server.
UA_NodeGetHandleList	FB	Library block to register node IDs on the server.
UA_NodeReleaseHandleList	FB	Library block to enable registered node IDs on the server.
UA_ReadList	FB	Library block to read a registered node.
UA_WriteList	FB	Library block to write a registered node.
UaConnect	DB	Data block containing the connection information for establishing a connection with a server.
UaNodesToReadWrite	DB	Data block containing the node information of nodes to be read or written.
UaData	DB	Data block containing the process data read or to be written.
UaInterface	DB	Data block containing the tags to control the user program.

2.2.2 Function principle of the user program

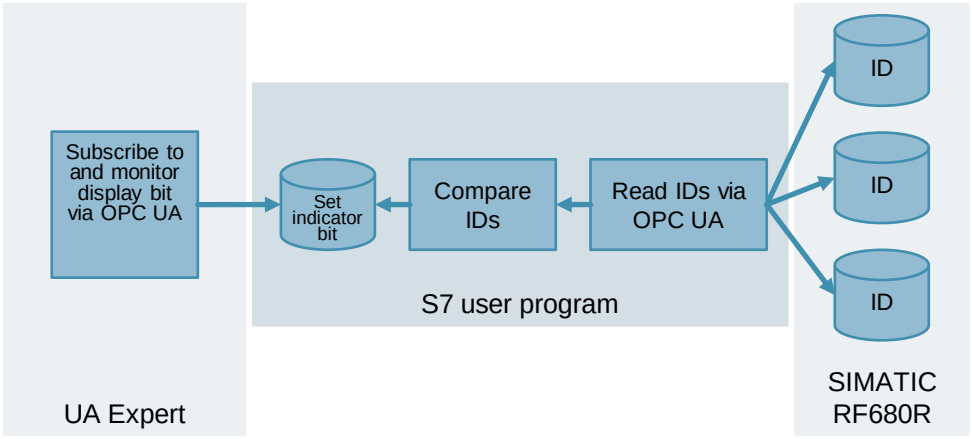
For each OPC UA server to be accessed by the CP443-1 OPC UA, there is one call of the higher-level user block “UaClientAccess” performed in OB 1. The input parameters of the function block are used to establish and terminate a connection to the OPC UA server and to assign a read or write job.

In this application example, “UaClientAccess” is used to read process data from a SIMATIC RF680R. The read data are stored in the “UaData” data block and contain the transponder IDs last read by the reader. These IDs are compared in OB 1 of the S7 user program. If the transferred data match, a display bit is set that is stored in the “UaInterface” data block.

The configured OPC UA server of the CP443-1 OPC UA provides this display bit to a higher-level OPC UA client. In this application example, the program “UA Expert” is used. The program is used to subscribe to the display bit via OPC UA, which can then be monitored continuously.

The following figure displays the function principle of the S7 user program:

Figure 2-4



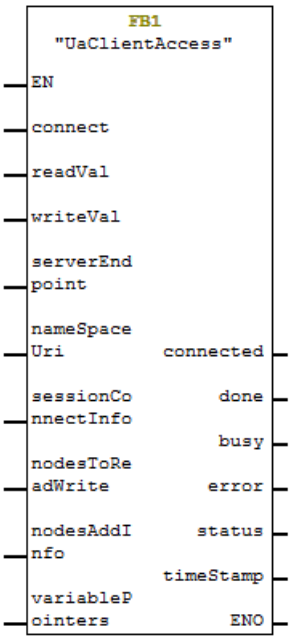
2.2.3 Program details of the block “UaClientAccess”

The function block “UaClientAccess” was created in SCL and contains two step sequences controlling the OPC UA client functionality. The step sequence is responsible for establishing and terminating the connection via the library blocks “UA_Connect” and “UA_Disconnect”. The second step sequence controls the data access, i.e. reading and writing tags. The following chronological sequence has been implemented:

1. The “UA_GetNamspaceIndex” block reads the namespace index of the given namespace URI.
2. The “UA_NodeGetHandleList” block registers the node ID on the server to be read/written. It returns a node handle.
3. The “UA_ReadList” or “UA_WriteList” block reads or writes on a tag using the node handle.
4. The “UA_NodeRealeaseHandleList” block removes the registration of the node handle on the server.

The figure below shows the “UaClientAccess” function block:

Figure 2-5



Block interface

Input parameters:

Table 2-2

Name	Data type	Description
connect	Bool	TRUE: Connection to the OPC UA server is established. FALSE: Connection to the OPC UA server is terminated.
readVal	Bool	Starts a read job at positive edge.
writeVal	Bool	Starts a write job at positive edge.
serverEndpoint	String	The ServerEndpointUrl (server address) for establishing a connection with an OPC UA

Name	Data type	Description
		server.
nameSpaceUri	String	The namespace URI of the OPC UA server for the tags to be read/written. Example: "http://www.siemens.com/SimaticIdent/RF600R/"
sessionConnectInfo	"UASessionConnectInfo"	Additional connection information.[1]
nodesToReadWrite	Array[1..1] of "UANodeID	Addressing of the tag nodes to be read/written.[2]
nodesAddInfo	Array[1..1] of "UANodeAdditionalInfo"	Specification of the tag nodes to be read/written.[3]
variablePointers	Array[1..1] of "UAAnyPointer	Reference to the memory area of the tag to be read/written within the PLC.[4]

Output parameter:

Table 2-3

Name	Data type	Description
connected	Bool	TRUE: Connection to the OPC UA server established successfully. FALSE: Connection to the OPC UA server terminated successfully.
done	Bool	TRUE: Job successfully executed.
busy	Bool	TRUE: Job in progress.
error	Bool	TRUE: Job aborted with error. See “Status” output parameter.
status	DWord	Output of error number, if "error" = TRUE. [5]
timeStamp	Array[1..1] of "UATimeStamp"	Outputs the time stamp for read tag nodes.

Call environment

The “UaClientAccess” block is an asynchronously working instruction; i.e., processing must be able to extend across several jobs (do not interconnect “EN” input parameter or set permanently to “true”) and therefore take place in a cyclic OB.

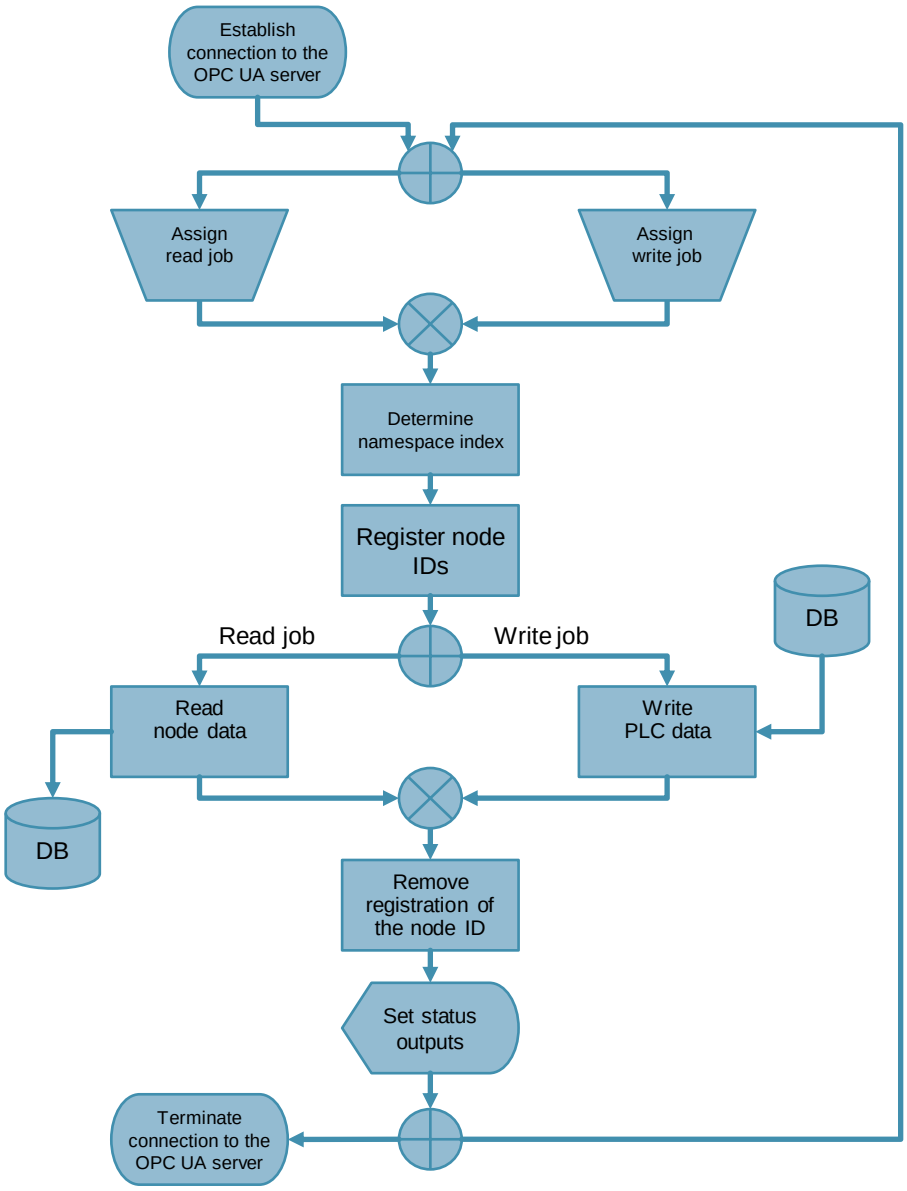
Note Only one OPC UA job can be performed at once per session. Each time the “UaClientAccess” block is called, a new session is started.

