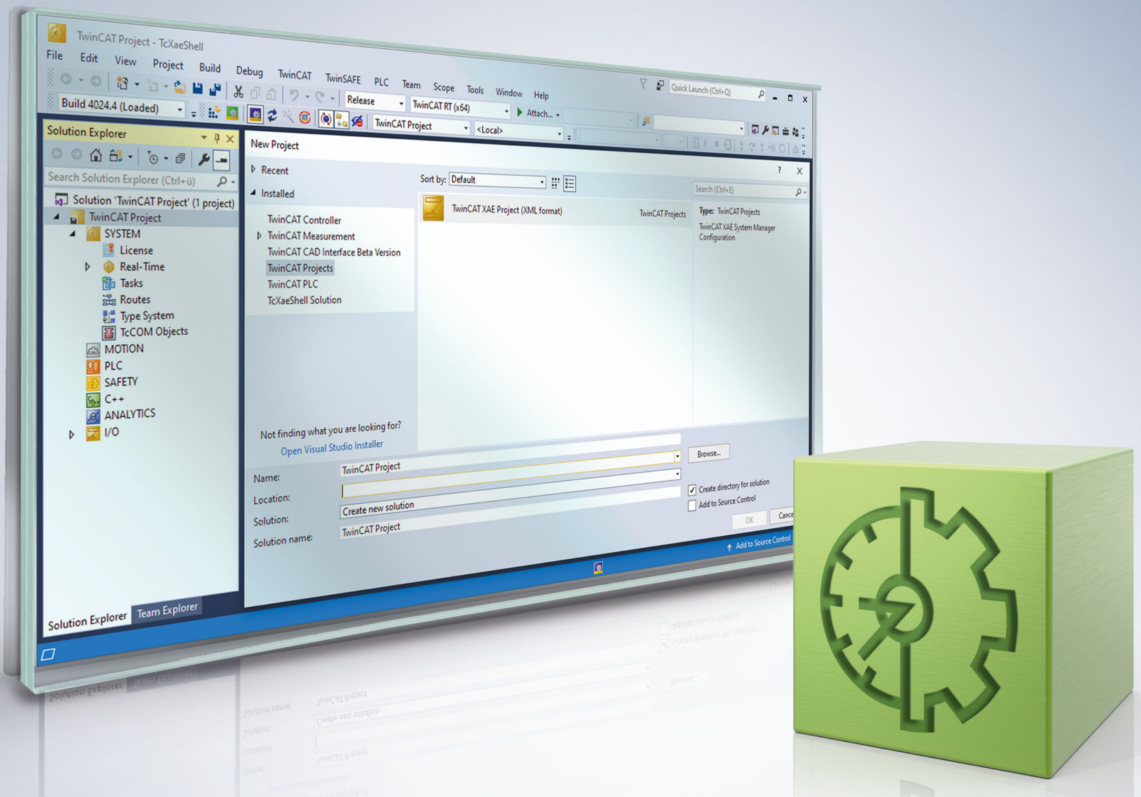
BECKHOFF New Automation Technology

Manual | EN

TE1000

TwinCAT 3 | Software Protection



2024-04-11 | Version: 1.17.3

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all 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 notice.
No claims to modify products that have already been supplied may be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

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

EtherCAT

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

Copyright

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are 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 technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER
Hazard with high risk of death or serious injury.

⚠ WARNING
Hazard with medium risk of death or serious injury.

⚠ CAUTION
There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE
The environment, equipment, or data may be damaged.

Information on handling the product

ℹ This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Introduction

The TwinCAT 3 Engineering is equipped with various functions for the protection of the PLC application software:

- Configurable access restrictions to the PLC source code through the definition of user groups and the assignment of access levels ("Object Protection Level")
- Know-how protection through encryption of PLC source code and boot file
- Cloning protection through the use of the TwinCAT 3 license technology for the OEM application software (requires a Beckhoff IPC/EPC or TwinCAT 3 dongle)

Through the use of the TwinCAT 3 license technology the OEM can additionally generate licenses himself for functional extensions of his application software and market them (requires a Beckhoff IPC/EPC or TwinCAT 3 dongle).

These functions are currently only available for the PLC area of TwinCAT 3.

An OEM certificate signed by Beckhoff is required in order to be able to use the functions to protect the application software. You can find details in the chapter TwinCAT OEM certificates [▶ 19].

The central switching point of the access protection is a user database.

Contents

Introduction [▶ 8]	This section provides general information on the system requirements, the software access protection, the user database and the software protection configurator.
Quick start [▶ 13]	This section provides you with a quick introduction to the two most important topics: - Regulation of access to source code - Protection against unauthorized use of software functions with own licenses
TwinCAT OEM certificates [▶ 19]	This section describes how you can apply for, install and extend the OEM certificate required to protect application software.
User databases (user DBs) [▶ 33]	This section describes how to create user databases and connect them with the project.
Setting up basic protection of PLC application software [▶ 75]	This section describes how to protect the OEM application software. In particular, the subjects of user access rights, encryption, signing and OEM application licenses are explained in detail.

2.1 General system requirements

Operating system:

- At least Windows 7 (or its Embedded version) is required in order to be able to use all the functions for the protection of the application software. Windows XP and Windows CE (Windows Embedded Compact) do not support either the encryption of the boot file or OEM licenses.

Reliable protection only when using the latest TwinCAT 3 version

For reliable protection (e.g. secure encryption), always use the latest TwinCAT 3 version. This provides the maximum security.

Use at least TwinCAT 3.1 Build 4024.x.
For security reasons, do not use an older version!

2.2 The three pillars of software access protection

The three pillars of software access protection are:

- Encryption *(= no longer readable*)
- Signing *(= no longer exchangeable)*
- Assignment of access rights (-> "Object Protection Level [▶ 70]")

Protecting a project from unauthorized access therefore includes the following measures:

- Encryption and signing of the project components
- Defining the access rights to the project components
- **Important: encryption and signing of the associated project file**

The encryption without setting the correct access level will protect the corresponding file at operating system level, but would still allow access via the TwinCAT 3 Engineering.

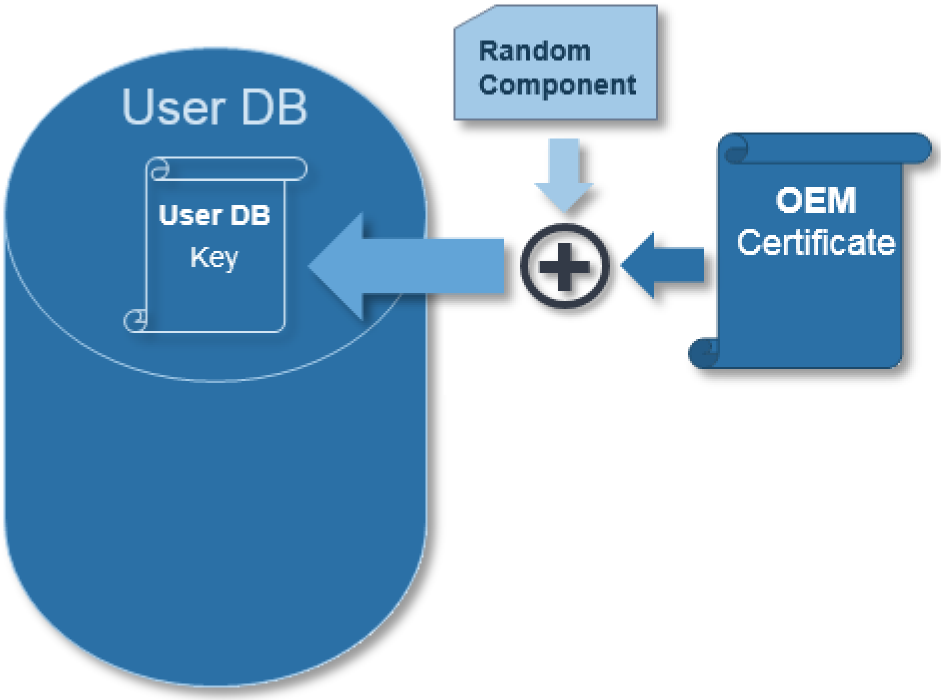
Conversely, the setting of a correct access level would define the access within the TwinCAT 3 Engineering, but access to the source code via the operating system level would still be possible.

Without a signature, a project file or a project component could be exchanged for another file with the same name.

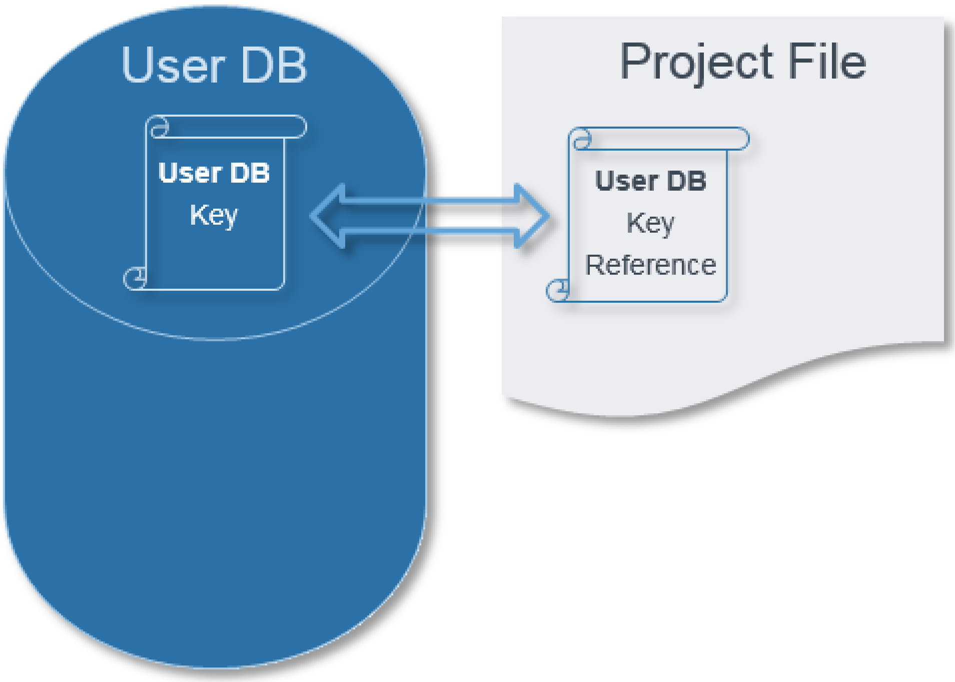
2.3 User database as a central switching point

Access to the PLC project components is regulated via a user database (User DB) [▶ 33].

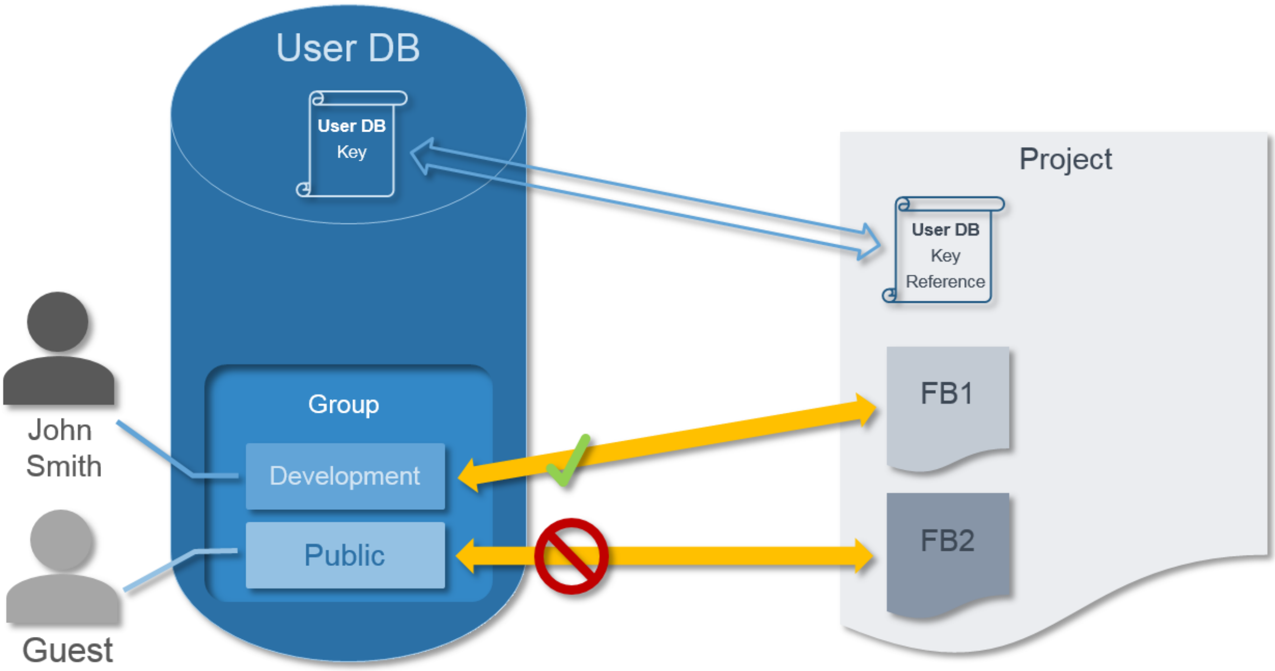
The content of the database is protected against unauthorized changes through signing by the administrator. In order to clearly identify the database, it is provided with a so-called "User DB Key" – a unique identifier made up of components of the OEM certificate and a random component. The random component ensures that every User DB created is given a unique User DB Key.



If a project is linked by an authorized user with a specific User DB, its User DB Key is stored in the project. Afterwards this project can only be opened in conjunction with this User DB.



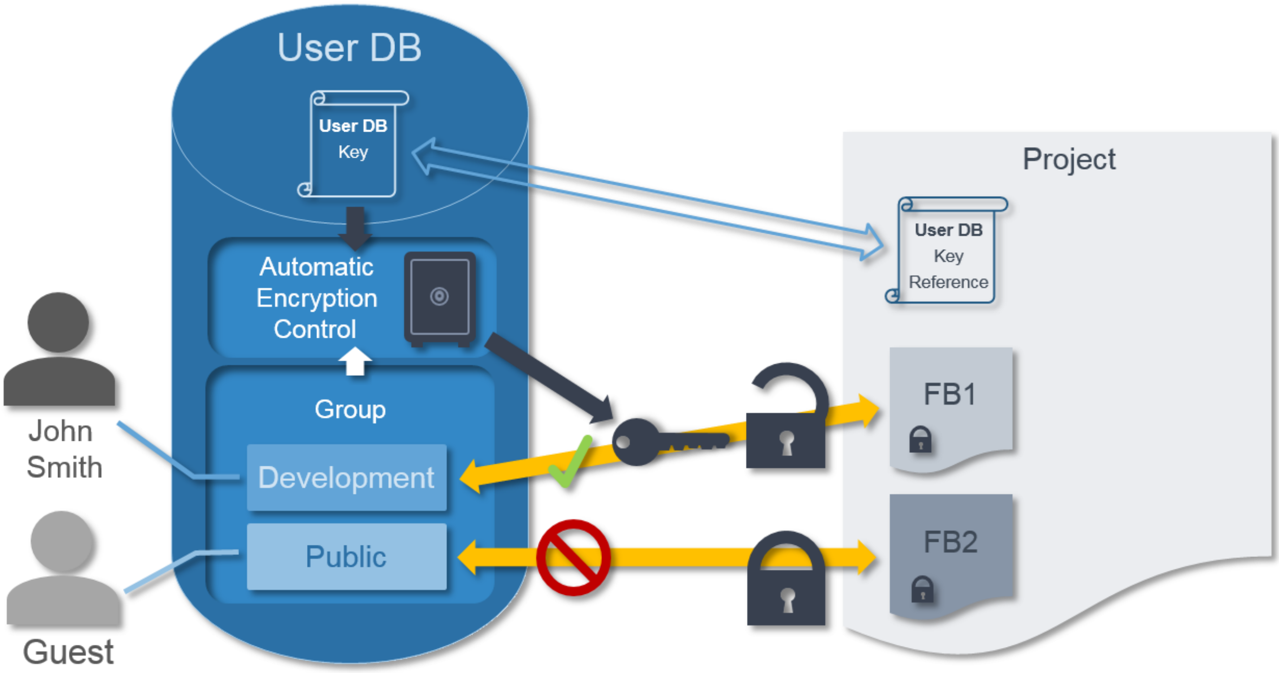
The Introduction [▶ 56] of a user is regulated within the User DB via groups.



This means that at first, the access rights are specified within TwinCAT 3 Engineering. However, access to the PLC source code or the exchange of project files would still be possible via the operating system level. Therefore, apart from the regulation of the access rights, there are two further protective measures in the TwinCAT 3 Engineering: the signing and encryption of the project file.

The signing of the project file ensures that the project file cannot be exchanged for another file with the same name at operating system level. The signature data of the file are saved in the higher-level project node. The project must be connected with the user database.

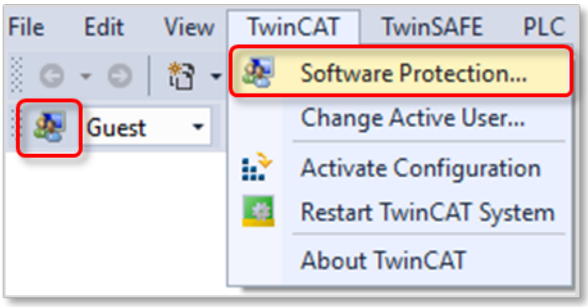
The key used for the encryption of the project file is saved in the user database. The corresponding user database must therefore always be present on the **engineering computer** (directory: *c:\Twincat\3.1\CustomConfig\userDBs*).



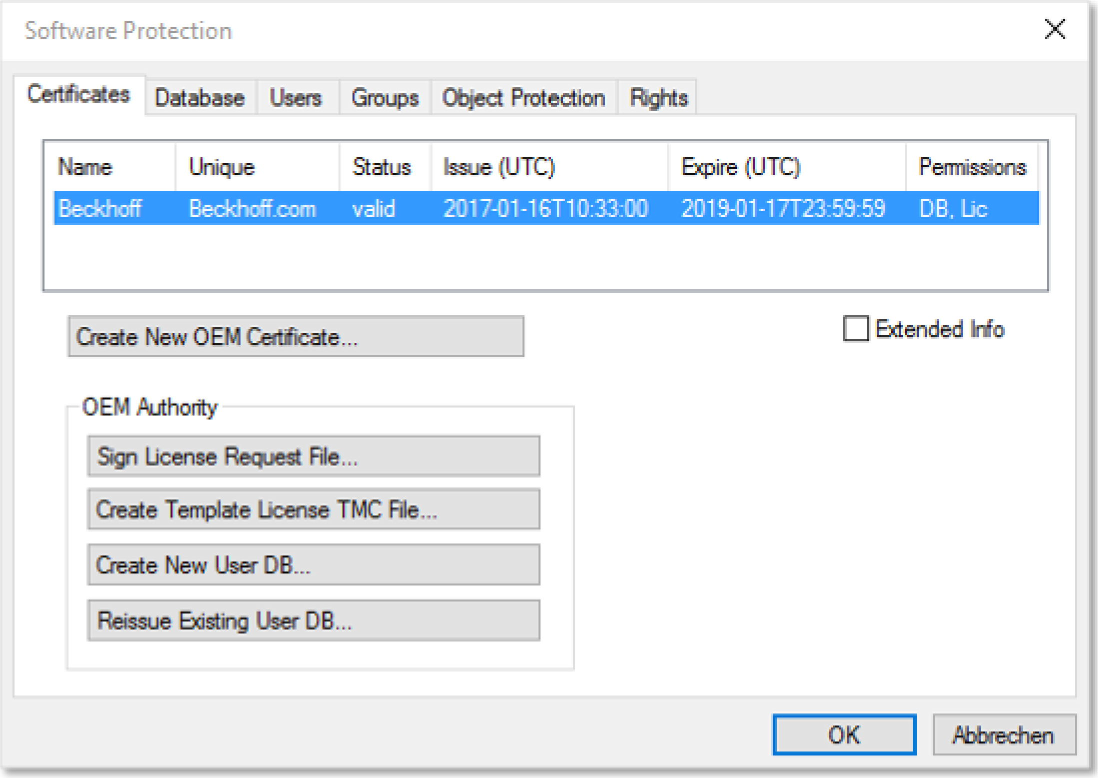
2.4 Software protection configurator

A central configurator is available to you for the configuration of the general software protection functions.

To start the **Software Protection** configurator, select the **Software Protection** command in the **TwinCAT** menu or click on the associated button in the **TwinCAT XAE Software Protection** toolbar. To add the toolbar to the user interface, activate it in the menu **View > Toolbars**.



The configurator then opens, in which you can configure the protection of the application software.



Name	Unique	Status	Issue (UTC)	Expire (UTC)	Permissions
Beckhoff	Beckhoff.com	valid	2017-01-16T10:33:00	2019-01-17T23:59:59	DB, Lic

Hints about configuration can be found in the following sections:

- TwinCAT OEM certificates [▶ 19]
- User databases (user DBs) [▶ 33]
- Setting up basic protection of PLC application software [▶ 75]

3 Quick start

3.1 Control of access to the PLC source code

From version Build 4024, TwinCAT 3 offers the option to encrypt PLC source code and to control access to the PLC source code via rights management. The central element is a user database (User DB), which is created with the inclusion of the OEM certificate (as the verification basis).

Note: The OEM certificate is only required to create the user database, not to use or modify it.

Prerequisite for using this function: Issue of a TwinCAT OEM certificate [▶ 19]

System requirements

- TwinCAT 3 OEM certificate TC0007 (Crypto version 1 or 2)
- Operating system: at least Windows 7 (also as Embedded version)
- TwinCAT version: at least TwinCAT 3.1 Build 4024

Reliable protection only when using the latest TwinCAT 3 version

For reliable protection (e.g. secure encryption), always use the latest TwinCAT 3 version. This provides the maximum security.

Use at least TwinCAT 3.1 Build 4024.x.
For security reasons, do not use an older version!

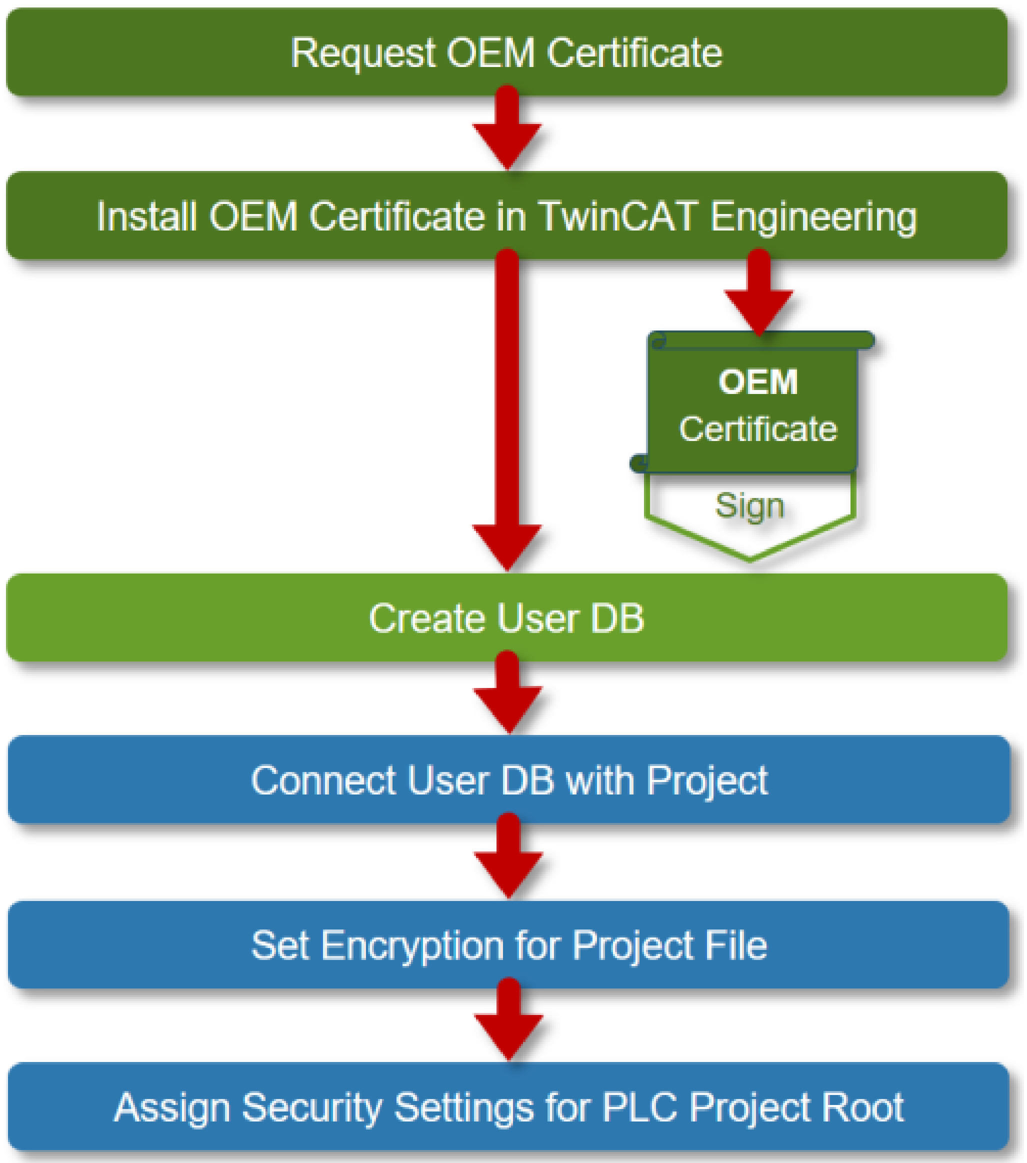
General notes

- Observe the general information regarding OEM certificates.
- The OEM certificate is only required once to create the User DB.
- Changes to the User DB only need to be signed by the administrator of the User DB (without using the OEM certificate).
- It is essential for the administrator of the User DB to have a strong password. Otherwise the User DB is easy to attack.
- The validity of the User DB is independent of the validity period of the OEM certificate. The User DB therefore remains valid even after the expiry of the validity period of the OEM certificate and can also be modified afterwards.
- Comments on the later extension of the certificate (after 2 years) can be found here Extending an OEM certificate [▶ 31].
- **Important:** store the password of the OEM certificate and the administrator of the User DB in a secure place. Beckhoff cannot restore the passwords if they are lost!
- The OEM certificate is not required on the target systems and should not be stored there for security reasons!

Procedure

The following procedure describes the simplest case:

- There is one user ("Admin") who has full access to the project
- All others ("Guest") are not allowed to view or modify the project.
- The administrator authenticates himself via a (secure!) password


Request OEM Certificate
Install OEM Certificate in TwinCAT Engineering
OEM Certificate
Sign
Create User DB
Connect User DB with Project
Set Encryption for Project File
Assign Security Settings for PLC Project Root

Links to documentation

1. Requesting (ordering) an OEM certificate [▶ 28]
2. Installing the OEM certificate [▶ 30]
3. Creating a user database [▶ 33]
   For the simplest standard case, you only need to define the administrator's name and his password and execute no further settings (e.g. no creation of additional users) in the User DB.
4. Linking the user database to a project [▶ 69]
5. Setting the encryption of the project file [▶ 76]
6. Setting access rights for PLC project root [▶ 70]

3.2 OEM licenses: protection against unauthorized use of software functions

With the help of TwinCAT 3 license technology a PLC application can be protected against unauthorized use/cloning through binding to hardware (Beckhoff IPC or TwinCAT dongle). With the same licensing technology, additional functionalities of the PLC application can also be activated for end customers.

Prerequisite for using this function: Issue of a TwinCAT OEM certificate [▶ 19]

System requirements

- TwinCAT 3 OEM certificate
  (only for the **creation** of a license type and the **signing** of license files, not for the **use** of an OEM license)
- Operating system: at least Windows 7 (also as embedded version)
  (Windows CE / Windows Embedded Compact is not supported!)
- Beckhoff IPC or TwinCAT 3 license dongle
- TwinCAT version: at least TwinCAT 3.1 Build 4022
- TC3 PLC Lib Tc2_Utilities v3.3.24 (or higher)

Note: a User DB is not required for the use of OEM licenses.

Reliable protection can only be guaranteed when a Beckhoff IPC or TwinCAT license dongle is used

For safe protection, always use a Beckhoff IPC or a TwinCAT 3 license dongle. The use of OEM licenses on non-Beckhoff computers without TwinCAT 3 license dongle is insecure and is not supported!

Reliable protection only when using the latest TwinCAT 3 version

For reliable protection (e.g. secure encryption), always use the latest TwinCAT 3 version. This provides the maximum security.

Use at least TwinCAT 3.1 Build 4024.x.
For security reasons, do not use an older version!

General notes

If you use OEM licenses make sure you encrypt your boot project!

