Managing SSL certificates in Codesys 3

Level 4	1 – Fundamental – No previous experience necessary 2 – Basic – Basic knowledge recommended 3 – Advanced – Reasonable knowledge required 4 – Expert – Good experience recommended

Powering Business Worldwide

Services

For service and support, please contact your local sales organisation.

Contact details: Eaton.com/contacts
Service page: Eaton.com/aftersales

Original Application Note

Original document is the German version of this document.

Translation

All non-German language versions of this document are translations of the original application note.

1. Edition 2025, publication date 07/2025

DANGER!
DANGEROUS ELECTRICAL VOLTAGE!

Before commencing the installation

- Installation requires qualified electrician
- Disconnect the power supply of the device.
- Ensure that devices cannot be accidentally retriggered
- Verify isolation from the supply
- Ground and short-circuit.
- Cover or enclose neighbouring units that are live.
- Follow the engineering instructions (IL) of the device concerned.
- Only suitably qualified personnel in accordance with EN 50110-1/-2 (VDE 0105 part 100) may work on this device/ system.
- Before installation and before touching the device ensure that you are free of electrostatic charge.
- The functional earth (FE, PES) must be connected to the protective earth (PE) or to the potential equalizing. The system installer is responsible for implementing this connection.
- Connecting cables and signal lines should be installed so that inductive or capacitive interference do not impair the automation functions.
- Install automation devices and related operating elements in such a way that they are well protected against unintentional operation.
- Suitable safety hardware and software measures should be implemented for the I/O interface so that a line or wire breakage on the signal side does not result in undefined states in the automation devices.
- Ensure a reliable electrical isolation of the low voltage for the 24 V supply. Only use power supply units complying with IEC 60364-4-41 or HD 384.4.41 S2 (VDE 0100 part 410).
- Deviations of the mains voltage from the nominal value must not exceed the tolerance limits given in the technical data, otherwise this may cause malfunction and dangerous operation.
- Emergency-Stop devices complying with IEC/EN 60204-1 must be effective in all operating modes of the automation devices. Unlatching the emergency switching off devices must not cause restart.
- Built-in devices for enclosures or cabinets must only be run and operated in an installed state, desk-top devices or portable devices only when the housing is closed.
- Measures should be taken to ensure the proper restart of programs interrupted after a voltage dip or failure. This should not cause dangerous operating states even for a short time. If necessary, emergency switching off devices should be implemented
- Wherever faults in the automation system may cause damage to persons or property, external measures must be implemented to ensure a safe operating state in the event of a fault or malfunction (for example, by means of separate limit switches, mechanical interlocks, etc.).

Disclaimer

The information, recommendations, descriptions, and safety notations in this document are based on Eaton's experience and judgment and may not cover all contingencies. If further information is required, an Eaton sales office should be consulted. Sale of the product shown in this literature is subject to the terms and conditions outlined in the applicable Terms and Conditions for Sale of Eaton or other contractual agreement between Eaton and the purchaser. THERE ARE NO UNDERSTANDINGS, AGREEMENTS, WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE OR MERCHANTABILITY, OTHER THAN THOSE SPECIFICALLY SET OUT IN ANY EXISTING CONTRACT BETWEEN THE PARTIES. ANY SUCH CONTRACT STATES THE ENTIRE OBLIGATION OF EATON. THE CONTENTS OF THIS DOCUMENT SHALL NOT BECOME PART OF OR MODIFY ANY CONTRACT BETWEEN THE PARTIES. As far as applicable mandatory law allows so, in no event will Eaton be responsible to the purchaser or user in contract, in tort (including negligence), strict liability, or otherwise for any special, indirect, incidental, or consequential damage or loss whatsoever, including but not limited to damage or loss of use of equipment, plant or power system, cost of capital, loss of power, additional expenses in the use of existing power facilities, or claims against the purchaser or user by its customers resulting from the use of the information, recommendations, and descriptions contained herein. The information contained in this manual is subject to change without notice.

Content

1 SSL – Secure Sockets Layer / TLS – Transport Layer Security

SSL stands for "Secure Sockets Layer" and is a standard technology for establishing an encrypted connection between a server and a client. Often the new version TLS (Transport Layer Security) is also called SSL, but the updated version offers higher security. With the help of SSL technology, two systems can transmit data in encrypted form so that the messages cannot be read by third parties. This requires certificates and keys, which in turn use encryption algorithms to encrypt the data.

1.1 Asymmetric and Symmetric key

Two encryption methods are used to establish SSL-encrypted communication.

- **Asymmetric encryption / public-key encryption**

Uses two different keys for encryption and decryption. The key for encrypting a message (public key) can be used by anyone. However, this encrypted message can then only be decrypted and read with the key for decryption (private key). The most common encryption algorithm is RSA, ECC or Diffie-Hellman.

- **Symmetric encryption / Pre-shared key encryption**

Uses one key for encryption and decryption. This key must be available to the sender and the receiver so that messages can be exchanged in a meaningful way.

The encrypted communication structure can thus be roughly summarized in four steps.

1. After the request from the client, the server sends a copy of its SSL certificate including an asymmetric public key to the client.
2. Client verifies the certificate and creates a symmetric key, encrypts it with the asymmetric public key and sends it to the server.
3. Server decrypts the message with its asymmetric private key to obtain the symmetric key and confirms this to the client.
4. Server and client now encrypt and decrypt all transmitted data with the symmetric key. This enables encrypted communication for this connection. When a new connection is established, a new key is created.

1.2 SSL certificates

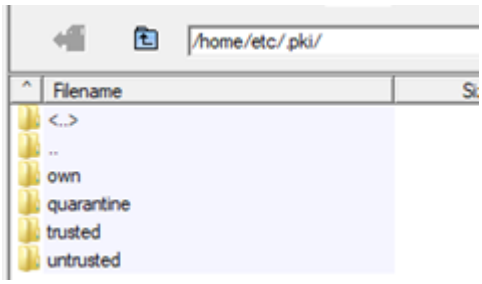
SSL certificates are small files with a digital binding of the key to an organization. On a web server, the certificate causes the https protocol to be used, allowing a secure connection from a web server to a browser. In the browser, a secure connection is visually represented by the lock.