Functional sequence

The figure below shows the sequence of functions within the “UaClientAccess” function block:

[1] A description of the UDT “UASessionConnectInfo” can be found in the manual on the CP \5\.
[2] A description of the UDT “UANodeId” can be found in the manual on the CP \5\.
[3] A description of the UDT “UANodeAdditionalInfo” can be found in the manual on the CP \5\.
[4] A description of the UDT “UAAnyPointer” can be found in the manual on the CP \5\.
[5] A detailed description of the error number can be found in the manual on the CP \5\.

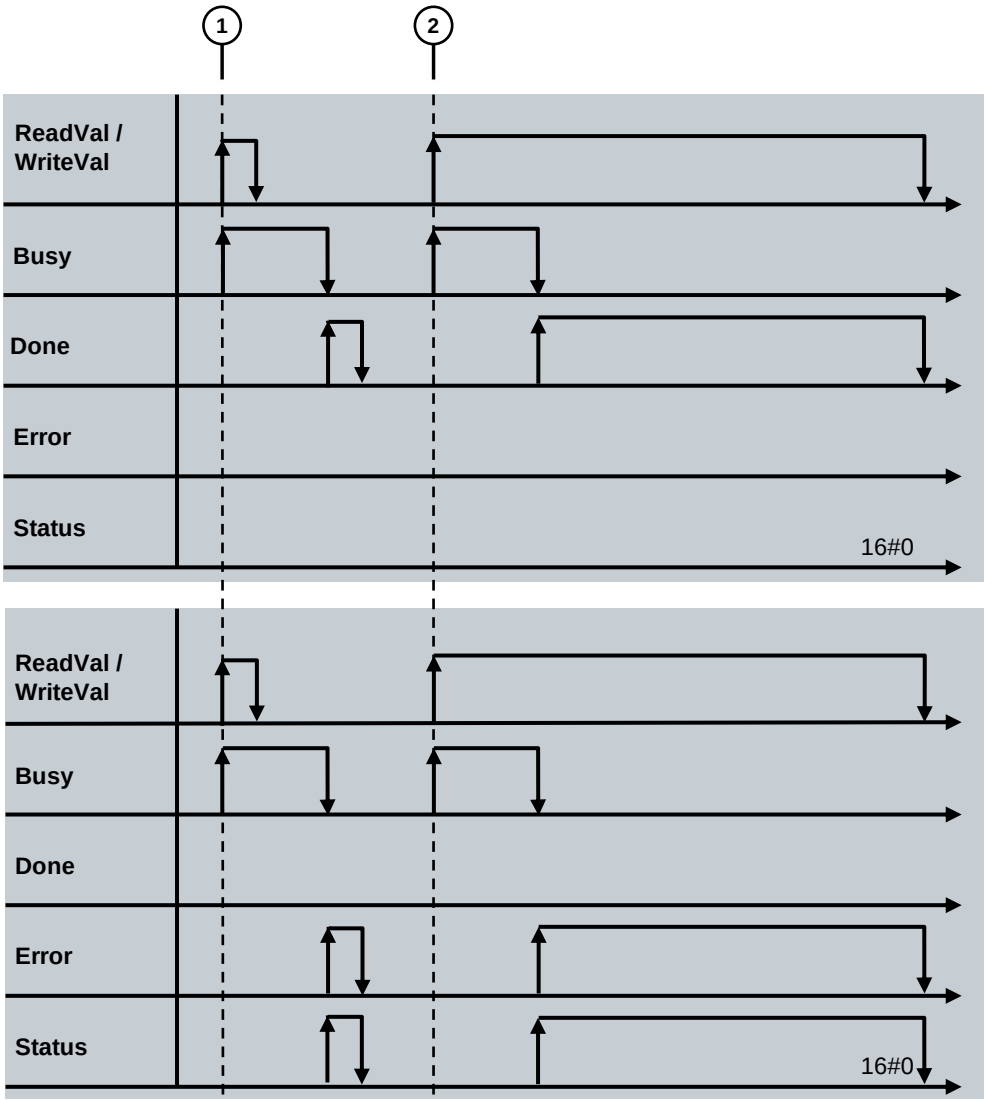
Figure 2-6



Functional sequence diagram

The figure below shows the functional sequence diagram of the “UaClientAccess” function block, in which the output parameter states “Done”, “Busy”, “Error” and “Status” are displayed chronologically depending on the setting of the input parameters “ReadVal” and “WriteVal”.

Figure 2-7



1. “ReadVal” or “WriteVal” are only set for one cycle. The block starts its processing and sets “Busy”. Once the operation is completed, “Busy” is reset.
   - When the processing was successful, “Done” is set for one cycle.
   - In case of an error, “Error” and the respective “Status” is set for one cycle.
2. “ReadVal” or “WriteVal” is set and remains in this state for several cycles. The block starts its processing and sets “Busy”. Once the operation is completed, “Busy” is reset.
   - When the processing was successful, “Done” is set as long as “ReadVal” or “WriteVal” maintain their state.
   - In case of an error, “Error” and the respective “Status” is set as long as “ReadVal” or “WriteVal” maintain their state.

Note When establishing a connection the “Connect” input parameter, the output parameters “Done”, “Error” and “Status” are set for one cycle only.

2.2.4 Program details on the OPC UA library blocks

The library blocks called in the higher-level function block “UaClientAccess” and which control the OPC UA client functionality, are contained in the “SIMATIC_NET_CP” block library.

Further information about the blocks can be found in the manual of CP443-1 OPC UA (\5\).

The block library download can be found in the delivery release of “SIMATIC_NET_CP” library V5.5.4 (\4\).

Note For the blocks to fully function, you need the following additional system functions:

- SFC20 (BLKMOV)
- SFC49 (LGC_GADR)
- SFC51 (RDSYSST)
- SFC64 (TIME_TCK)

2.2.5 Configuring the OPC UA client connection

The connection parameters required for the “UaClientAccess” block are stored in the “UaConnect” data block of the included project.

To configure the client connection, you have to enter the correct values in the cells in the “Actual value” column.

Note The example project included in the delivery contains three sets of all required parameter tags, because three different connections are established with calling up the “UaClientAccess” function block three times.

Specifying server addresses

Figure 2-8

Address	Name	Type	Initia	Actual value
0.0	serverEndpointUrls[1]	STRING [254]	''	'opc.tcp://192.168.0.254:4840'
256.0	serverEndpointUrls[2]	STRING [254]	''	'opc.tcp://192.168.0.254:4840'
512.0	serverEndpointUrls[3]	STRING [254]	''	'opc.tcp://192.168.0.254:4840'

Enter the server URLs in the “EndpointUrls[x]” lines. The “UaClientAccess” block establishes the connection to the OPC UA server entered in this field.

Example of a server URL:

```
'opc.tcp://192.168.0.254:4840'
```

Select connection parameters

Figure 2-9

768.0	sessionConnectInfos[1].SessionName	STRING [254]	''	'mySession1'
1024.0	sessionConnectInfos[1].ApplicationName	STRING [254]	''	''
1280.0	sessionConnectInfos[1].SecurityMsgMode	WORD	W#16#0	W#16#3
1282.0	sessionConnectInfos[1].SecurityPolicy	WORD	W#16#0	W#16#3
1284.0	sessionConnectInfos[1].CertificateStore	STRING [254]	''	''
1540.0	sessionConnectInfos[1].ClientCertificateName	STRING [254]	''	''
1796.0	sessionConnectInfos[1].ServerUri	STRING [254]	' '	' '
2052.0	sessionConnectInfos[1].CheckServerCertificate	BOOL	FALSE	FALSE
2054.0	sessionConnectInfos[1].TransportProfile	WORD	W#16#0	W#16#1
2056.0	sessionConnectInfos[1].UserIdentityToken.UserIdentit	WORD	W#16#0	W#16#0
2058.0	sessionConnectInfos[1].UserIdentityToken.TokenParam1	STRING [254]	''	''
2314.0	sessionConnectInfos[1].UserIdentityToken.TokenParam2	STRING [254]	''	''
2570.0	sessionConnectInfos[1].VendorSpecificParameter	WORD	W#16#0	W#16#3FFB
2572.0	sessionConnectInfos[1].SessionTimeout	TIME	T#0MS	T#20M
2576.0	sessionConnectInfos[1].MonitorConnection	TIME	T#0MS	T#15S
2580.0	sessionConnectInfos[1].LocaleIDs[1]	STRING [6]	''	''
2588.0	sessionConnectInfos[1].LocaleIDs[2]	STRING [6]	''	''
2596.0	sessionConnectInfos[1].LocaleIDs[3]	STRING [6]	''	''
2604.0	sessionConnectInfos[1].LocaleIDs[4]	STRING [6]	''	''
2612.0	sessionConnectInfos[1].LocaleIDs[5]	STRING [6]	''	''

Enter the connection parameters in the “EndpointUrls[x]” lines. The “UaClientAccess” block starts establishing a connection to an OPC UA server with the entered parameters.