Remember that the license ID [▸ 85] queried via FB_CheckLicense [▸ 92] in the binary code can easily be found and (with a little effort) manipulated with a hex editor. Therefore, be sure to encrypt your boot project [▸ 77] (safest), or at least disguise the queried license ID in the source code as best as possible.

- A user database is not required for the application licensing.
- The license is validated by the TwinCAT 3 runtime (XAR). The TwinCAT 3 runtime must therefore be installed on the IPC.
- The validity of the application license is independent of the validity period of the OEM certificate. The application license thus remains valid even after the validity of the OEM certificate has expired.
- The use of OEM application licenses always requires a TwinCAT 3 dongle or a Beckhoff IPC.
- For IPCs with a platform level >= 90 (non-Beckhoff IPCs) a TwinCAT-3 dongle must always be used as a "License Device" for security reasons!

Typical applications

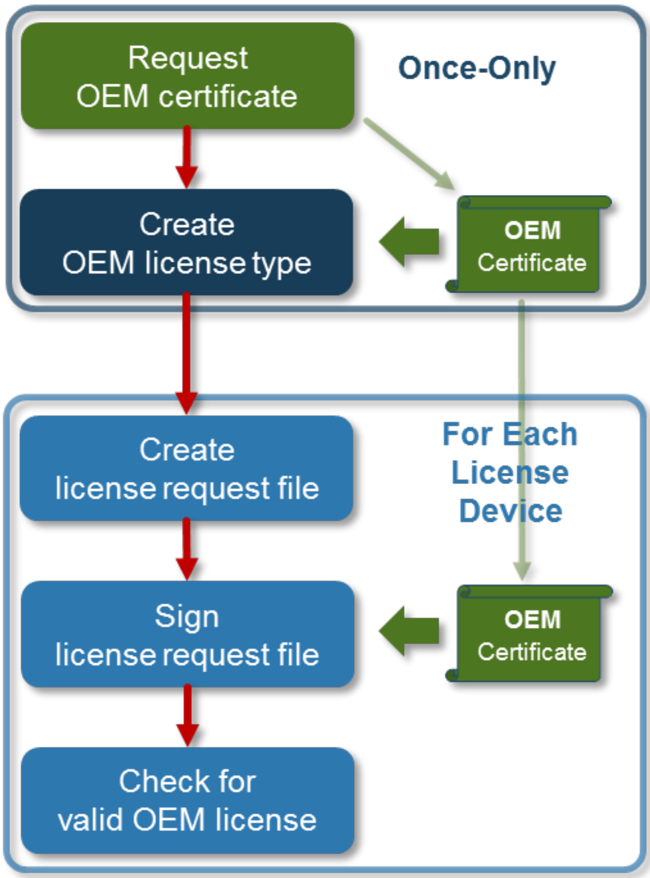
- The application is to be protected against cloning by binding it to hardware (TwinCAT 3 dongle or Beckhoff IPC).
- Additional functions in the application are to be enabled by an associated license.

Procedure

First of all, the TwinCAT 3 Engineering must be configured for the generation of application licenses. Among other things you need a small tool for this that is not part of the standard scope of delivery of the TwinCAT 3 Engineering.

The preparation of the TwinCAT 3 Engineering for application licenses is described in the section Preparation of the TwinCAT 3 Engineering [▸ 85].

The principle of the licensing process is illustrated in the following graphic:



Request OEM certificate

The basis for the licensing is an OEM certificate signed by Beckhoff with which the license is issued (by signing the License Request File).

How to apply for and install this certificate is described in the section Creating the "OEM Certificate Request File" [▸ 22].

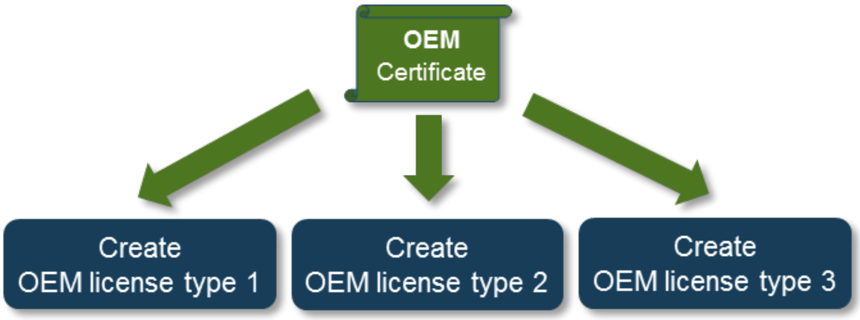
Be sure to use a strong password for your OEM certificate!

Create OEM license type

With the aid of data (OEM GUID) from the OEM certificate, a description file is generated for a license type. This license description file is the basis for the creation of a “License Request File” (see next step).

The process for generating a license description file is described in the section Creating a license description file for an OEM application license [▸ 85].

With an OEM certificate any number of license description files can be generated:



Create license request file

Now you can generate a “license request file” for a specific “license device” (TwinCAT 3 dongle or Beckhoff IPC).

The process for generating the file is described in the section Creating License Request Files for an OEM application license [▶ 88].

Application licenses for a non-Beckhoff IPC (platform level 90 or higher) always require a TwinCAT-3 dongle for security reasons!

Sign license request file

Sign
license request file
OEM
Certificate

The "license request file" generated must be signed with the OEM certificate and thus becomes a "license response file". This is the actual license file that is bound to the specific device that was specified when creating the "license request file".

The procedure to sign the “license request file” with the OEM certificate is described in the section Manual creation via the TwinCAT Engineering [▶ 89].

Subsequently, the license generated must be made available on the “license device” (TwinCAT 3 dongle or Beckhoff IPC) (see Importing License Response Files for an OEM application license [▶ 90]).

Version 3.3.24 of the TC3 PLC Lib Tc2_Utilities, which provides various function blocks for license handling, is available from TwinCAT 3 Build 4022.16. Among other things, it includes function blocks with whose help license files can be stored directly in a PLC application on a TwinCAT 3 dongle or downloaded from the latter. (See documentation on TC3 PLC Lib Tc2 Utilities)

You can download the required TC3 PLC Lib Tc2_Utilities: https://infosys.beckhoff.com/content/1033/tc3_security_management/Resources/5299845387/.zip

Check for valid OEM license

At the start (and during the runtime), the TwinCAT 3 runtime checks whether the application license is valid. You can query the result of this check with the PLC function block FB_CheckLicense (see Querying the OEM application license in a PLC application [▶ 91]).

In your PLC application you can then react as required to the result of the license validation check and can thus control the reaction to the presence or absence of your application license.

4 TwinCAT OEM certificates

A TwinCAT OEM certificate signed by Beckhoff is required in order to be able to use the application software protection functions.

The TwinCAT OEM certificate is exclusively intended for use together with TwinCAT.

With TwinCAT Build 4024, the TwinCAT OEM certificate version TC0008 can additionally be used to sign TwinCAT *.tmx files created with TwinCAT 3 in C++.

With the launch of TwinCAT 3.1 Build 4024, several new features relating to TwinCAT OEM certificates were introduced, compared to Build 4022:

- Update to a newer encryption version for the internal certificate data
- Introduction of an extended certificate version TC0008, with which C++ TwinCAT driver software created in TwinCAT 3 can also be signed
- This certificate version requires secure validation of the applicant data, since it is used in the Windows environment.
- The process of applying for a TwinCAT OEM certificate was modified for this purpose. **All OEM certificates must** be officially **ordered** to validate address and contact information. (However, the issuing of a TwinCAT OEM certificate remains free of charge.)
- TwinCAT OEM Certificates Extended Validation (TC0008) are only issued to existing Beckhoff customers.

Order numbers for TwinCAT OEM certificates

TC0007: TwinCAT OEM Certificate Standard (TwinCAT Software Protection)

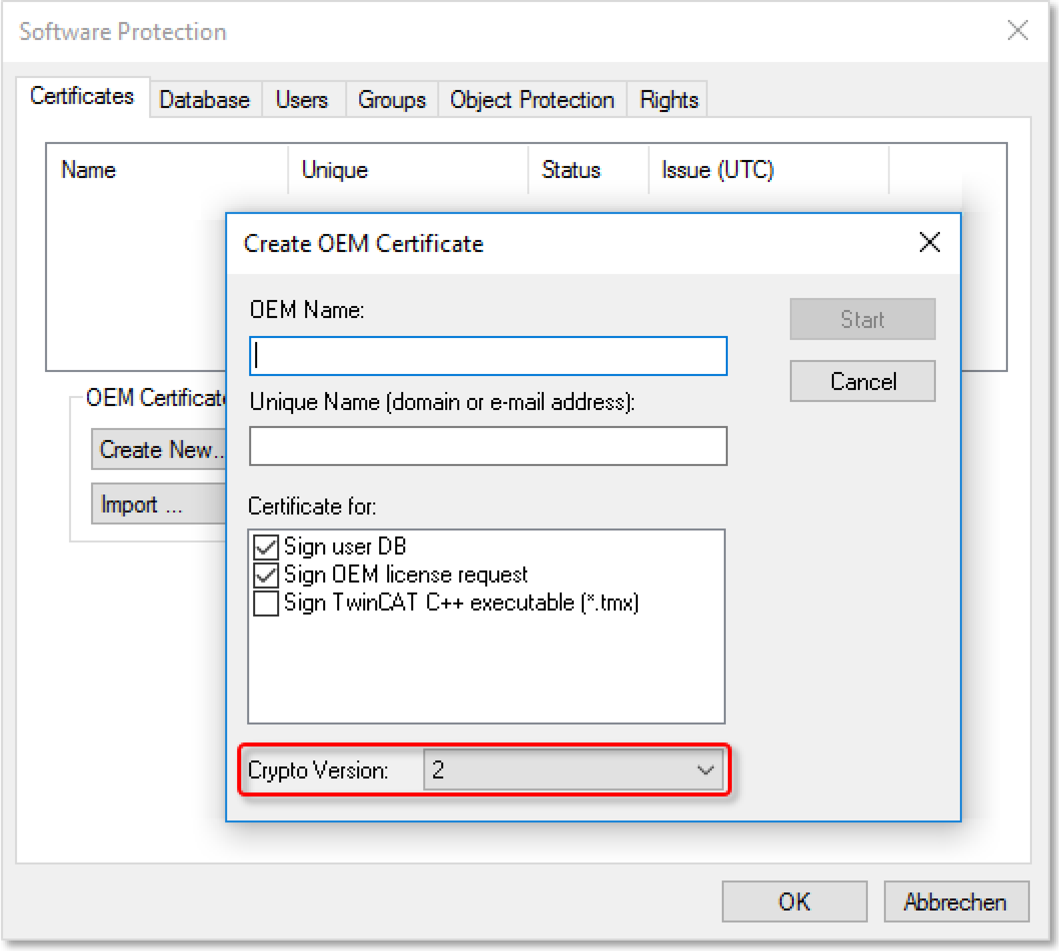
TC0008: TwinCAT OEM Certificate Extended Validation (like TC0007, additionally signing of TwinCAT driver software created with TwinCAT 3 in C++)

Only valid for TwinCAT 3.1 Build 4024.0: creation of a User DB requires Crypto Version 1

In the TwinCAT version Build **4024.0**, a user database [▸ 33] for the TwinCAT Software Protection may only be created with an OEM certificate with Crypto version 1!

Please note:

- **TC0008** includes all functionalities of TC0007
- The standard certificate version **TC0007** can optionally be issued with the encryption version of TwinCAT 3.1 Build 4022 or 4024.
- The certificate version **TC0008** with extended validation can only be issued with the newer encryption version of TwinCAT 3.1 Build **4024**.
- The encryption version of the certificate is defined by the user when the "OEM Certificate Request File" is created [▸ 22] (not when the order is placed!):



Create OEM Certificate

OEM Name:

Unique Name (domain or e-mail address):

Certificate for:

☑ Sign user DB
☑ Sign OEM license request
□ Sign TwinCAT C++ executable (*.tmx)

Crypto Version: 2

Compatibility of OEM certificates: Build 4022 <-> Build 4024:

- The encryption version (=1) of build 4022 (e.g. an existing OEM certificate created with build 4022 or UserDBs or OEM application licenses created with it) can also be used with build 4024 (the other way round it only works with encryption version 1!)
- A TwinCAT OEM certificate (Standard only) with encryption version **1** of Build **4024** (or UserDBs or OEM application licenses generated with it) can be used with TwinCAT 3.1 Build **4022**. (-> build 4022 can decrypt the certificate data of encryption version 1)
- A TwinCAT OEM certificate with encryption version **2** of Build **4024** (or UserDBs or OEM application licenses generated with it) can **not** be used with TwinCAT 3.1 Build **4022**! (-> build 4022 cannot decrypt the certificate data of encryption version 2!)
- TwinCAT OEM certificates with different encryption versions can be used in parallel in TwinCAT 3.1 Build **4024**: an OEM certificate with the encryption version of TwinCAT 3.1 Build 4022 for protecting user software, and a second OEM certificate with the encryption version of TwinCAT 3.1 Build 4024 for signing TwinCAT driver software.

Storage instructions for the application area: protection of OEM application software

The OEM key included in all certificate versions facilitates the use of the functions for protecting the TwinCAT 3 application software:

- Creating a user database (user DB) for user access control
- Create OEM application license description files
  (basis for issuing OEM application licenses)
- Issuing (signing) of OEM application licenses

The OEM Standard certificate (TC0007) is only required for these three purposes.

On which computer has the OEM certificate TC0007 to be stored?

The OEM certificate should only be located on the computer on which the three activities listed above are performed.

The OEM certificate TC0007 is not required:

- for the use of a User DB
- for the program sequence
- for the use of OEM application licenses

For security reasons, the certificate should not be delivered on control computers or installed randomly on computers with TwinCAT Engineering.

When using OEM licenses, the OEM certificate is only required once to **issue** the license (since it is used to sign the license file).

Storage instructions for the application area: signing TwinCAT driver software

The OEM key included in the certificate version TC0008 (TwinCAT OEM Certificate Extended Validation) can additionally be used to sign TwinCAT driver software created with TwinCAT 3 in C++.

If you use TC0008 only for this purpose, the following applies:

On which computer has the OEM certificate TC0008 to be stored?

The OEM certificate should only be located on the computer on which TwinCAT driver software created with TwinCAT 3 in C++ is signed.

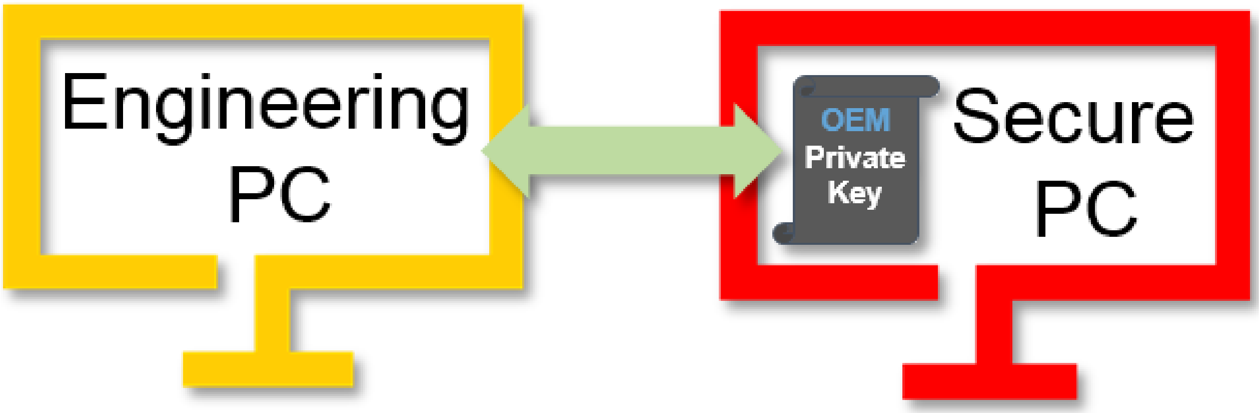
If you also use TC0008 for TwinCAT Software Protection, the relevant instructions for the computers on which the certificate may / should be stored also apply.

The OEM certificate TC0008 is not required for running the TwinCAT driver software signed with it.

The certificate should not be delivered on control computers or installed randomly on computers with TwinCAT Engineering.

Use of a secure PC

Use a secure PC for activities that require handling of the password for the OEM certificate private key, in order to prevent password sniffing.



Validity of the TwinCAT OEM certificate

For reasons of security, the validity of the OEM certificate is limited to two years.

The OEM may apply for an extension of the certificate before the two-year period has expired (or afterwards), in order to be able to continue working without interruption. (See Extending an OEM certificate [▶ 31])

What happens if the certificate has expired?

The following functions are **no** longer available with an invalid (expired) OEM certificate:

- Creating a user database
- Creating OEM application license description files
- Issuing (signing) of OEM application licenses
- Signing C++ executables (build 4024) with the OEM certificate

All other functions continue to work:

- Program execution is still possible.
- Issued OEM licenses remain valid.
- C++ executables signed with TC0008 continue to run (Build 4024).
- The user database remains valid, and the administrator can continue to modify/adapt the database. (It is no longer possible to create a new user database.)

4.1 Creating the "OEM Certificate Request File"

TwinCAT OEM certificates are only issued for existing Beckhoff customers.

Please get in touch with your Beckhoff sales contact for further information.

System requirements

- Min. TwinCAT 3.1 Build 4024
- Min. Windows 10 or TwinCAT/BSD (on the target system)

Do not use special characters (ä, é, ...) for company name and password!

The algorithm for processing the OEM certificate in TwinCAT cannot process special characters.

Order numbers for TwinCAT OEM certificates

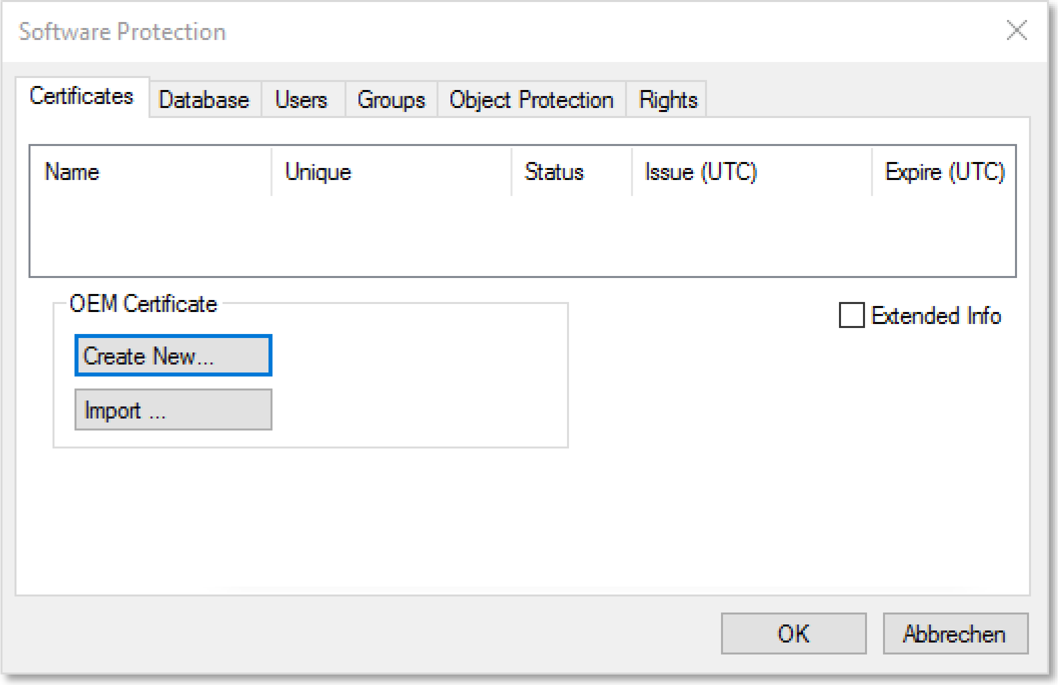
TC0007: TwinCAT OEM Certificate Standard (TwinCAT Software Protection)

TC0008: TwinCAT OEM Certificate Extended Validation (like TC0007, additionally signing of TwinCAT driver software created with TwinCAT 3 in C++)

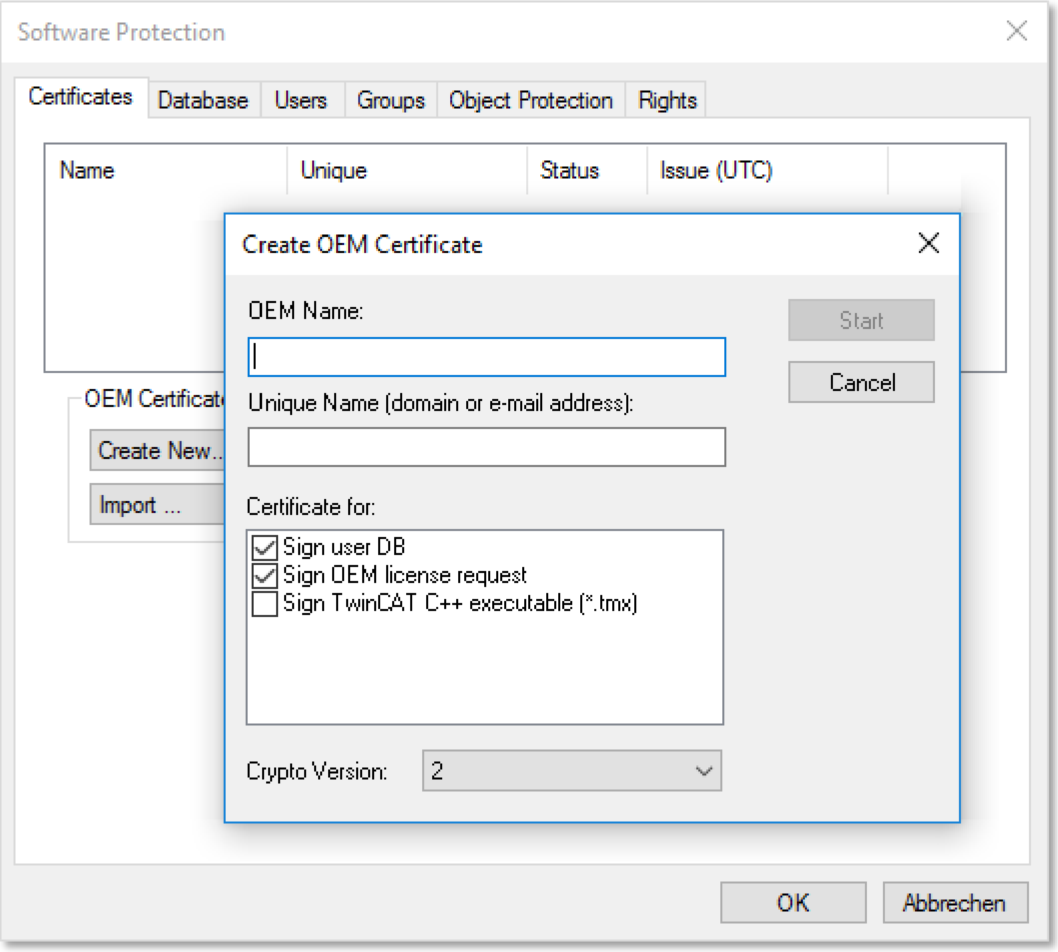
✓ The Software protection configurator [▶ 11] has been opened.

1. Select the **Certificates** tab.

2. Click **Create New…**.



⇨ The **Create OEM Certificate** input window opens.



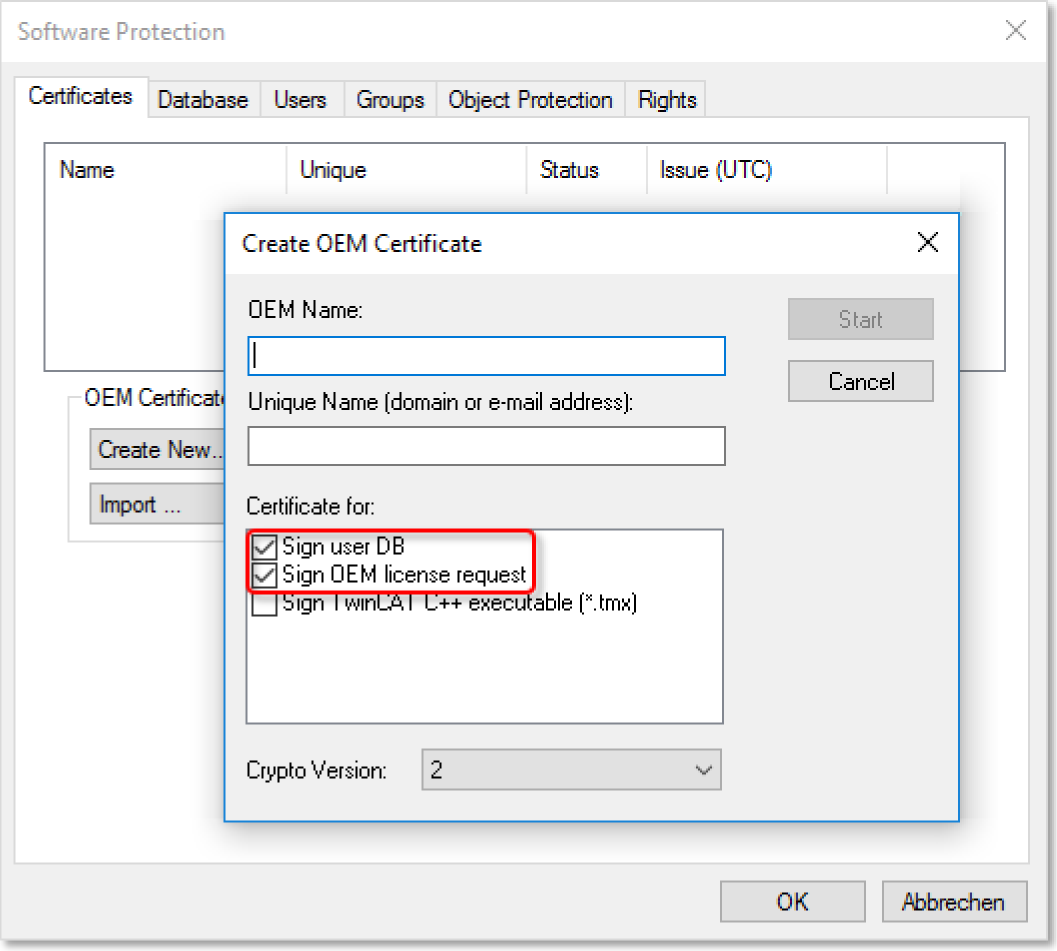
3. Enter the required data for an "OEM Certificate Request File":

- Enter your company name in the **OEM Name** text field. The name must have a clear reference to your company or your business unit.

Do not use special characters (ä, é, ...) for company name and password!

The algorithm for processing the OEM certificate in TwinCAT cannot process special characters.

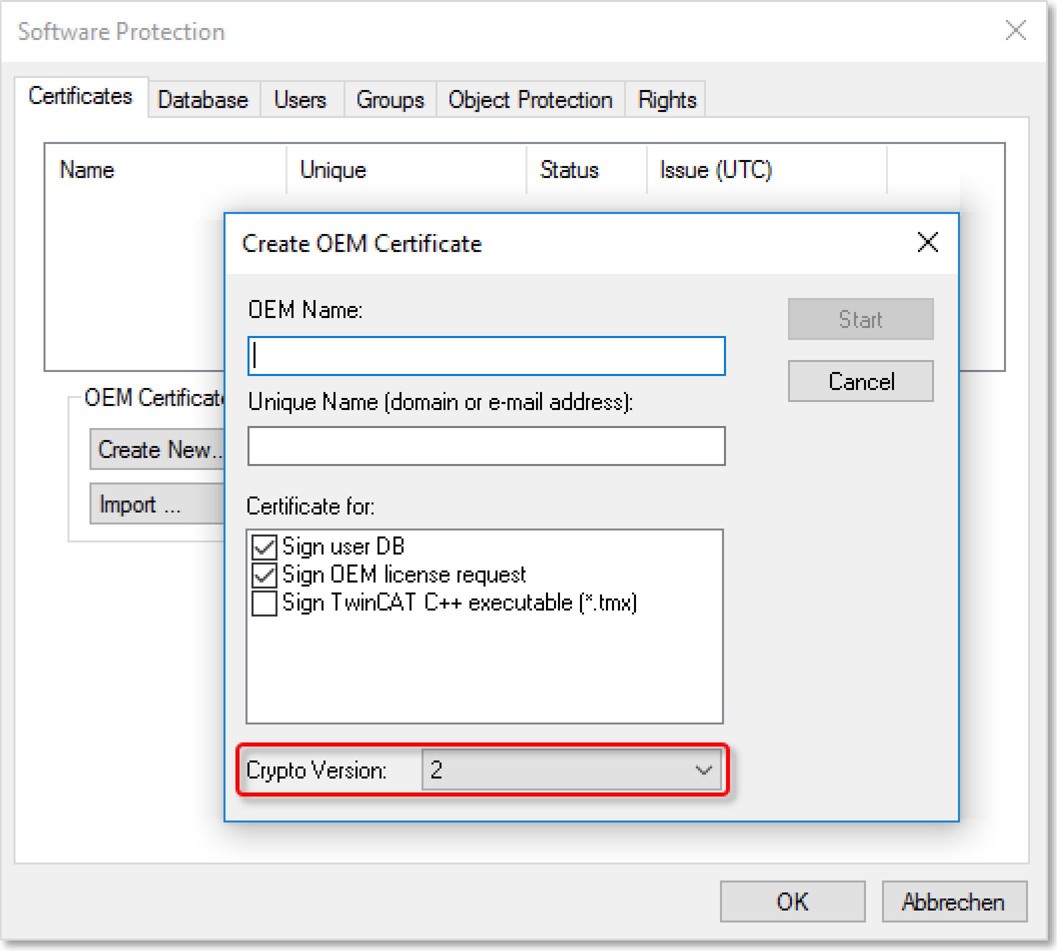
- Enter a **Unique Name**. The "OEM Unique Name" must be a unique name that uniquely identifies the owner of the certificate worldwide, preferably the URL of your company's website or your email address. The email address must be a company email address, i.e. it must be possible to assign it unambiguously to a company.
- For a **Standard certificate** (TC0007), make sure that at most these two checkboxes are checked for the application area of the certificate:



☑ Sign user DB
☑ Sign OEM license request
□ Sign TwinCAT C++ executable (*.tmx)

Crypto Version: 2

- The current Crypto version (for the encrypted content of the certificate) is "2". You should only select the older Crypto version "1" if you want to use this certificate with TwinCAT 3.1 Build 4022.x. **Information:** Crypto version "1" can only be selected for a Standard certificate (TC0007).