2 Creating certificates in CODESYS

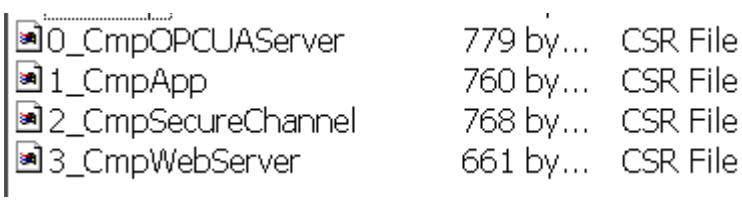
CODESYS offers the possibility to configure individual security features. The "Security Screen" is available for this purpose, in which the use of certificates and encryption can be set to enable access to project data only by means of a digital signature. CODESYS uses certificates in X.509 format for the security features, which are stored in the PKI directory. To enable encryption, the devices must support OpenSSL and have the CmpSecurityManager component.

XV300:

XC300:



There are two types of certificates that can be used by the Runtime in this context. On the one hand self-signed certificates, which can be generated e.g. via the PLC Shell, or CA signed certificates, which are generated or signed by an official authority (e.g. VeriSign, GeoTrust, ...). For this the command "cert-createcsr" must be executed for the corresponding application in the PLC Shell, whereby a CSR (Certificate Signing Request) is created and stored in the directory cert/export.



The CSR file can be transferred via file transfer and sent to a certification authority. After the certificate has been signed, it can be transferred back to the device via file transfer into the directory cert/import and imported into the runtime with the PLC shell command "cert-import".

In the following the available PLC Shell commands for security are listed:

- cert-getapplist

Shows for which components a certificate is already registered. In addition one receives information concerning the runtime of the certificate and the ID number of the components. This is needed for the use of other commands.

```
cert-getapplist

Nr.    ComponentName            CommonName                    CertAvailable DateNotBefore
-------------------------------------------------------------------------------------------
0      CmpOPCUAServer           OPCUAServer@WM01              FALSE         --
1      CmpApp                   WM01                  FALSE         --
2      CmpSecureChannel         WM01                  TRUE          2021-1-13T22:58:57.0
3      CmpWebServer             WM01                  TRUE          2021-1-13T22:58:57.0
```

- cert-genselfsigned <number of the component in the applist>

This command can be used to generate self-signed certificates for the respective components. A successful generation of the certificates can be seen in the PLC log.

```
cert-genselfsigned 0

Generate selfsigned certificate with given index (0). Check logger to see when finished.
```

- cert-gendhparams [len in bits]

Generates the parameters for the Diffie-Hellman key exchange. Attention: This process can take several minutes!

```
cert-gendhparams

Generation of the parameter started. This may take SEVERAL hours.
Please do not turn off the PLC.
Check the logger if the generation has finished.
```

- cert-getcertlist [<trustlevel>]

Lists all certificates of the selected trust level. If this is not specified, all certificates are displayed. Possible trust levels are

- untrusted (not trustworthy)
- trusted (trustworthy)
- own (certificates of the controller)
- quarantine (certificates could not be validated)

```
cert-getcertlist


------------------------------------------------------
List of own certificates
------------------------------------------------------
Number: 0
Thumbprint: 401e6fb676da07ec5516f4765b62512c74be970c
Subjects:
 - commonName: OPCUAServer@WM01
Key usage(s): Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment, Certificate Sign
Extended key usage(s): TLS Web Server Authentication
Valid from: 27.1.2021 0:49:47
Valid until: 27.1.2022 0:49:47
------------------------------------------------------
Number: 1
Thumbprint: 8b5793f43754450e1a2c69a5e70543e723ebf5bf
Subjects:
 - commonName: WM01
 - unstructuredName: Vendor: Eaton Automation
 - unstructuredName: Device: Plc3 V3.5.16.20-5427 192.168.119.133
 - serialNumber: 101200001813
Extended key usage(s): TLS Web Server Authentication
Valid from: 13.1.2021 22:58:57
Valid until: 12.2.2021 22:58:57
```

- cert-createcsr [<number retrieved by “cert-getapplist”>]

Generates CSR files for the components.

```
cert-createcsr 2

Create CSR for application with given index. Check logger to see when finished.
```

- cert-import <trustlevel> <filename>

Imports the specified certificate from the cert/import directory

```
cert-import own signature.cer

Import Succeeded.
```

- cert-export <trustlevel> [<number retrieved by "cert-getcertlist">]

Exports the specified certificate to the cert/export directory.

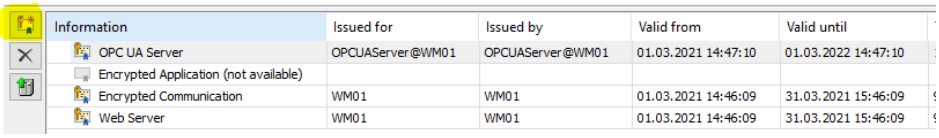
```
cert-export own 0

Export certificate Nr. 0 of specified trustlevel.
Certificate Nr. 0 exported to $cert$/export/4c5761fe84cfa941ad5da34b22d0dccd59a8d8be.cer.
```

cert-remove <trustlevel> <number retrieved by "cert-getcertlist" or "all">

Removes the listed certificates.

```
cert-remove own 0

Remove certificate Nr. 0 of specified trustlevel.
Successfully removed certificate Nr. 0.
```

In principle, certificates can also be created directly in the Security Screen. To do this, select the corresponding component and click the "Create new certificate" button:



3 Encrypted application

To encrypt the download or an online change of an application on a controller, the option "Force encryption of download, online changes and boot applications" can be activated. This means that applications on the controller can only be exchanged with a valid certificate. This requires an X.509 certificate for the CmpApp component. This certificate must then be present in the CODESYS software during a download.

