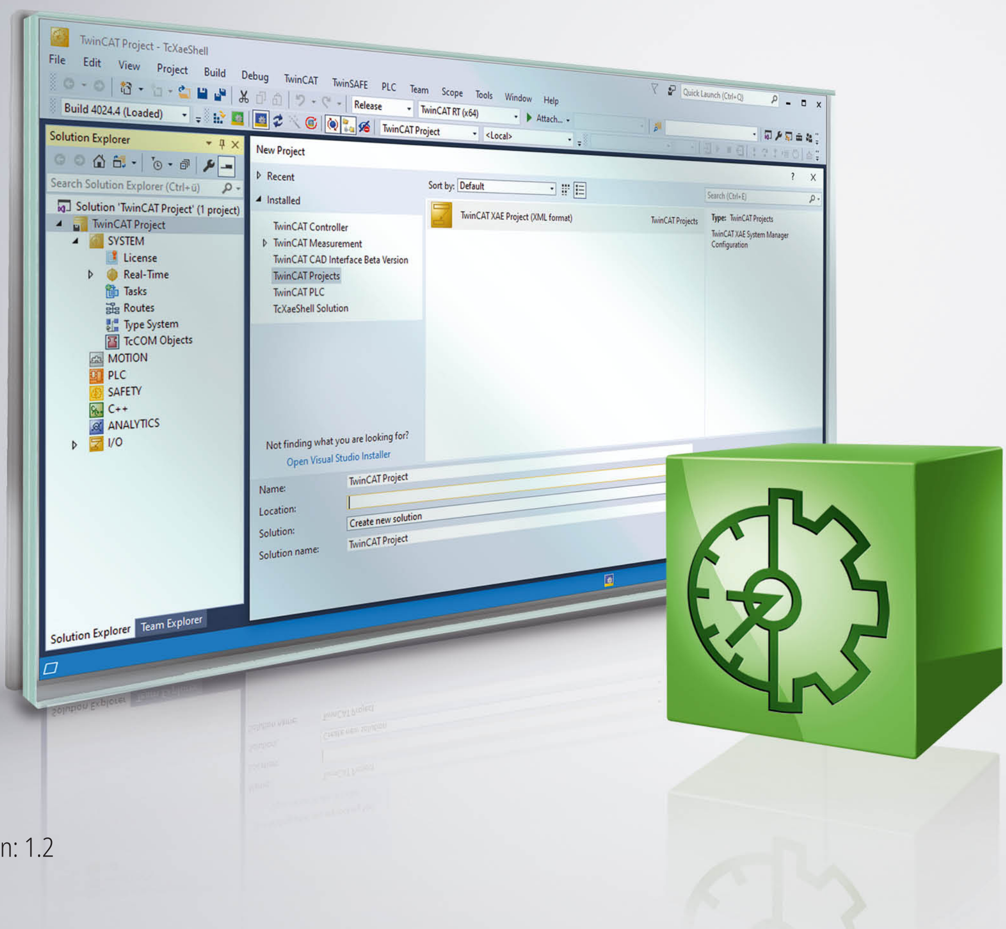
BECKHOFF New Automation Technology

Manual | EN

TE1000

TwinCAT 3 Secure ADS



11/9/2020 | Version: 1.2

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

EtherCAT

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

Tip or pointer

This symbol indicates information that contributes to better understanding.

2 General description

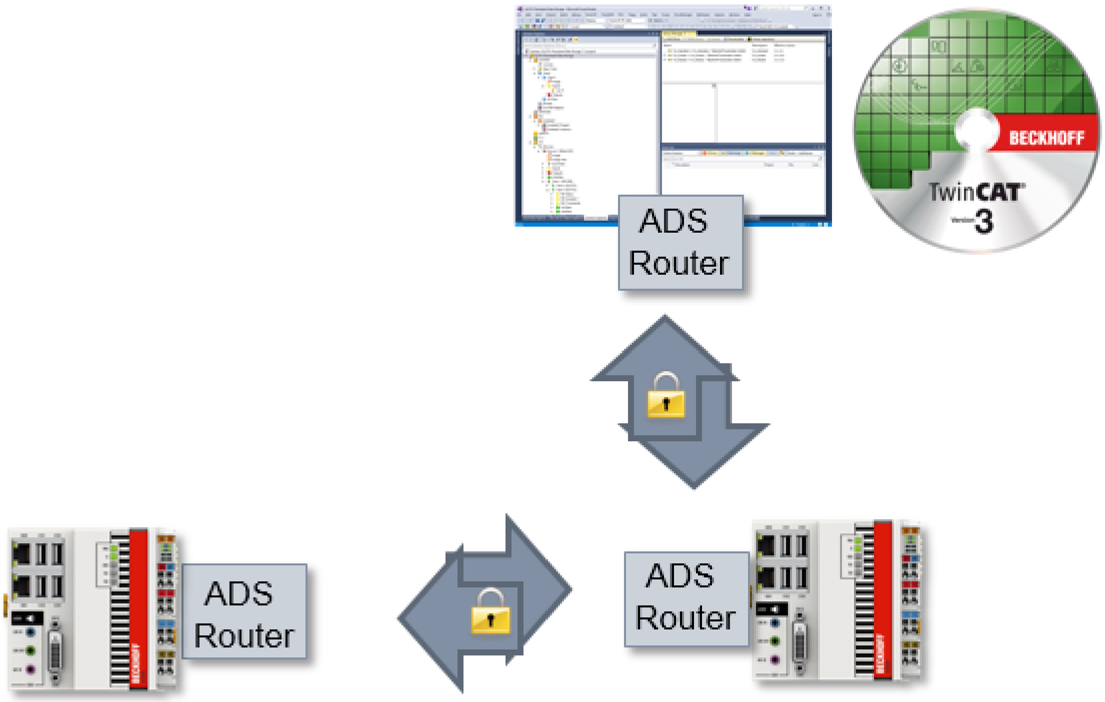
From TwinCAT 3.1 Build 4024.0

The functionality described here is available from TwinCAT 3.1. 4024.0.