Create OEM Certificate

OEM Name:

Unique Name (domain or e-mail address):

Certificate for:

☑ Sign user DB
☑ Sign OEM license request
☐ Sign TwinCAT C++ executable (*.tmx)

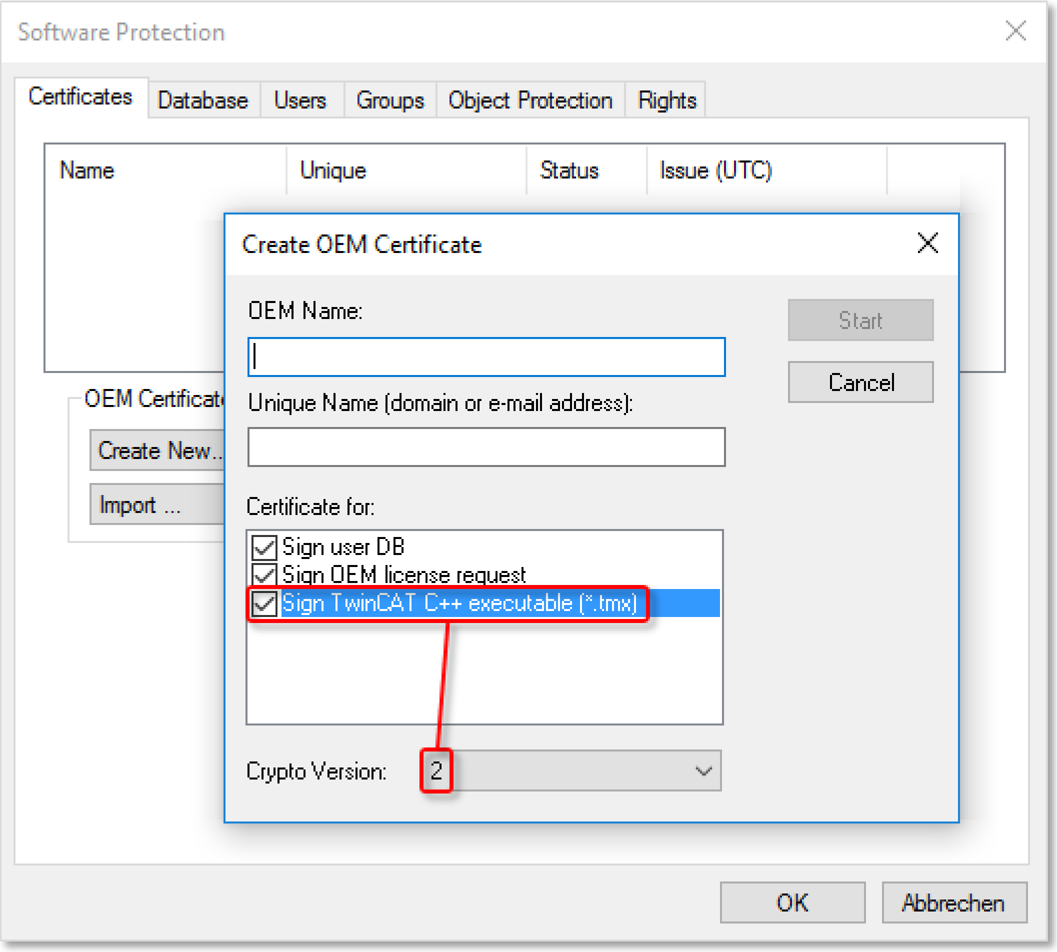
Crypto Version: 2

Only valid for TwinCAT 3.1 Build 4024.0: creation of a User DB requires Crypto Version 1

In the TwinCAT version Build **4024.0**, a user database [▸ 33] for the TwinCAT Software Protection may only be created with an OEM certificate with Crypto version 1!

- The checkbox "**Sign TwinCAT C++ executables** (*.tmx)" should only be checked for the purpose of applying for a certificate with "**Extended Validation**" (TC0008) that can be used for signing C++ executables generated with TwinCAT 3 (including Matlab/Simulink). This certificate version requires

more complex validation of your contact data (and therefore more time) during the certificate ordering process and should therefore only be selected if you really need this option:



Create OEM Certificate

OEM Name:

Unique Name (domain or e-mail address):

Certificate for:

☑ Sign user DB
☑ Sign OEM license request
☑ Sign TwinCAT C++ executable (*.tmx)

Crypto Version: 2

- A certificate with "Extended Validation" (TC0008) always requires Crypto version "2". (Please note: This certificate version cannot be used with TwinCAT 3 Build 4022.x!)

1. Once you have entered the data, click **Start** and select a directory to save the file. **Important**: You can simply accept the suggested directory "c:\twincat\3.1\customconfig\certificates". You need the newly created file in this directory in order to be able to read out the file fingerprint for this file in a subsequent step.
   ⇨ A dialog for selecting a password for the OEM Private Key opens.
2. Issue a password for the OEM Private Key**.**

Do not use special characters (ä, é, ...) for company name and password!

The algorithm for processing the OEM certificate in TwinCAT cannot process special characters.

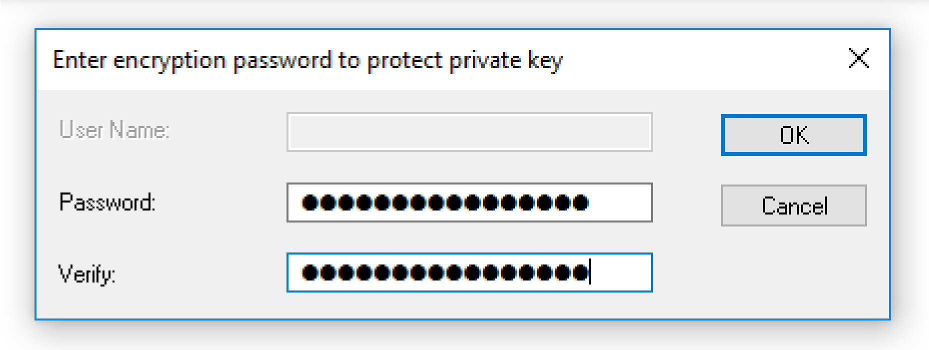
Important: Password security!

Be sure to use a strong password for your OEM certificate!
Protect your password with appropriate measures so that it cannot fall into the wrong hands!

Password cannot be restored if lost

Beckhoff is unable to recover or reset your password. If you forget or lose the password for your OEM certificate, you can no longer use it and have to request a new OEM certificate.

1. Confirm the password by entering it again and close the dialog with **OK**.



Enter encryption password to protect private key

User Name:

Password: ●●●●●●●●●●●●●●●●

Verify: ●●●●●●●●●●●●●●●●

⇨ The file is saved.

The "OEM Certificate Request File" generated in this way must now be signed by the Beckhoff certificate section in order to be valid. The procedure is described in chapter "Requesting an OEM certificate [▶ 28]".

4.2 Determining the file fingerprint of the OEM certificate file

You need this functionality to request a **TwinCAT OEM Certificate Extended Validation** (TC0008).

System requirements

This functionality requires TwinCAT 3.1 Build 4024 of higher.

The OEM Certificate Request File becomes the TwinCAT OEM certificate once it is signed by Beckhoff. The files do not differ except for this signature. For this reason, the term "TwinCAT OEM certificate file" is used for both file versions in the following sections.

Reading the "file fingerprint" of an OEM certificate file via TwinCAT 3 Engineering

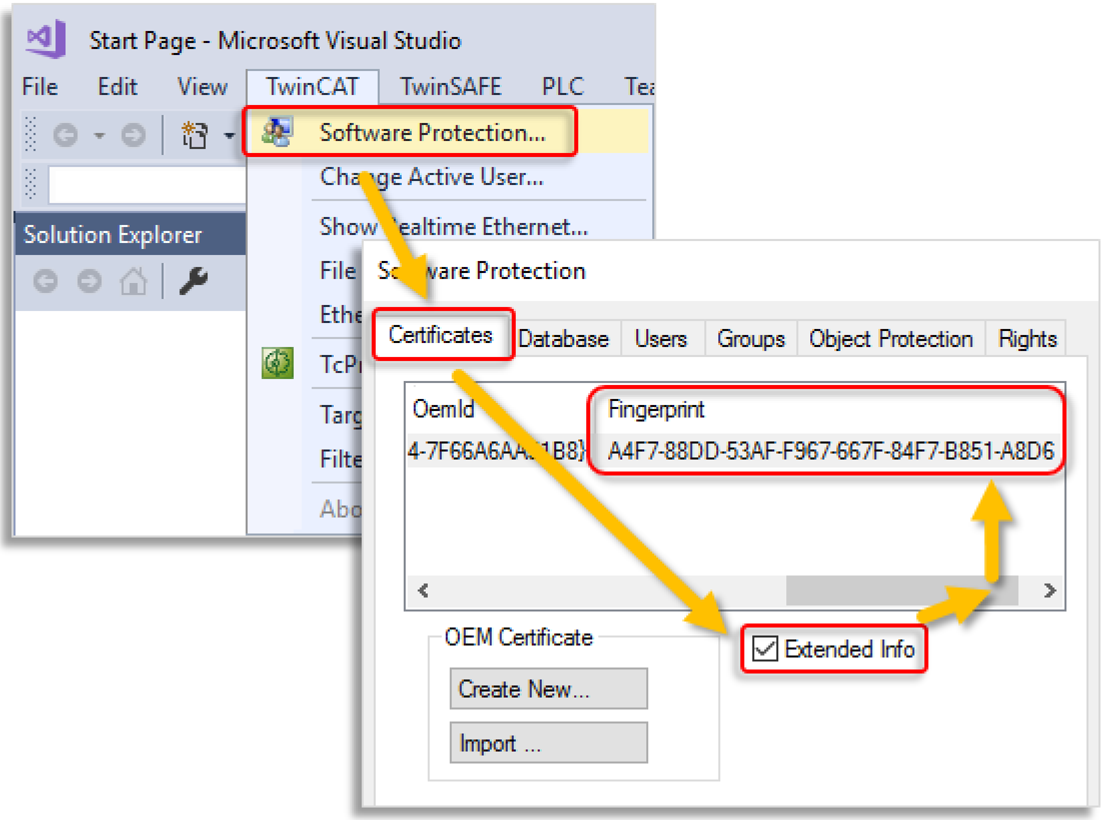
For this function it is necessary that the OEM certificate file is located in this directory: "c:\twincat\3.1\customconfig\certificates".

This directory contains your OEM certificate, if you already have a certificate and want to renew it.

If you did not change the suggested directory when creating the "OEM Certificate Request File", the file is already in this directory.

Procedure:

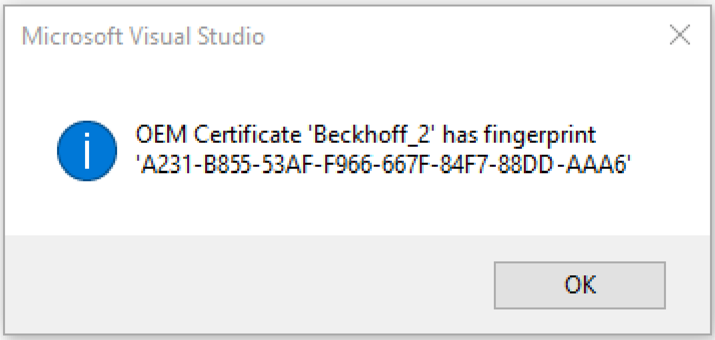
1. Call up the TwinCAT 3 Software Protection configurator.



OemId	Fingerprint
4-7F66A6AA...1B8}	A4F7-88DD-53AF-F967-667F-84F7-B851-A8D6

☑ Extended Info

2. Select the **Certificates** tab.
3. Check the **Extended Info** checkbox.
4. In the window scroll to the right until you see the **Fingerprint** column. (As an alternative, you can simply double-click the certificate line. The Fingerprint file is then displayed in a pop-up window:



OEM Certificate 'Beckhoff_2' has fingerprint 'A231-B855-53AF-F966-667F-84F7-88DD-AAA6'

The shortcut [Ctrl] + [C] can be used to copy the fingerprint data from the message window to the Windows clipboard.

4.3 Requesting the OEM certificate

Please refer to the information in the previous chapters and, if applicable, to the information on extending an OEM certificate:

TwinCAT OEM certificates [▸ 19]

Creating the "OEM Certificate Request File" [▸ 22]

Extending an OEM certificate [▸ 31]

Order process for TwinCAT OEM certificates

An official order is required to request a TwinCAT OEM certificate. Please get in touch with your Beckhoff sales contact.

Notes:

- The issuing and extension of a TwinCAT OEM certificate is free of charge.
- TwinCAT OEM certificates are only issued for existing Beckhoff customers.
- For a new OEM certificate, create the Creating the "OEM Certificate Request File" [▸ 22] in TwinCAT Engineering.
- If the certificate is extended, the existing OEM certificate file is simply re-signed and is valid for another 2 years. (In this case, do not create an "OEM Certificate Request File".)
- The "OEM Certificate Request File" becomes the TwinCAT OEM certificate once it is signed by Beckhoff. The files do not differ except for this signature.
- In the following text the term "OEM certificate file" is used for both variants for simplicity's sake.
- Since a TwinCAT OEM certificate is a digital ID card, it is necessary to verify the contact data of the inquirer.
- The two OEM certificate versions represent different security levels, so the verification processes differ somewhat.
- You should only order TC0008 (TwinCAT OEM Certificate Extended Validation) if you really need it (signing of TwinCAT 3 C++ executables).

Order numbers for TwinCAT OEM certificates

TC0007: TwinCAT OEM Certificate Standard (TwinCAT Software Protection)

TC0008: TwinCAT OEM Certificate Extended Validation (like TC0007, additionally signing of TwinCAT driver software created with TwinCAT 3 in C++)

Overview of the ordering and validation process

Your email address must be a company email account (freemailers such as GMail or similar are not permitted) and correspond with the company name of the inquirer.

1. Contact your Beckhoff sales contact and announce the request of a TwinCAT 3 OEM certificate. Order "TC0007" or "TC0008".
2. Important: as the inquirer, please provide your contact details as the delivery address (= contact name and email address) and the area of use of the certificate (company name, address).
3. The contact details provided in the order will be verified and you (the inquirer named in the delivery address) will be contacted by Beckhoff Sales.
4. When requesting a new OEM certificate, create an Creating the "OEM Certificate Request File" [▸ 22] in TwinCAT 3 Engineering.
5. TC0008 only: determine the "File Fingerprint" of the OEM certificate file using TwinCAT Engineering (see Determining the file fingerprint of the OEM certificate file [▸ 27]). Please inform the Beckhoff sales contact of this File Fingerprint as part of your contact data verification. The transmission of the File Fingerprint must be done by a different communication channel than the one used for sending the OEM certificate request file.
6. Now send the "OEM certificate file" to the Beckhoff sales contact.
7. After signing the certificate file at the Beckhoff headquarters, you will receive it by e-mail from your contact person.

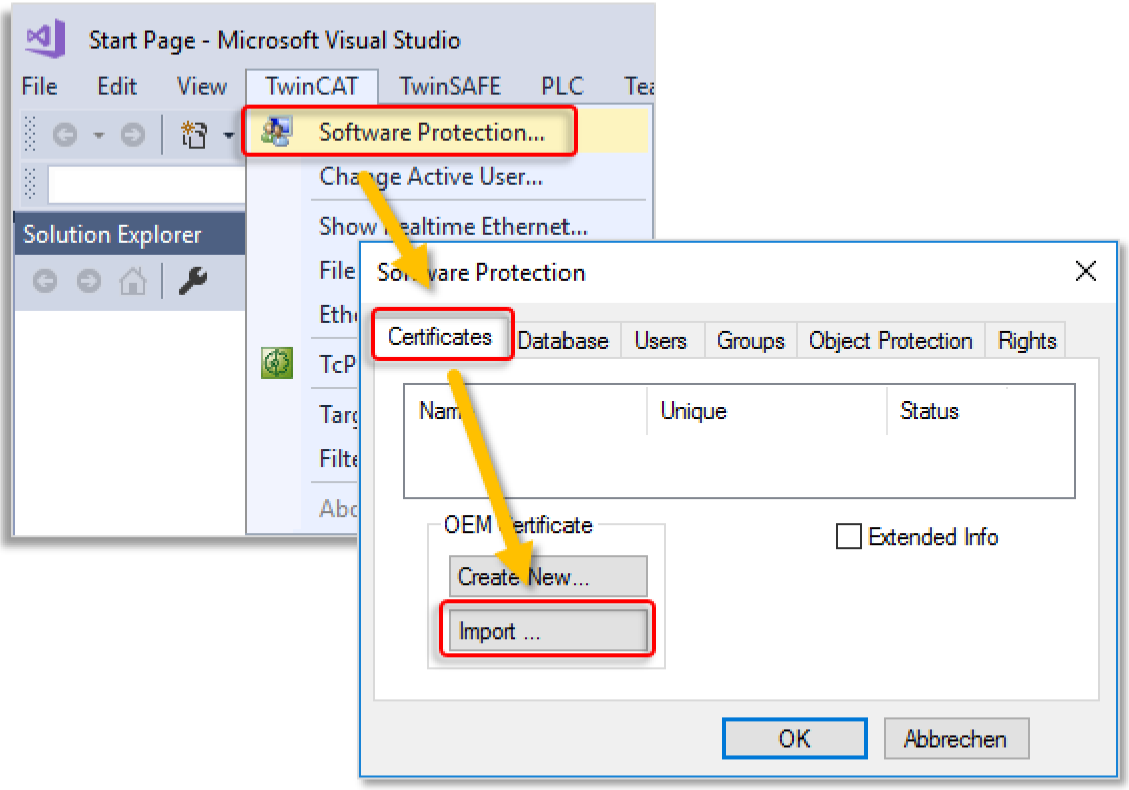
Please note that it may take a few days to validate your contact details and issue the certificate.

4.4 Installing the OEM certificate

TwinCAT root directory <TwinCAT_ROOT>

Up to and including TwinCAT 3.1.4024: **C:\TwinCAT**
From TwinCAT 3.1.4026: **C:\ProgramData\Beckhoff\TwinCAT**

Once you have received the signed certificate, import it via the Software Protection Control Center:



Note: This import function is only available in TwinCAT 3.1 build 4024 or higher.

Alternatively, you can store the file manually on your Engineering system in the directory *<TwinCAT_ROOT>\3.1\customconfig\certificates*.

After restarting the TwinCAT Engineering the certificate is listed in the software protection configurator on the **Certificates** tab.

Check whether the certificate is shown as "valid" there.

Storage instructions for the application area: protection of OEM application software

The OEM key included in all certificate versions facilitates the use of the functions for protecting the TwinCAT 3 application software:

- Creating a user database (user DB) for user access control
- Create OEM application license description files
(basis for issuing OEM application licenses)
- Issuing (signing) of OEM application licenses

The OEM Standard certificate (TC0007) is only required for these three purposes.

On which computer has the OEM certificate TC0007 to be stored?

The OEM certificate should only be located on the computer on which the three activities listed above are performed.

The OEM certificate TC0007 is not required:

- for the use of a User DB
- for the program sequence
- for the use of OEM application licenses

For security reasons, the certificate should not be delivered on control computers or installed randomly on computers with TwinCAT Engineering.

When using OEM licenses, the OEM certificate is only required once to **issue** the license (since it is used to sign the license file).

Storage instructions for the application area: signing TwinCAT driver software

The OEM key included in the certificate version TC0008 (TwinCAT OEM Certificate Extended Validation) can additionally be used to sign TwinCAT driver software created with TwinCAT 3 in C++.

If you use TC0008 only for this purpose, the following applies:

On which computer has the OEM certificate TC0008 to be stored?

The OEM certificate should only be located on the computer on which TwinCAT driver software created with TwinCAT 3 in C++ is signed.

If you also use TC0008 for TwinCAT Software Protection, the relevant instructions for the computers on which the certificate may / should be stored also apply.

The OEM certificate TC0008 is not required for running the TwinCAT driver software signed with it.

The certificate should not be delivered on control computers or installed randomly on computers with TwinCAT Engineering.

4.5 Extending an OEM certificate

To extend an OEM certificate, the same process applies as for requesting a new certificate. In this case, the certificate must also be ordered (the order numbers for a certificate extension are the same as for a new certificate request).

In contrast to a new certificate, you do not generate a new "OEM Certificate Request File" but send your existing certificate to the Beckhoff certificate section for renewal. Please notify us in the email that this is a certificate extension and not a new issue. Otherwise, the same criteria apply regarding the content of the email as for the application for a new certificate.

The existing certificate will be re-signed and is then valid for another two years.

Since the certificate only receives a new signature, it is fully compatible with the original version.

4.6 Updating an existing OEM certificate?

Unfortunately, an existing OEM certificate cannot be updated (new crypto version or different area of application). In this case it is always necessary to issue a new OEM certificate. It is, however, possible to extend the validity period of the OEM certificate by re-signing.

What are the consequences of a new OEM certificate for applications with existing or new TwinCAT UserDBs [▶ 33], OEM license description files [▶ 85] or OEM application licenses [▶ 89]?

TwinCAT User DB

- Use case: an existing User DB is to be reused, and at the same time a new OEM certificate is to be used. No problem: the existing User DBs can still be used and modified, since an OEM certificate is not required for either case. This also applies to the switch from build 4022 to build 4024 (and the User DB of build 4022).
- Use case: an existing User DB (created with old OEM certificate 1) is to be replaced by a new User DB (created with new OEM certificate 2). Provided that the requirement stated below for the crypto version is taken into account, there is no problem. However, the project must be linked once with the new User

DB [▶ 69]. A simple exchange at file level is not possible, i.e. the replaced User DB must always be reassigned to the project, because the new User DB has a different user DB key.

Note: all security settings for the project will be lost!

- A User DB created on the basis of a certificate with crypto version 2 cannot be used under build 4022. (The information encrypted in the UserDB cannot be decrypted by build 4022.)

TwinCAT OEM application licenses

OEM license description files: in general, an OEM license description file must always be created with the same certificate that is used to sign the OEM application license. (Otherwise the OEM key in the license description file will not match the OEM key in the application license.)

This is independent of the TwinCAT version or the crypto version.

Comments:

- OEM license description files and OEM application licenses created with a crypto version 2 certificate cannot be used in build 4022.
- But OEM license description files and OEM application licenses created with a crypto version 1 certificate can be used in build 4024.

5 User databases (user DBs)

Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

No changes in settings of a user database when a project is open

No project may be open when changing the settings of a user database.

No change of user when a project is open

No project may be open when changing the user.

New from TwinCAT 3 Build 4024.8: Extensions for user databases

A user database can be extended by so-called "extensions" from this version onwards. You can find details in the chapter Extensions for user databases [▶ 41].

5.1 Creating a user database

Only valid for TwinCAT 3.1 Build 4024.0: creation of a User DB requires Crypto Version 1

In the TwinCAT version Build **4024.0**, a user database [▶ 33] for the TwinCAT Software Protection may only be created with an OEM certificate with Crypto version 1!

Directory for storing user databases

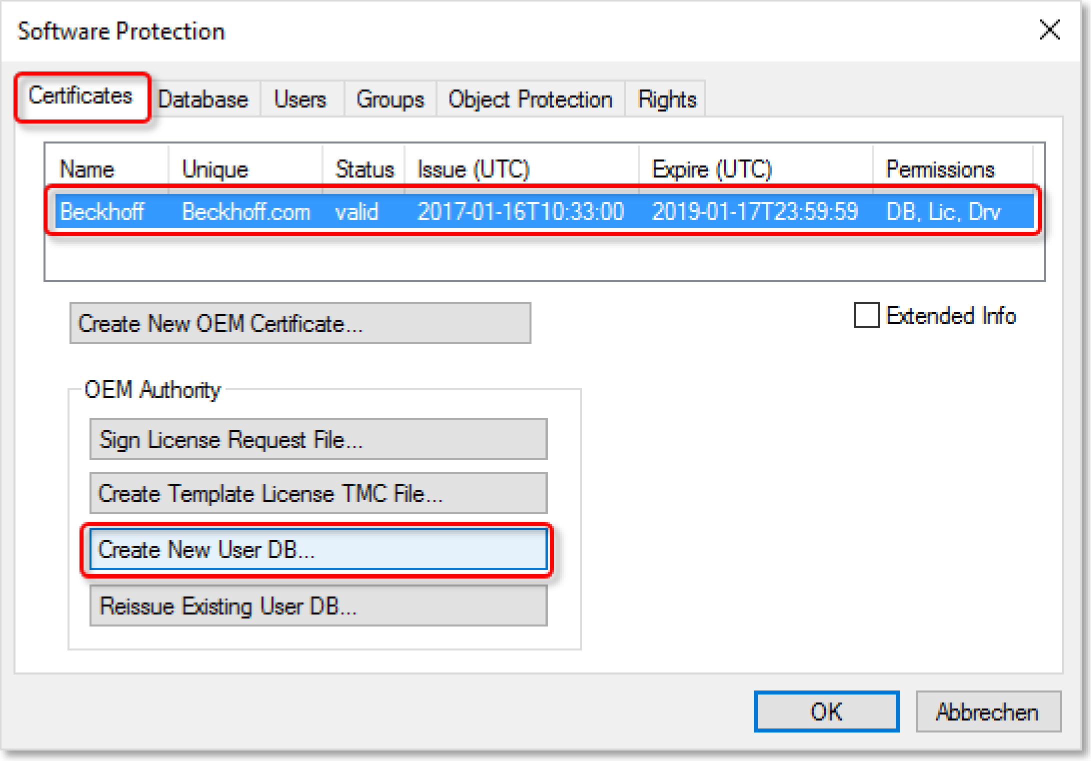
User databases must be stored in the following directory in order to be used in the TwinCAT Engineering: C:\TwinCAT\3.1\CustomConfig\UserDBs

Depending on the TwinCAT version, two different ways of creating a user database are available to you.

✓ User database can only be created or edited if no project is open. Close any open projects.

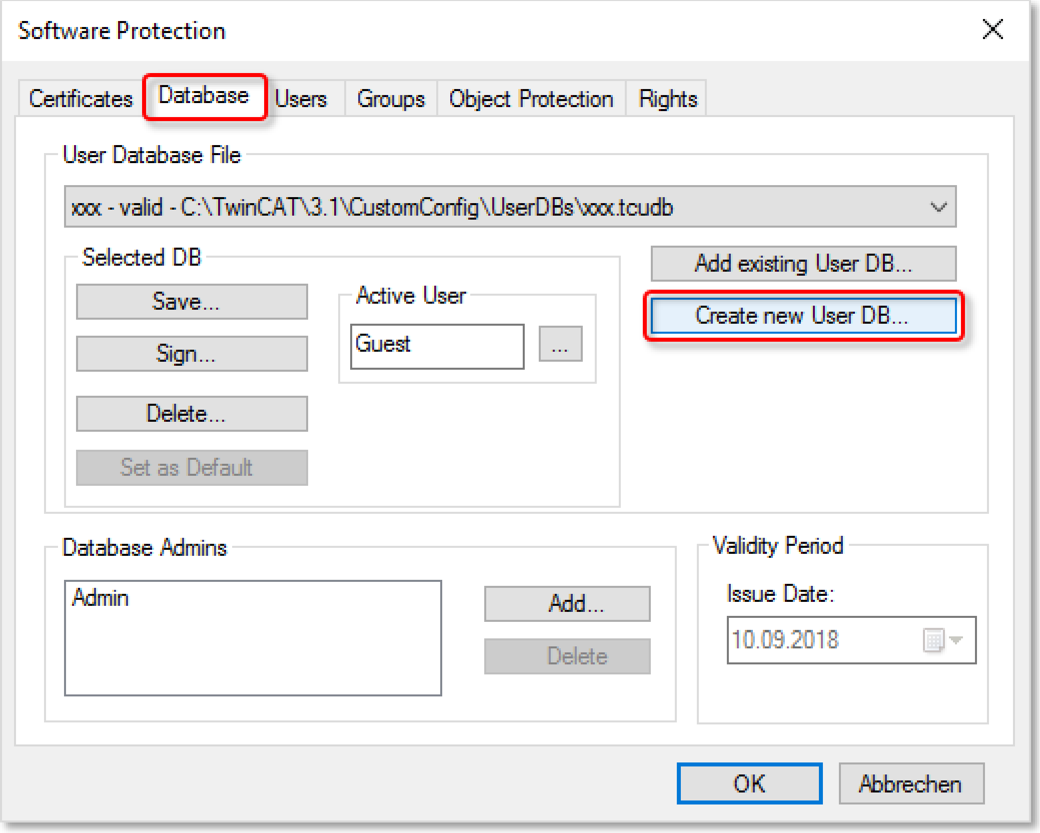
✓ The Software protection configurator [▶ 11] is opened.