First, it is checked whether a certificate has already been generated:

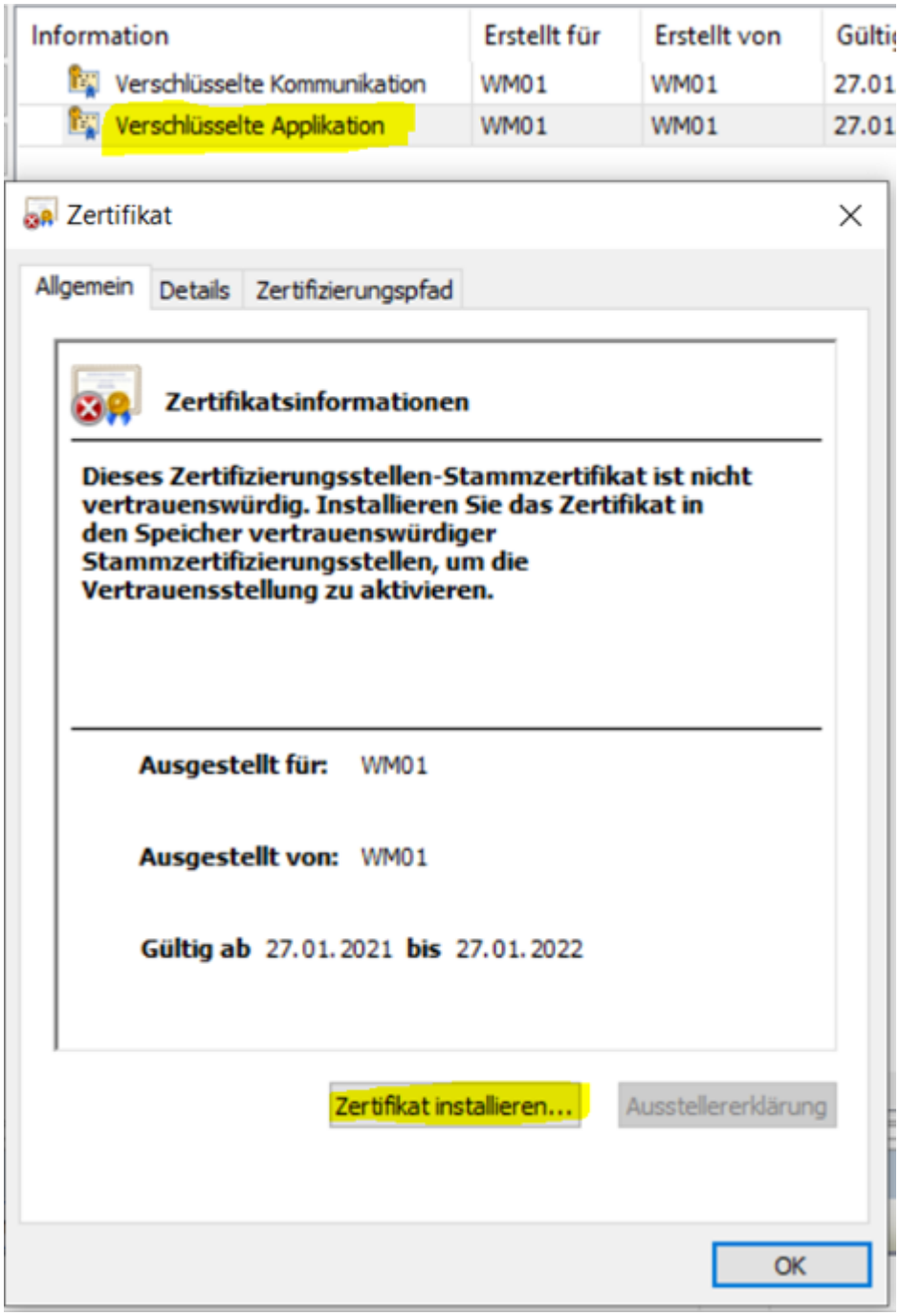
```
cert-getapplist

Nr.    ComponentName              CommonName                    CertAvailable DateNotBefore
       DateNotAfter               Thumbprint
-------------------------------------------------------------------------------------------------
-------------------------------------------------------------
0      CmpOPCUAServer             OPCUAServer@WM01              FALSE         --
                    --
1      CmpApp                     WM01                          FALSE         --
          --
2      CmpSecureChannel           WM01                          TRUE          2021-1-27T2:49:36.0
27T2:49:36.0          87a2a53ade19f4be88c603877d2965013be014ee
3      CmpWebServer               WM01                          TRUE          2021-1-27T2:49:36.0
27T2:49:36.0          87a2a53ade19f4be88c603877d2965013be014ee
```

If this is not the case, a new certificate can be generated with the command "cert-genselfsigned 1". A successful generation is reported in the Log tab.

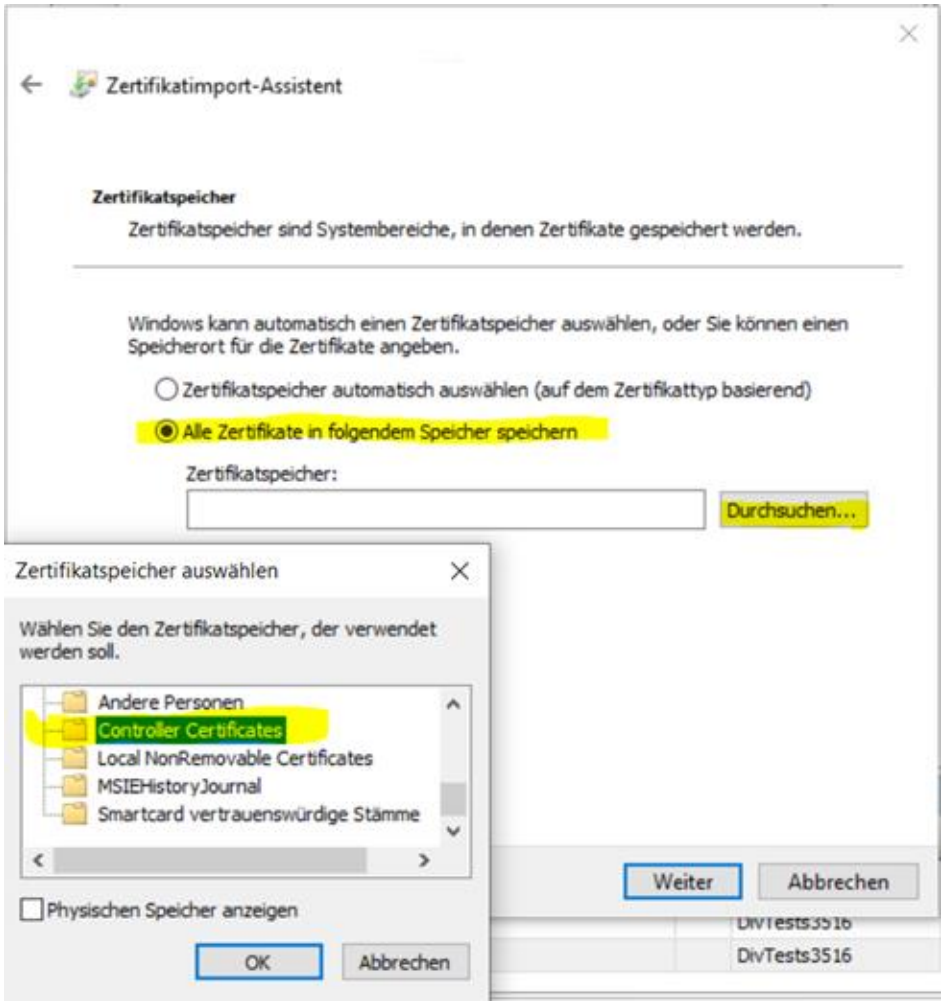
```
cert-genselfsigned 1

Generate selfsigned certificate with given index (1). Check logger to see when finished.
```

The generated certificate must now be transferred to the PC so that it can be integrated in the CODESYS software. The existing certificates on the device can now also be viewed in the Security Screen in the devices tab.