Secure ADS is an additional transport channel from the point of view of the ADS protocol. Precisely the same ADS commands are transmitted via a secure connection as via other communication protocols.

To this end a connection encrypted by means of TLSv1.2 is established from one TwinCAT router to another.

Due to the implementation inside the TwinCAT router, existing applications do not need to be modified. They can be made to use the encrypted connection by simply parameterizing the used route.



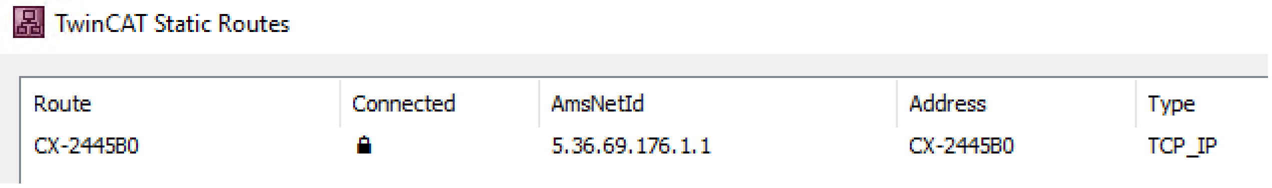
This documentation illustrates the different options of Secure ADS, in particular with regard to the provision of the keys.

Detection of a Secure ADS route

TwinCAT displays a Secure ADS route with a lock icon.

It is displayed at the appropriate points:

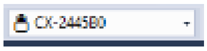
- Route overview of a system



TwinCAT Static Routes

Route	Connected	AmsNetId	Address	Type
CX-2445B0	🔒	5.36.69.176.1.1	CX-2445B0	TCP_IP

- When selecting the target system in the XAE engineering environment:

- 

CX-2445B0

3 Limitations

From TwinCAT 3.1 Build 4024.0

The functionality described here is available from TwinCAT 3.1. 4024.0.

- Secure ADS is available only between ADS routers.
- Like all other ADS connections, Secure SDS connections represent full access for the connected systems as is also described in the Security Advisory 2017-01.
  This access is configurable per system through unidirectional [▶ 11] ADS routes.

4 Requirements

From TwinCAT 3.1 Build 4024.0

The functionality described here is available from TwinCAT 3.1. 4024.0.

- Secure ADS is a component of TC1000 and can be used without license costs.
- The devices used require network communication. Incoming Secure ADS is communicated via the TCP port 8016.
- Appropriate certificates may need to be generated and signed for TLS encryption.

5 Technical introduction

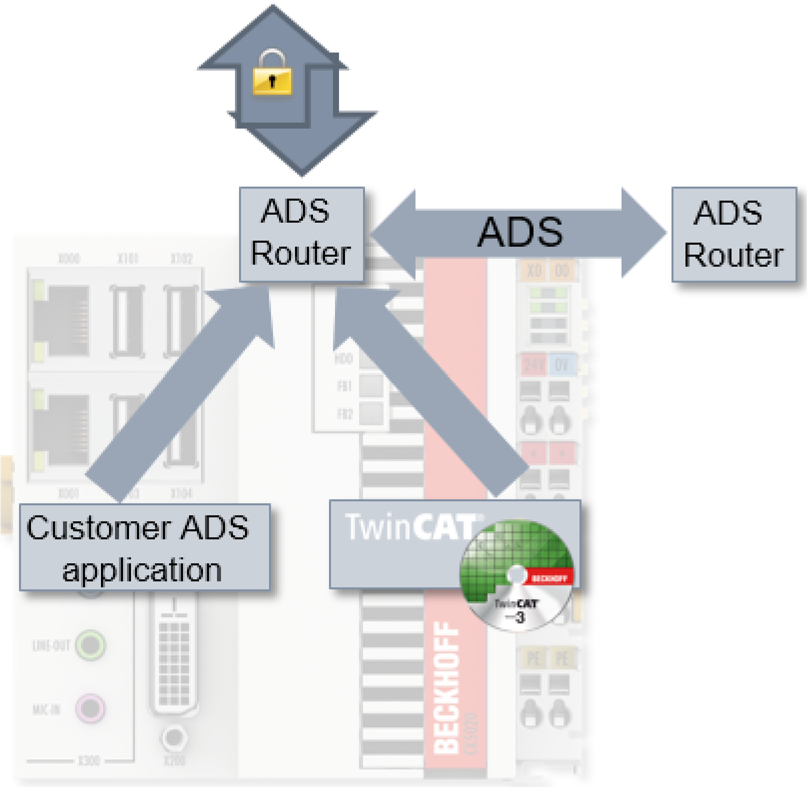
In this section the basic mode of operation is described, irrespective of the specific configuration.

Secure ADS introduced an additional communication channel for the familiar ADS protocol. This can be used by programs without them having to be adapted for the new communication channel.

From the point of view of security, therefore, it is a transport encryption, but not an end-to-end encryption between the components, because all applications running locally on a device can use this encrypted connection together – exactly as with ADS routes also.

Local realization

Secure ADS is part of the ADS router and is also configured here. The ADS router establishes an encrypted connection to another TwinCAT router and makes it available to the applications. Care must therefore be taken that the ADS devices do not themselves communicate applications in encrypted form, but that this takes place between the routers.



Transparent retrofitting

The realization of Secure ADS inside the TwinCAT router makes the retrofitting of applications possible. None of the ADS applications (client and server) – this also includes applications written by the customer – need to be recompiled.