1. If you are using a TwinCAT version < build 4022.25, open the **Certificates** tab, select the OEM certificate and then click on **Create New User DB...**



Name	Unique	Status	Issue (UTC)	Expire (UTC)	Permissions
Beckhoff	Beckhoff.com	valid	2017-01-16T10:33:00	2019-01-17T23:59:59	DB, Lic, Drv

2. If you are using a TwinCAT version >= build 4022.25, the **Create New User DB...** button is additionally available on the **Database** tab. Here you select the OEM certificate directly in the input mask. Click on **Create New User DB…**



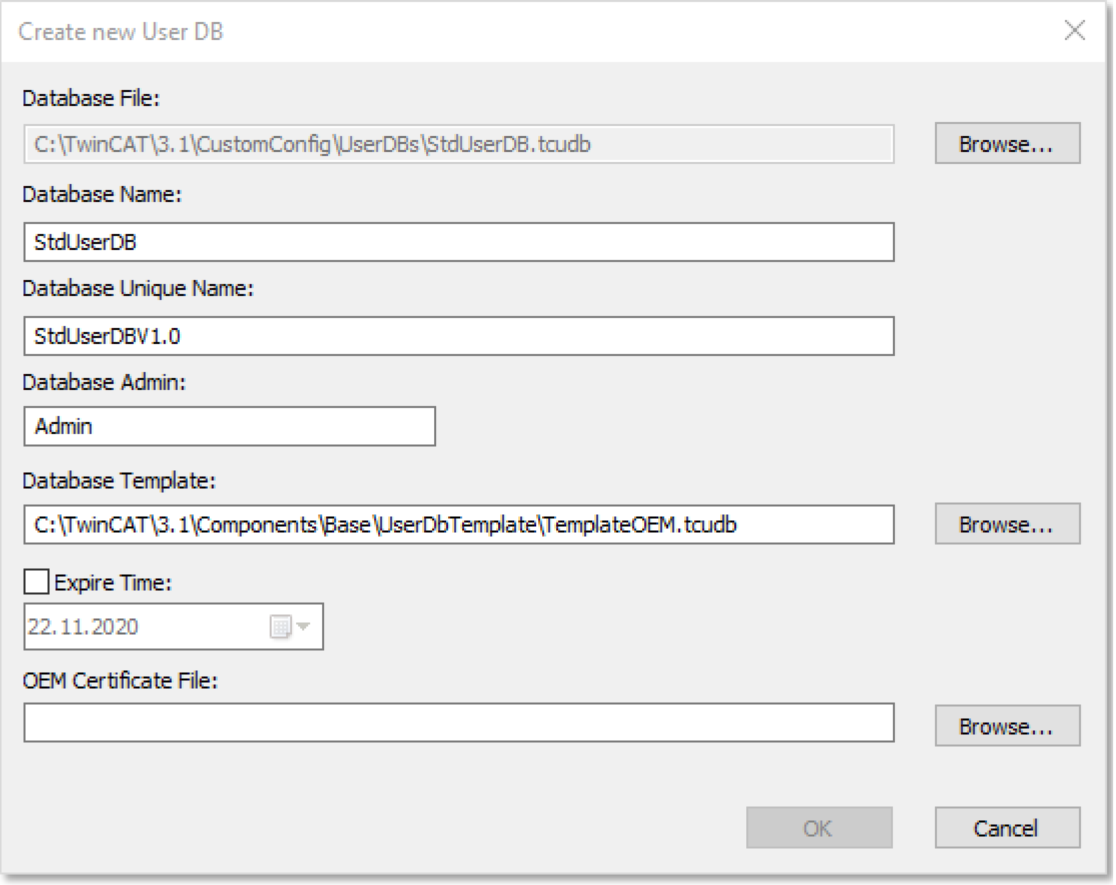
User Database File: xxx - valid - C:\TwinCAT\3.1\CustomConfig\UserDBs\xxx.tcudb

Active User: Guest

Database Admins: Admin

Validity Period – Issue Date: 10.09.2018

⇨ The **Create new User DB** dialog opens.



Create new User DB

Database File: C:\TwinCAT\3.1\CustomConfig\UserDBs\StdUserDB.tcudb

Database Name: StdUserDB

Database Unique Name: StdUserDBV1.0

Database Admin: Admin

Database Template: C:\TwinCAT\3.1\Components\Base\UserDbTemplate\TemplateOEM.tcudb

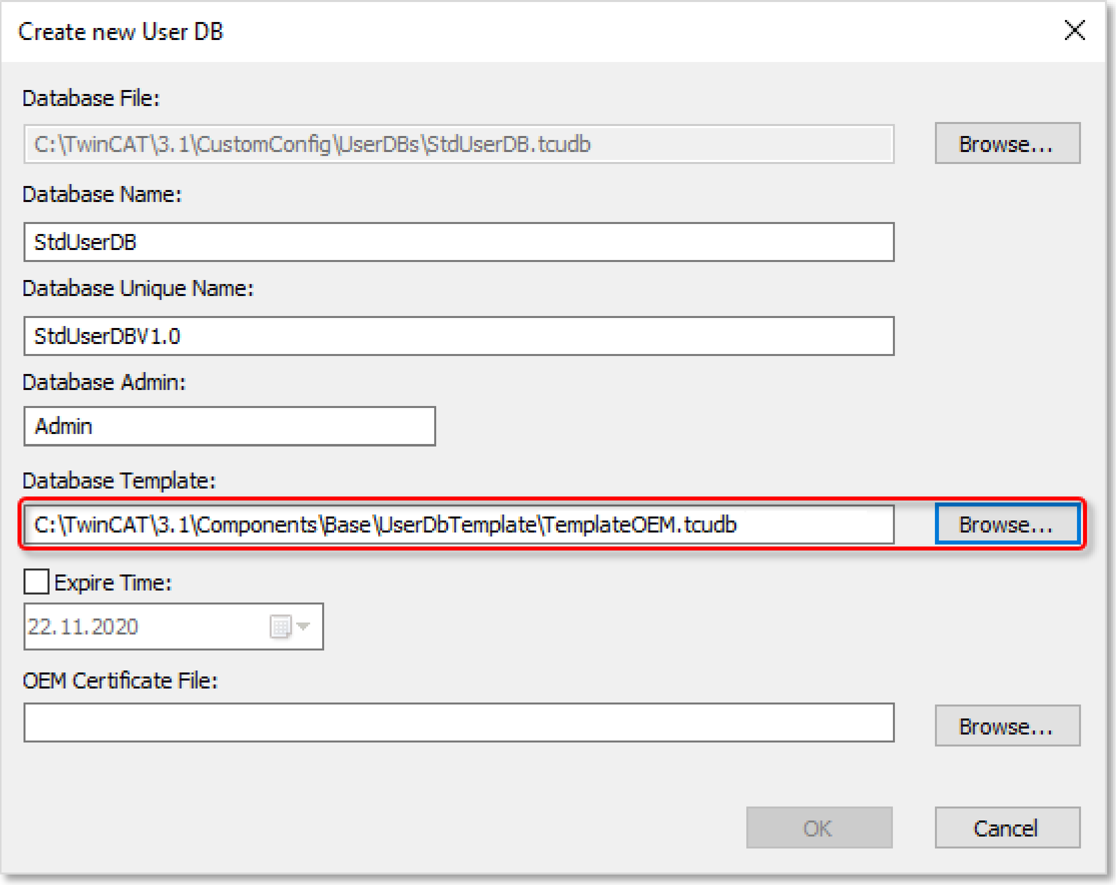
□ Expire Time: 22.11.2020

OEM Certificate File:

3. Enter a name for the database (**Database Name**). This name is used in the program to display the selected database.
4. Specify a **Database Unique Name** (for example, with a version number) that enables the unambiguous identification of that database (version) within your organization.
5. Enter a name for the administrator of the database. The **Database Admin** created here is used only to sign the database and cannot be used to log in or to make changes to the database. To make changes to the database, at least one database user must be a member of the administrator group.
6. Define the template for the new database.
   You should use the *TemplateOEM.tcudb* template as an easy basis. If your TwinCAT version doesn't contain the template yet, you can download it here: https://infosys.beckhoff.com/content/1033/tc3_security_management/Resources/5943612299/.zip.

   To select a different template, click **Browse...** next to the **Database Template** box and select the file you want from the Explorer window.

⇨ The template is displayed in the **Database Template** box.



Create new User DB

Database File: C:\TwinCAT\3.1\CustomConfig\UserDBs\StdUserDB.tcudb

Database Name: StdUserDB

Database Unique Name: StdUserDBV1.0

Database Admin: Admin

Database Template: C:\TwinCAT\3.1\Components\Base\UserDbTemplate\TemplateOEM.tcudb

□ Expire Time: 22.11.2020

OEM Certificate File:

Notice You can also create your own templates for a database, for example, based on a database that you have already created.

7. The database created must initially be signed with a valid OEM certificate. Data from the OEM certificate is also used to generate the User DB key, which unambiguously identifies the individual database.

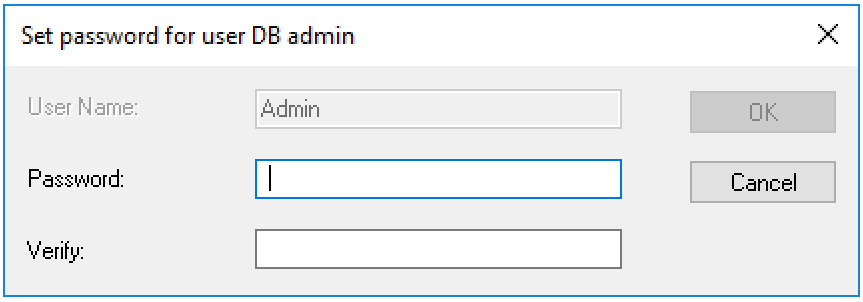
   If the desired certificate is not set in the **OEM Certificate File** box, select the OEM certificate by clicking on **Browse...**

   The standard directory for the OEM certificate is: *c:\twincat\3.1\customconfig\certificates*.

   ⇨ The certificate is displayed in the **OEM Certificate File** box.

8. Click on **OK**.

   ⇨ You will now be prompted in a dialog to specify a password for the (signing) administrator of the database.



Set password for user DB admin

User Name: Admin

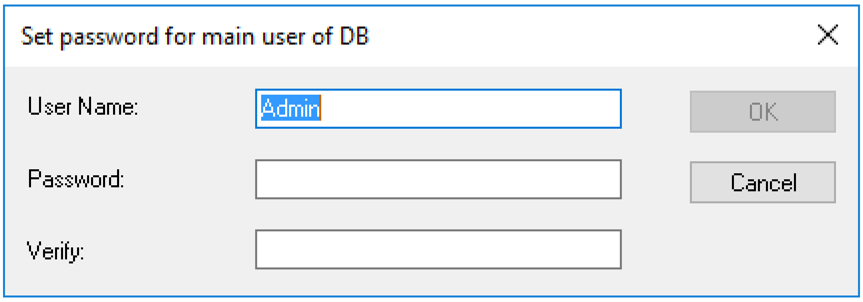
Password:

Verify:

9. Enter a password and confirm it by entering it again. Be sure to use a strong password, as otherwise the database will be easy to attack!

10. Click on **OK**.

11. Build 4024 only: You will now be prompted to create the second (content-managing) administrator of the database:



Set password for main user of DB

User Name: Admin

Password:

Verify:

You can provide them with the same user name and password as the signing administrator. This makes it easier to manage the database. The user name of the previously created signing administrator is therefore suggested here as the default value.

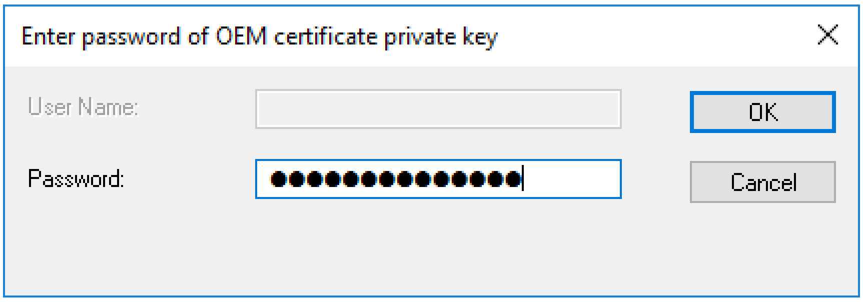
However, you can also separate the functions of content management (= this administrator) and release of changes (= signing administrator) if you wish.

Notice **You can create additional administrators or make changes later, after the database has been created.**
Notice **You don't have to create any more users if you only need a single user who is allowed to do everything. This is the case, for example, if you simply want to encrypt your project and make no other distinction in access rights.**

12. Click on **OK**.

⇨ The database is saved. In a dialog you will be requested to enter the password for the OEM Private Key, with which the database has to be signed in order to be used.

13. Enter the password of the OEM certificate and confirm the dialog with **OK**.



Enter password of OEM certificate private key

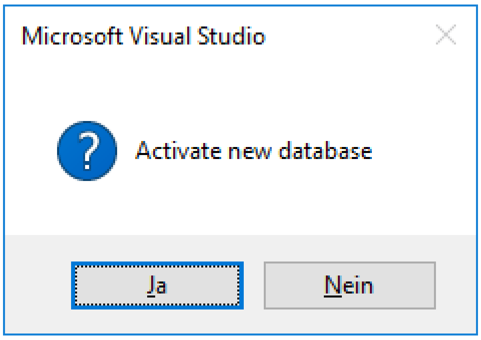
User Name:

Password: ●●●●●●●●●●●●●●

Note: From now on you no longer need the OEM certificate when you work with this database (for example, to make changes to the content).

⇨ A further dialog opens with the question whether the database should also be set as the current database in Visual Studio ("activated").

14. If so, confirm the dialog with **OK**.



Microsoft Visual Studio

Activate new database

⇨ This sets the new database as the **current** database in Visual Studio.

The currently set database is used for the (new) connection of a project to a database.

The database assigned to a project is saved in the project (file name and User DB key).

The database's location is C:\TwinCAT\3.1\CustomConfig\UserDBs.

If you want to define this (or another) database as the default database (which is to be used by default when starting Visual Studio), set it on the **Database** tab of the configuration window. The procedure is described in the next chapter.

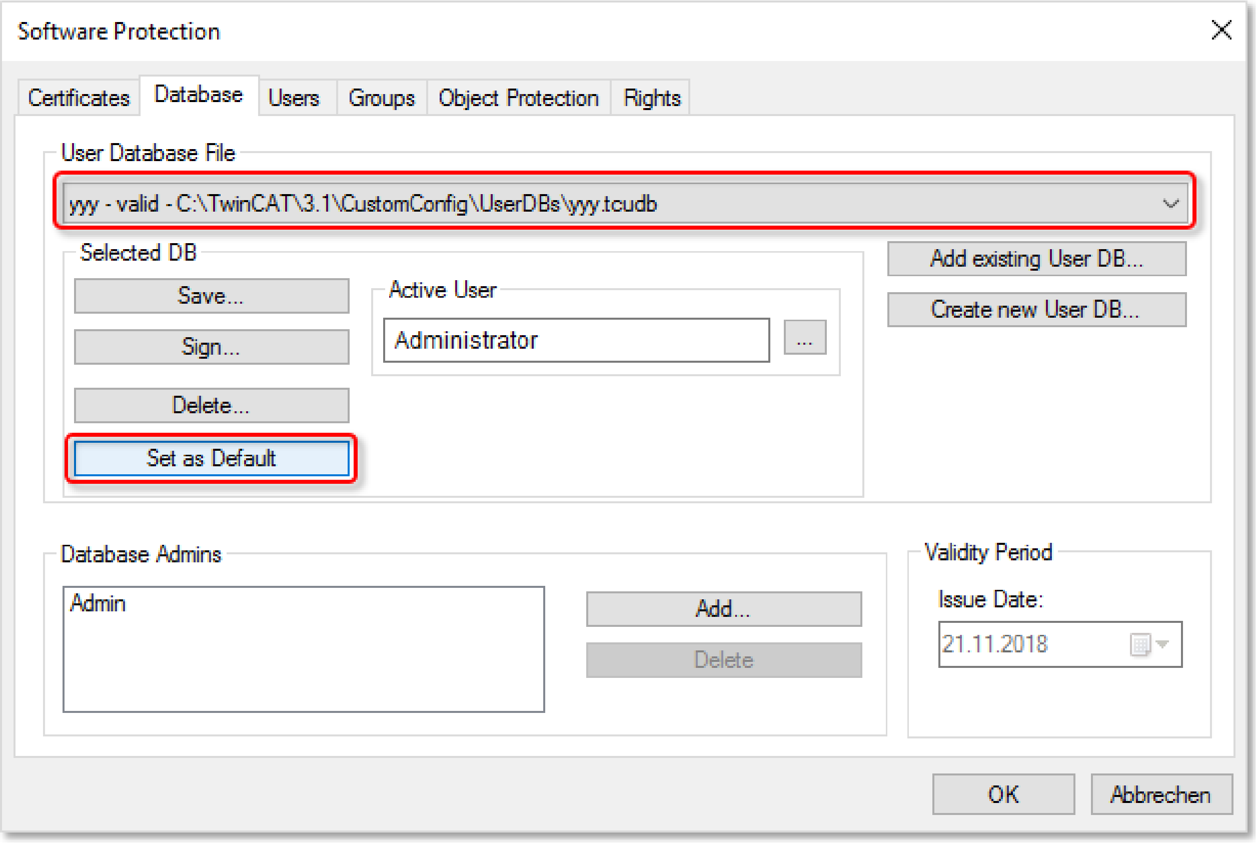
5.2 Setting default settings for the user database in Visual Studio

Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

Specifying the default settings when Visual Studio starts

If you want to set a database as the default database (which is to be used by default when starting Visual Studio), select the desired database on the **Database** tab of the Software Protection configuration window and click **Set as Default**.



User Database File: yyy - valid - C:\TwinCAT\3.1\CustomConfig\UserDBs\yyy.tcudb

Active User: Administrator

Database Admins: Admin

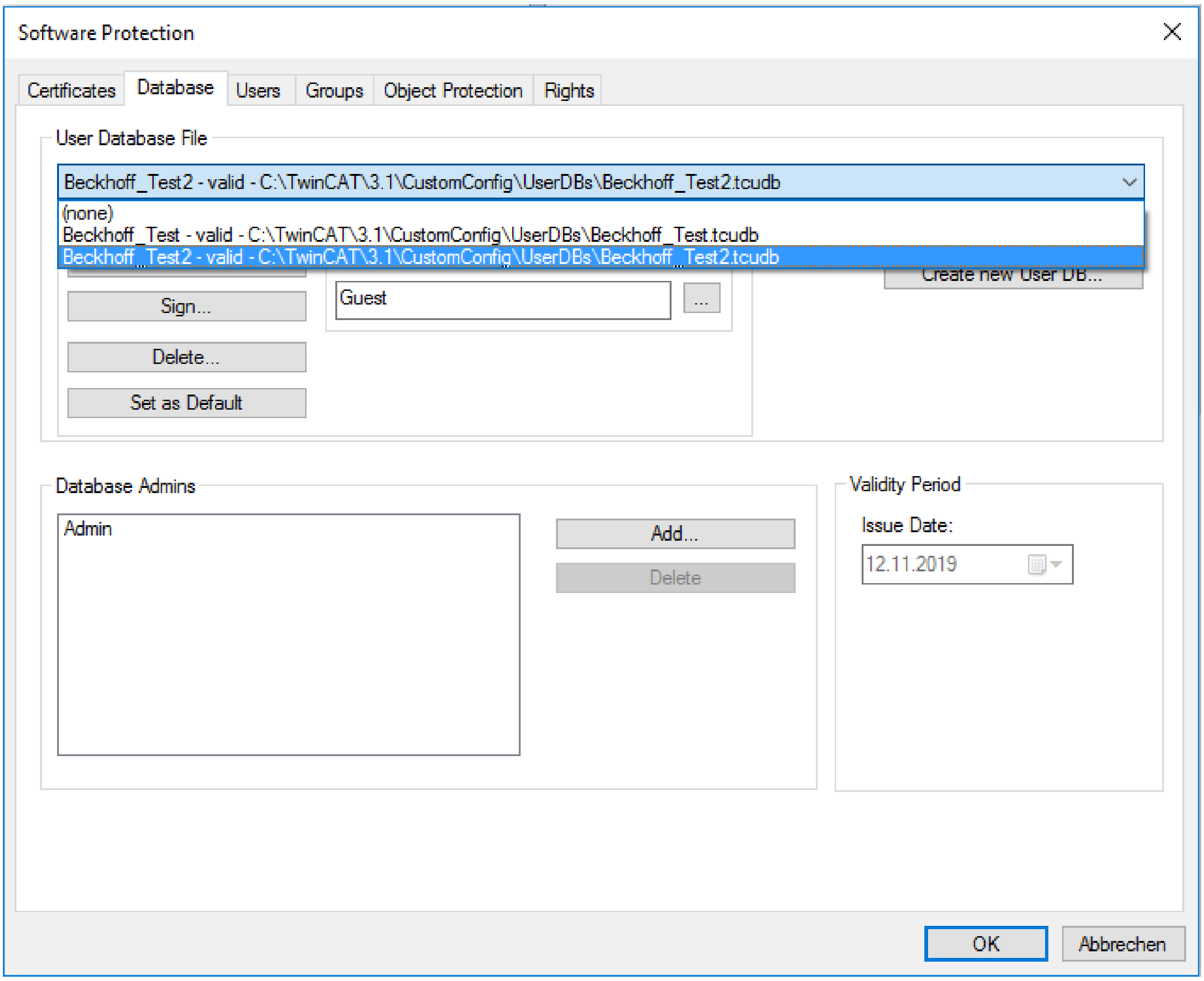
Issue Date: 21.11.2018

You can also specify which user should be automatically active after Visual Studio starts. You specify this in the **Active User** text box.

5.3 Select the current user database in Visual Studio

Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.



Software Protection

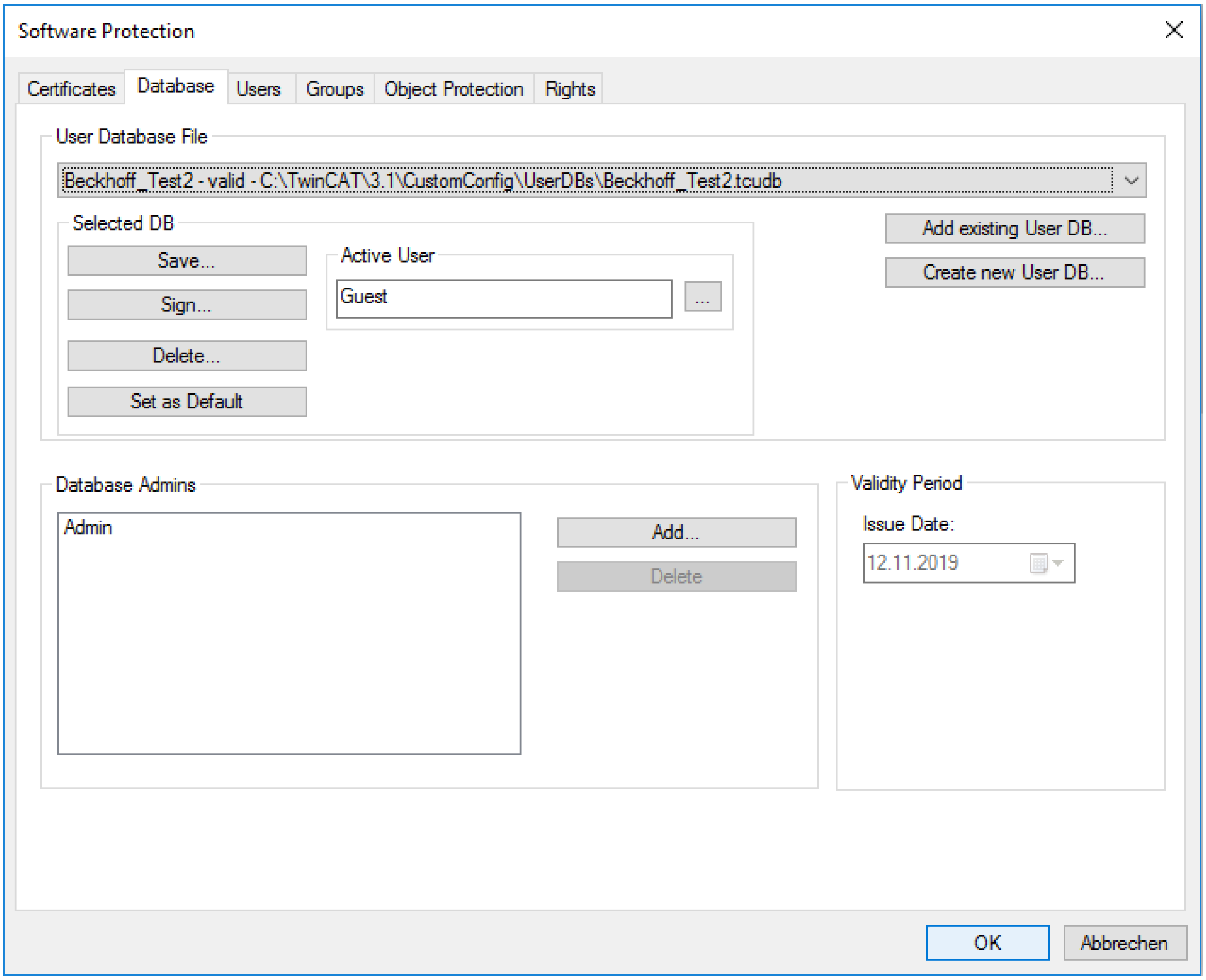
User Database File: Beckhoff_Test2 - valid - C:\TwinCAT\3.1\CustomConfig\UserDBs\Beckhoff_Test2.tcudb

(none)
Beckhoff_Test - valid - C:\TwinCAT\3.1\CustomConfig\UserDBs\Beckhoff_Test.tcudb
Beckhoff_Test2 - valid - C:\TwinCAT\3.1\CustomConfig\UserDBs\Beckhoff_Test2.tcudb

Guest

Database Admins: Admin

Validity Period – Issue Date: 12.11.2019



User Database File: Beckhoff_Test2 - valid - C:\TwinCAT\3.1\CustomConfig\UserDBs\Beckhoff_Test2.tcudb

Active User: Guest

Database Admins: Admin

Issue Date: 12.11.2019

5.4 Default user in the user database

After creating the user database using the template "TemplateOEM", it contains these two default users:

- Guest (may not do anything)
- Administrator* (may do everything)
  * The administrator's name is set when creating the user database [▶ 33].

If it is the simplest use case – there is one user who is allowed to do everything (administrator) and one user who is not allowed to anything (guest), then there is no need to create more users and you can use the newly-created user database without further configuration. In this case, you can continue right here: Linking the user database to the project [▶ 69]

If you want to cover more complex access scenarios, you can expand the user database as needed. This is described in the chapter Expanding the user database [▶ 49].

5.5 Extensions for user databases

Requirement: TwinCAT 3 Build 4024.8 or higher

The functions described below require at least TwinCAT 3 Build 4024.8.

Directory for storing user databases

User databases must be stored in the following directory in order to be used in the TwinCAT Engineering: C:\TwinCAT\3.1\CustomConfig\UserDBs

Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

Introduction

From Build 4024.8 onwards, the TwinCAT Software Protection supports extension files for the user database, so-called "User DB Extensions".