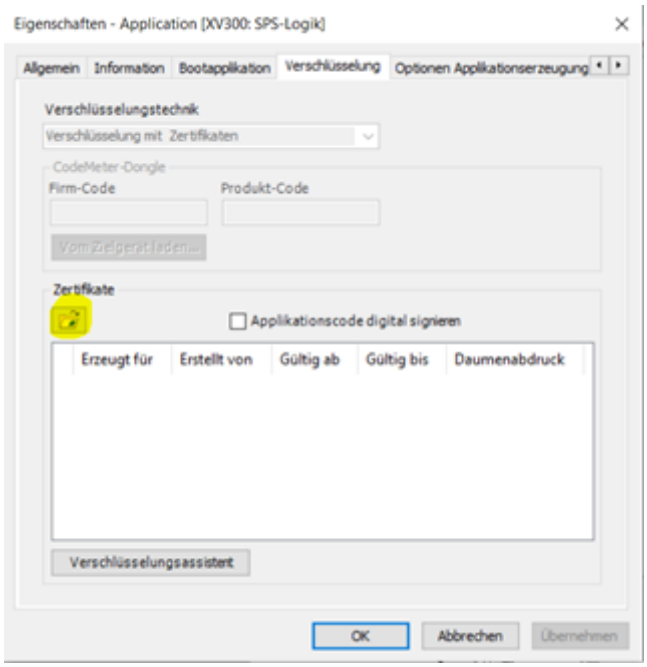
Double-click on the "Encrypted application" certificate to open the standard Windows dialog for certificates.

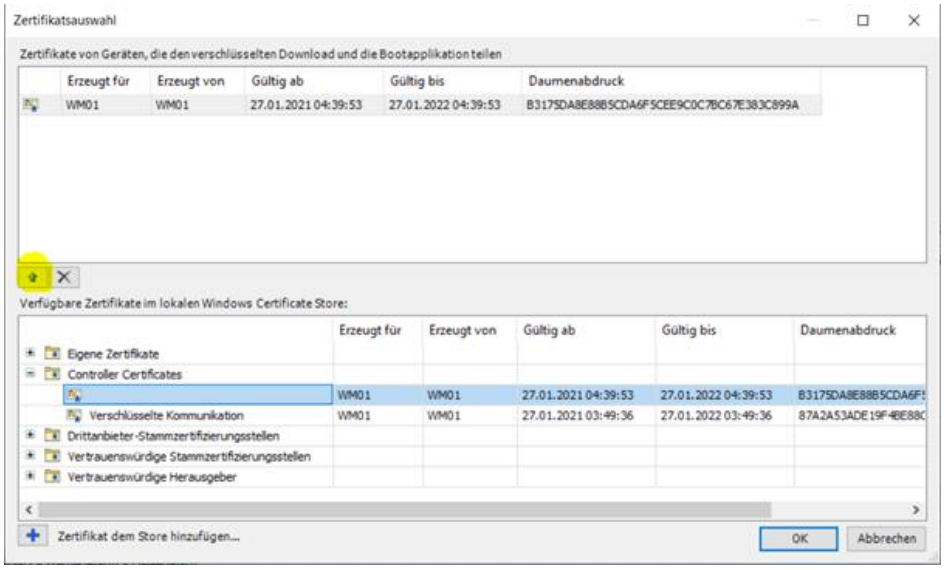


The certificate is stored in the "Controller Certificates" certificate store.

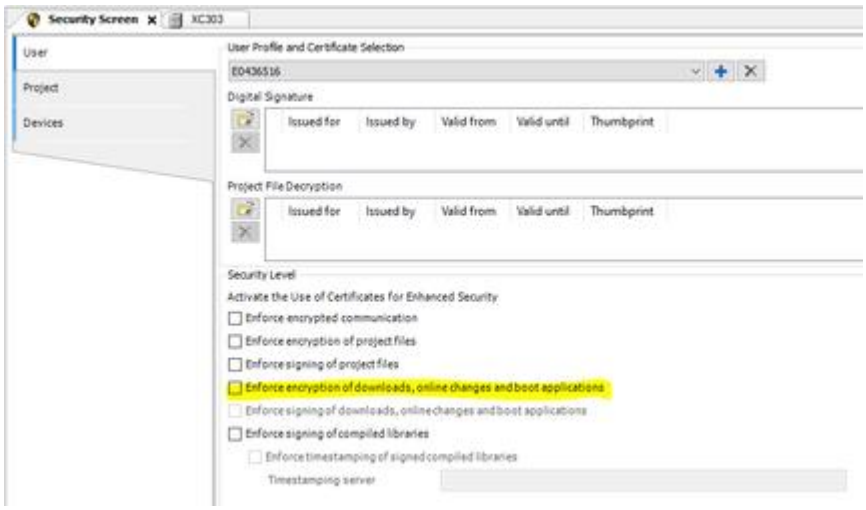


After successful import, this certificate is available for encrypting the download, online change and boot application. Now this certificate must also be selected in CODESYS. In the Security Screen in the Project tab, the certificate is assigned to the project. A double click on the application opens the properties of the application, where the certificate must be selected.