The parameter tag “sessionConnectInfo” consists of the UDT “UASessionConnectInfo”. The following parameters are relevant for you:

Table 2-4

Parameter	Description
SessionName	Name of the session
SecurityMsgMode	The desired security procedure: • 0 = best possible procedure • 1 = no security procedure • 2 = Only authentication • 3 = Authentication and encryption
SecurityPolicy	The desired security profile: • 0 = best possible security profile • 1 = no security profile • 2 = Basic128Rsa15 • 3 = Basic256 • 4 = Basic256Sha256
VendorSpecificParameter	Input of the logic address of the CP 443-1 OPC UA. You can find it in the STEP 7 properties dialog of the CP as an input address in the “Addresses” register.
UserIdentityToken. UserIdentityTokenType	0 = no authentication 1 = authentication If the communication partner (server) requires the authentication via user name and password, you set this parameter to 1 and the following two according to the requirements of the server.
UserIdentityToken.TokenParam1	User name
UserIdentityToken.TokenParam2	Password
SessionTimeout	Maximum period of time to maintain a session without data traffic (in milliseconds): If this value is exceeded, the session is terminated. In this case you have to re-establish the connection by calling UA_Connect.

Parameter	Description
MonitorConnection	Connection monitoring time (in milliseconds): Period without data traffic, after which the client checks the connection to the server by sending a telegram.
LocaleIDs[1..5]	Optional language and region id according to RFC 3066. 0 = no or unknown LocaleID.

All other fields are either set automatically by STEP 7 or have to remain on the preconfigured value.

2.2.6 Configuring OPC UA tag connections

The tag parameters required for the “UaClientAccess” block are stored in the “UaNodesToReadWrite” data block of the included project.

To configure the tag parameters, you have to enter the correct values in the cells in the “Actual value” column.

Note The example project included in the delivery contains three sets of all required tag parameters, because three different tags are read when calling up the “UaClientAccess” function block three times.

Specifying node IDs

Figure 2-10

Address	Name	Type	Initia	Actual value
0.0	nodeIds1[1].NamespaceIndex	WORD	W#16#0	W#16#0
2.0	nodeIds1[1].Identifier	STRING [254]	''	'6030'
258.0	nodeIds1[1].IdentifierType	WORD	W#16#0	W#16#2

Enter the node IDs of the OPC UA node to be read or written in the “nodeIdsX[1]” lines.

The parameter tag “nodeIds” consists of the UDT “UANodeID”. The parameters contained in the UDT are explained in the following table:

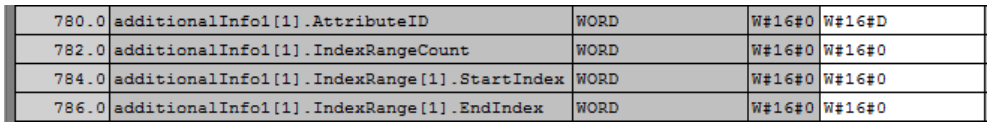
Table 2-5

Parameter	Description
NamespaceIndex	Namespace index of the server with the node.
Identifier	Specifies the node ID in the namespace index. Example for "IdentifierType" = 2: '6030'
IdentifierType	Specifies the format and scope of validity (normally the server) of the identifier. Supported types: • 1: String • 2: Numeric

Note The NamespaceIndex does not have to be entered when using the “UaClientAccess” function block, because the namespace index is queried by the client via the “UA_NamespaceGetIndex” function called up in the block.

Entering additional parameters

Figure 2-11



780.0	additionalInfo1[1].AttributeID	WORD	W#16#0	W#16#D
782.0	additionalInfo1[1].IndexRangeCount	WORD	W#16#0	W#16#0
784.0	additionalInfo1[1].IndexRange[1].StartIndex	WORD	W#16#0	W#16#0
786.0	additionalInfo1[1].IndexRange[1].EndIndex	WORD	W#16#0	W#16#0

Enter additional parameters in the “additionalInfoX[1]” lines. The “UaClientAccess” block requires these parameters to be able to specifically read or write to arrays. These additional parameters are irrelevant for elementary data types.

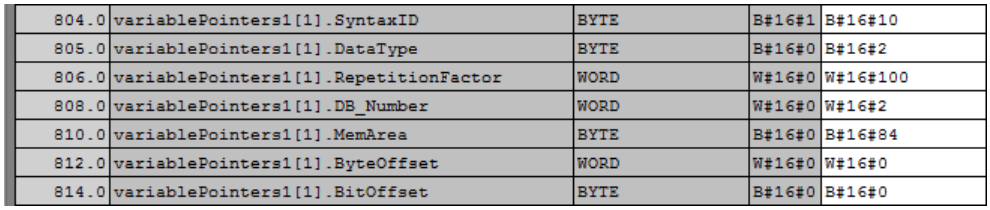
The parameter tag “additionalInfo” consists of the UDT “UANodeAdditionalInfo”. The parameters contained in the UDT are explained in the following table:

Table 2-6

Parameter	Description
AttributeID	Attribute of the items. Only attribute 13 (UAAI_Value) for the value of the item is supported.
IndexRangeCount	Number of index areas. For items with ARRAY data type, the following applies: • 0: One single index. The complete array is read/written. • 1: A sub-area of an array defined by "IndexRange".
IndexRange[1].StartIndex	Index where reading starts.
IndexRange[1].Endindex	Index where reading stops.

Defining tag pointers

Figure 2-12



804.0	variablePointers1[1].SyntaxID	BYTE	B#16#1	B#16#10
805.0	variablePointers1[1].DataType	BYTE	B#16#0	B#16#2
806.0	variablePointers1[1].RepetitionFactor	WORD	W#16#0	W#16#100
808.0	variablePointers1[1].DB_Number	WORD	W#16#0	W#16#2
810.0	variablePointers1[1].MemArea	BYTE	B#16#0	B#16#84
812.0	variablePointers1[1].ByteOffset	WORD	W#16#0	W#16#0
814.0	variablePointers1[1].BitOffset	BYTE	B#16#0	B#16#0

Enter additional parameters in the “variablePointersX[1]” lines. The “UaClientAccess” block requires these parameters to correctly address the PLC memory area of the tags to be read or written.

The tag pointer “variablePointers” consists of the UDT “UAAnyPointer”. The parameters contained in the UDT are explained in the following table:

Table 2-7

Parameter	Description
SyntaxID	The value for SyntaxID is always 10.
DataType	Data types of the target nodes. See “Data types” table below.
RepetitionFactor	Repetition factor (set to 256 for DataType “String”).
DB_Number	Number of the data block (DB): Alternatively, enter the number of the DB or a memory area. If you enter a DB, then enter a zero under "MemArea".

Parameter	Description
MemArea	Memory area: Alternatively, enter the number of the DB or a memory area. If you enter a memory area, then enter a zero under "DB_Number". See “Memory area” table below.
ByteOffset	Byte offset in the specified memory area where data access starts.
BitOffset	Bit offset in the specified memory area.

The table below explains the data type coding in the “DataType” parameter of the UDT “UAAnyPointer”:

Table 2-8 Data types

Hex code	S7 data type	Description
B#16#01	BOOL	Bit
B#16#02	BYTE	Bytes (8 bits). Is also used by STRING as a subordinate data type with RepetitionFactor = 256
B#16#03	CHAR	Characters (8 bits)
B#16#04	WORD	Word (16 bits)
B#16#05	INT	Integer (16 bits)
B#16#06	DWORD	Word (32 bits)
B#16#07	DINT	Integer (32 bits)
B#16#08	REAL	Floating-point number (32 bits)
B#16#09	DATE	Date
B#16#0A	TIME_OF_DAY	Time
B#16#0B	TIME	Time
B#16#0C	S5TIME	Data type S5TIME
B#16#0E	DATE_AND_TIME	Date and time (62 Bits)

The table below explains the memory area coding in the “MemArea” parameter of the UDT “UAAnyPointer”:

Table 2-9 Memory area

Hex code	Area	Description
B#16#80	P	IO memory area
B#16#81	E	Input memory area
B#16#82	A	Output memory area
B#16#83	M	Flag memory area
B#16#84	DB	Data block