- A user database can be extended with these "extensions".
- An extension is an additional XML file that is the same as the main user database in terms of structure, but can only be used together with the main user database. An extension is therefore not usable on its own without the main user database.
- An extension can contain only the definition of users, but not the definition of groups or Object Protection Levels.
- An extension does not have its own administrator. The signing administrator of the main user database is also the signing administrator of the associated extension.
- An existing extension file can be added or removed at file level. This does not require configuration in the associated main user database (i.e. administrator rights are not required).
- The extensions are placed in a subdirectory with the name of the main user database (below the directory that contains the main user database): C:\TwinCAT\3.1\CustomConfig\UserDBs\<UserDB name>.
- An extension is usually limited in time.
- A user database can have any number of extensions.

Notice **A secure time limitation requires a tamper-proof time reference.**

Application:

- The main user database stores static information (such as definitions of groups or Object Protection Levels).
- The extension stores time-limited information (users), such as for service purposes.

A user database can easily be replaced by another version (with the same name and user DB key) at file level. To make changes in the user database tamper-proof (protection against being replaced by an older version without the changes), a completely new user database (with a different User DB key) would have to be created and linked to the project again. However, this is often not feasible in practice. This can be solved easily and elegantly with extensions of the user database:

- The main user database (with static information such as definitions of groups or Object Protection Levels) is permanently linked to the project.
- The (time-limited) extension contains all the information (users) that could change over time.

Notice **In simple scenarios (few users) this could also be solved with a time-limited user database (without the use of extensions). However, for more complex scenarios, especially in the service area, this is not a practical solution.**

Scenarios with different user groups / Object Protection Levels are simpler to realize with extensions. For example, in-house developers can be summarized in their own extension, which is simply not copied to the target system during delivery. This allows areas (or individual users) to be added or removed as needed without having to adapt the entire user database.

This also enables a significant simplification of the versioning of a user database.

Sample application from the service area

- The Main User DB contains only the most necessary information (signing and changing administrators, definition of OPLs and groups).
- The extension is created specifically for the service assignment and contains only the service employee as the user. The extension is time-limited for the period of the service assignment.
- The service employee brings the extension with him on the service notebook (or a USB stick).

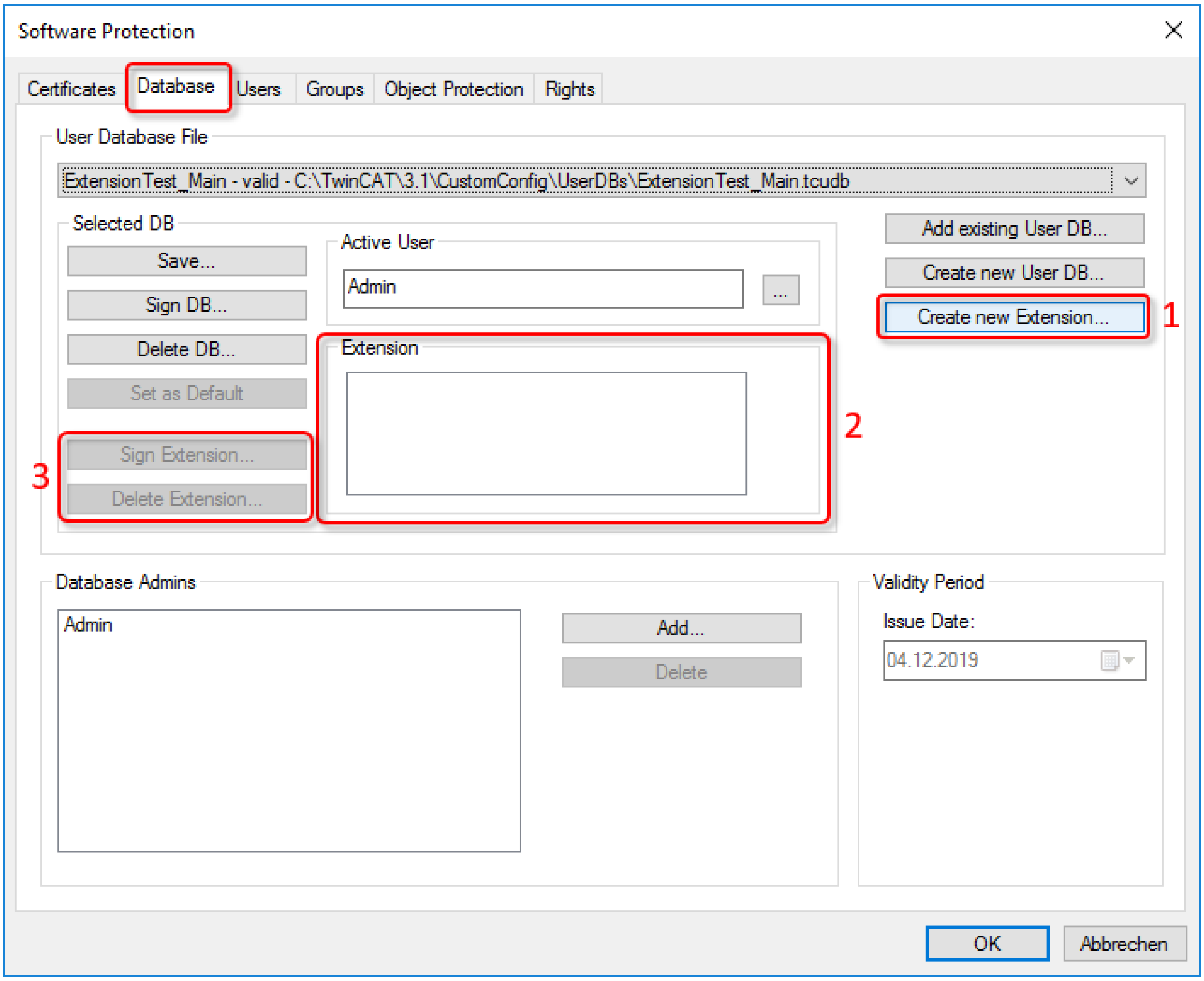
Time limitation of a user database / extension

A tamper-proof time limitation requires a tamper-proof time reference!

5.5.1 Associated elements in the Software Protection configuration console

For the management of extensions and the users defined in them, the elements described below are available in the user interface of the Software Protection configuration console.

Database tab :



User Database File: ExtensionTest_Main - valid - C:\TwinCAT\3.1\CustomConfig\UserDBs\ExtensionTest_Main.tcudb

Active User: Admin

Database Admins: Admin

Validity Period

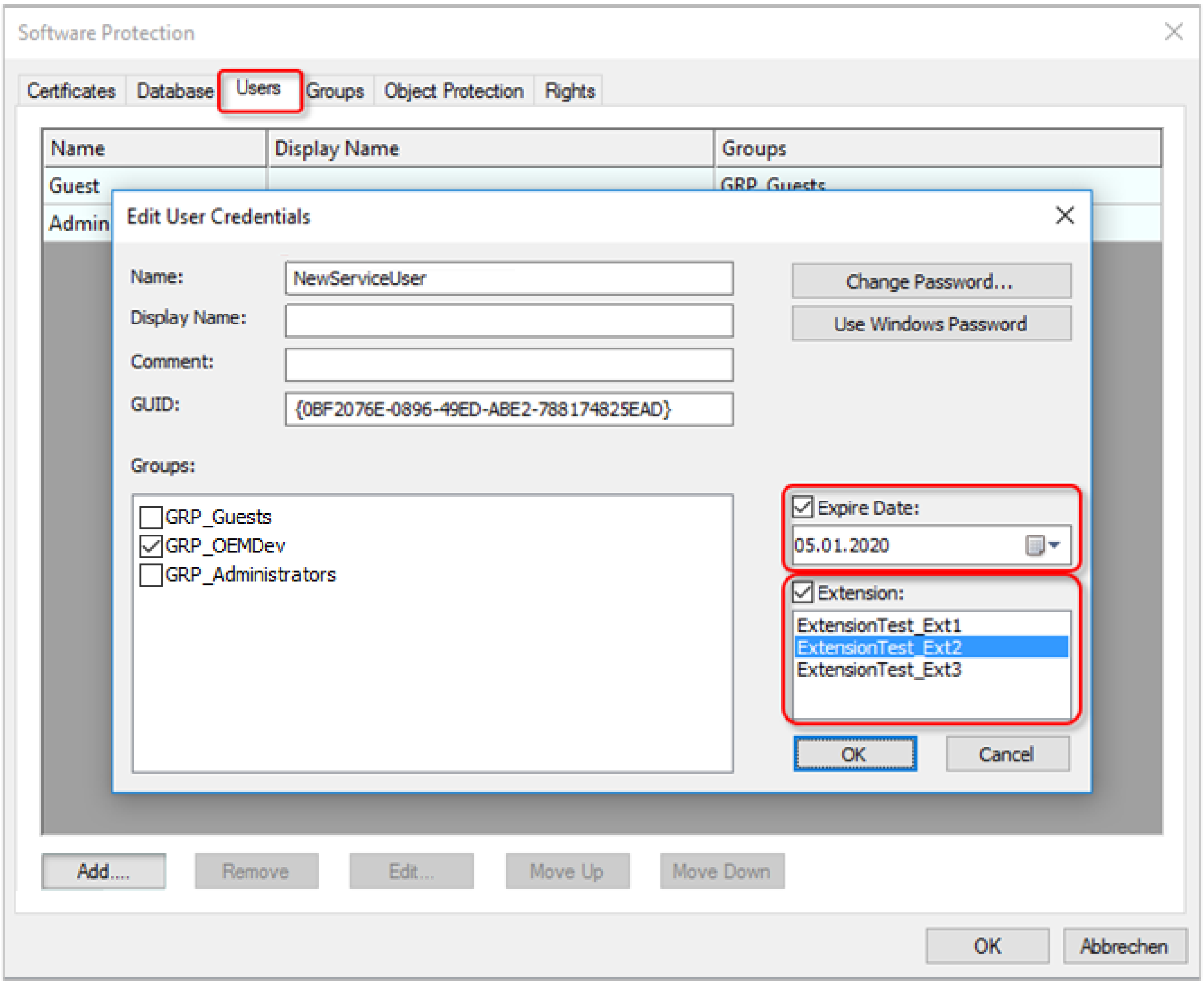
Issue Date: 04.12.2019

1: Creating a new extension for the currently selected User DB

2: List of existing extensions

3: Signing or deleting the extension selected in (2)

Users tab:



Name: NewServiceUser

Display Name:

Comment:

GUID: {0BF2076E-0896-49ED-ABE2-788174825EAD}

Groups:

- □ GRP_Guests
- ☑ GRP_OEMDev
- □ GRP_Administrators

☑ Expire Date: 05.01.2020

☑ Extension:

- ExtensionTest_Ext1
- ExtensionTest_Ext2
- ExtensionTest_Ext3

A user account can be assigned to an existing extension and an expiration date for the account can be set.

5.5.2 Manual procedure in the TwinCAT 3 Engineering

Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

5.5.2.1 Creating Extensions

Allow operating system access for authorized users only

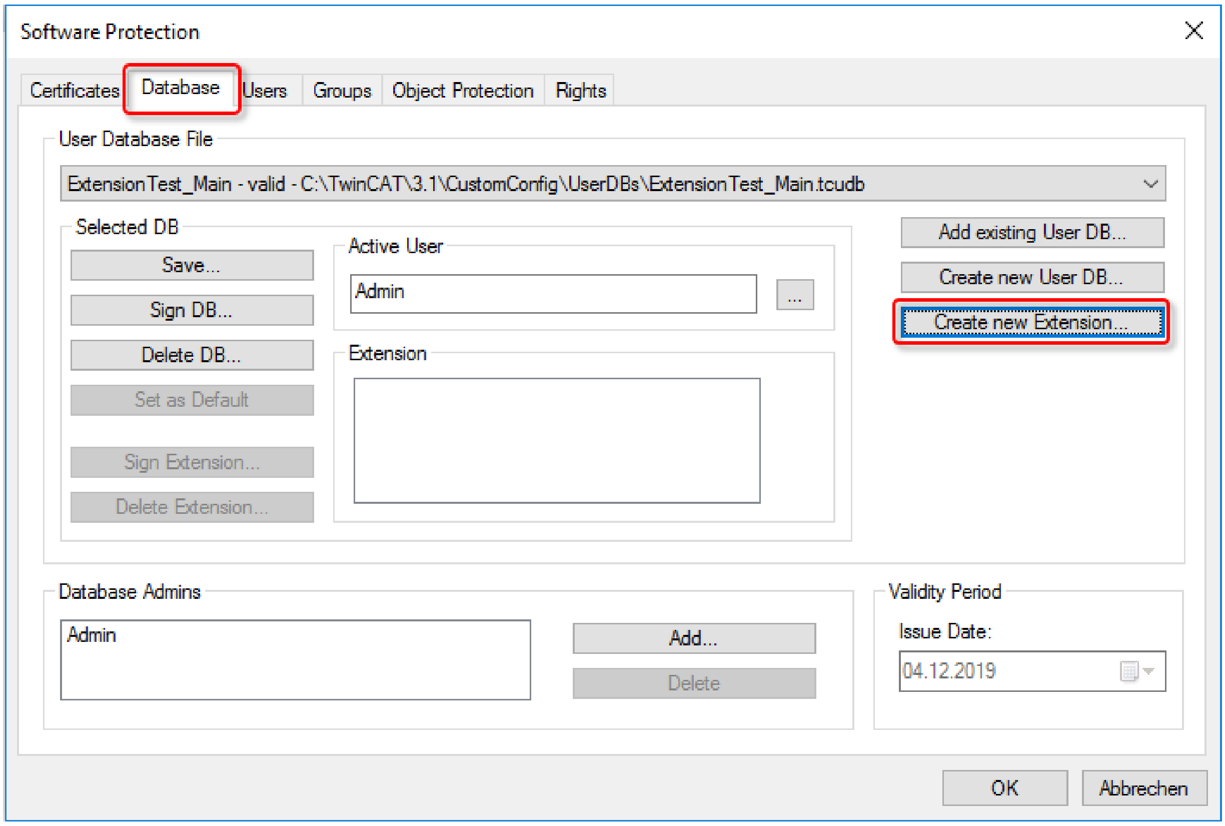
The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

Changes in an extension must be signed

The changes in an extension (including the initial creation of the extension) must be signed by the signing administrator, otherwise the extension will be invalid.

Creating a new extension

✓ The current user must have (editing) administrator rights!



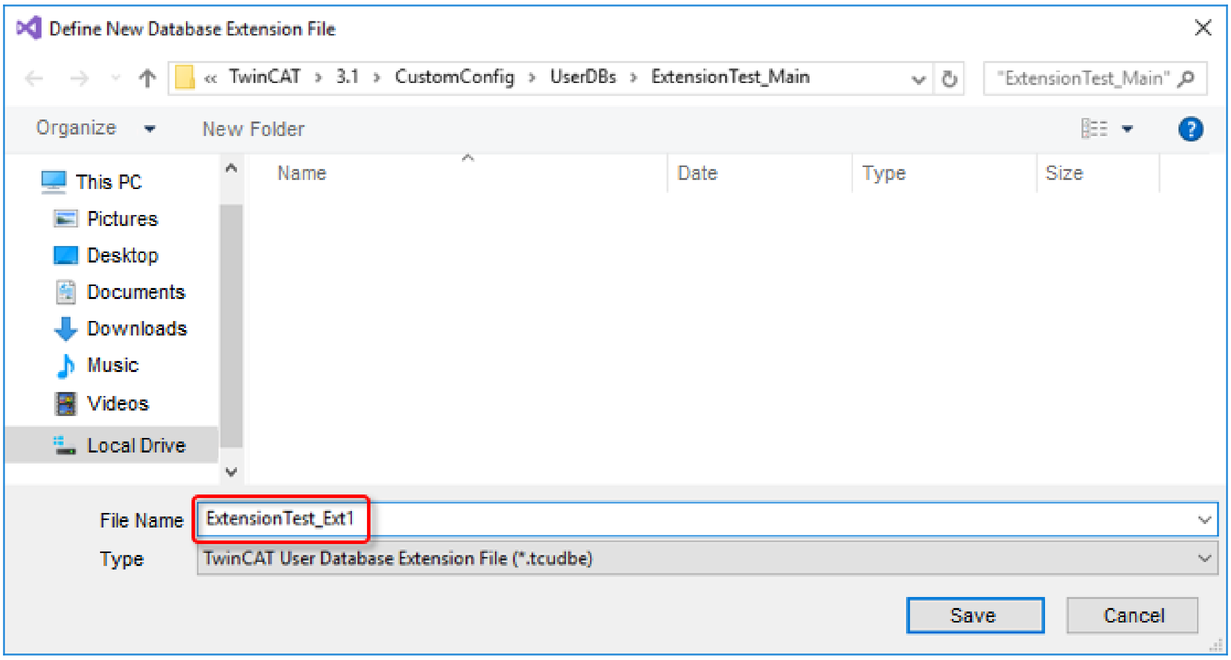
User Database File: ExtensionTest_Main - valid - C:\TwinCAT\3.1\CustomConfig\UserDBs\ExtensionTest_Main.tcudb

Active User: Admin

Database Admins: Admin

Issue Date: 04.12.2019

1. Clicking on **Create new Extension** opens a dialog for creating a new extension:



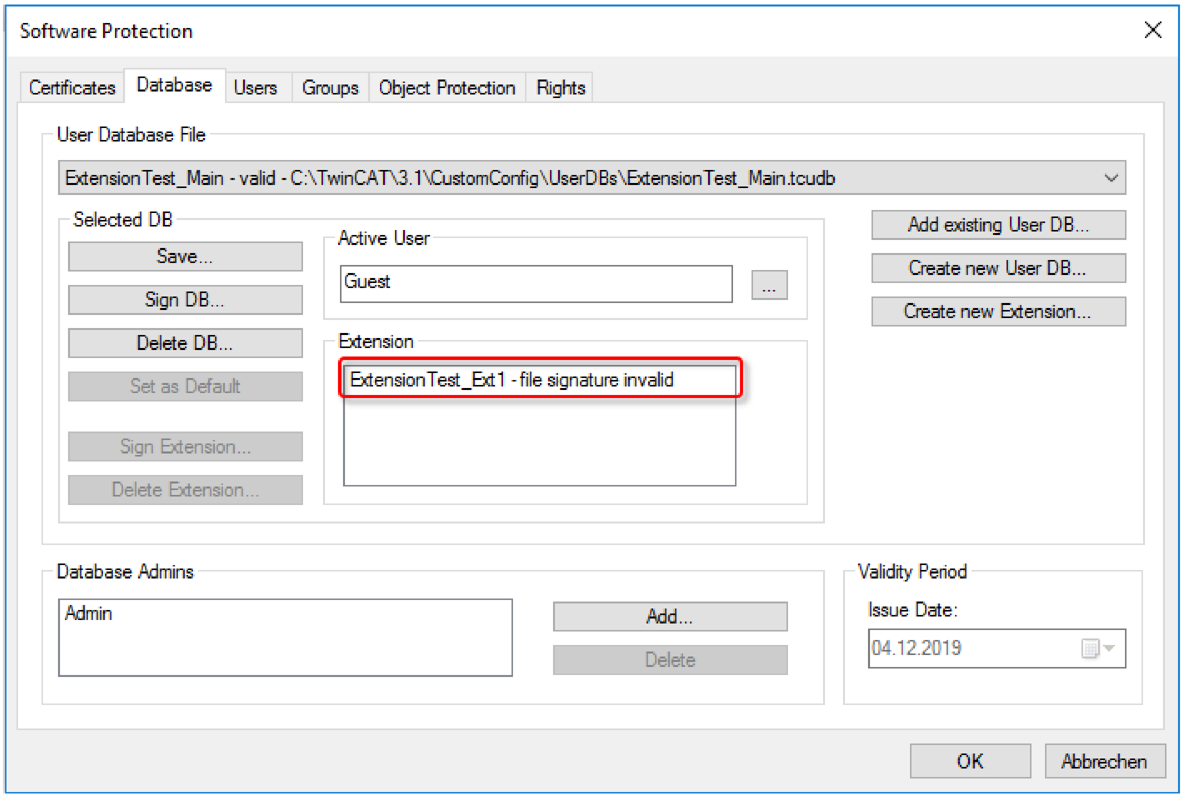
« TwinCAT › 3.1 › CustomConfig › UserDBs › ExtensionTest_Main

File Name: ExtensionTest_Ext1

Type: TwinCAT User Database Extension File (*.tcudbe)

Notice The specified directory may not be changed!

⇨ The new extension has now been created, but initially with the status "invalid", because it has not yet been signed:



User Database File: ExtensionTest_Main - valid - C:\TwinCAT\3.1\CustomConfig\UserDBs\ExtensionTest_Main.tcudb

Active User: Guest

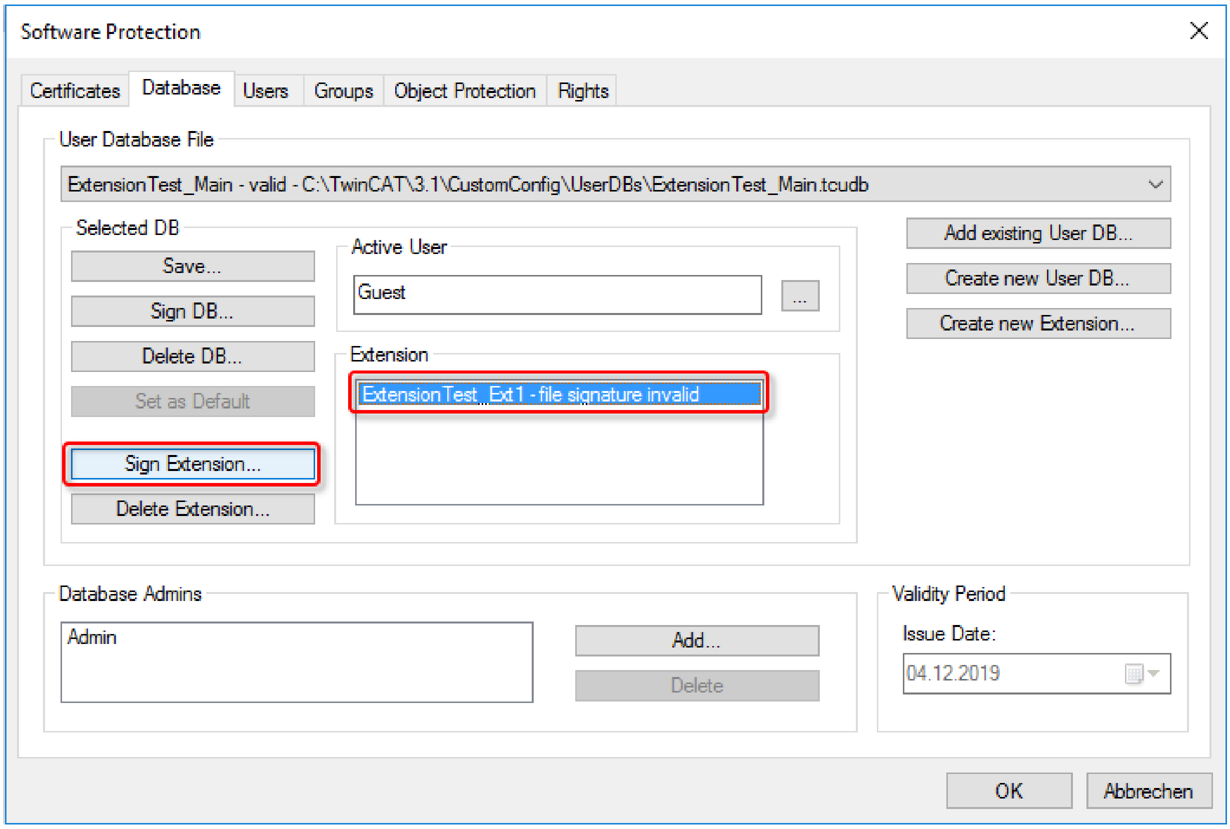
Extension: ExtensionTest_Ext1 - file signature invalid

Database Admins: Admin

Validity Period

Issue Date: 04.12.2019

1. To do this, the extension is selected on the **Database** tab in the list of extensions by clicking it, and it can then be signed by the signing administrator (the Main User DB) ...



User Database File: ExtensionTest_Main - valid - C:\TwinCAT\3.1\CustomConfig\UserDBs\ExtensionTest_Main.tcudb

Active User: Guest

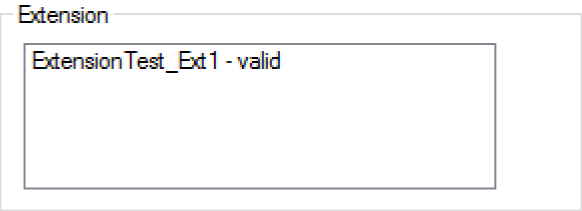
Extension: ExtensionTest_Ext1 - file signature invalid

Database Admins: Admin

Validity Period

Issue Date: 04.12.2019

2. ... and thus validated:



Extension

ExtensionTest_Ext1 - valid

The extension is now created as an empty envelope and must be filled with content (users).

5.5.2.2 Creating users in extensions

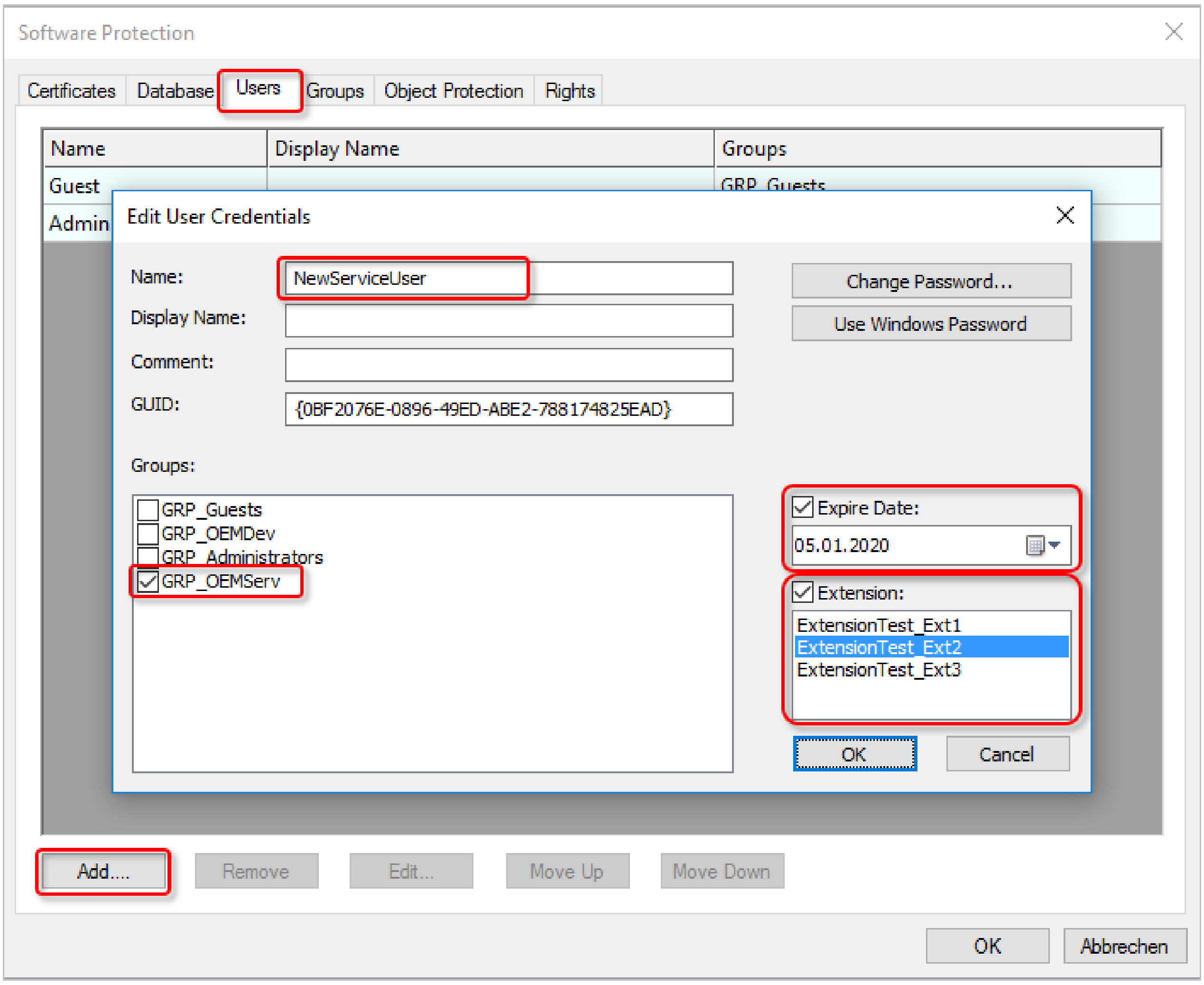
Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

Changes in an extension must be signed

The changes in an extension (including the initial creation of the extension) must be signed by the signing administrator, otherwise the extension will be invalid.