The ADS applications use ADS routes to identify the communication partner. This ADS route is independent of the transport channel and is described in the TwinCAT router.

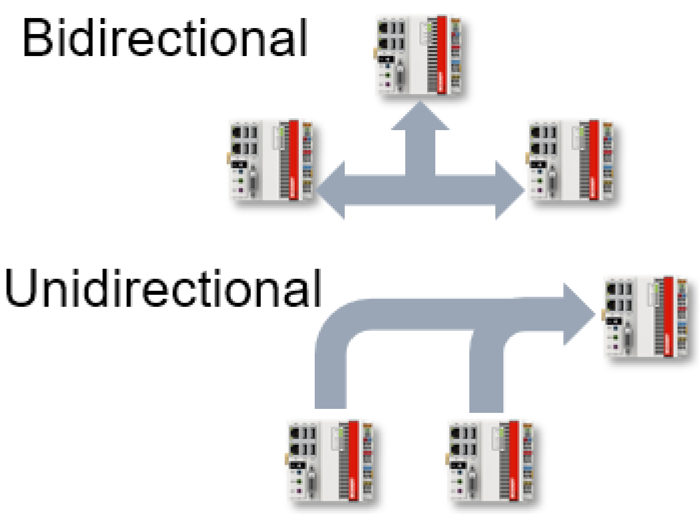
If the used route is switched to a Secure ADS connection, the ADS traffic is transported in encrypted form.

5.1 Directed ADS communication

One of the properties of ADS routes is that they can be directed. This property was supplemented within the scope of Secure ADS, but is generally available for routes.

Once they have been opened at network level, ADS routes are used for communication on both sides by the respective ADS applications. This behavior is very efficient, but may be undesirable. For example, an engineering computer (XAE) is supposed to have access to a runtime (XAR) system in the normal case, but it is not necessary for an XAR system to access the XAE system via ADS.
Therefore, this direction can be limited in that a corresponding system (the XAE in the example) does not accept any ADS request commands via the route.

The chapter Configuration [▶ 13] describes the procedure for limiting the properties.



5.2 Server

A normal ADS route is established by both devices as soon as it is required.
Once a route has been established it is used in both directions.
A server configuration is offered as an extension for Secure ADS. Such a configuration represents the basis for setting up specific routes.

```
<TcConfig>
  <RemoteConnections>
    <Server>
      …
    </Server>
  </RemoteConnections>
</TcConfig>
```

For PSK [▶ 15] and certificates provided by the customer [▶ 16] this is used to store the initial configuration on one side.
When setting up the specific route, the server entries are then checked to see if rights exist. If this is the case, a normal route will be set up.

Also see about this

- Configuration [▶ 13]

5.3 Key exchange

Secure ADS offers three ways of providing the keys required for encryption; these are described here with their advantages and disadvantages.

What they all have in common is that the respective device has to be isolated with respect to access to the secrets (Pre-Shared Keys, certificates). If these secrets are compromised, the system has to be set up again in order to restore the integrity of the complete system.

Self-Signed Certificates (SSC)

When starting for the first time (e.g. after the installation), TwinCAT generates a self-signed certificate.

The use of such certificates has the advantage that they are generated and are available locally. In order to establish a basis for trust, however, a check of the certificates must be performed among all communication devices.

These certificates are thus suitable for the initial commissioning or also for static machines that can make do without dynamics in the system structure or the entity authorized to access.

From TwinCAT 4024.0 these certificates will be provided as standard when used. The chapter Configuration [▶ 14] describes how they are used to establish an ADS route.

Validity periods of the certificates

The certificates generated have a fixed validity period from 1/1/2000 to 1/1/2061. From the point of view of security this is too long, meaning that organizational measures have to be taken to meet the security demands. With this excessively long validity period, Beckhoff ensures that communication does not fail, even if, for example, incorrect times are set in the local system.

If this behavior is not desired, you can generate and use your own certificates (see Certificates provided by the customer).

Pre-Shared Keys (PSK)

Pre-Shared Keys can be stored in a TwinCAT system. These are used to authorize the incoming ADS routes when establishing the connection.

As the Pre-Shared Keys have to be configured they are particularly suitable for granting access, for example, to maintenance staff. The Pre-Shared Keys can be bound to a specific person.

Pre-Shared Keys do not have a validity period like that foreseen for certificates. They are also stored directly in files so that they are not stored as a hash value (as is usually the case with passwords). They are therefore not protected against direct viewing.

The chapter Configuration [▶ 15] describes how Pre-Shared Keys are used on both sides of the communication.

Certificates provided by the customer (CA with certificates)

Secure ADS also provides customers with the option of generating and managing their own certificates.

As a result, dynamic constellations in particular are easily mappable, because there can be a common Certificate Authority (CA). All devices that trust this CA can communicate in encrypted form with one another with no further configuration, even if they have never encountered one another before.

The chapter Configuration [▶ 16] describes how these certificates can be integrated into TwinCAT.

NOTE

Expiry of the certificates

Certificates have an expiry date. Organizational measures must be taken to replace certificates before their expiry.