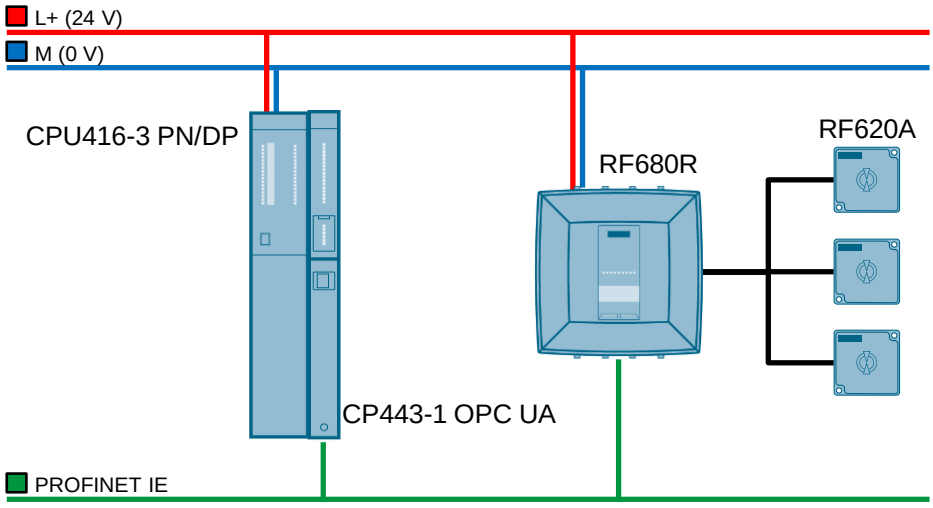
2.3 Commissioning

The following step-by-step instructions show you how to commission the application example.

2.3.1 Hardware setup

The figure below shows a schematic overview of the hardware configuration of this application example:

Figure 2-13



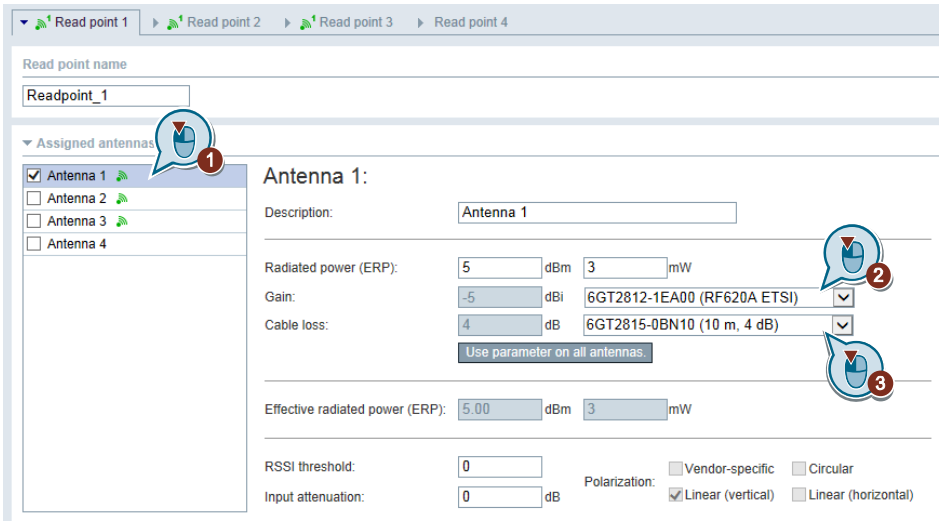
Note The setup guidelines for S7-400 (\6\) and RF680R (\7\) must generally be followed.

1. Plug the CPU into a suitable module rack.
2. Plug the CP into the next free slot on the rack next to the CPU.
3. Connect the CPU with suitable power supply.
4. Connect the reader with suitable power supply.
5. Connect the three reader antennae with the “ANT1”, “ANT2” and “ANT3” connections of the reader.
6. Connect the “X1P1” connection of the CP443-1 OPC UA with the “X1P1” connection of the readers via Ethernet.

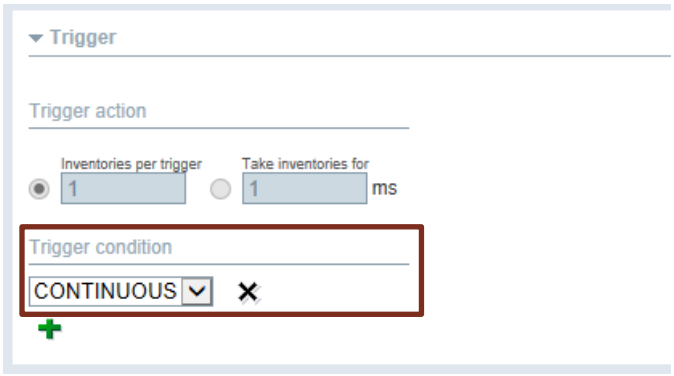
2.3.2 Preparing the OPC UA server of the RF680R

The SIMATIC RF680R servers as an OPC UA field device in this example. The reader provides Web Based Management (WBM) for configuration of the module. Three read points and the OPC UA server of the reader have to be activated via the WBM. Proceed as follows:

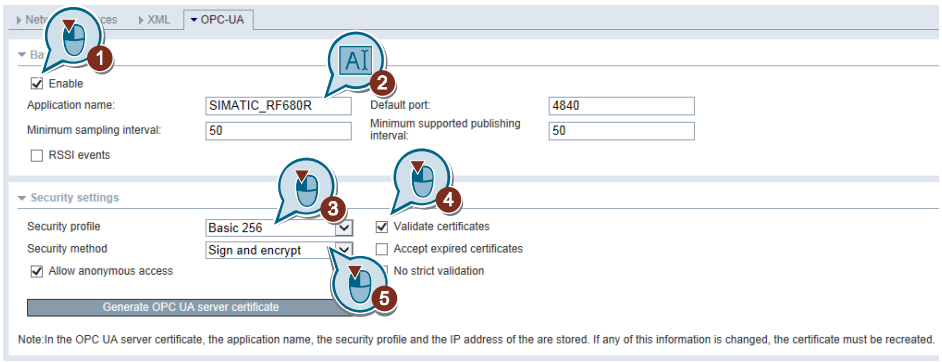
1. Open your Web browser.
2. Enter the IP address (factory default: 192.168.0.254) of the reader into the address bar of the browser and confirm with “Enter”.
3. Go to “Settings” > “Read point” in the menu.
4. Navigate to the “Read point 1” task card and activate “Antenna 1”. In the “Gain” drop-down list, select the antenna you have connected. Then, in the “Cable loss” drop-down list, select the antenna cable you are using.



5. Set the “Trigger condition” in the “Trigger” area to “CONTINUOUS” to have the reader perform read operations via the read point constantly.



6. Repeat steps 4 and 5 for “Read point 2” and “Read point 3”. However, only activate “Antenna 2” or “Antenna 3” for these two read points.
7. Navigate to “Settings” > “Communication” in the menu.
8. Go to the “OPC UA” task card.
9. Activate the “Enable” check box to activate the OPC UA server of the reader. Adding a name for the OPC UA server in the “Application name” field. Set the “Security profile” to “Basic 256” and activate the “Validate certificates” check box. Select the “Security method” “Sign and encrypt”.



10. Download the configuration to the reader with the “Transfer configuration to reader” button.

11. Click on “Generate OPC UA server certificate”.

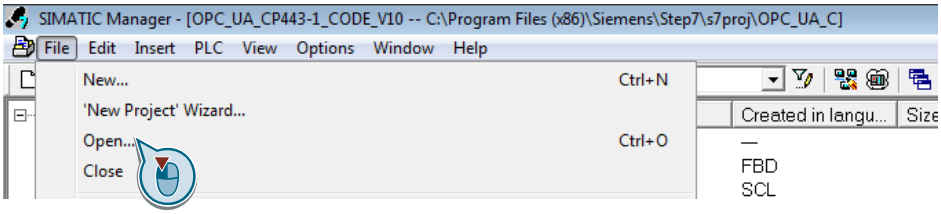
2.3.3 Preparing the STEP 7 project

Note The access data for the “Security login” of the hardware configuration of the sample project are as follows:

User name: User
Password: Siemens.1

Load the prepared STEP 7 sample project into your controller. The configuration steps described in chapter 2.1 Configuring the CP443-1 as OPC UA server and client have been made for you in this project. Proceed as follows:

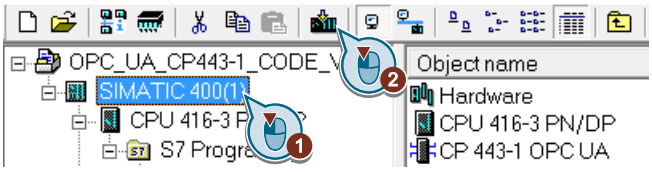
1. Download the "109743832_CP443-1_OPC_UA_V55_CODE_V10.zip" project onto your hard drive. The download can be found on the HTML page of this entry (\2\).
2. Unzip the zip file.
3. Start the SIMATIC Manager.
4. Navigate to “File > Open…”.



5. In the following dialog, click “Browse” and go to the storage location of the unpacked ZIP archive.
6. Select the unzipped project and confirm with “OK”.
7. The STEP 7 project is now open in SIMATIC Manager.