Example: On the **Users** tab, a new user ("NewServiceUser") is created, assigned to the group "GRP_OEMServ" as well as to an extension and the desired time limit is set:



Name: NewServiceUser

Display Name:

Comment:

GUID: {0BF2076E-0896-49ED-ABE2-788174825EAD}

Groups:

□ GRP_Guests
□ GRP_OEMDev
□ GRP_Administrators
☑ GRP_OEMServ

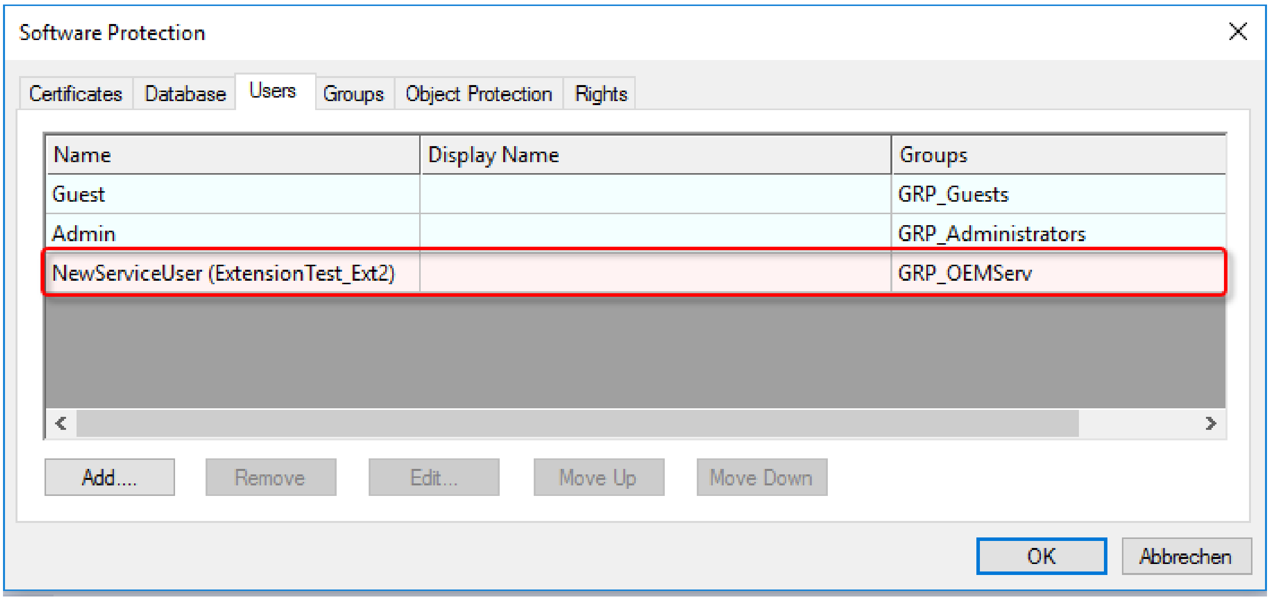
☑ Expire Date: 05.01.2020

☑ Extension:

ExtensionTest_Ext1
ExtensionTest_Ext2
ExtensionTest_Ext3

The new user is thus created in the selected extension.

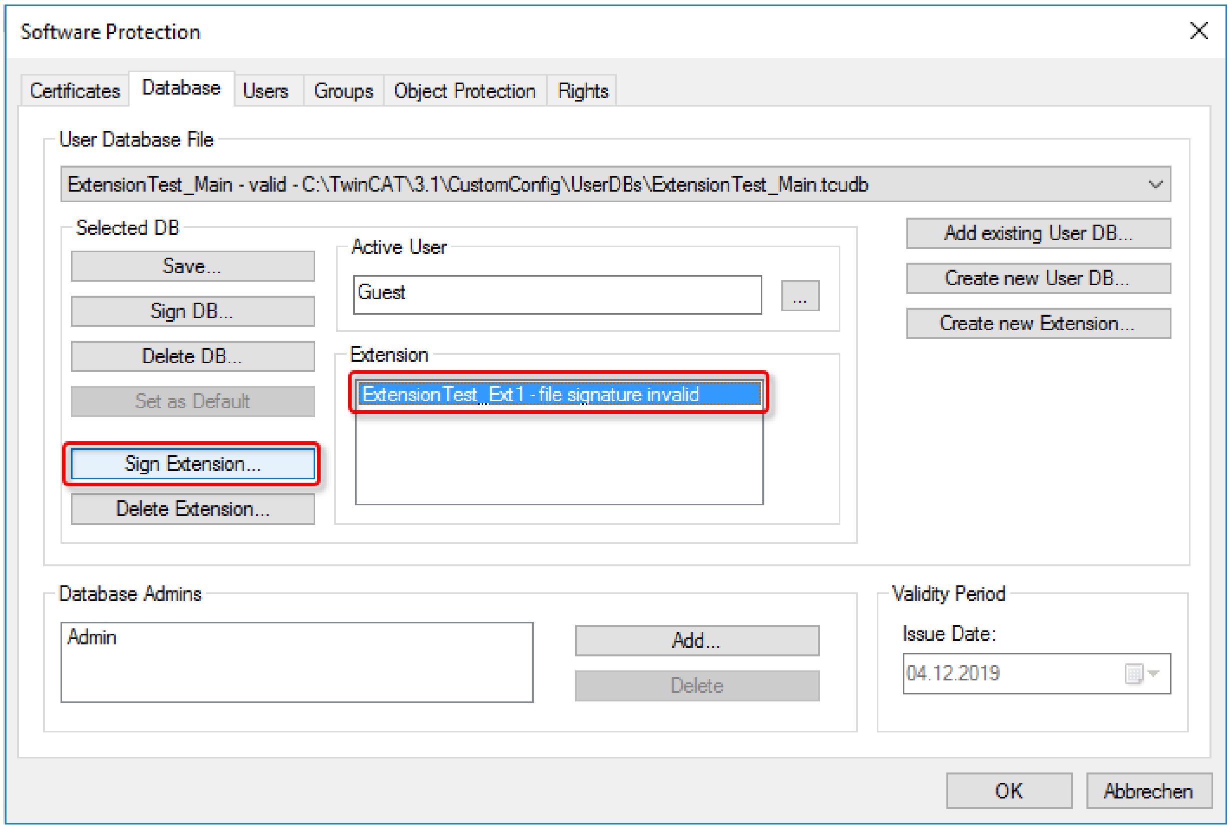
The user's affiliation to an extension is represented in the list of existing user accounts by a different color and the naming of the extension in brackets behind the user name:



Name	Display Name	Groups
Guest		GRP_Guests
Admin		GRP_Administrators
NewServiceUser (ExtensionTest_Ext2)		GRP_OEMServ

Now the changes in the extension still need to be signed in order for them to be valid.

1. To do this, the extension is selected on the **Database** tab in the list of extensions by clicking it, and it can then be signed by the signing administrator (the Main User DB) ...



User Database File: ExtensionTest_Main - valid - C:\TwinCAT\3.1\CustomConfig\UserDBs\ExtensionTest_Main.tcudb

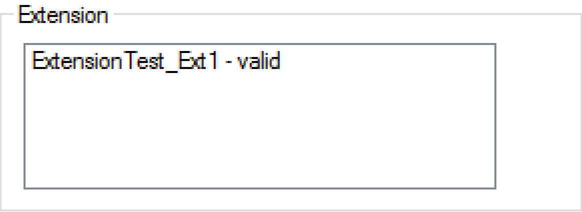
Active User: Guest

Extension: ExtensionTest_Ext1 - file signature invalid

Database Admins: Admin

Issue Date: 04.12.2019

2. ... and thus validated:



Extension

ExtensionTest_Ext1 - valid

5.5.3 Automated procedure

A command line tool is planned for the automated creation of extensions and users in extensions, which can then be integrated into your own infrastructure.

5.6 Expanding the user database

Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

Changes in the database must be signed when saving

The changes in the database must be signed by the signing administrator, otherwise the database will be invalid. Signing is automatically queried during the saving process.

Database template "TemplateOEM"

The "T*emplateOEM*" database template is designed to cover the most common (simple) use cases without having to define your own group rights. These are:

- Two users: One may do everything, the other nothing [▶ 41].
- Adding/changing database administrators [▶ 49]
- Separating the database administrator and developer functions [▶ 53]
- Adding more users with the "Developer" group assignment [▶ 54]

Separate group rights [▶ 56] must be defined for further application cases.

Download link: Planning table for group rights and Object Protection Level

An Excel table for the simple planning of group rights and access rights group sets (Object Protection Level) can be downloaded https://infosys.beckhoff.com/content/1033/tc3_security_management/Resources/8882888971/.zip.

5.6.1 Adding/changing database administrators

Allow operating system access for authorized users only

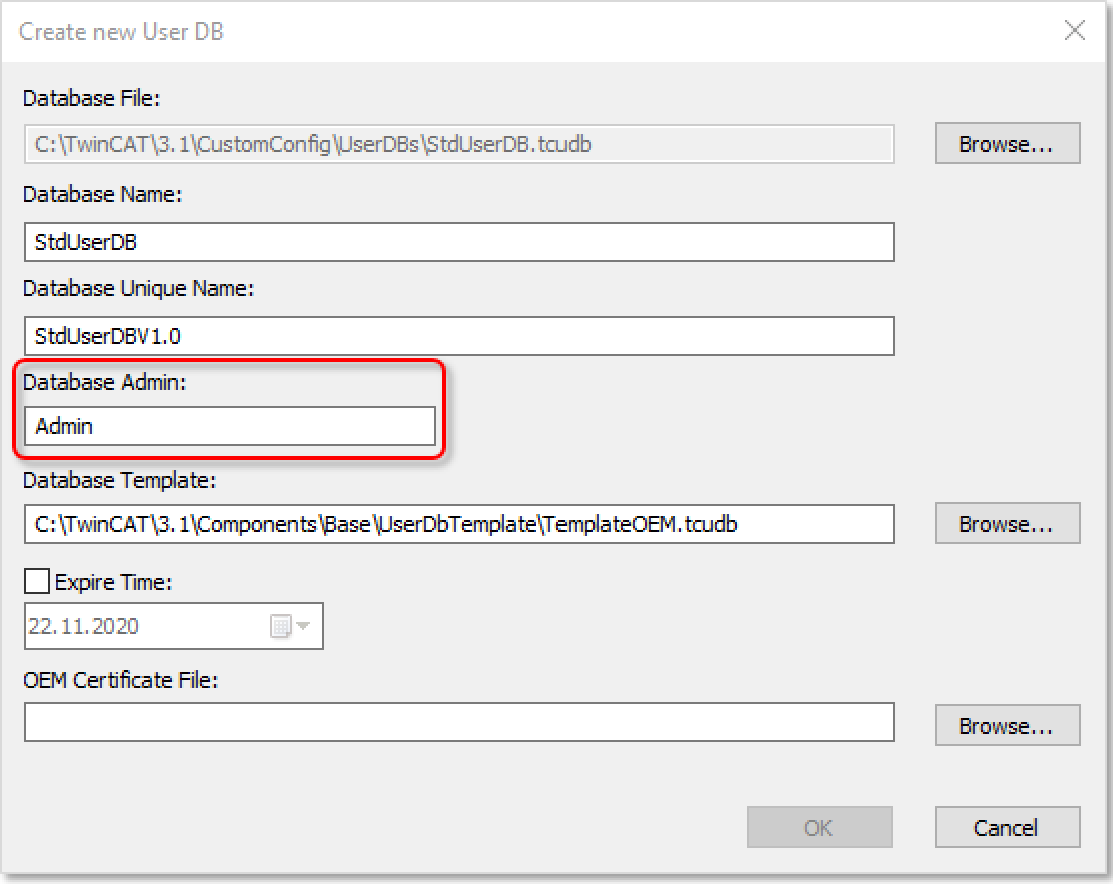
The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

This description is designed for **Build 4024**.

The user database includes two administrators with different task areas:

1. Signing (releasing) changes to the database
2. Changing the contents of the database

The first (signing) administrator is created directly when the user database is created:



Create new User DB

Database File: C:\TwinCAT\3.1\CustomConfig\UserDBs\StdUserDB.tcudb

Database Name: StdUserDB

Database Unique Name: StdUserDBV1.0

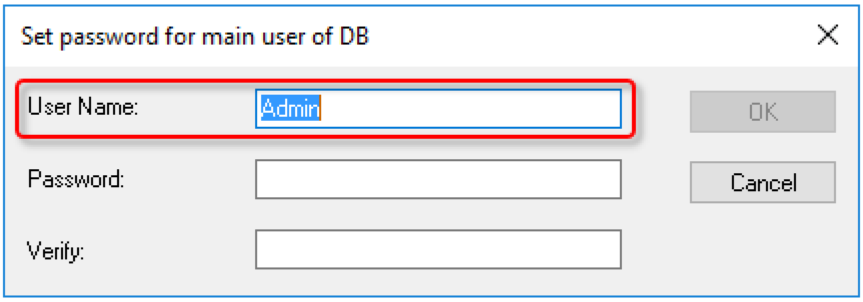
Database Admin: Admin

Database Template: C:\TwinCAT\3.1\Components\Base\UserDbTemplate\TemplateOEM.tcudb

□ Expire Time: 22.11.2020

OEM Certificate File:

Following the creation of the first database administrator, TwinCAT 3 creates the second (editing) administrator as user in the database ("main user") and suggests the name of the first (= signing) administrator as the user name. This allows both administrator functions to be easily combined and created and used with the same username and password if required:



Set password for main user of DB

User Name: Admin

Password:

Verify:

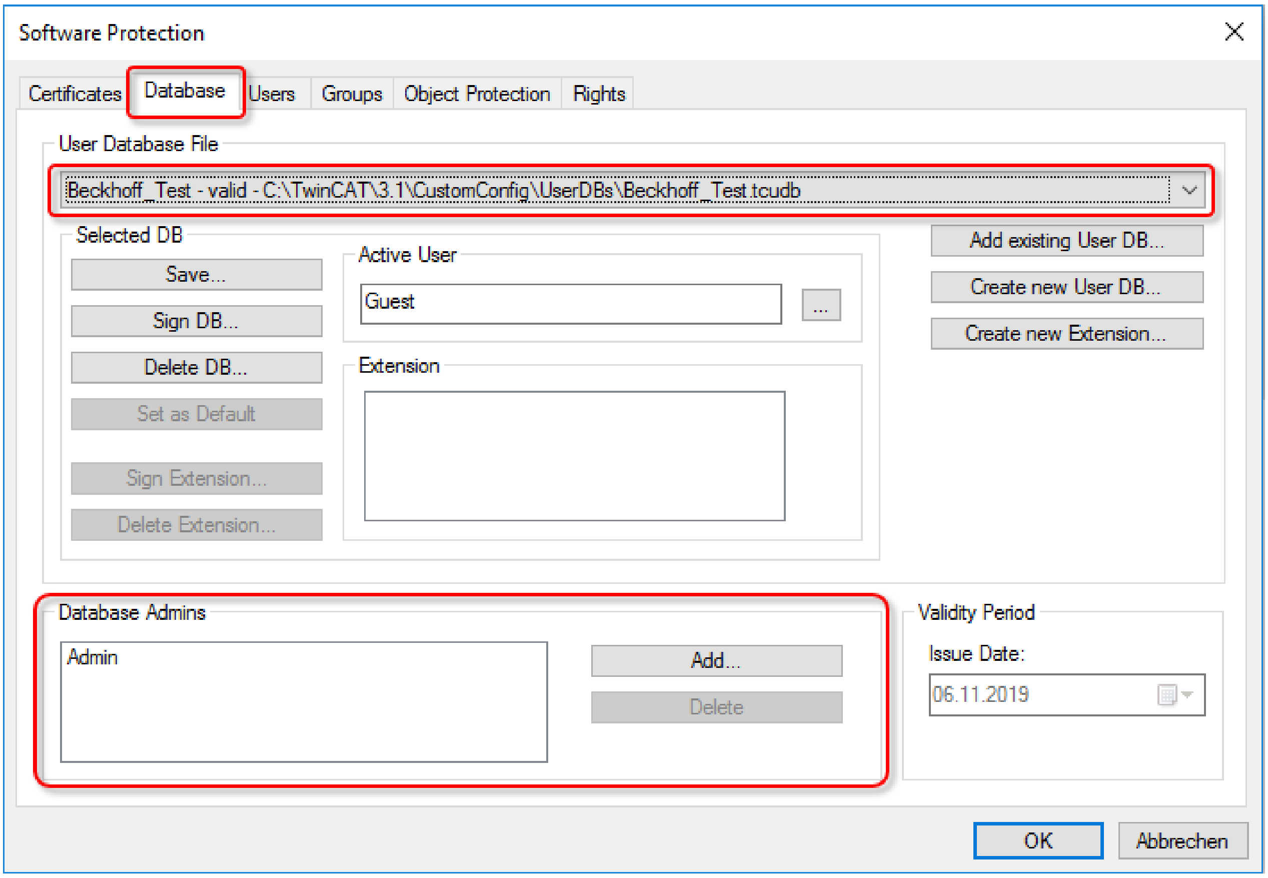
Build 4022:

This input window does not yet exist there. Therefore, following the creation of the user database, the editing database administrator must be created manually as a user and assigned to the "GRP_Administrators" group.

Creating new signing database administrators

Further signing administrators can be created on the **Database** tab in the Software Protection configuration console.

The desired database must be selected; a new administrator can then be created or an existing one deleted in the **Database Admin** window area:



User Database File: Beckhoff_Test - valid - C:\TwinCAT\3.1\CustomConfig\UserDBs\Beckhoff_Test.tcudb

Selected DB

Active User: Guest

Extension

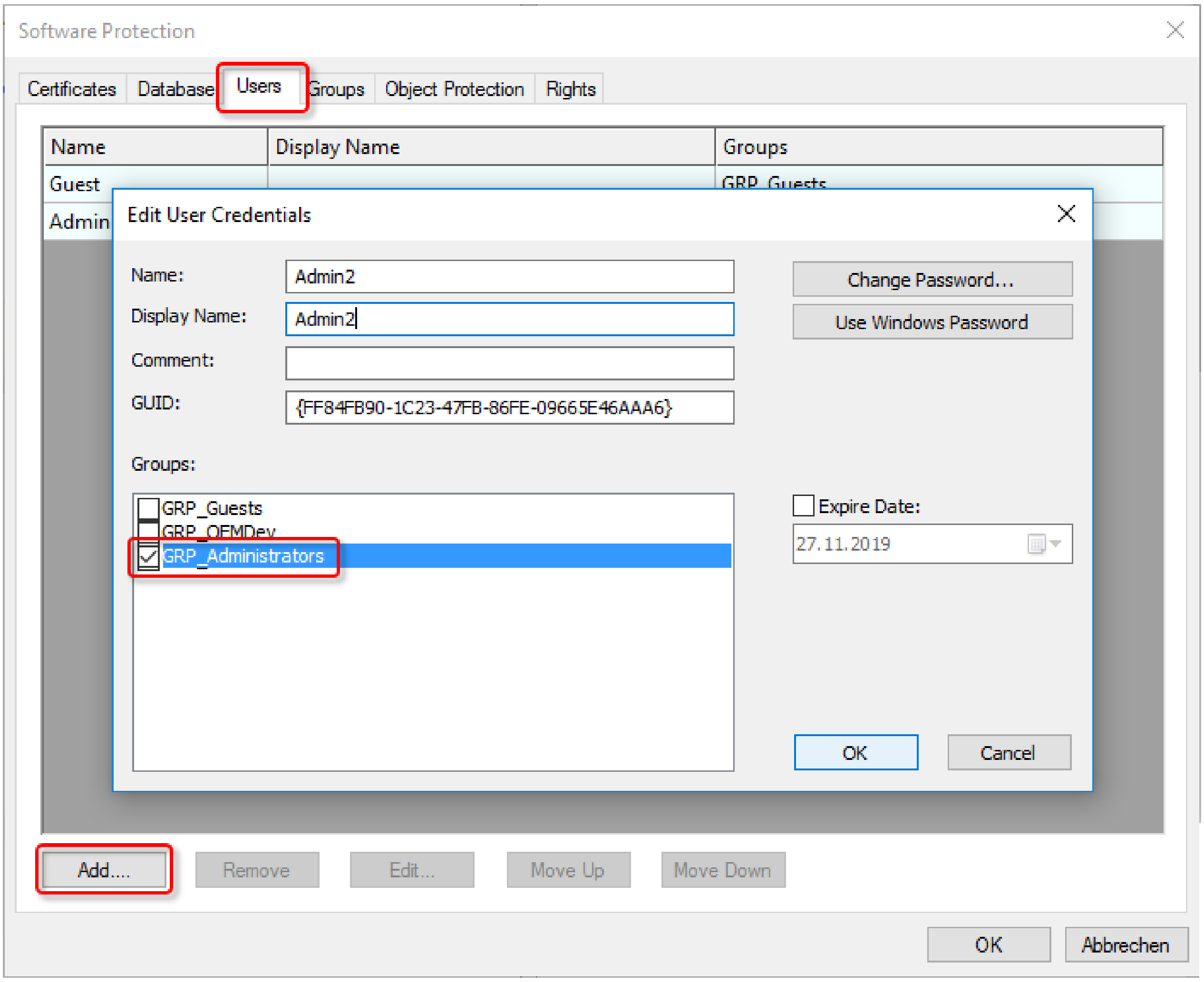
Database Admins: Admin

Validity Period

Issue Date: 06.11.2019

Creating new editing database administrators

Further editing administrators can be created on the **Users** tab in the Software Protection configuration console:



Name: Admin2

Display Name: Admin2

Comment:

GUID: {FF84FB90-1C23-47FB-86FE-09665E46AAA6}

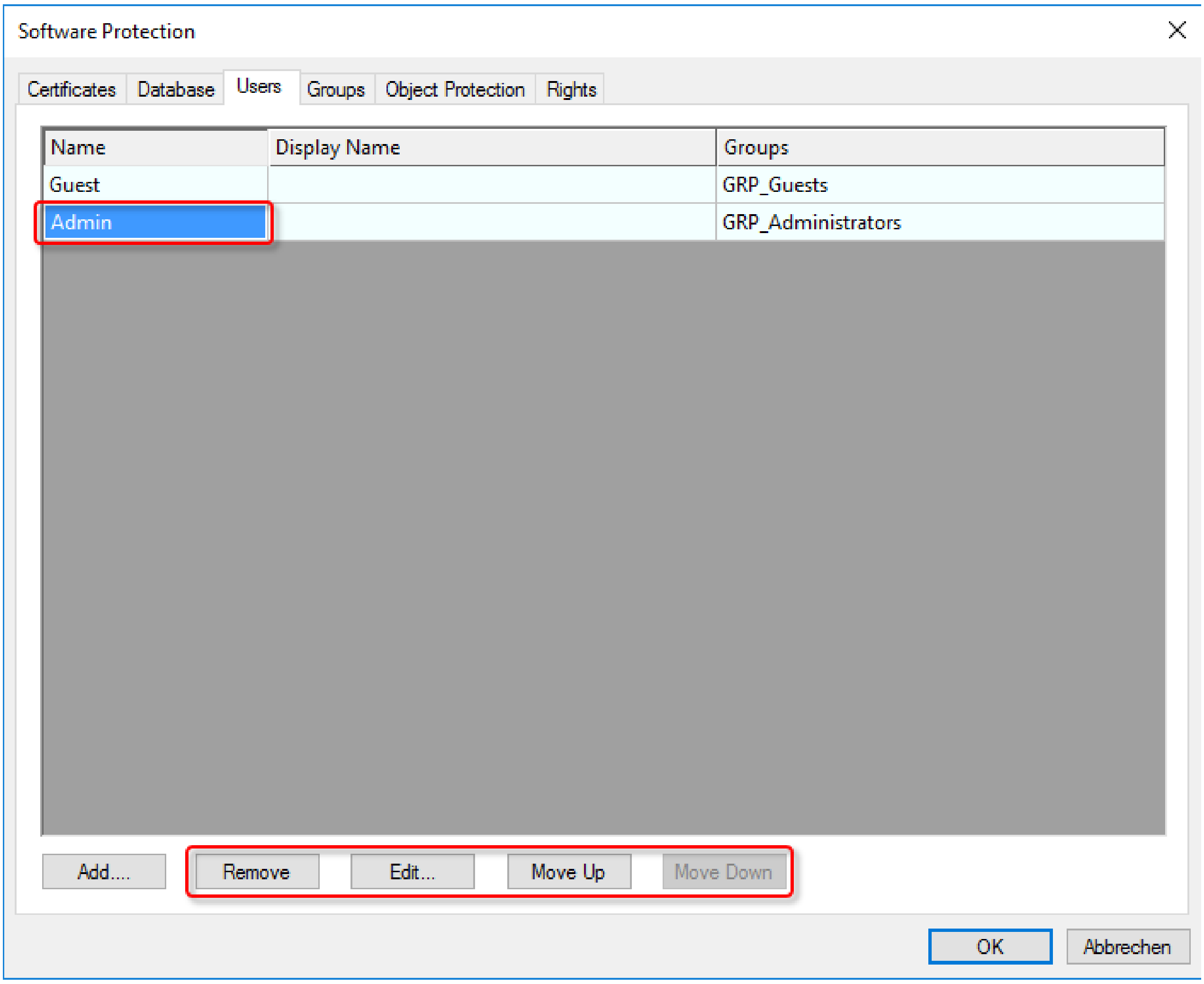
□ GRP_Guests
□ GRP_OEMDev
☑ GRP_Administrators

□ Expire Date: 27.11.2019

The new user must be assigned to the "GRP_Administrators" group.

The user can also be a Windows account (domain user); in this case, the associated Windows password can be used for automatic login.

After selecting a user, it can also be deleted, changed or moved up or down in the list:



Name	Display Name	Groups
Guest		GRP_Guests
Admin		GRP_Administrators

There must always be a user with administrator rights!

If you do not have a user with administrator rights in the user database, you will not be able to make any further changes to the database (including adding a new administrator!). Therefore, there must always be at least one user with (editing) administrator rights! (The signing administrator is not sufficient because he is not allowed to make changes to the user database).

5.6.2 Separating the database administrator and developer functions

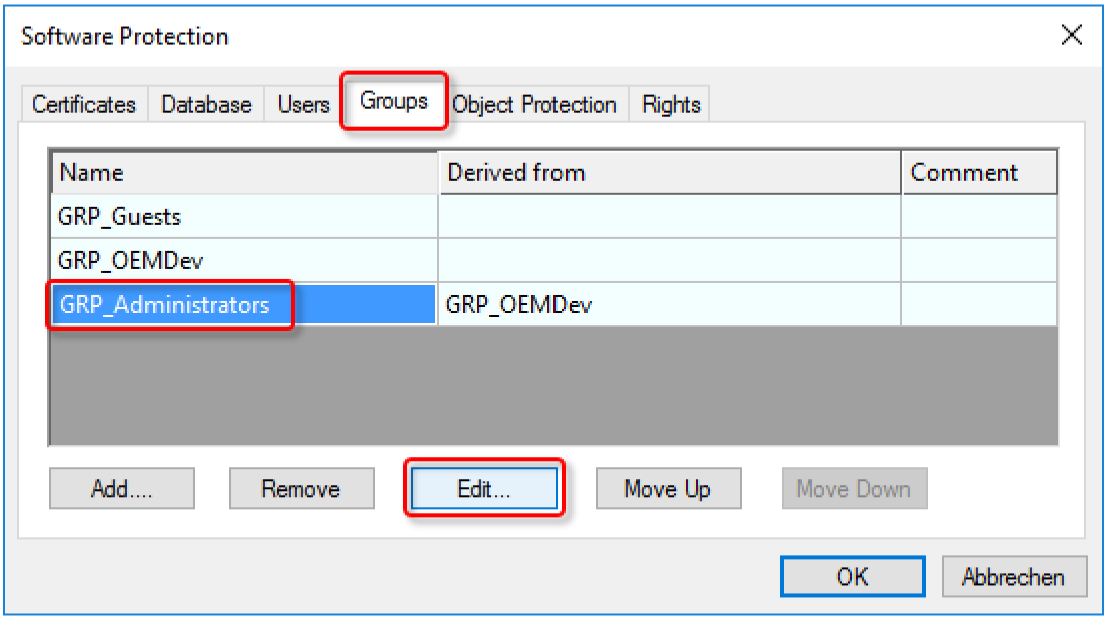
Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

By default, the "GRP_Administrators" group also inherits the rights of the "GRP_OEMDev" (Developers) group.