The transfer of a boot application, a download of the project or an online change are now encrypted. For this the following option must be activated:



4 Encrypted online communication

The security functions enable encrypted online communication to a PLC Runtime. The encryption is done via the TLS protocol within the CODESYS communication protocol. An X.509 certificate and a key for the CmpSecureChannel component are required for this. The configuration takes place in the Security Screen. First, however, a certificate for CmpSecureChannel must be generated on the controller. To do this, the "cert-getapplist" command is first executed in the PLC shell to check whether a certificate has already been created for this component:

```
cert-getapplist

Nr.    ComponentName              CommonName                    CertAvailable DateNotBefore
-------------------------------------------------------------------------------------------
-----
0      CmpOPCUAServer             OPCUAServer@WM01              FALSE         --
1      CmpApp                     WM01                  FALSE       --
2      CmpSecureChannel           WM01                  FALSE       --
3      CmpWebServer               WM01                  FALSE       --
```

If there is no certificate for the CmpSecureChannel component yet, then a certificate can be generated with the "cert-genselfsigned 2" command.

```
cert-genselfsigned 2

Generate selfsigned certificate with given index (2). Check logger to see when finished.
```

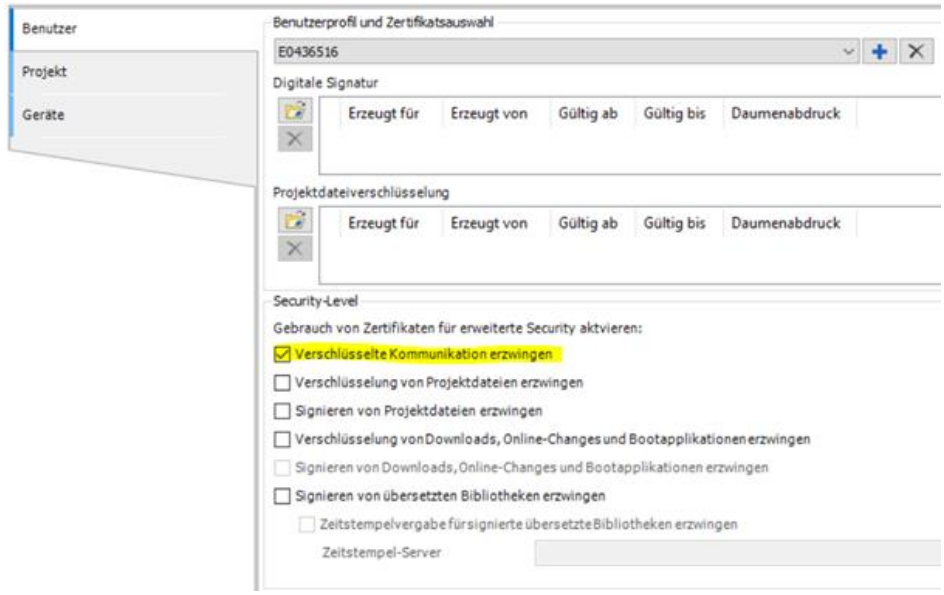
Once the certificate has been successfully generated, this will be visible in the "Log" tab.

Gewichtung	Zeitstempel	Beschreibung	Komponente
ℹ	27.01.2021 03:49:48	[2] SelfSigned cert created, subject='commonName=WM01 unstructuredName=Vendor: Eaton Automation unstructuredNa...	CmpOpenSSL
ℹ	27.01.2021 03:49:36	SysTaskCreate(617d5c6/102225350): X509AsyncTask (TaskFrame) Prio 255 Interval 0	SysTask
ℹ	27.01.2021 03:46:50	SysTaskCreate(6fcb1da/117223898): X509AsyncTask (TaskFrame) Prio 255 Interval 0	SysTask

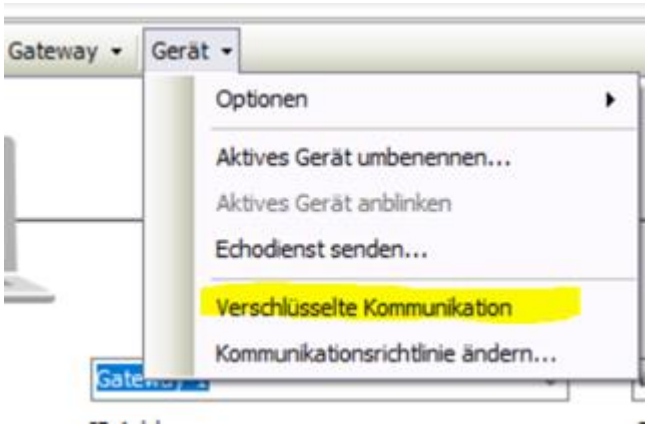
By re-executing the command "cert-getapplist" a check is also possible in the PLC Shell:

```
Nr.   ComponentName          CommonName                CertAvailable DateNotBefore
-----------------------------------------------------------------------------------
-----
0     CmpOPCUAServer         OPCUAServer@WM01          FALSE         --
1     CmpApp                 WM01                FALSE         --
2     CmpSecureChannel       WM01                TRUE          2021-1-27T2:49:36.0
3     CmpWebServer           WM01                TRUE          2021-1-27T2:49:36.0
```

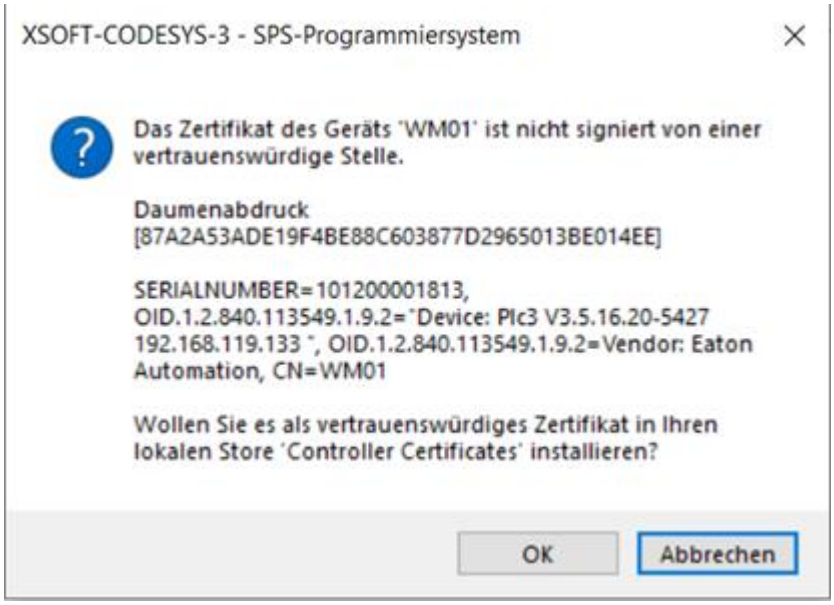
Now the option "Force encrypted communication" can be activated in the security screen.