Note The configuration of the client has been prepared for you matching the configuration of the reader in chapter 2.3.2 Preparing the OPC UA server of the RF680R in this sample project.

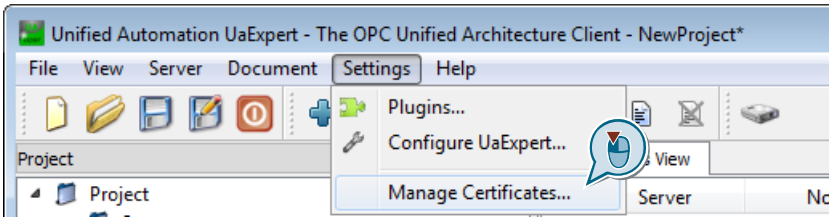
8. Select the SIMATIC 400 station in the SIMATIC Manager project tree and load the project to your controller with “Download”.



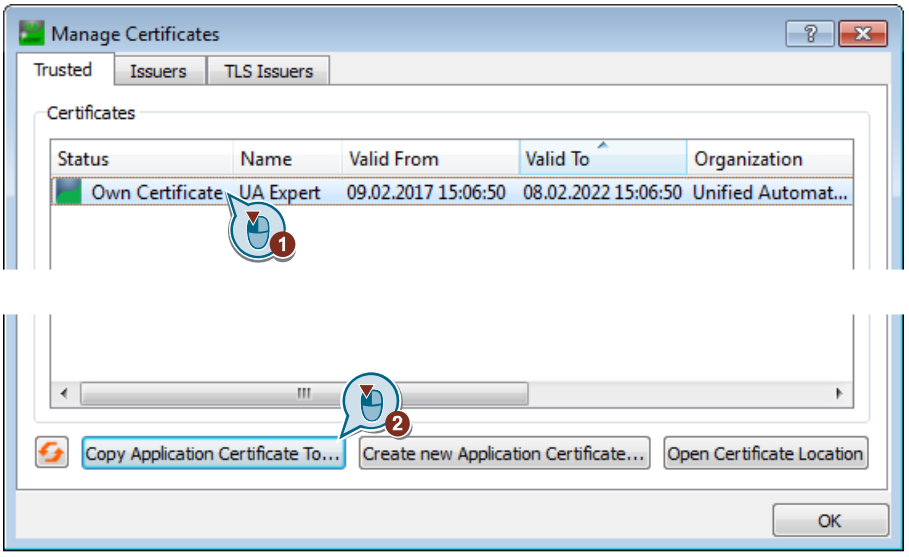
2.3.4 Preparing OPC UA client “UA Expert”

The program “UA Expert” serves as a higher-level OPC UA client in this application example, like it could be used on a MES or ERP level.

1. Download the installer for the UA Expert from the homepage of Unified Automation (\9\).
2. Start the installation and follow the instructions of the installer.
3. Run the program UA Expert after successfully installing the software.
4. In the main menu, go to “Settings” > “Manage Certificates…”.



5. In the following dialog, select the entry “Own Certificate” and press the “Copy Application Certificate To…” button.



6. In the next dialog, select a storage path and confirm with “Save”.
7. Import the certificate of UA Expert you have just saved as described in chapter 2.1.5 Managing certificates to your STEP 7 project.

2.4 Operation

The following chapters explain how to operate this application example.

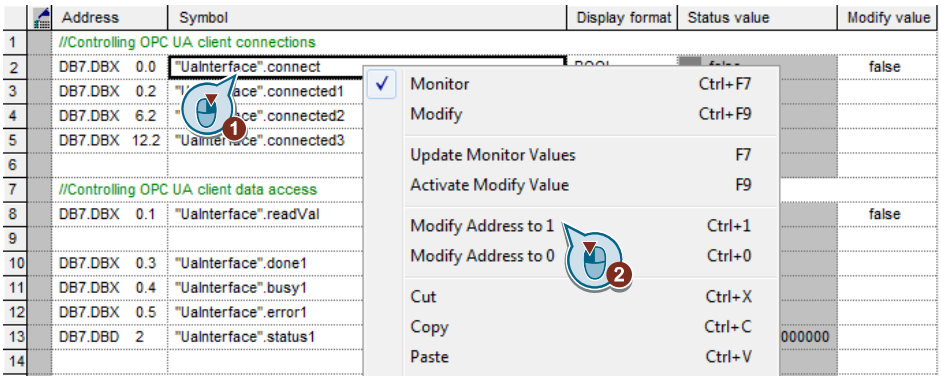
2.4.1 Reading process data from the field device

The transponder IDs read by the RFID reader RF680R are made available via its integrated OPC UA server. The client functionality of the CP 443-1 OPC UA is used to read the transponder IDs and to save them as a string in a data block. The S7 user program of the S7-400 compares the IDs and displays the result with a Bool tag (display bit).

To read the IDs, proceed as follows:

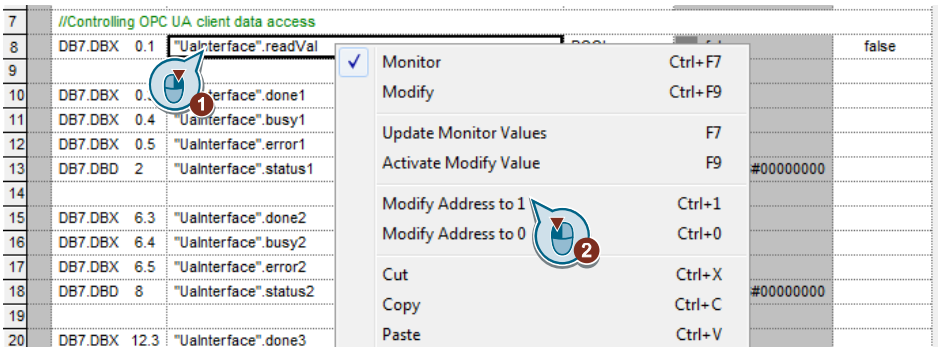
1. In the project tree of the SIMATIC Manager, go to “SIMATIC 400(1)” > “CPU 416-3 PN/DP” > “S7 Program(5)” > “Blocks”.
2. Open the “ControllOpcUa” watch table with a double click.
3. Confirm the “Monitor variable” button.

4. Right-click in the line of the “UaInterface.connect” symbol and click on “Modify Address to 1” in the context menu.

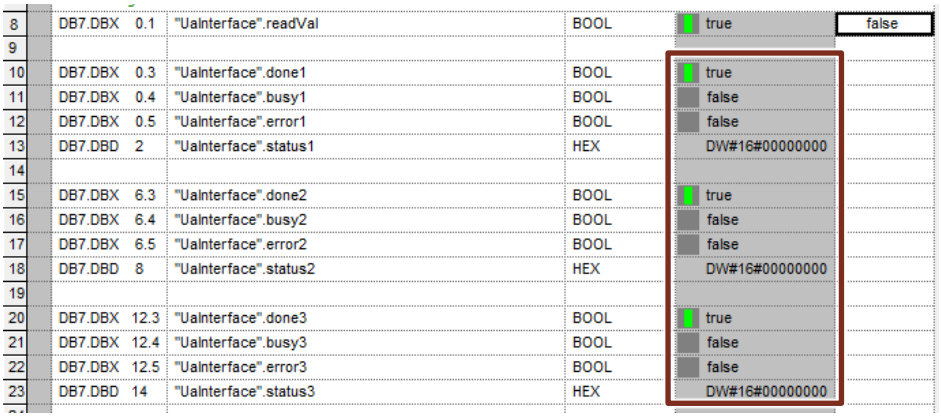


5. The CP443-1 OPC UA will now establish three connections to the RFID reader. After the connection has been established successfully, the “UaInterface.connected[1-3]” symbols are set to TRUE.

6. Hold any transponders in front of the three read points of the reader. Orange lit indicator LEDs on the reader indicate the transponders are detected by the reader.
7. Then right-click in the line of the “UaInterface.readVal” symbol and click on “Modify Address to 1” in the context menu.



8. The CP443-1 OPC UA now reads the transponder IDs of the three read points one after the other. Successful jobs are indicated by the “UaInterface.done[1-3]” symbols set to TRUE.

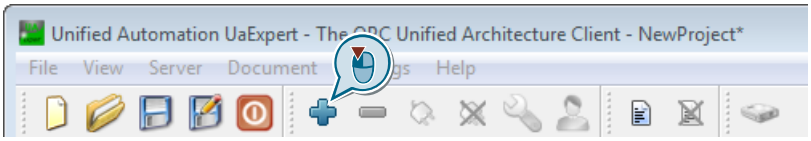


9. When the three transponder IDs are identical, the “UaInterface.compResult” symbol is set to TRUE. When the IDs are not identical, the symbol is set to FALSE.