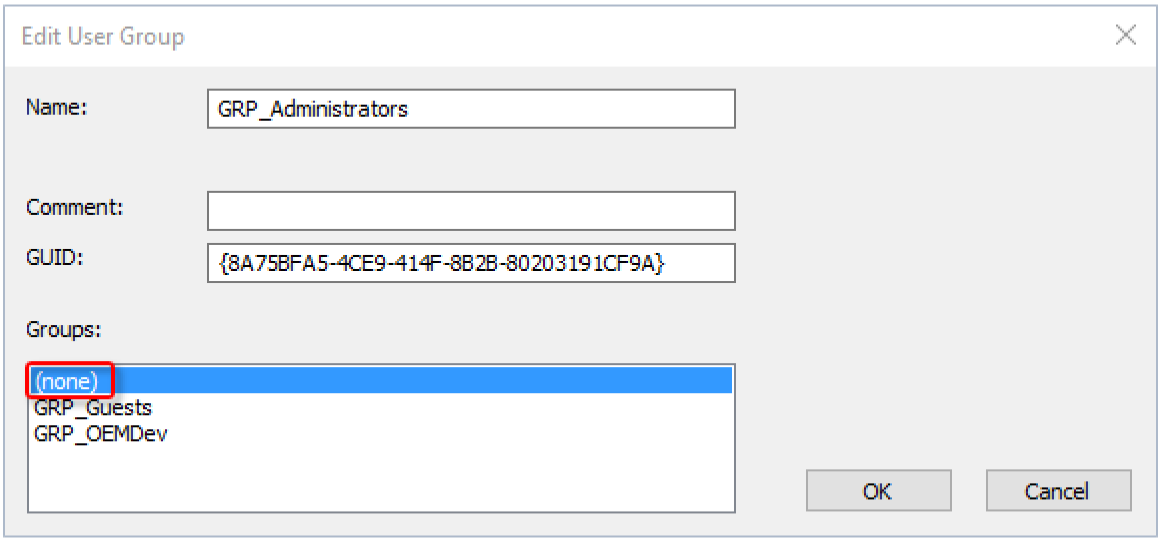
If the (editing) administrator of the user database does not have rights to modify the TwinCAT Solution, only the membership of the "GRP_OEMDev" group in the "GRP_Administrators" group needs to be changed.

To do this, select the "GRP_Administrators" group on the **Groups** tab in the Software Protection configuration console and then click the **Edit** button:



Name	Derived from	Comment
GRP_Guests		
GRP_OEMDev		
GRP_Administrators	GRP_OEMDev	

The desired group membership (or "None") can then be selected:



Name: GRP_Administrators

Comment:

GUID: {8A75BFA5-4CE9-414F-8B2B-80203191CF9A}

Groups:
(none)
GRP_Guests
GRP_OEMDev

An (editing) administrator can now change the user database, but no longer has the rights of the "GRP_OEMDev" group (Developers).

5.6.3 Adding users to a group

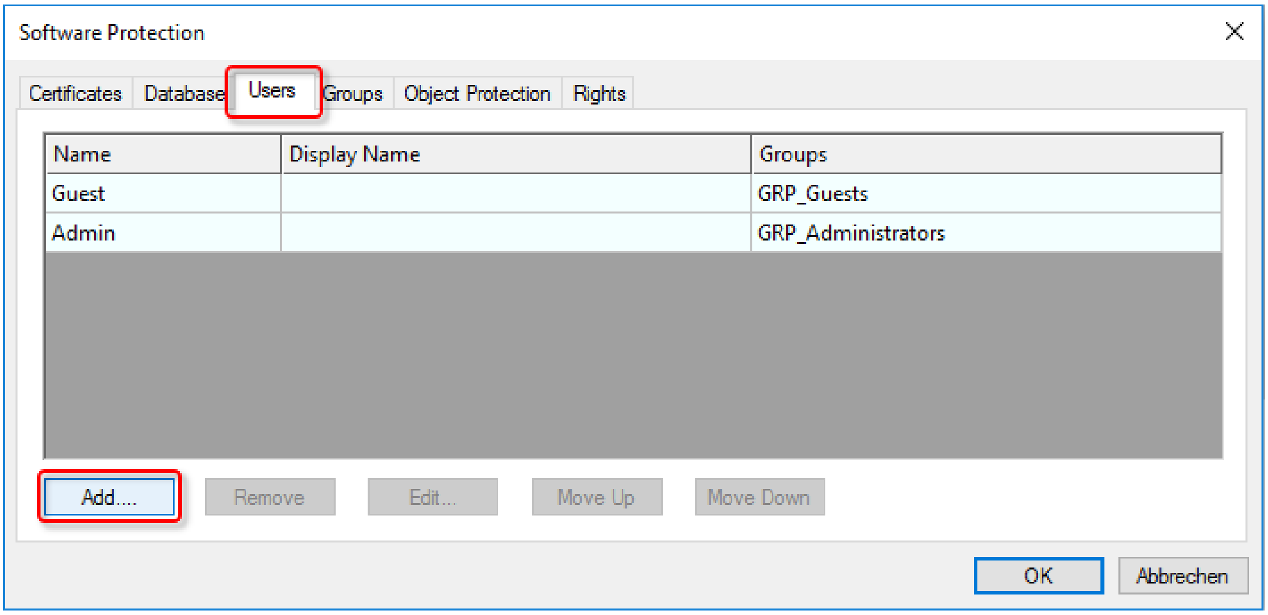
Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

In this example, a user is added to the "GRP_OEMDev" group. The procedure also applies analogously to other groups.

A user can also be a member of several groups.

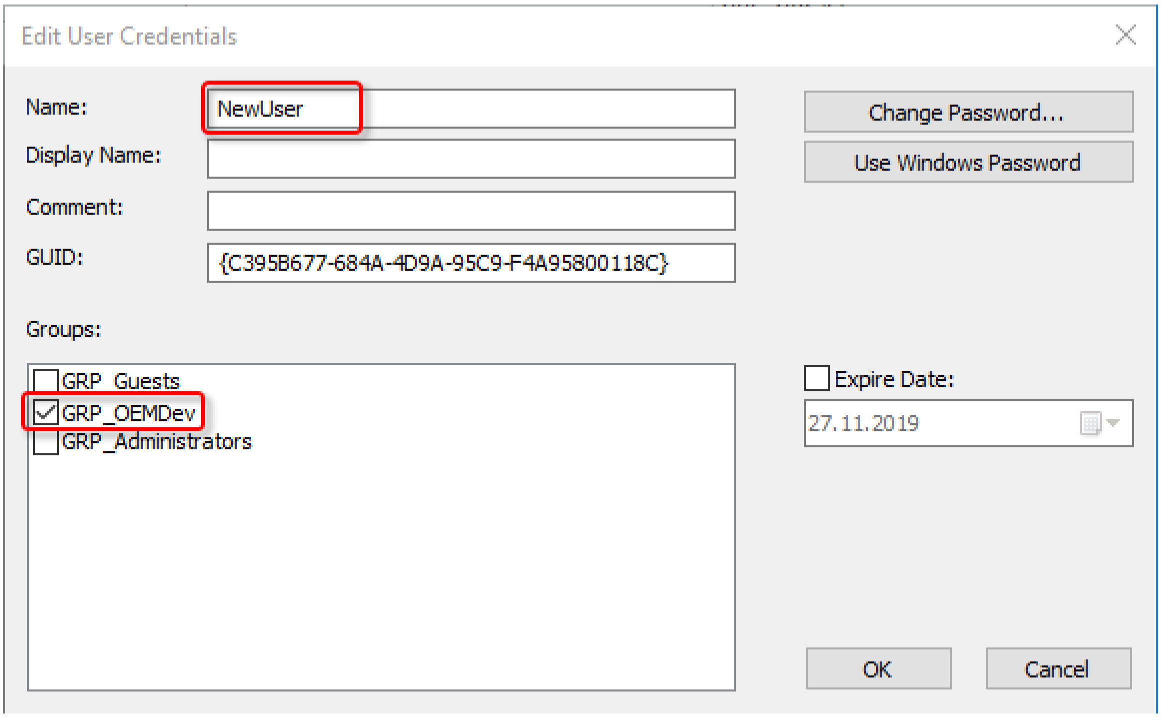
Click the **Add...** button on the **Users** tab in the configuration console:



Software Protection

Name	Display Name	Groups
Guest		GRP_Guests
Admin		GRP_Administrators

A new user can now be created and the desired group membership can be set:



Edit User Credentials

Name: NewUser

Display Name:

Comment:

GUID: {C395B677-684A-4D9A-95C9-F4A95800118C}

Groups:

□ GRP_Guests
☑ GRP_OEMDev
□ GRP_Administrators

□ Expire Date: 27.11.2019

The user can also be a Windows account (domain user); in this case, the associated Windows password can be used for automatic login.

From Build 4024.8

Users can also be created in so-called "extensions [▶ 47]" of user databases. This is described here [▶ 41].

The creation of new user groups and the adaptation of existing ones is explained in the chapter "Creating and editing user groups [▶ 61]".

5.6.4 Defining your own group access rights

Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

Download link: Planning table for group rights and Object Protection Level

An Excel table for the simple planning of group rights and access rights group sets (Object Protection Level) can be downloaded https://infosys.beckhoff.com/content/1033/tc3_security_management/Resources/8882888971/.zip.

5.6.4.1 Introduction

System requirements

Operating system:

- At least Windows 7 (or its Embedded version) is required in order to be able to use all the functions for the protection of the application software. Windows XP and Windows CE (Windows Embedded Compact) do not support either the encryption of the boot file or OEM licenses.

TwinCAT version:

- The functionalities described require TwinCAT 3.1 build 4022 or higher.

Reliable protection only when using the latest TwinCAT 3 version

For reliable protection (e.g. secure encryption), always use the latest TwinCAT 3 version. This provides the maximum security.

Use at least TwinCAT 3.1 Build 4024.x.
For security reasons, do not use an older version!

Download link: Planning table for group rights and Object Protection Level

An Excel table for the simple planning of group rights and access rights group sets (Object Protection Level) can be downloaded https://infosys.beckhoff.com/content/1033/tc3_security_management/Resources/8882888971/.zip.

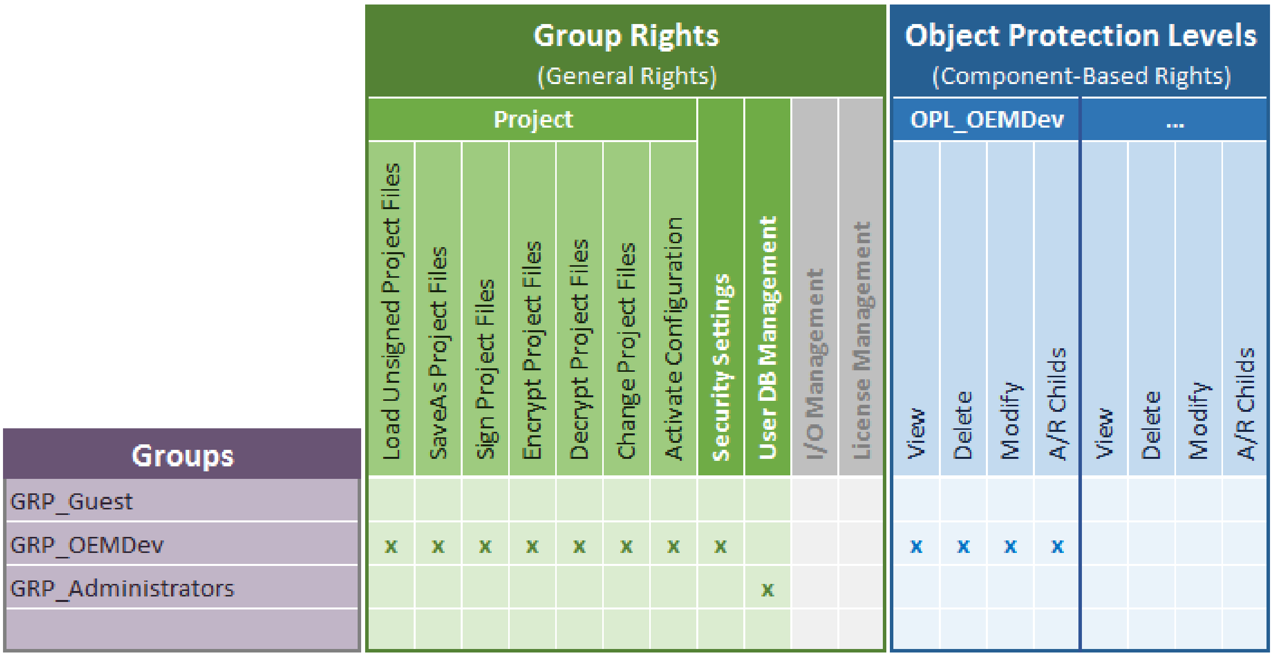
TwinCAT user access rights

- Access rights are assigned to groups in the TwinCAT 3 Engineering.
- Users can be assigned to several groups.
- Groups can be members of **another** group.

Note: For a better overview, it is recommended not to assign groups to another group, but to assign rights to the group completely independently.

Rights are divided into two main categories in the TwinCAT 3 Engineering:

1. General rights in the project (e.g. the right to sign files). These are assigned to user groups individually because they always apply to the entire project.
2. Component-specific rights ("View", "Delete", "Modify", and "Add/Remove Children").
   Because these can vary for different components of a project depending on the group membership, they are organized into a "rights set" that summarizes the individual rights of all groups under one designation.



	Group Rights (General Rights)											Object Protection Levels (Component-Based Rights)							
	Project											OPL_OEMDev				...			
Groups	Load Unsigned Project Files	SaveAs Project Files	Sign Project Files	Encrypt Project Files	Decrypt Project Files	Change Project Files	Activate Configuration	Security Settings	User DB Management	I/O Management	License Management	View	Delete	Modify	A/R Childs	View	Delete	Modify	A/R Childs
GRP_Guest																			
GRP_OEMDev	x	x	x	x	x	x	x	x				x	x	x	x				
GRP_Administrators									x										

Rights grayed out in the above illustration are provided "for future use" in the current version and are not yet implemented.

Such a rights set is called an "Object Protection Level" and represents a matrix of the existing groups and their rights for an object. With an Object Protection Level, individual project components can be conveniently provided with prefabricated rights sets for each group at once, and these do not have to be assigned in groups to each project component.

If the objects of a project are not different in terms of the set of access rights (the simplest use case), the definition and use of a single Object Protection Level is sufficient. This is then assigned to all objects in the project.

In the example above, the Developer group is allowed to do everything except make changes to the database, the Administrator group is only allowed to make changes to the database, and the Guest group is not allowed to do anything (not even load the project).

Keep in mind the membership of groups in other groups!

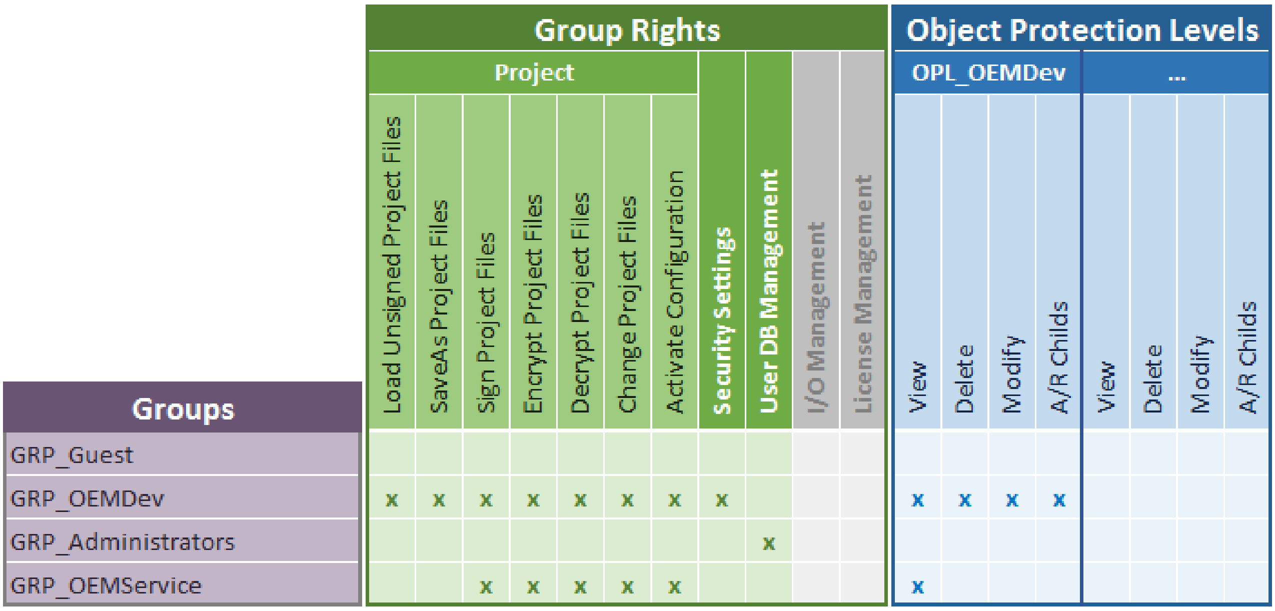
Sample 1

In the following sample, a new group called "GRP_OEMService" is to be added.

(The creation of a new group and the assignment of rights is described here [▶ 61]).

The new group is allowed to see everything, but not change anything, and may activate the project.

In order to view the project, the group must have the "Decrypt Project Files" right (otherwise Visual Studio will not be able to load the encrypted parts of the project).



Groups	Group Rights											Object Protection Levels							
	Project											OPL_OEMDev				...			
	Load Unsigned Project Files	SaveAs Project Files	Sign Project Files	Encrypt Project Files	Decrypt Project Files	Change Project Files	Activate Configuration	Security Settings	User DB Management	I/O Management	License Management	View	Delete	Modify	A/R Childs	View	Delete	Modify	A/R Childs
GRP_Guest																			
GRP_OEMDev	x	x	x	x	x	x	x	x				x	x	x	x				
GRP_Administrators									x										
GRP_OEMService			x	x	x	x	x					x							

To activate the project it is necessary, in addition to the "Activate Configuration" right, to be able to modify the project file (because certain information is saved there when activated), as well as to save these changes in encrypted form. Therefore, the "Change Project File" and "Encrypt Project Files" rights are additionally required.

For component-specific rights, only "View" is necessary.

A new Object Protection Level does not need to be created, because this rights set should always apply to the entire project.

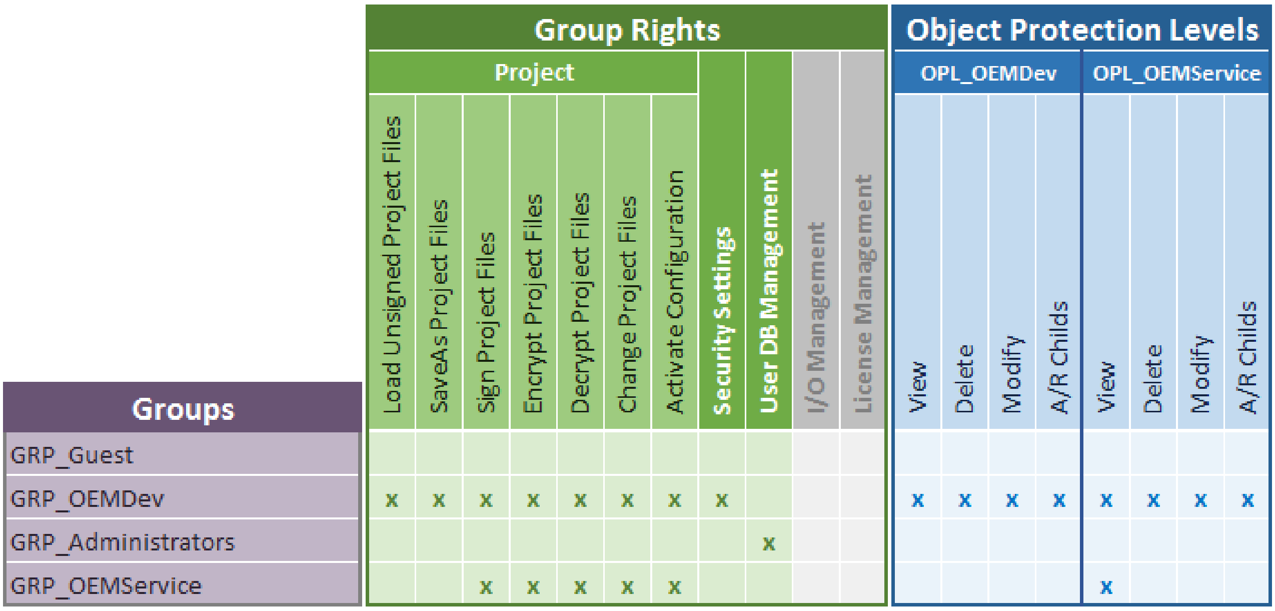
Sample 2

In the next sample, the "GRP_OEMService" group should only be able to view defined components of the project.

This requires the creation of a new group rights set, i.e. a new Object Protection Level (OPL), in order to be able to differentiate the respective rights assignment for a specific project component. We call the new OPL "OPL_OEMService".

(The creation of a new Object Protection Level is described here [▶ 66]).

The viewing right for the GRP_OEMService group is now removed from the "OPL_OEMDev" and added to the new "OPL_OEMService":



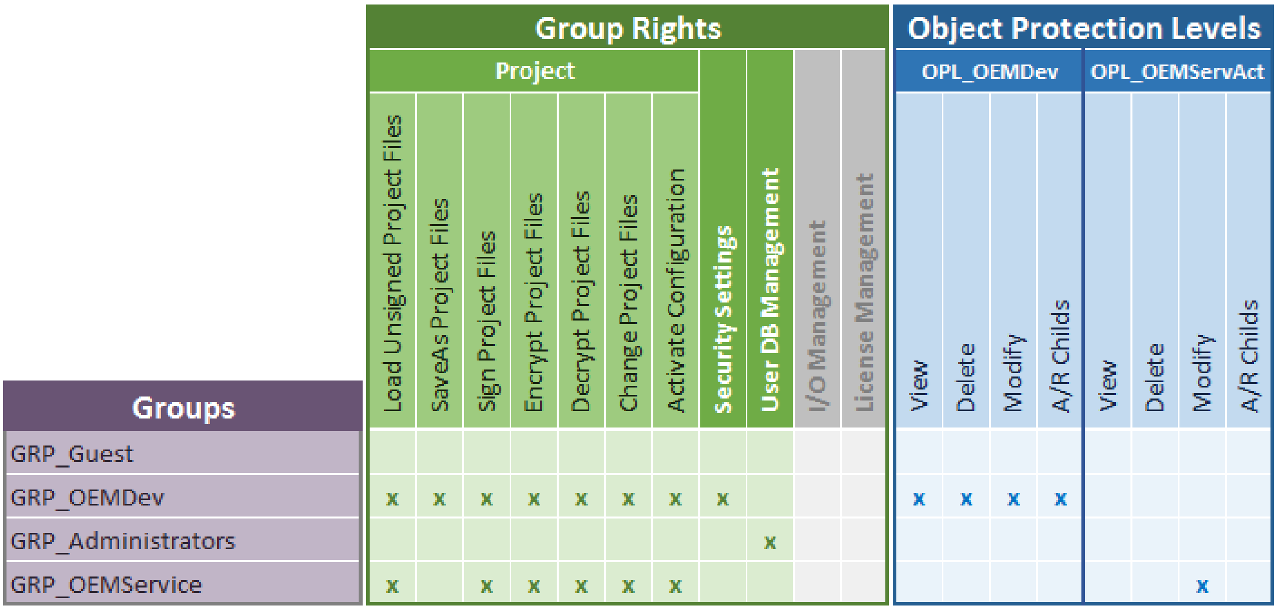
	Group Rights											Object Protection Levels							
	Project											OPL_OEMDev				OPL_OEMService			
Groups	Load Unsigned Project Files	SaveAs Project Files	Sign Project Files	Encrypt Project Files	Decrypt Project Files	Change Project Files	Activate Configuration	Security Settings	User DB Management	I/O Management	License Management	View	Delete	Modify	A/R Childs	View	Delete	Modify	A/R Childs
GRP_Guest																			
GRP_OEMDev	x	x	x	x	x	x	x	x				x	x	x	x	x	x	x	x
GRP_Administrators									x										
GRP_OEMService			x	x	x	x	x									x			

Since the group "GRP_OEMDev" is also allowed to do everything in the new "OPL_OEMService", all rights (View, Modify, ...) were also entered there for this group.

Sample 3

In the next sample, the group GRP_OEMService is additionally to be allowed to make changes to certain project components. (However, it may (still) not delete or add project components).

For this, another new Object Protection Level (OPL) must be created. We call it "OPL_OEMServiceEdit":



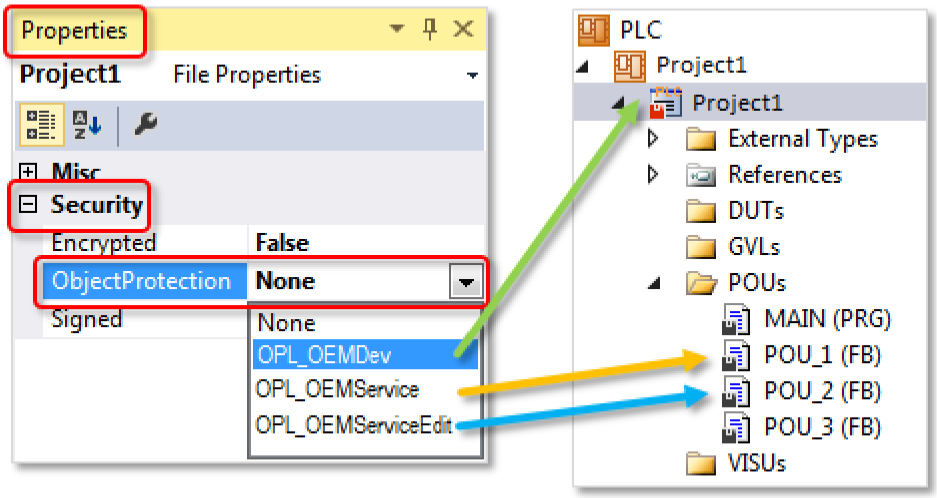
	Group Rights											Object Protection Levels							
	Project											OPL_OEMDev				OPL_OEMServAct			
Groups	Load Unsigned Project Files	SaveAs Project Files	Sign Project Files	Encrypt Project Files	Decrypt Project Files	Change Project Files	Activate Configuration	Security Settings	User DB Management	I/O Management	License Management	View	Delete	Modify	A/R Childs	View	Delete	Modify	A/R Childs
GRP_Guest																			
GRP_OEMDev	x	x	x	x	x	x	x	x				x	x	x	x				
GRP_Administrators									x										
GRP_OEMService	x		x	x	x	x	x											x	

Compared to OPL_OEMService, only the "Modify" right is added here, the rest is identical.

Project components assigned to OPL_OEMServiceEdit can now also be changed by users of the GRP_OEMService group.

Assignment of the Object Protection Levels in the project

Now we only need to assign the OPLs created in the previous samples to the project components. (How exactly the assignment of the OPL takes place in the TwinCAT Engineering is described here [▸ 70]).



Properties

Project1 File Properties

Misc

Security

Encrypted False

ObjectProtection None

Signed

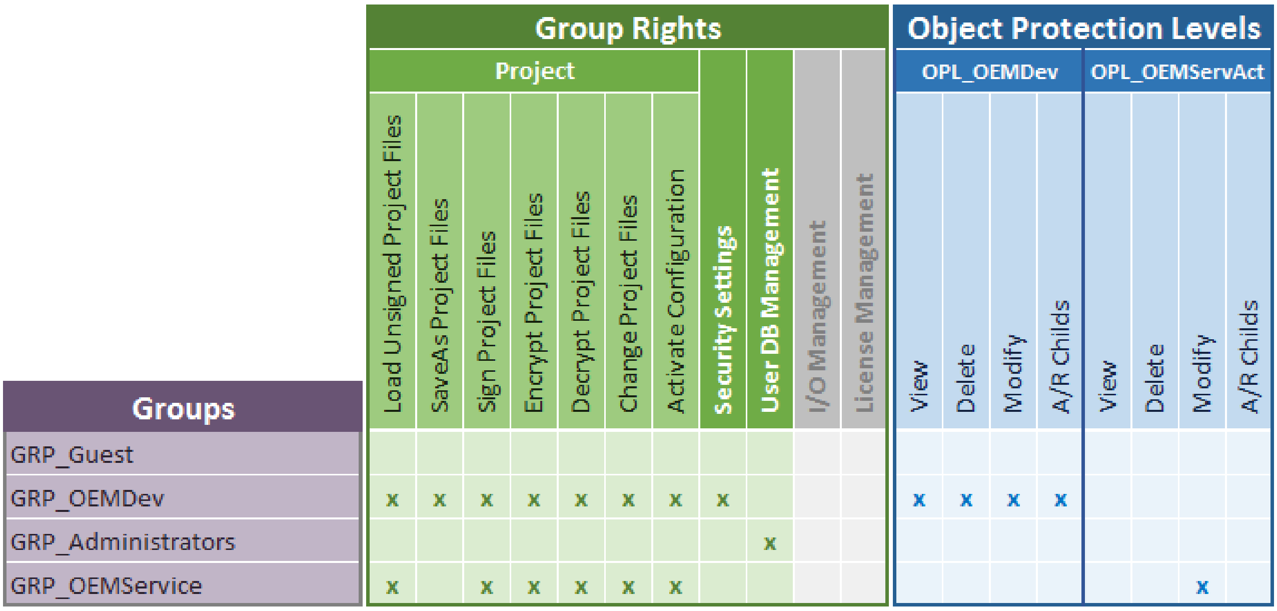
OPL is inherited

The OPL assigned to the root of the PLC project is inherited into the underlying nodes. Only the nodes that require a setting other than the PLC project root must be individually configured with the required OPL.