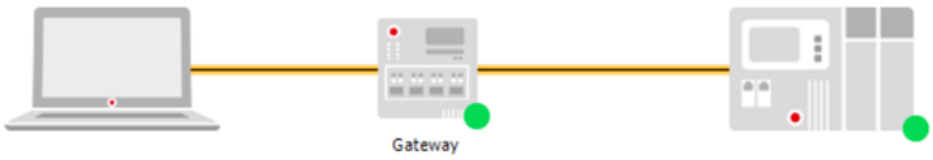
This option applies to all controllers. If communication is only to be encrypted to individual controllers, it is possible to activate this in the "Communication" tab:



When logging in to the controller for the first time, a message appears stating that the controller's certificate is not signed by a trusted authority. In addition, the user is offered to install the certificate in the local store "Controller Certificates". If this dialog is confirmed, then the controller is logged in and future connections to the controller are encrypted with this certificate.



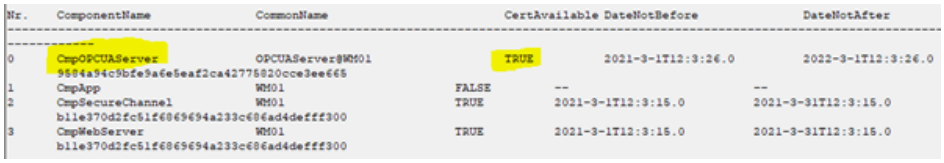
An encrypted connection is displayed accordingly in color in the Communication tab:



5 Encrypted OPC-UA Server Access

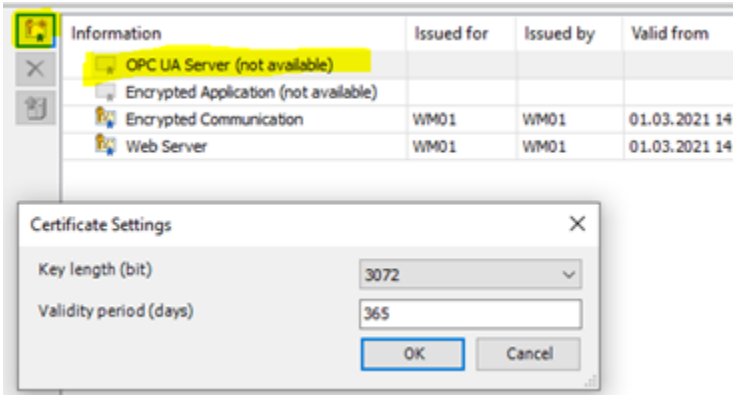
For the encrypted communication to the OPCUA-Server a X.509 certificate is required. With cert-getapplist it is first checked whether a certificate has already been created for the OPUA server.

With the command cert-genselfsigned a certificate for this component can be created again.

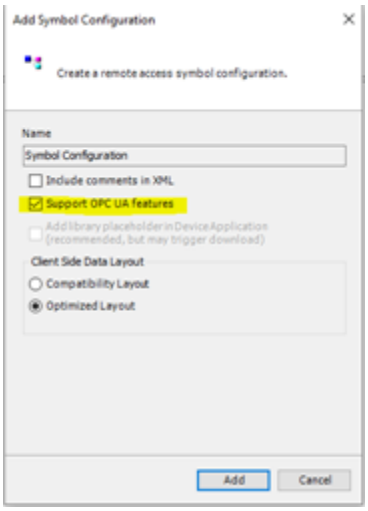


Nr.	ComponentName	CommonName	CertAvailable	DateNotBefore	DateNotAfter
0	CmpOPCUAServer 9584a94c9bfe9a6e5eaf2ca42775820cce3ee665	OPCUAServer@WM01	TRUE	2021-3-1T12:3:26.0	2022-3-1T12:3:26.0
1	CmpApp	WM01	FALSE	--	--
2	CmpSecureChannel b11e370d2fc51f6869694a233c686ad4defff300	WM01	TRUE	2021-3-1T12:3:15.0	2021-3-31T12:3:15.0
3	CmpWebServer b11e370d2fc51f6869694a233c686ad4defff300	WM01	TRUE	2021-3-1T12:3:15.0	2021-3-31T12:3:15.0

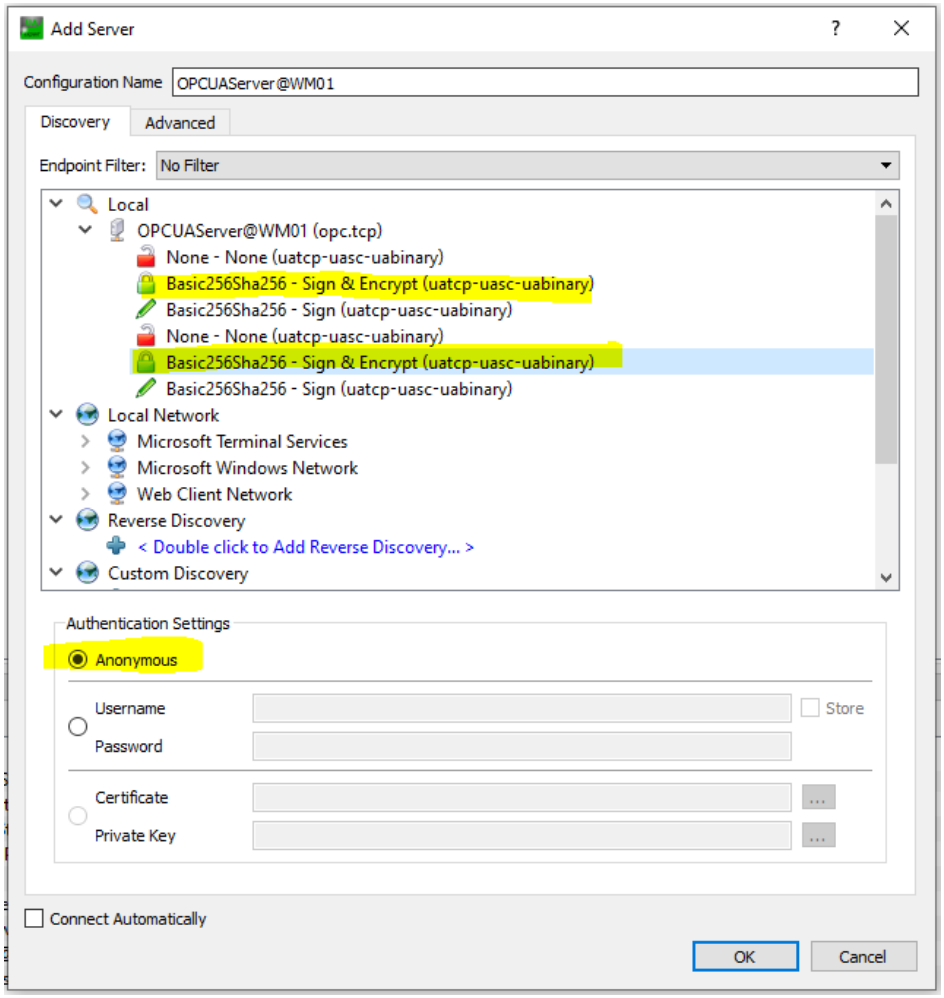
Alternatively, the certificate can also be created in the Security Screen:



After the creation the PLC runtime must be restarted. On the CODESYS side, a symbol configuration must now be added with the "Support OPC UA functionalities" option activated.