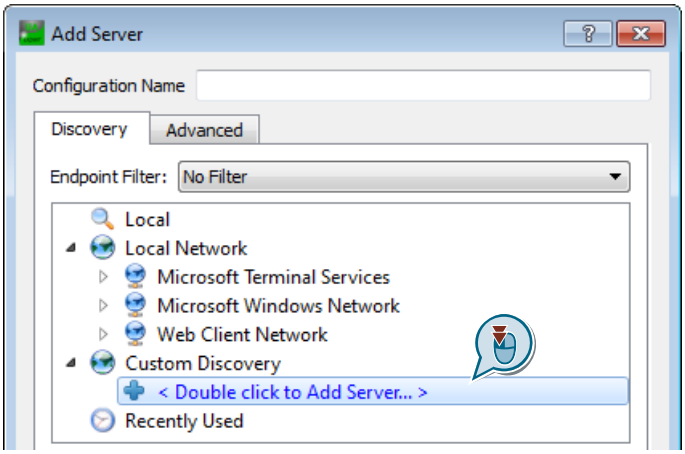
2.4.2 Reading process data from the server of the CP443-1 OPC UA with “UA Expert”

The higher-level OPC UA client “UA Expert” is used to read the result of the comparison of the transponder IDs from the server of the CP443-1 OPC UA via OPC UA. Proceed as follows:

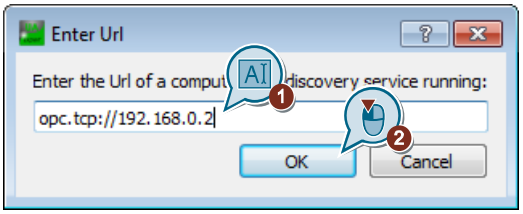
1. Start the UA Expert.
2. Then click on the “Add server” button.



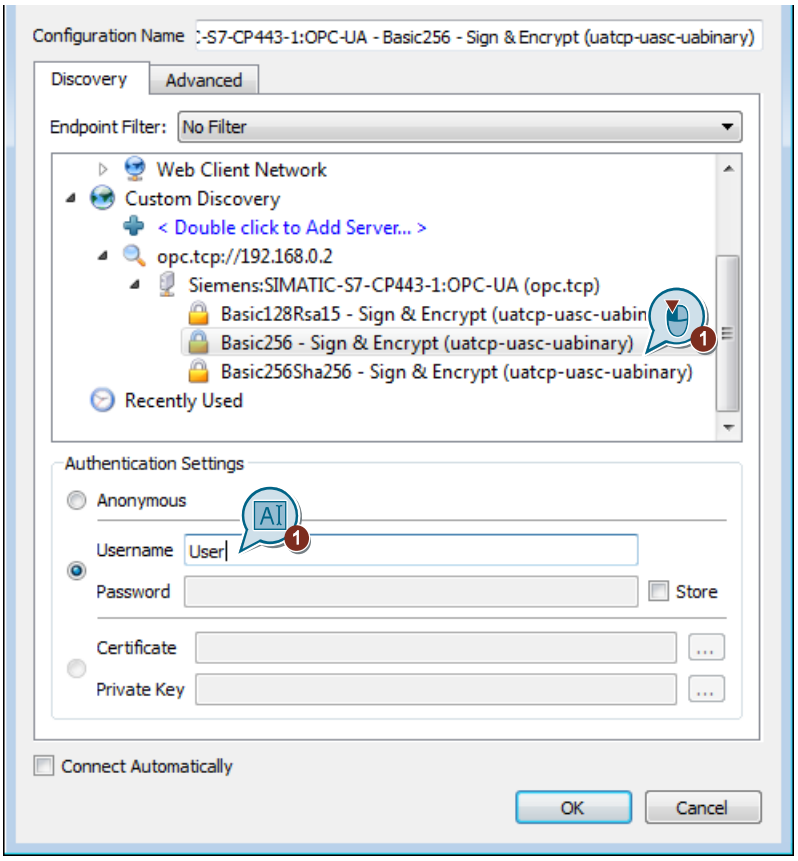
3. Double-click in the following dialog on “< Double click to Add Server… >” in the “Custom Discovery” area.



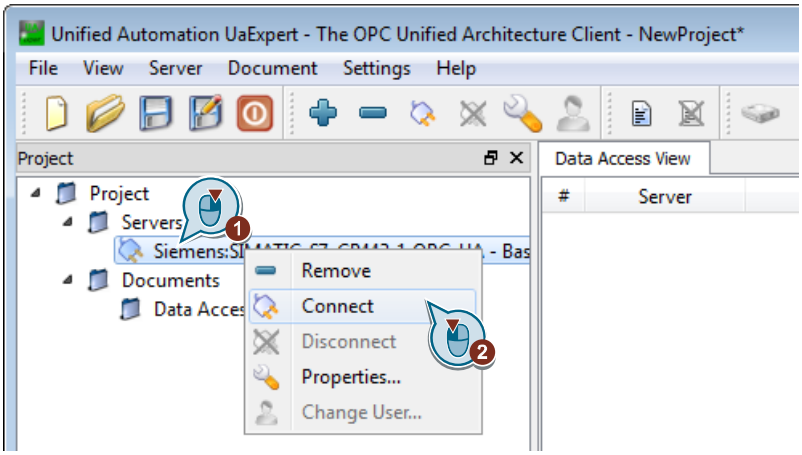
4. Enter the URL of the OPC UA server of the CP443-1 OPC UA (in the example: 192.168.0.2) and confirm with “OK”.



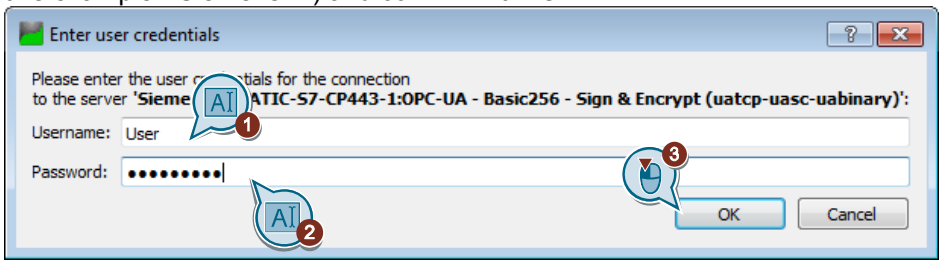
5. Select an OPC UA server endpoint, enter a user name in the “Username” field (e.g. “User”) for authentication and confirm with “OK”.



6. In the project explorer, right-click on the server you just added and then click on “Connect” in the context menu.

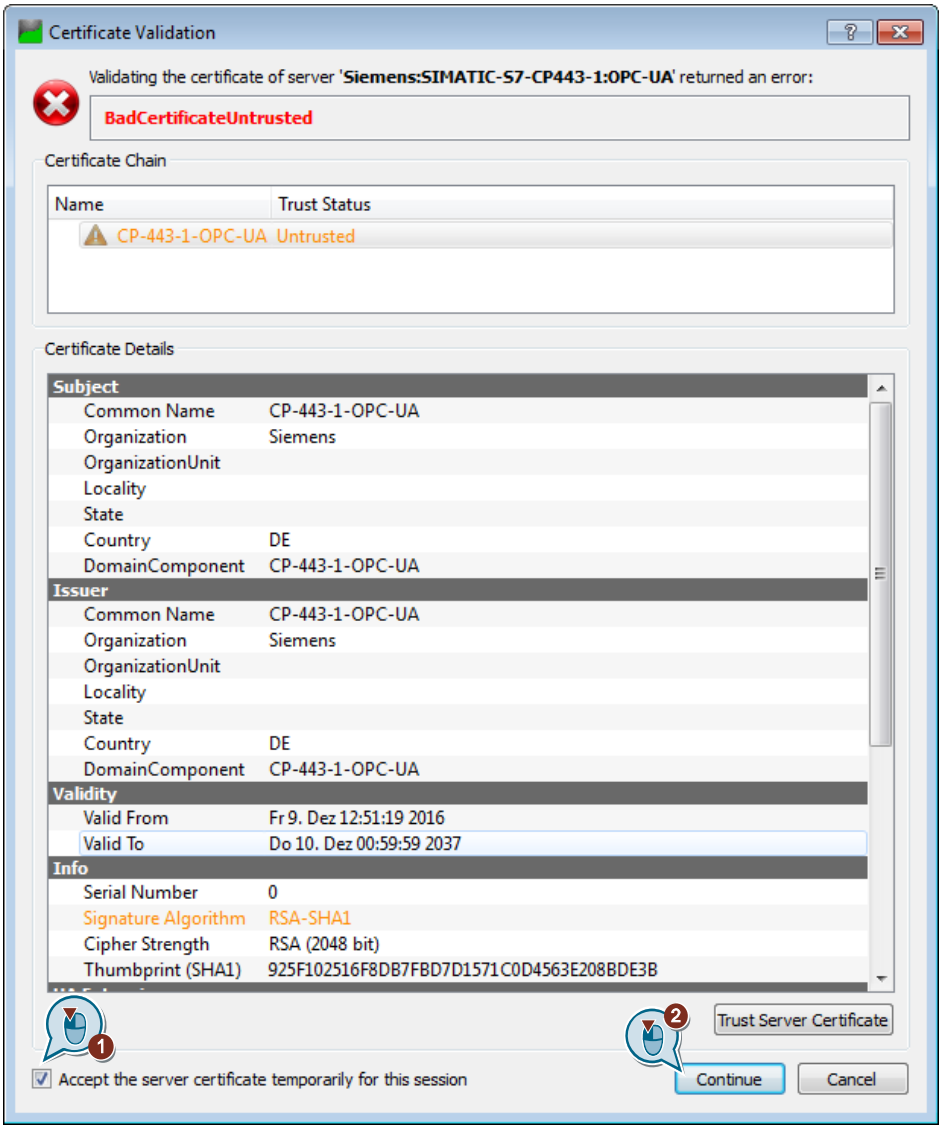


7. Enter the password of the OPC UA user of the CP443-1 OPC UA server (in this example: “Siemens.1”) and confirm with “OK”.