6 Configuration

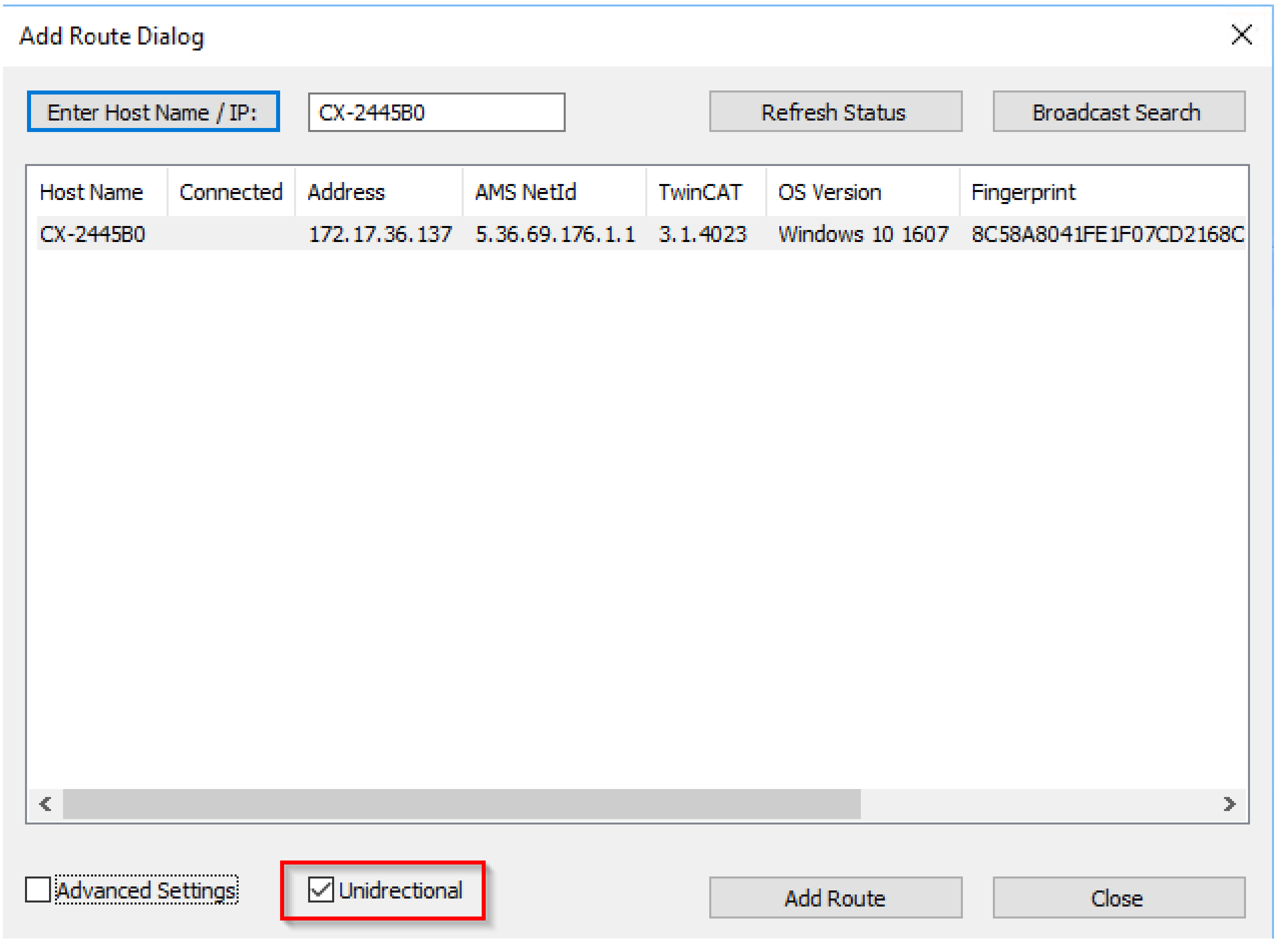
Secure ADS offers three ways of providing the keys required for the encryption. At this point the configurations will be described separately from one another.

While the Server vs. Route configuration is described within the three ways, directed ADFS connections [▶ 13] are illustrated independently.

6.1 Directed ADS communication

The configuration of a directed ADS communication takes place using the checkbox **Unidirectional** when creating the route.

If this checkbox is set, TwinCAT will not accept any ADS command calls from the opposite target system via the associated route. TwinCAT itself sends ADS command calls (requests) and receives responses.



Enter Host Name / IP: CX-2445B0

Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Fingerprint
CX-2445B0		172.17.36.137	5.36.69.176.1.1	3.1.4023	Windows 10 1607	8C58A8041FE1F07CD2168C

☐ Advanced Settings ☑ Unidirectional

In the XML configuration this setting is made via the attribute `Unidirectional="true"`:

```
<RemoteConnections>
<Route Unidirectional="true">
<Name>CX-123456</Name>
<Address>CX-123456</Address>
<NetId>5.36.69.176.1.1</NetId>
<Type>TCP_IP</Type>
<Flags>128</Flags>
<Tls>
…
</Tls>
</Route>
</RemoteConnections>
```

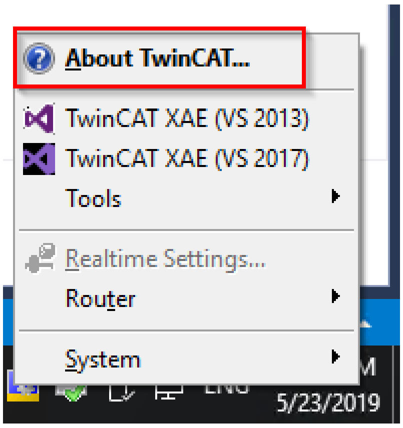
6.2 Self-Signed Certificates (SSC)