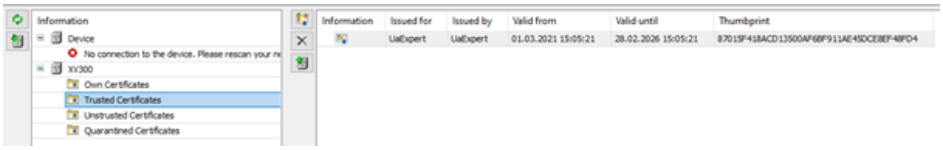


Now the OPCUA client can be configured. In this example the client UaExpert from Unified Automation is used. First, a server is added. The UaExpert client lists the possible connections to the servers. Here one selects the encrypted Basic256 (Sign&Encrypt" connection. Access can be made either via the device name or the device IP address.

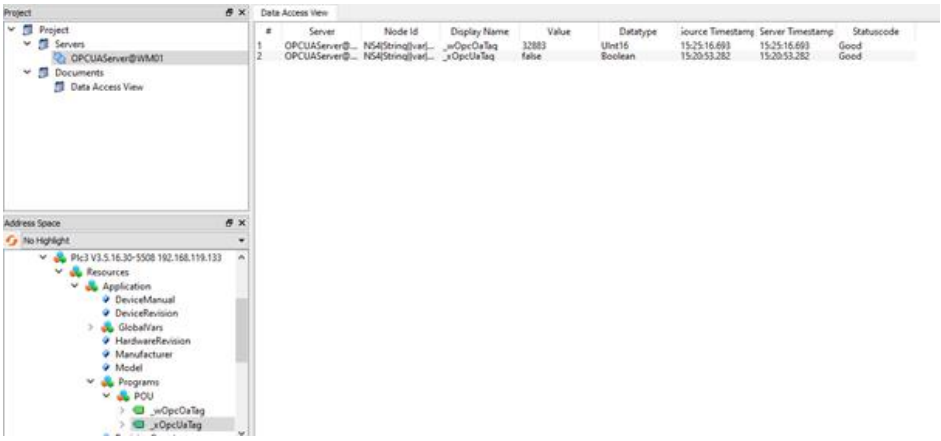


After the first connection, the following message is displayed, indicating that the existing certificate is a self-signed certificate and cannot be verified. By confirming the message the certificate is recognized as valid and the client certificate is transferred to the controller.

Then this client certificate must be moved from the quarantined directory to the trusted directory:



If you now connect to the OPCUA client again, an encrypted connection is established and the configured variables from the symbol configuration can be observed and manipulated.



By default the PLC accepts all incoming connections to the OPCUA server. To prevent this and to force an encrypted connection, the CommunicationMode in the PLC3.cfg file must be adjusted:

```
[CmpOPCUAServer]
SECURITY.CommunicationMode=SIGNED_AND_ENCRYPTED
```

6 Encrypted Webserver-Access

For encrypted communication to the web server (https), an X.509 certificate is required for TLS web server authentication. With cert-getapplist it is first checked whether a certificate has already been created for the web server.

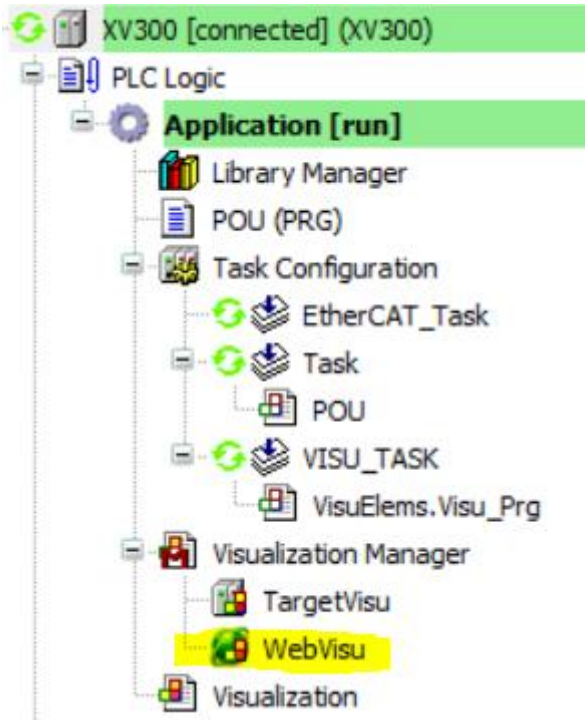
```
cert-getapplist

Nr.   ComponentName          CommonName                  CertAvailable DateNotBefore
      DateNotAfter           Thumbprint
-------------------------------------------------------------------------------------------
-------------------------------------------------------------
0     CmpOPCUAServer         OPCUAServer@WM01            FALSE         --
                        --
1     CmpApp                 WM01                  FALSE         --
                  --
2     CmpSecureChannel       WM01                  FALSE         --
                  --
3     CmpWebServer           WM01                  FALSE         --
                  --
```

If this is not the case, then a certificate for this component can be created with the cert-genselfsigned command. By default, a certificate for the web server is already present on the devices. However, the validity of this certificate is limited to a short period of time. When a new certificate is created, the validity period can be configured accordingly.

```
----
cert-genselfsigned 3 365

Generate selfsigned certificate with given index (3). Check logger to see when finished.
```