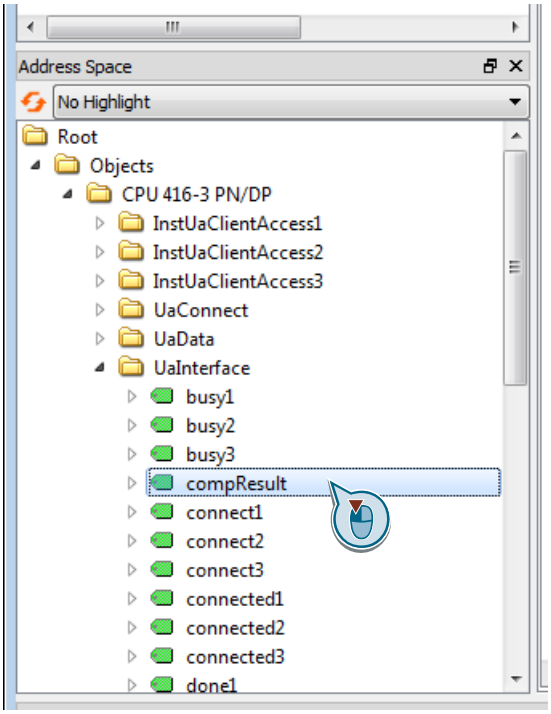
8. UA Expert now connects to the server of the CP443-1 OPC UA and authenticates itself with the entered user ID. The client UA Expert does not

know the server certificate upon the first login. Therefore, the OPC UA server sends its certificate to the client when establishing the connection. Accept the server certificate by checking the box “Accept the server certificate temporarily for this session” and then confirm with “Continue”. As an alternative, you can use the “Trust Server Certificate” button to accept the certificate for all future sessions as default.



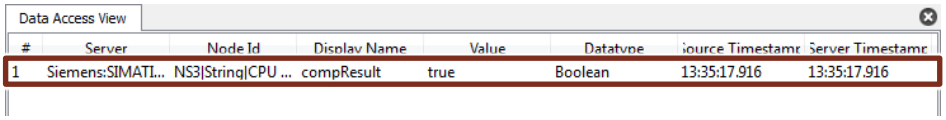
You are no connected with the OPC UA server of the CP443-1 OPC UA.

9. In the “Address Space” of the server, navigate to “Root” > “Objects” > “CPU 416-3 PN/DP” > “UaInterface” and drag the “compResult” tag into the “Data Access View” area.



Note As in this sample project, all tags of the CPU management for the OPC UA server of the CP 443-1 OPC UA are enabled, you can see all available data blocks within the CPU in UA Expert.

10. You can now view the evaluation of the transponder IDs in the “Data Access View” area. The value of the “compResult” display bit is displayed in the “Value” column.



3 Valuable Information

3.1 Basics of OPC UA

3.1.1 General OPC UA information

Overview

In recent years, the OPC Foundation (an interest grouping of well-known manufacturers for the definition of standard interfaces) has defined a large number of software interfaces to standardize the information flow from the process level to the management level. According to the different requirements within an industrial application, different OPC specifications have been developed in the past: Data Access (DA), Alarm & Events (A&E), Historical Data Access (HDA) and Data eXchange (DX). Access to process data is described in the DA specification, A&E describes an interface for event-based information, including acknowledgement, HDA describes functions for archived data and DX defines a lateral server to server communication.

Based on the experience with these classic OPC interfaces, the OPC Foundation defined a new platform, called OPC Unified Architecture (UA). The aim of this standard is the generic description and uniform access to all information which is to be exchanged between systems or applications. This includes the functionality of all previous OPC interfaces. Furthermore, this has generated the option of natively integrating the interface into the appropriate system, irrespective of which operating system the system is operated on and irrespective of the programming language in which the system was created.

More information can be found on the homepage or the OPC Foundation (\8\).

What is OPC?

In the past, OPC was a collection of software interfaces for data exchange between PC applications and process devices. These software interfaces have been defined according to the rules of Microsoft COM (Component Object Model) and can therefore be easily integrated into Microsoft operating systems. COM or DCOM (Distributed COM) provides the functionality of inter process communication and organizes the information exchange between applications, even across network boundaries (DCOM). Using mechanisms of the Microsoft operating system, an OPC client (COM client) can use it to exchange information with an OPC server (COM server).

The OPC server provides process information of a device at its interface. The OPC client connects itself with the OPC server and can access the offered data.

The use of COM or DCOM causes OPC servers and clients to run only on a Windows PC or in the local network and that the communication to the respective automation system has to be realized mainly via proprietary protocols. Additional tunneling tools often have to be used for the network communication between client and server in order to get through firewalls or to avoid the complicated DCOM configuration. The interface can furthermore only be accessed natively with C++ applications; .NET or JAVA applications can only gain access via a wrapper layer. In practice these restrictions lead to additional communication and software layers which increase the configuration workload and the complexity.

Due to the widespread use OPC, the standard is increasingly used for the general connection of automation systems and no longer only for the original application as driver interface in HMI and SCADA systems to access process information.

To solve the mentioned restrictions in real-life situations and to fulfill the additional requirements, the OPC Foundation has defined a new platform in the last 7 years, called OPC Unified Architecture, which offers a uniform basis for the exchange of

information between components and systems. OPC UA is available as an IEC 62541 standard and therefore also forms the basis for other international standards.

OPC UA offers the following features:

- Summary of all previous OPC features and information such as DA, A&E and HDA in a generic interface.
- Use of open and platform-independent protocols for inter-process or network communication.
- Internet access and communication by means of firewalls.
- Integrated access control and security mechanisms on protocol and application level.
- Extensive representation options for object-oriented models; objects can have tags and methods and can fire events.
- Expandable type system for objects and complex data types.
- Transport mechanisms and modeling rules form the basis for other standards.
- Scalability of small embedded systems up to business applications and from simple DA address spaces up to complex, object-oriented models.

3.1.2 OPC UA address space

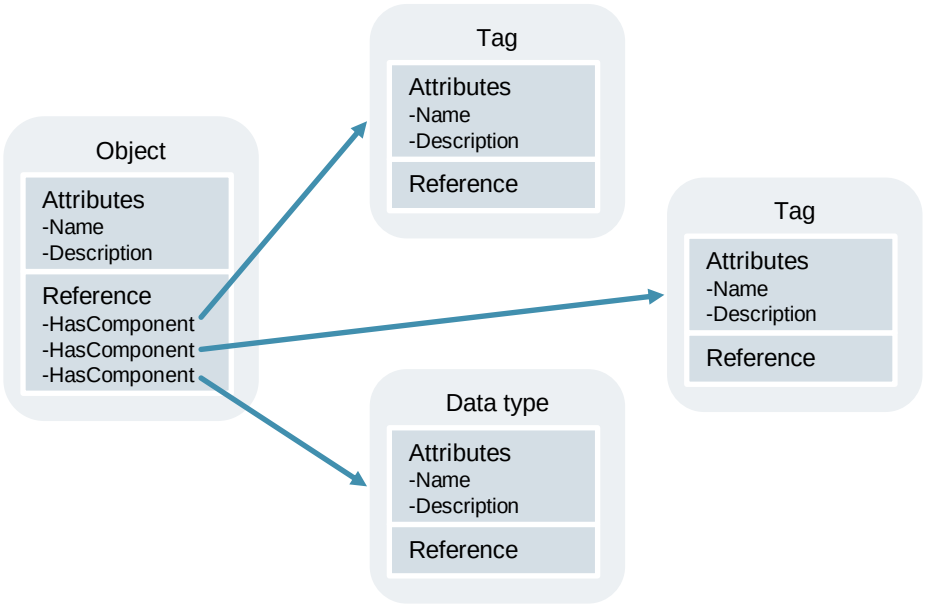
The following descriptions explain the address space of an OPC UA server.

Nodes in the address space

A node in the OPC UA address space is of a certain type, such as, for example, object, tag or method and is described by a list of attributes. All nodes have joint attributes such as name or description and specific attributes such as, for example, the value of a tag. The list of attributes cannot be extended. Additional information on the node can be added as property. Properties are a special type of tag. The nodes are interconnected with references. The references are typified. There are two main groups: Hierarchical references, such as, for example, HasComponent for the components of an object or non-hierarchical references such as, for example, HasTypeDefinition for a connection of an object instance to an object type.