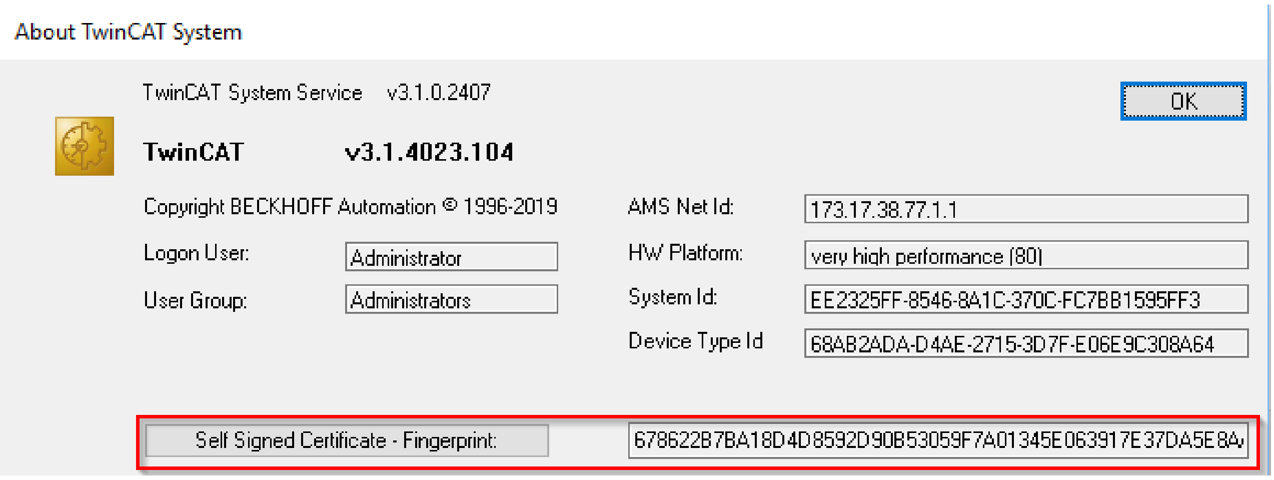
When setting up the connection, Self-Signed Certificates require the checking of the communication device, as no trust basis automatically exists.

This check is made possible in TwinCAT by the fingerprint of the opposite system.

Displaying the SSC fingerprint on a system

The fingerprint of your own system is displayed in the **About TwinCAT** dialog:





About TwinCAT System

TwinCAT System Service v3.1.0.2407

TwinCAT v3.1.4023.104

Copyright BECKHOFF Automation © 1996-2019

Logon User: Administrator

User Group: Administrators

AMS Net Id: 173.17.38.77.1.1

HW Platform: very high performance (80)

System Id: EE2325FF-8546-8A1C-370C-FC7BB1595FF3

Device Type Id 68AB2ADA-D4AE-2715-3D7F-E06E9C308A64

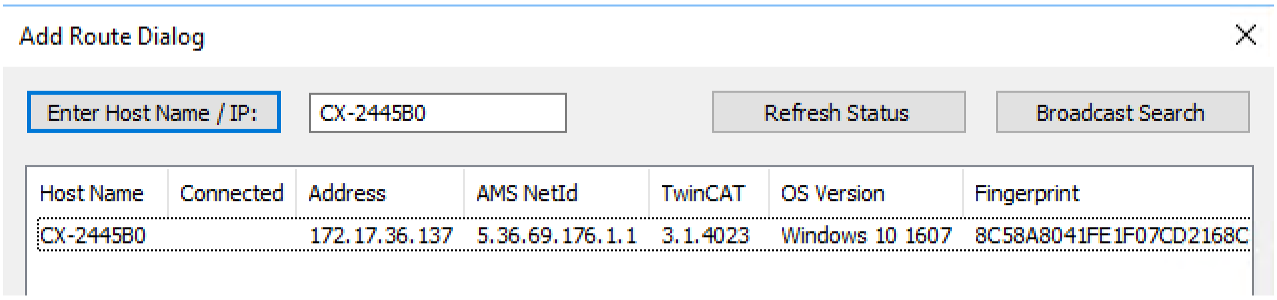
Self Signed Certificate - Fingerprint: 678622B7BA18D4D8592D90B53059F7A01345E063917E37DA5E8A

The button **Self Signed Certificate - Fingerprint:** copies the fingerprint listed on the right to the clipboard.

This dialog does not exist for CE systems. The fingerprint can be displayed here in the file `\Hard Disk \TwinCAT\3.1\Target\TcSelfSigned.xml`.

Establishment of the connection

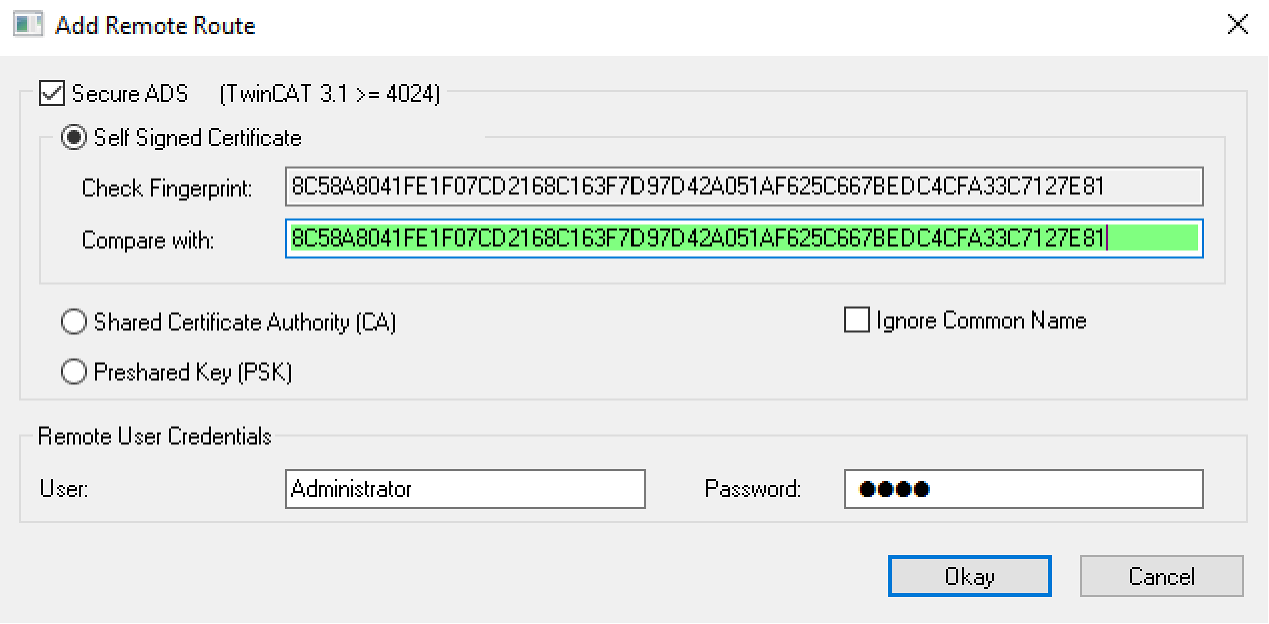
The fingerprint is displayed purely for information and cryptographically unsecured following the discovery:



Enter Host Name / IP: CX-2445B0

Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Fingerprint
CX-2445B0		172.17.36.137	5.36.69.176.1.1	3.1.4023	Windows 10 1607	8C58A8041FE1F07CD2168C

The final checking of the fingerprint takes place when setting up the route:



Add Remote Route

☑ Secure ADS (TwinCAT 3.1 >= 4024)

● Self Signed Certificate

Check Fingerprint: 8C58A8041FE1F07CD2168C163F7D97D42A051AF625C667BEDC4CFA33C7127E81

Compare with: 8C58A8041FE1F07CD2168C163F7D97D42A051AF625C667BEDC4CFA33C7127E81

○ Shared Certificate Authority (CA) □ Ignore Common Name

○ Preshared Key (PSK)

Remote User Credentials

User: Administrator Password: ●●●●

The **Compare with** field can be used, for example, with copy & paste for checking: If the same fingerprint is entered there the field appears green, otherwise it is red.

Thus, an RDP connection, for example, can be used to copy the fingerprint of a system to the clipboard via the **Self Signed Certificate - Fingerprint** button and to enter it here.

So that the target system will accept the route establishment, a system login with corresponding administrator rights that is valid there is used.
These login data are already transmitted in encrypted form.

With CE systems the host name is always entered with TwinCAT 3.1 4024.5, even if **IP address** was selected when creating the route. Therefore, if a network without a functioning host name lookup is to be used, the host name must be changed manually by the IP address in the file `\Hard Disk\TwinCAT\3.1\Target\StaticRoutes.xml`.

6.3 Pre-Shared Keys (PSK)

Pre-Shared Keys are set up on one side as a server and on the other side for authentication and authorization.

Setting up Pre-Shared Keys as a server

Pre-Shared Keys are normally used with server connections.
The configuration takes place via an entry in the route configuration.

To do this, the following entries can be made in the file *C:\TwinCAT\3.x\Target\StaticRoutes.xml* :

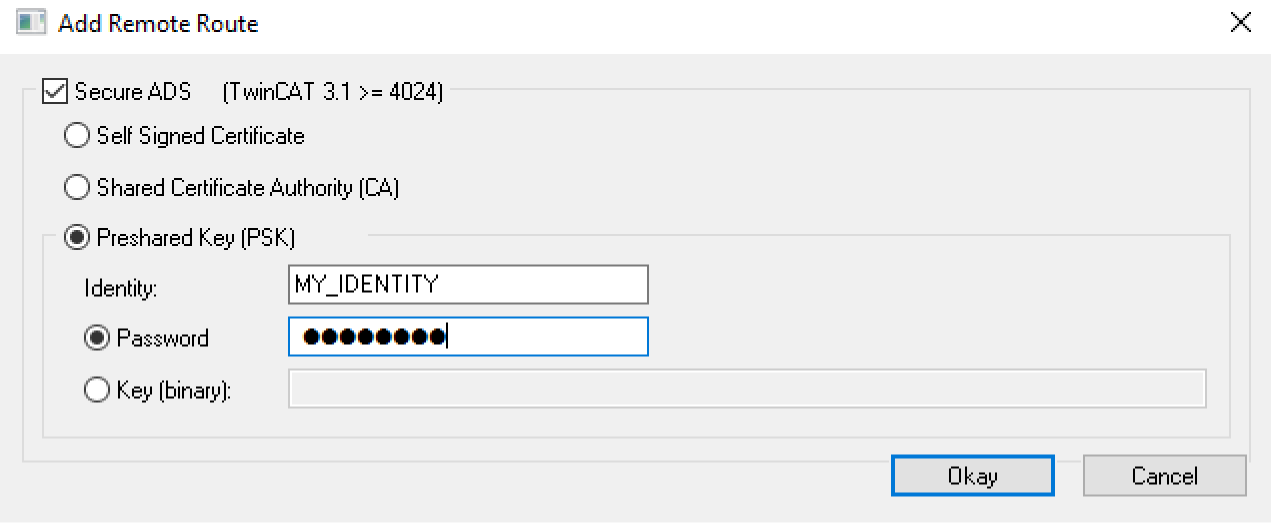
```
<?xml version="1.0"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<RemoteConnections>
<Server>
<Tls>
<Psk>
<Identity>MY_IDENTITY</Identity>
<Pwd>MySecret</Pwd>
</Psk>
<Psk>
<Identity>MY_IDENTITY2</Identity>
<Pwd>MyOtherSecret</Pwd>
</Psk>
</Tls>
```

```
</Server>
</RemoteConnections>
</TcConfig>
```

Saved changes are accepted when the TwinCAT router is initialized, which takes place, for example, during the transition RUN->CONFIG or CONFIG->CONFIG.