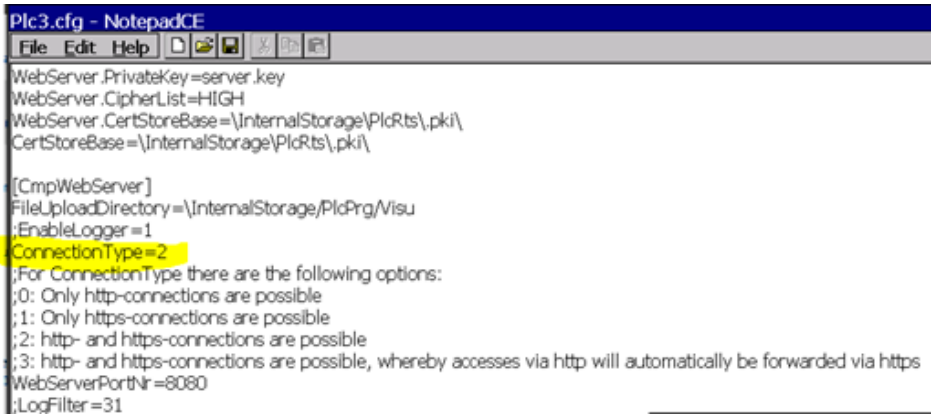
```
Nr.    ComponentName              CommonName                    CertAvailable DateNotBefore
           DateNotAfter                Thumbprint
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------
0      CmpOPCUAServer             OPCUAServer@WM01              FALSE         --
           --                          --
1      CmpApp                     WM01                    FALSE         --
       --                         --
2      CmpSecureChannel           WM01                    TRUE          2018-1-24T23:0:19.0
       2018-2-23T23:0:19.0        2a453b12f8836769b4ee40ce10a299ca499dbdc4
3      CmpWebServer               WM01                    TRUE          2018-1-24T23:3:54.0
       2019-1-24T23:3:54.0        0c31a21fe022b7cde5351cfdcef1f592ec6767f8
```

In the CODESYS project, the creation and transfer of a web visualization is required:

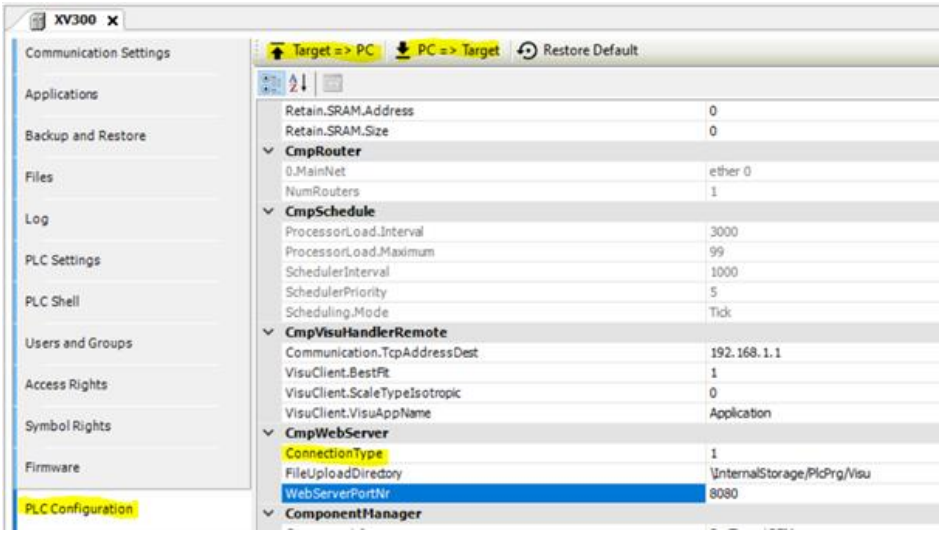


Afterwards the web visualization can be accessed via the URL https://<IP address>/Name_of_HTML_file.html.

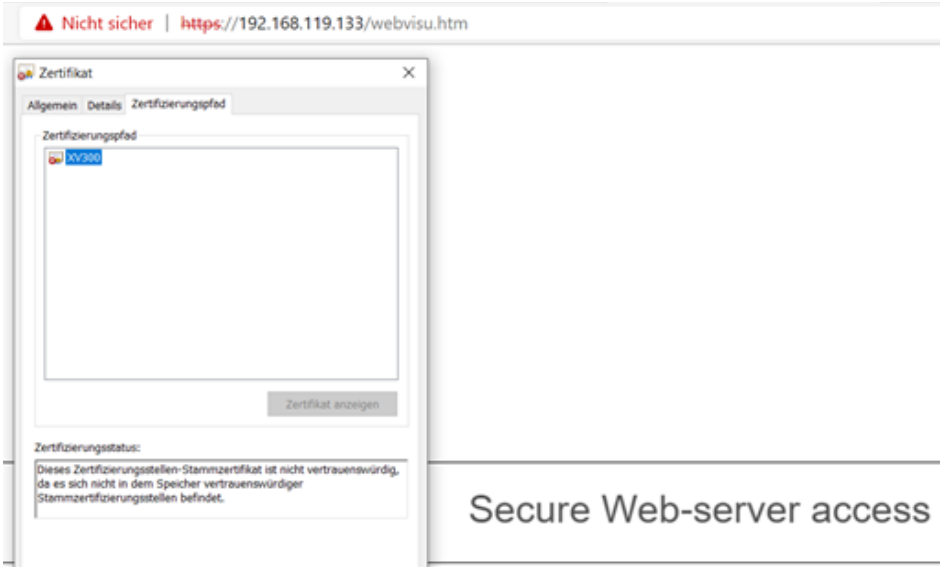
So that also compellingly a https connection can be used, the appropriate entry must take place in the PLC3.cfg. By default http and https connections are allowed. If only https connections are to be allowed, the ConnectionType must be set to 1 or 3.



The change can be made in the CODESYS software for an XV100/XV300. After changing and transferring, the device must be restarted.



As long as you use self-signed certificates and do not have them signed by an official authority or create your own root certificate, the browser displays the message "Not secure".



Eaton is an intelligent power management company dedicated to improving the quality of life and protecting the environment for people everywhere. We are guided by our commitment to do business right, to operate sustainably and to help our customers manage power – today and well into the future.
By capitalizing on the global growth trends of electrification and digitalization, we're accelerating the planet's transition to renewable energy, helping to solve the world's most urgent power management challenges, and doing what's best for our stakeholders and all of society.

Founded in 1911, Eaton has been listed on the NYSE for nearly a century.
We reported revenues of $19.6 billion in 2021 and serve customers in more than 170 countries.

For more information, visit Eaton.com. Follow us on Twitter and LinkedIn.

Eaton Industries GmbH
Hein-Moeller-Str. 7- 11
D-53115 Bonn

07/2025 AP050017EN

Powering Business Worldwide