The following figure shows an example for nodes and the connecting references:

Figure 3-1



Available types of nodes in the address space

The following table shows the node types defined in the standard:

Table 3-1

Node type	Description
Object	An object is used as typified container or folder for tags, methods and events.
Tag	Tags represent the data of objects or the properties of a node as attributes.
Method	Methods are components of objects and can have a list of input or output parameters. The parameters are described via defined attributes.
View	Views represent a part of the address space. The node is used as access point and as filter when browsing.
Object type	Object types supply information on the structure or the components of an object.
Tag type	Tag types typically describe which attributes or data types can be found in an instance of a tag.
Reference type	Reference types define the possible types of references between nodes.
Data type	Data types describe the content of the value in a tag.

Name spaces and node IDs

Each node in the OPC UA address space is uniquely identified by a node ID. This node ID is made up of a namespace to distinguish codes from different subsystems and a code which can either be a numerical value, a string or a GUID.

Strings are typically used for the ID. This is analog to OPC Data Access, where the item ID as identifier is also a string. Numerical values are used for statistical namespaces such as, for example, type system. OPC UA defines a namespace with associated namespace index for the nodes defined by the OPC Foundation. The OPC UA servers additionally define one or several namespaces with index. The namespaces defined by the servers are variable and can change. This is why

it is recommended to request the current namespace for the client when establishing the session.

The figure below explains the structure of a node ID:

Figure 3-2

Table 3-2

No.	Description
1.	Namespace index
2.	Node ID type (s=String; i=Numeric: g=GUID)
3.	Identifier (Id)

Attributes of the nodes

The table below explains the most important attributes defining a node:

Table 3-3

Attribute	Node type	Description
Node ID	All	The unique node ID with namespace index
Namespace index	All	The namespace index that is assigned to the node.
Identifier Type	All	The node ID type
Identifier	All	The unique node ID within the namespace index
Browse Name	All	The browse name
Display Name	All	The display name
Node Class	All	The node class (object, tag, data type)
Description	All	Short description of the node
Type Definition	All	Reference for data type description of the tag
Write Mask	All	Write rights to node attributes (0=no, 1=yes) without consideration of user groups
User Write Mask	All	Write rights to node attributes (0=no, 1=yes) without consideration of the current user
Data Type	Tag	Data type of the tag
Value Rank	Tag	Value type of the tag (none, scalar, vector, array)
Array Dimensions	Tag	Number of array dimensions
Access Level	Tag	Access authorization (read, write, read/write) to the node
Minimum Sampling Interval	Tag	The smallest possible sampling interval of the tag on the server side
Historizing	Tag	Course of time of the tag available on server (yes, no)

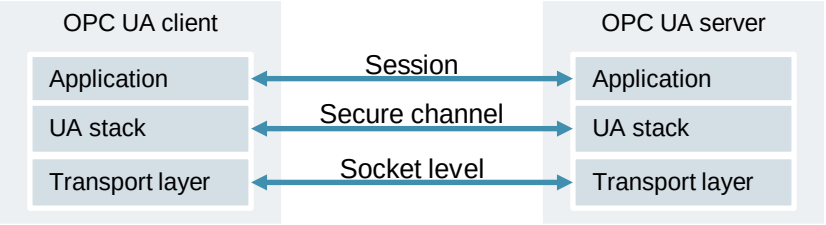
3.1.3 OPC UA Security

The following explanations outline the security concept of OPC UA.

Security layers

The following figure gives an overview of the security layers of OPC UA:

Figure 3-3



The user authentication is carried out via the **Session**. This is done, for example, through a user name and a password or via certificates.

Via a **Secure Channel** the applications are mutually authenticated and a message-based security of the communication is performed. Each message is signed and encrypted to ensure the integrity and secrecy of the messages. Basis of these mechanisms are certificates (X509) which uniquely identify the applications based on a Public Key Infrastructure (PKI) system.

On the **socket level**, a connection-oriented security of the socket connection via Secure Socket Layer (SSL) or via Virtual Private Network (VPN) can be used in addition or as an alternative to the secure channel.

Configuration options for the security

The following table describes the different configuration options for the security mechanisms:

Table 3-4

Option	Description
Security Policy	**None** – In the secure channel no security is used. **Basic128Rsa15** – Set of encryption algorithms. **Basic256** – Set of expandable encryption algorithms.
Message Security Mode	**None** – The messages are not secured. **Sign** – The messages are signed. **Sign&Encrypt** – The messages are signed and encrypted.
User Authentication	**Anonymous** – User authentication is not necessary. **User Password** – The user authentication is performed using user names and password. **Certificate** – The user authentication is performed using a certificate.

Certificate exchange between client and server

When all applications involved, implement the guidelines of the OPC UA regarding the security configuration, only one manual step (4) is necessary at the server for the exchange of certificates, since the certificates are automatically exchanged between the applications and the certificates only have to be accepted by an administrator.

The following figure illustrates the certificate exchange between client and server:

Figure 3-4

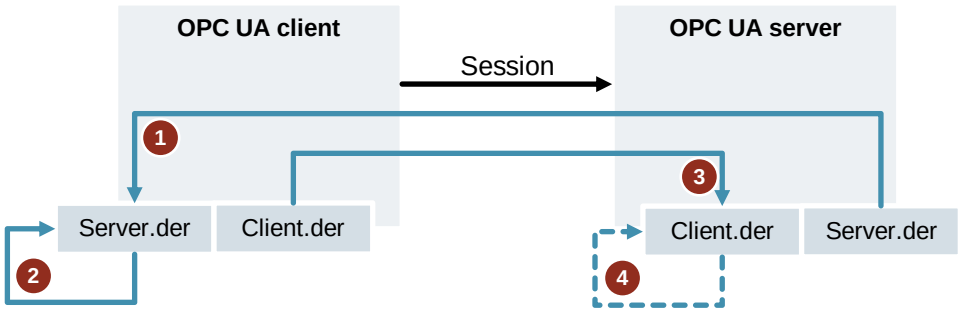


Table 3-5

No.	Description
1.	When establishing a connection to the server (Session.Create), the client receives the server certificate via the server endpoint.
2.	The client program can then decide how it deals with the certificate: Reject or accept.
3.	In the same process the client sends its certificate to the server. The server rejects the certificate at first and then stores it in a reject folder.
4.	As a result, the client certificate has to be accepted manually by an administrator on the server. In most cases, this is done by an administrator copying the client certificate from a reject folder into a trusted folder.

Note

For the OPC UA server/client of the CP443-1 OPC UA, the client or server certificate has to be loaded via the SIMATIC Manager onto the controller, in order to accept it.

4 Annex

4.1 Service and support

Industry Online Support

Do you have any questions or need support?

Siemens Industry Online Support offers access to our entire service and support know-how as well as to our services.

Siemens Industry Online Support is the central address for information on our products, solutions and services.

Product information, manuals, downloads, FAQs and application examples – all information is accessible with just a few mouse clicks at:
https://support.industry.siemens.com/ .

Technical Support

Siemens Industry's Technical Support offers quick and competent support regarding all technical queries with numerous tailor-made offers – from basic support to individual support contracts.

Please address your requests to the Technical Support via the web form:
www.siemens.com/industry/supportrequest .

Service offer

Our service offer comprises, among other things, the following services:

- Product Training
- Plant Data Services
- Spare Parts Services
- Repair Services
- On Site and Maintenance Services
- Retrofit & Modernization Services
- Service Programs and Agreements

Detailed information on our service offer is available in the Service Catalog:
https://support.industry.siemens.com/cs/sc

Industry Online Support app

Thanks to the “Siemens Industry Online Support” app, you will get optimum support even when you are on the move. The app is available for Apple iOS, Android and Windows Phone.
https://support.industry.siemens.com/cs/ww/en/sc/2067

4.2 Links and Literature

Table 4-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to this entry page of this application example https://support.industry.siemens.com/cs/ww/de/view/109743832
\3\	Download page of the Security Configuration Tool https://support.industry.siemens.com/cs/ww/en/view/109744041
\4\	Download page of the SIMATIC_NET_CP block library https://support.industry.siemens.com/cs/ww/en/view/109738487
\5\	Operating instructions - Industrial Ethernet CP 443-1 OPC UA https://support.industry.siemens.com/cs/ww/en/view/109738422
\6\	SIMATIC S7-400 Automation System https://support.industry.siemens.com/cs/ww/en/view/1117740
\7\	SIMATIC Ident RFID Systems SIMATIC RF600 https://support.industry.siemens.com/cs/ww/en/view/109743277
\8\	Homepage of the OPC Foundation https://opcfoundation.org/
\9\	Download UA Expert https://www.unified-automation.com/products/development-tools/uaexpert.html?gclid=CMfumaOk_tECFUG4GwodbpEGKg

4.3 Change documentation

Table 4-2

Version	Date	Modifications
V1.0	05/2017	First version