Use of a Pre-Shared Key server

When adding a route, the entry **Pre-Shared Key (PSK)** is selected and the corresponding credentials are entered.



Add Remote Route

☑ Secure ADS (TwinCAT 3.1 >= 4024)

○ Self Signed Certificate

○ Shared Certificate Authority (CA)

● Preshared Key (PSK)

Identity: MY_IDENTITY

● Password ●●●●●●●●

○ Key (binary):

If this is successful, a specific route is stored in the target system and is used for the future establishment of connections.

6.4 Certificates provided by the customer (CA with certificates)

The configuration of certificates provided by the customer takes place via an entry in the route configuration.

To do this, the following entries can be made in the file *C:\TwinCAT\3.x\Target\StaticRoutes.xml* :

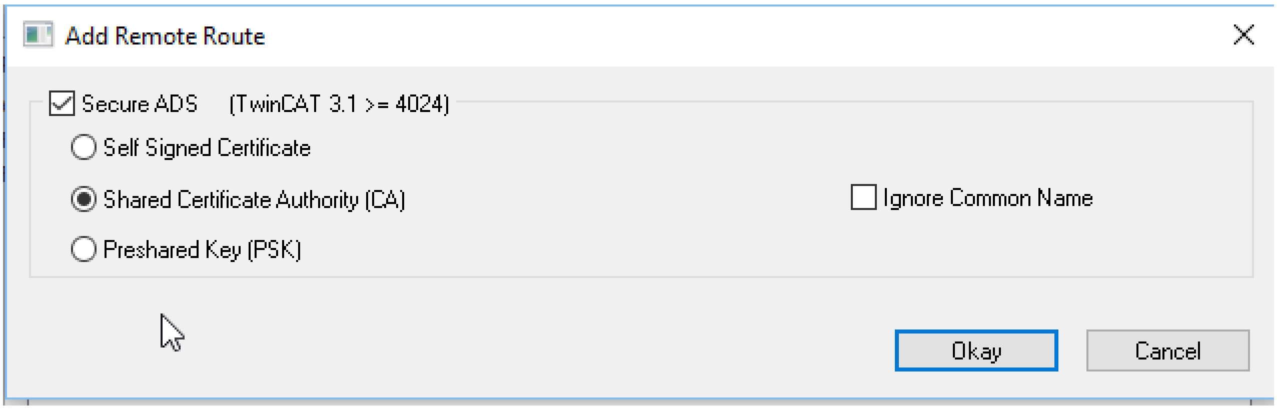
```
<?xml version="1.0"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<RemoteConnections>
<Server>
<Tls IgnoreCn="true"> <!--see below-->
  <Ca>C:\TwinCAT\3.1\Target\CACerts\rootCA.pem</Ca>
  <Cert>C:\TwinCAT\3.1\Target\CACerts\ipc.crt</Cert>
                <Key>C:\TwinCAT\3.1\Target\CACerts\ipc.key</Key>
              </Tls>
          </Server>
</RemoteConnections>
</TcConfig>
```

Saved changes are accepted when the TwinCAT router is initialized, which takes place, for example, during the transition RUN->CONFIG or CONFIG->CONFIG.

The certificates are X.509 certificates, which can be generated, for example, with OpenSSL. If the key (XML-Element `<Key>`) is to be protected by a password, this can be specified via the XML element `<KeyPwd>`. The .der and .pem formats are supported.

The "CommonName" of the certificate must correspond to the name used when establishing the connection (XML-Element `<Name>`). This behavior can be deactivated with the option `IgnoreCn=" true"`.

If both sides have suitable certificates of a common CA, the route can be created without further information using this dialog:



Add Remote Route

☑ Secure ADS (TwinCAT 3.1 >= 4024)

○ Self Signed Certificate

● Shared Certificate Authority (CA) □ Ignore Common Name

○ Preshared Key (PSK)

As described under Server [▶ 11], a specific route is created on both sides as a result of this.

6.5 Deactivating ADS

- The unencrypted ADS is transmitted via the TCP port 48898 (0xBF02)
- The discovery ("Broadcast Search") is transmitted via the UDP Port 48899 (0xBF03)

Both ports can be blocked in the firewall.

The target system can be configured with respect to the ports to be used.

The following keys are available below KEY_LOCAL_MACHINE\SOFTWARE\[WOW6432Node\]Beckhoff\TwinCAT3\System:

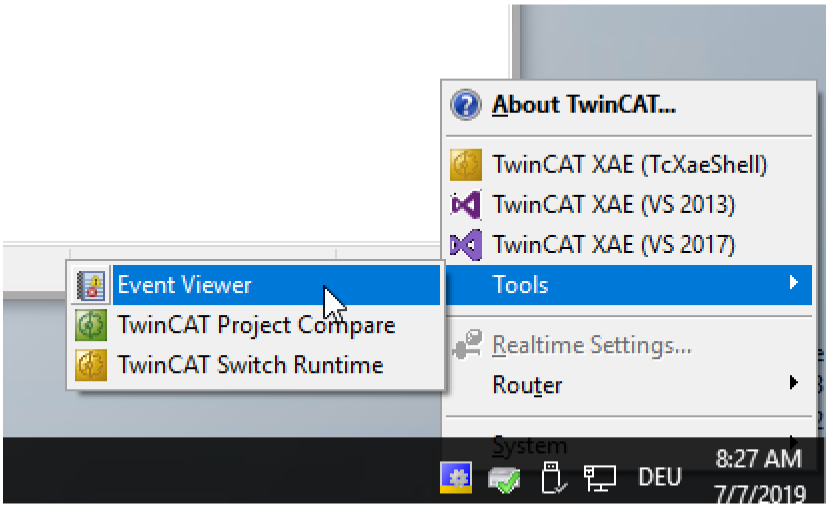
ADS Ports		
DisableAdsTcpListening	REG_DWORD	1 = prevents the opening of the TCP port 0xBF02 for unencrypted ADS.
DisableAdsTlsListening	REG_DWORD	1 = prevents the opening of the TCP port 8016 for Secure ADS
DisableAdsDiscovery	REG_DWORD	1 = prevents the opening of the UDP port 0xBF03 for the ADS discovery ("Broadcast Search")

The attribute `SecureOnly="True"` can additionally be used via the StaticRoutes.xml file. The ADS port 0xBF02 is thereby kept open, but no further ADS communication is allowed via the port.

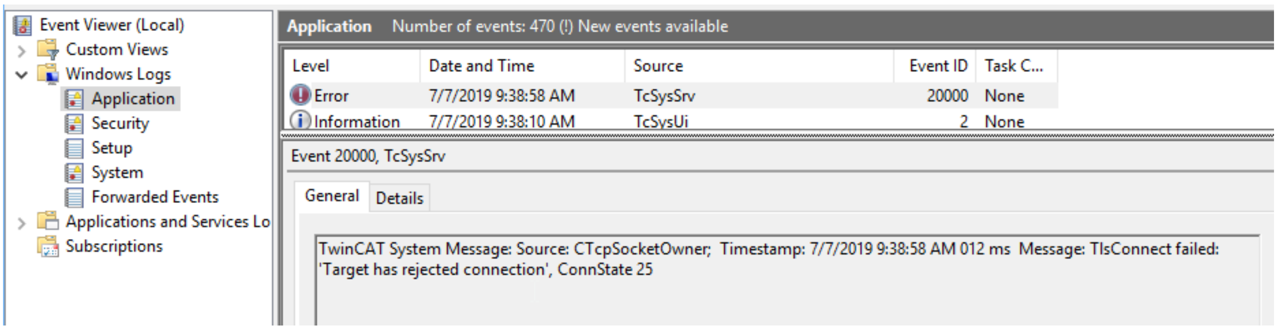
```
<RemoteConnections SecureOnly="True">
```

6.6 Logging

Secure ADS writes information about failed connection establishments in the Windows Event Log, which is available via the TwinCAT System Tray icon.



The messages can be found under the category **Windows Logs > Application**:



Application Number of events: 470 (!) New events available

Level	Date and Time	Source	Event ID	Task C...
Error	7/7/2019 9:38:58 AM	TcSysSrv	20000	None
Information	7/7/2019 9:38:10 AM	TcSysUi	2	None

Event 20000, TcSysSrv

TwinCAT System Message: Source: CTcpSocketOwner; Timestamp: 7/7/2019 9:38:58 AM 012 ms Message: TlsConnect failed: 'Target has rejected connection', ConnState 25

7 Sample

7.1 Certificates provided by the customer (CA with certificates)

At this point certificates are generated by means of Open SSL and can be used for the Secure ADS connection.

These instructions do not represent comprehensive advice on the creation and handling of certificates. In particular the validity periods must be observed, which necessitates organizational measures in order to ensure replacement before the expiry of the validities (in this case: 3600 days for CA and 360 days for the respective certificates).

In this example a Certificate Authority (CA) is generated that signs a certificate for both sides (called IPC and CX here) of the communication.

The meaning of the call parameters can be viewed in detail via "`openssl help`".

✓ OpenSSL is installed and is available from the command line.

1. Generate a key for the Certificate Authority that will be trusted later.
   ```
   openssl genrsa -out rootCA.key 2048
   ```
2. Generate the certificate with a validity period of 3600 days. Owner information is added via the parameter "-subj".
   ```
   openssl req -x509 -new -nodes -key rootCA.key -sha256 -subj "/C=DE/ST=NRW/
   L=Verl/O=Bk/OU=TCPM/CN=RootCA" -days 3600 -out rootCA.pem
   ```
3. Generate a key for the IPC
   ```
   openssl genrsa -out ipc.key 2048
   ```
4. Generate a Certificate Signing Request (CSR) for this key:
   Please note: The address specified as CN (IP address in this case) must be used as the name when establishing the connection. Alternatively, the route must be parameterized with IgnoreCN.
   ```
   openssl req -out ipc.csr -key ipc.key -subj "/C=DE/ST=NRW/L=Verl/O=Bk/
   OU=TCPM/CN=192.168.2.1" –new
   ```
5. Sign the CSR of the IPC with the CA with a validity of 360 days
   ```
   openssl x509 -req -in ipc.csr -CA rootCA.pem -CAkey rootCA.key -
   CAcreateserial -out ipc.crt -days 360 -sha256
   ```
   ⇨ The route can now be set up on the IPC using these files: rootCA.pem, ipc.key and ipc.pem
6. Generate a key for the CX
   ```
   openssl genrsa -out cx.key 2048
   ```
7. Generate a Certificate Signing Request (CSR) for this key:
   Please note: The address specified as CN (IP address in this case) must be used as the name when establishing the connection. Alternatively, the route must be parameterized with IgnoreCN.
   ```
   openssl req -out cx.csr -key cx.key -subj "/C=DE/ST=NRW/L=Verl/O=Bk/OU=TCPM/
   CN=192.168.2.2" –new
   ```
8. Sign the CSR of the IPC with the CA with a validity of 360 days
   ```
   openssl x509 -req -in cx.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial
   -out cx.crt -days 360 -sha256
   ```
   ⇨ The route can now be set up on the CX using these files: rootCA.pem, cx.key and cx.pem

⇨ The route can be used.

More Information:
www.beckhoff.com/te1000/

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com