Sample 4

The following sample considers the case where the service employee is allowed to activate a project but not view it.

Since a special rights configuration is only required for the root of the PLC project here, we need our own Object Protection Level. We call it "OPL_OEMServAct":

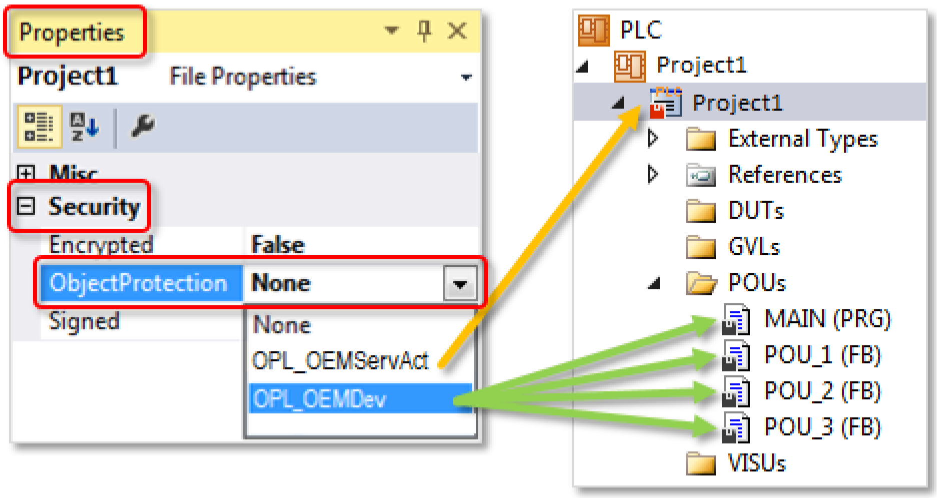


	Group Rights											Object Protection Levels							
	Project											OPL_OEMDev				OPL_OEMServAct			
Groups	Load Unsigned Project Files	SaveAs Project Files	Sign Project Files	Encrypt Project Files	Decrypt Project Files	Change Project Files	Activate Configuration	Security Settings	User DB Management	I/O Management	License Management	View	Delete	Modify	A/R Childs	View	Delete	Modify	A/R Childs
GRP_Guest																			
GRP_OEMDev	x	x	x	x	x	x	x	x				x	x	x	x				
GRP_Administrators									x										
GRP_OEMService	x		x	x	x	x	x											x	

Unlike in sample 2, the "GRP_OEMService" group has only modifying rights, but no viewing rights. "View" is not included in "Modify".

Visual Studio requires the "Modify" right for the project file, because changes must be made there when it is enabled.

When assigning the OPLs, the project root is now provided with the "OPL_OEMServAct".



However, since this property is passed on to the project components located below the root (unless explicit individual settings have been made there), the project components located below the root may have to be manually switched to another OPL individually. The convenient inheritance function of the PLC root properties cannot then be used in this case.

Documents about this

🗎 RightsOverview_TcSoftwareProtection.zip (Resources/zip/9007208137629963.zip)

5.6.4.2 Creating and editing users

Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

At least one user with administrator rights

In order to make changes to the database, at least one database user must belong to the administrator group. Therefore, always create at least one user with administrator rights.

The "Database Admin" defined when the user DB is created is used exclusively for signing the database. This account cannot be used for logging in or for changes to the database.

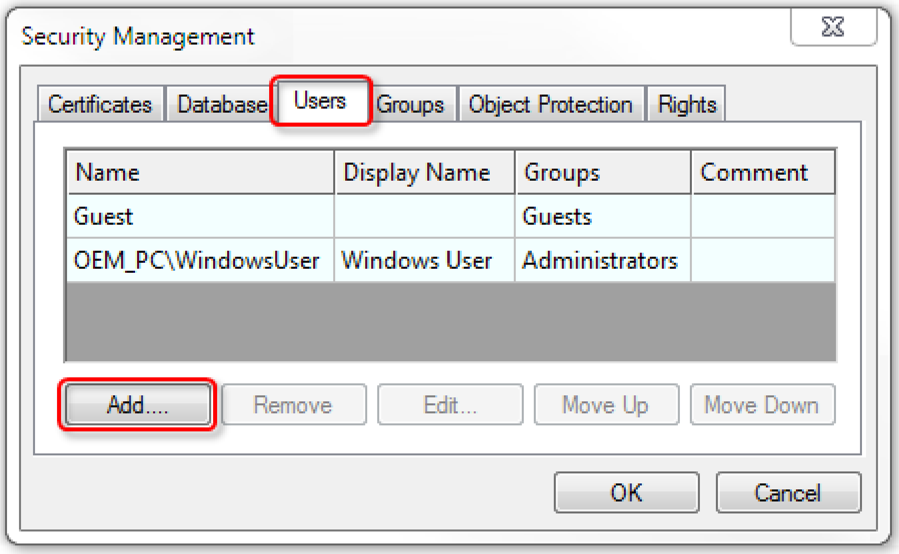
Creating and editing users

You can change the settings for existing users and create new users on the **Users** tab of the software protection configurator.

✓ User database can only be created or edited if no project is open. Close any open projects.

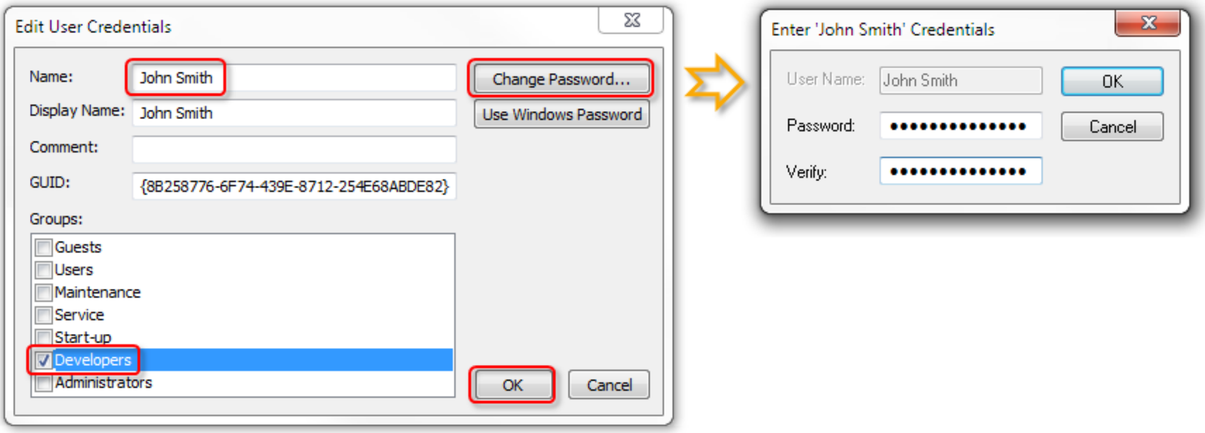
✓ The Software protection configurator [▶ 11] is opened.

1. Select the **Users** tab.



Name	Display Name	Groups	Comment
Guest		Guests	
OEM_PC\WindowsUser	Windows User	Administrators	

2. Click **Add** to add a new user.
   ⇨ The **Edit User Credentials** dialog opens.
3. Specify a **name** for the user and assign the user to a user group by ticking the corresponding check box (**Groups**).
4. For a Windows account the authentication can be done automatically via Windows. For all other users you have to specify a user-specific password. To do this, click on **Change Password**.
   ⇨ A dialog box for setting a password opens.
5. Assign a password for the user and confirm the password by repeating it.
6. Close the dialog with **OK**.



Name: John Smith
Display Name: John Smith
Comment:
GUID: {8B258776-6F74-439E-8712-254E68ABDE82}
Groups:
□ Guests
□ Users
□ Maintenance
□ Service
□ Start-up
☑ Developers
□ Administrators

User Name: John Smith
Password: ••••••••••••••
Verify: ••••••••••••••

   ⇨ The new user appears in the overview.
7. To edit an entry, select the user in the list and click **Edit**.
8. Close the **Edit User Credentials** dialog with **OK**.

⇨ A new user has been created in the system.

From Build 4024.8:

Users can also be created in so-called "extensions [▶ 47]" of user databases. This is described here [▶ 41].

All changes are only finally confirmed and valid on saving and signing the user database.

5.6.4.3 Creating and editing user groups

Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

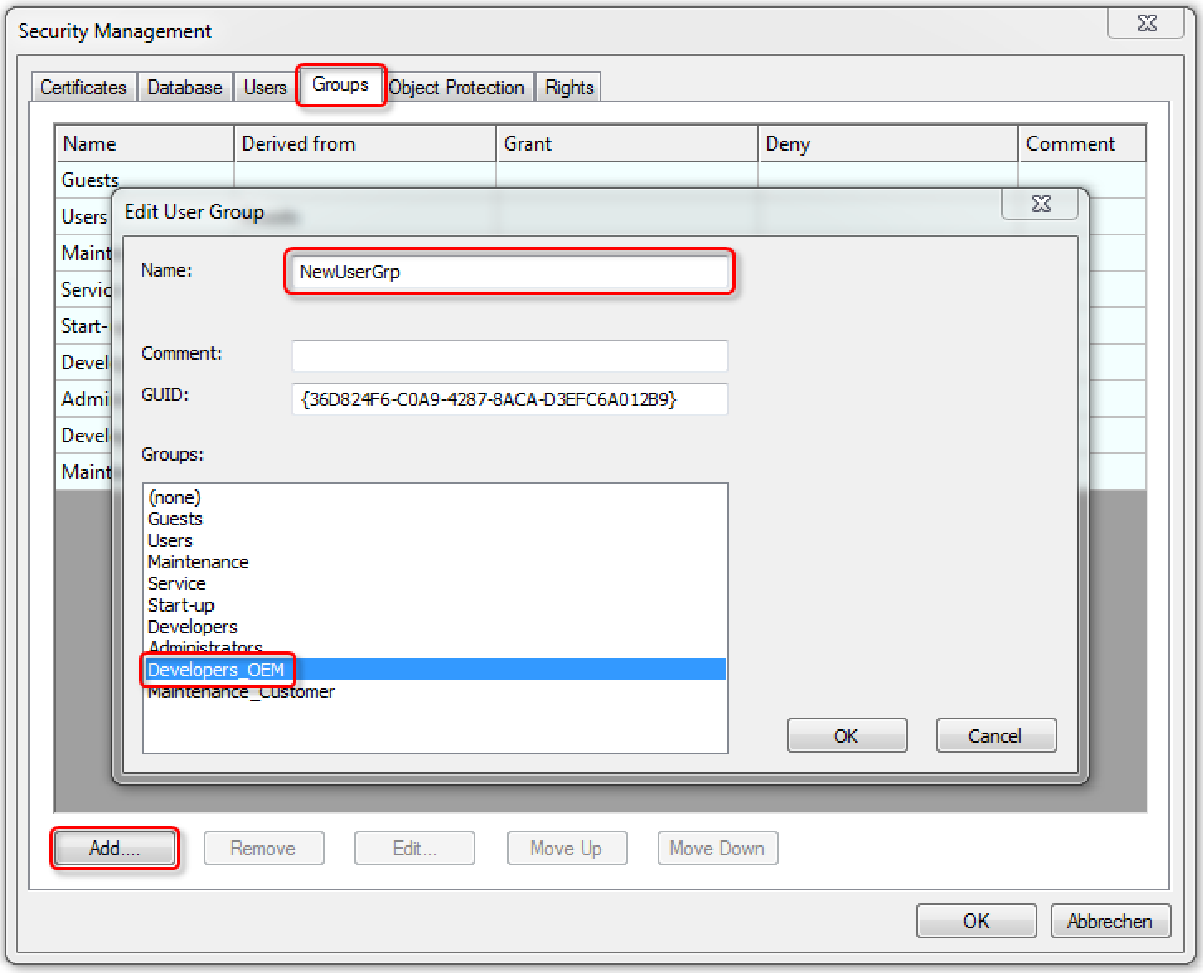
Creating and editing user groups

You can change the basic settings for existing user groups and create new user groups on the **Groups** tab of the software protection configurator.

Note: The rights assigned to a user group are adjusted on the "Rights" tab. The procedure is described in the following chapter [▶ 64].

✓ The Software protection configurator [▶ 11] is opened.

1. Select the **Groups** tab.



Name: NewUserGrp

Comment:

GUID: {36D824F6-C0A9-4287-8ACA-D3EFC6A012B9}

Groups:
(none)
Guests
Users
Maintenance
Service
Start-up
Developers
Administrators
Developers_OEM
Maintenance_Customer

2. Click on **Add** to create a new group.
   ⇨ The **Edit User Group** dialog opens.
3. Give the group a name (**Name**).
4. If the group is to inherit the rights of another group, select the corresponding group in the **Groups** section.
5. Close the dialog with **OK**.
   ⇨ The new group appears in the overview.
6. To edit an entry, select the user group in the list and click **Edit**.
7. Close the **Edit User Group** dialog with **OK**.

⇨ A new user group has been created in the system.

All changes are only finally confirmed and valid on saving and signing the user database.

In the **Rights** tab you can assign rights to user groups. Further information can be found in section Customizing the rights of user groups [▶ 64].

5.6.4.4 Adjusting the access rights of user groups

Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

The rights assigned to the user groups are managed on the **Rights** tab of the software protection configurator.

Download link: Planning table for group rights and Object Protection Level

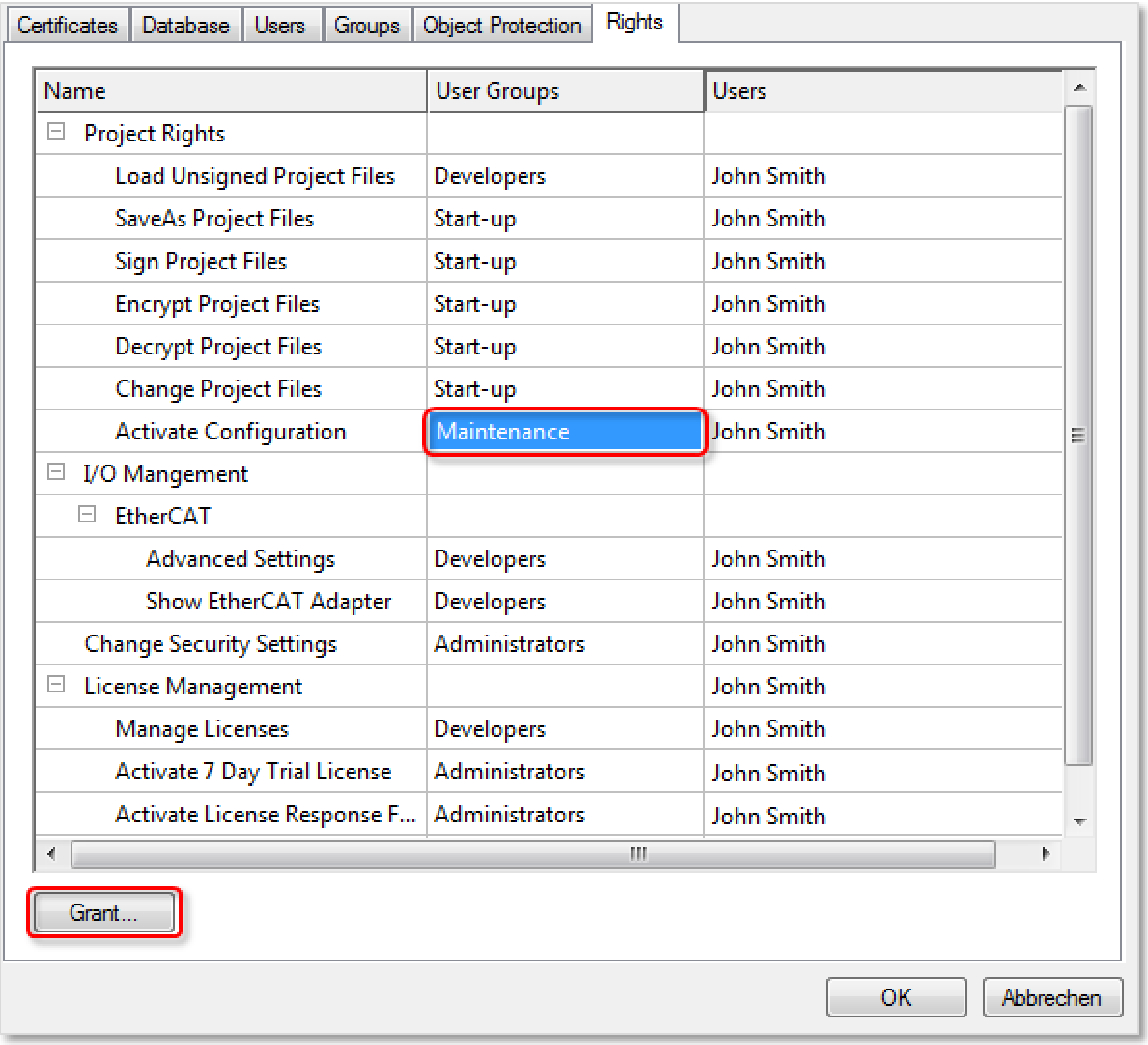
An Excel table for the simple planning of group rights and access rights group sets (Object Protection Level) can be downloaded https://infosys.beckhoff.com/content/1033/tc3_security_management/Resources/8882888971/.zip.

✓ User database can only be created or edited if no project is open. Close any open projects.

✓ The Software protection configurator [▶ 11] is opened.

1. Select the **Rights** tab.

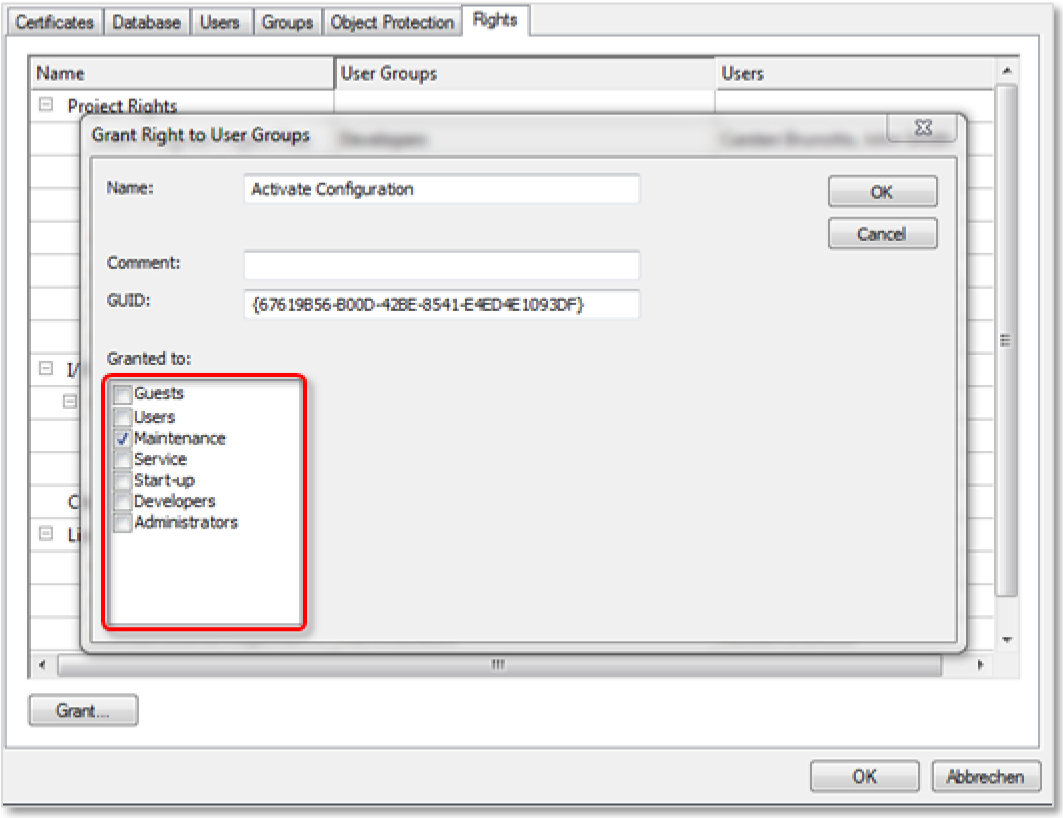
2. In the **UserGroups** column, mark the row with the desired right and click on the **Grant** button.



Name	User Groups	Users
⊟ Project Rights		
Load Unsigned Project Files	Developers	John Smith
SaveAs Project Files	Start-up	John Smith
Sign Project Files	Start-up	John Smith
Encrypt Project Files	Start-up	John Smith
Decrypt Project Files	Start-up	John Smith
Change Project Files	Start-up	John Smith
Activate Configuration	Maintenance	John Smith
⊟ I/O Mangement		
⊟ EtherCAT		
Advanced Settings	Developers	John Smith
Show EtherCAT Adapter	Developers	John Smith
Change Security Settings	Administrators	John Smith
⊟ License Management		John Smith
Manage Licenses	Developers	John Smith
Activate 7 Day Trial License	Administrators	John Smith
Activate License Response F...	Administrators	John Smith

⇨ The **Grant Right to User Groups** dialog opens.

3. Use the check boxes to select which user groups should have this right.



Name: Activate Configuration

Comment:

GUID: {67619B56-B00D-42BE-8541-E4ED4E1093DF}

Granted to:

- □ Guests
- □ Users
- ☑ Maintenance
- □ Service
- □ Start-up
- □ Developers
- □ Administrators

4. Click on **OK**.

⇨ The changes are applied (temporarily).

All changes are only finally confirmed and valid on saving and signing the user database.

5.6.4.5 Creating and editing access rights group sets (Object Protection Level)

Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

Download link: Planning table for group rights and Object Protection Level

An Excel table for the simple planning of group rights and access rights group sets (Object Protection Level) can be downloaded https://infosys.beckhoff.com/content/1033/tc3_security_management/Resources/8882888971/.zip.

✓ User database can only be created or edited if no project is open. Close any open projects.

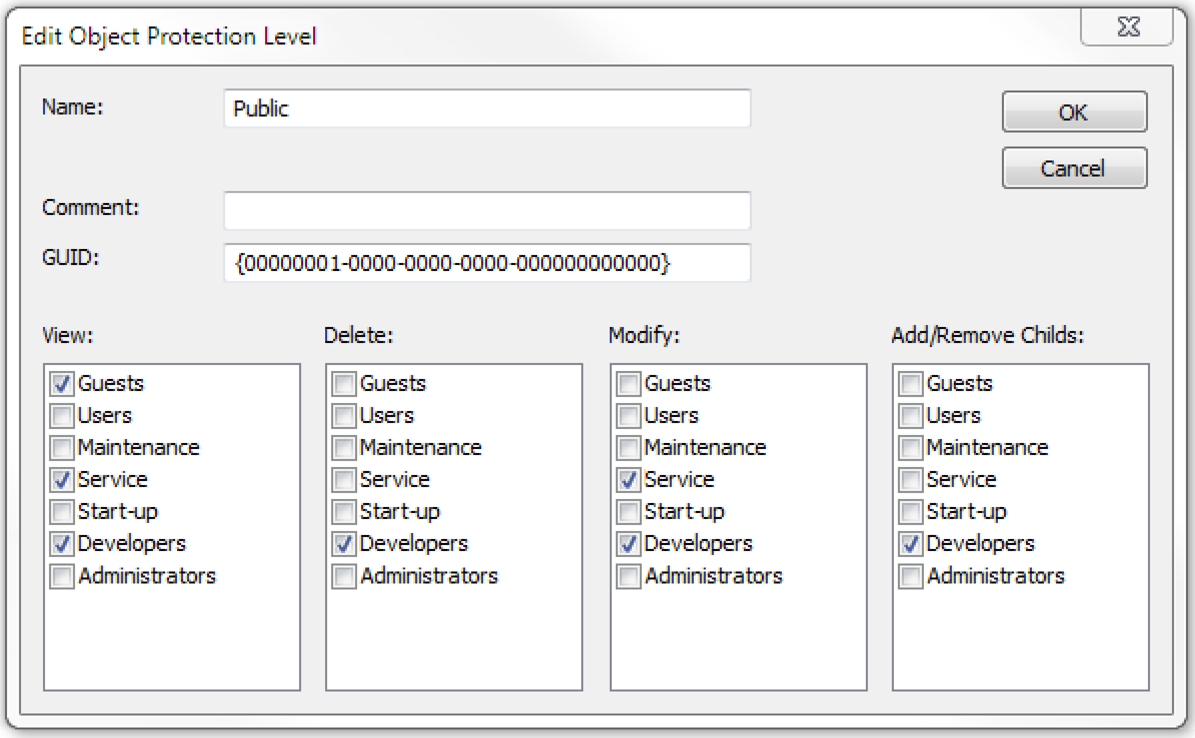
✓ The Software protection configurator [▶ 11] is opened.

1. Select the **Object Protection** tab.
2. Click on **Add**.

   ⇨ The **Edit Object Protection Level** dialog opens.

3. Assign the individual user rights, by ticking the respective check boxes, for all the groups defined under Security Management for this specific Object Protection Level.

   The following example shows the definition of the "Public" Object Protection Level:



Name: Public

Comment:

GUID: {00000001-0000-0000-0000-000000000000}

View:
☑ Guests
□ Users
□ Maintenance
☑ Service
□ Start-up
☑ Developers
□ Administrators

Delete:
□ Guests
□ Users
□ Maintenance
□ Service
□ Start-up
☑ Developers
□ Administrators

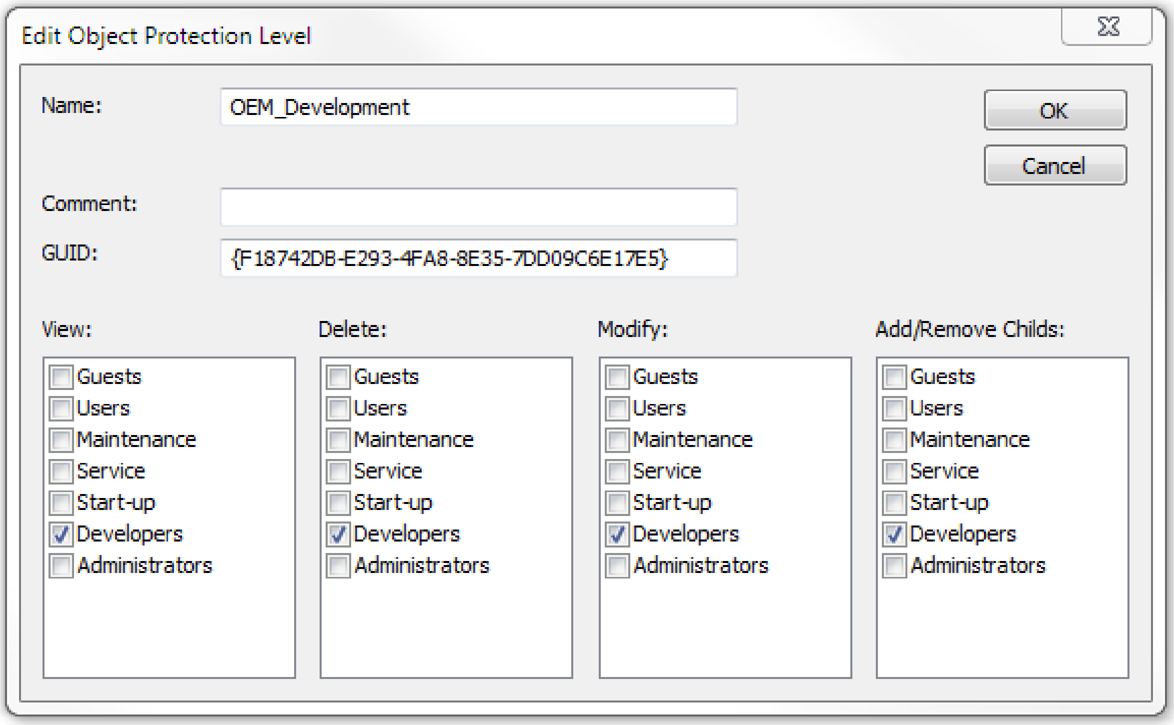
Modify:
□ Guests
□ Users
□ Maintenance
☑ Service
□ Start-up
☑ Developers
□ Administrators

Add/Remove Childs:
□ Guests
□ Users
□ Maintenance
□ Service
□ Start-up
☑ Developers
□ Administrators

- The "Guest" user group can read a TwinCAT object that is assigned this Object Protection Level but cannot change it.
- The "Service" user group can read and modify a TwinCAT object that is assigned this Object Protection Level but cannot delete it.

- The "Developers" user group has full access.

  In the following example only the "Developers" user group has access to the TwinCAT object. The other user groups have no rights at all.



Edit Object Protection Level

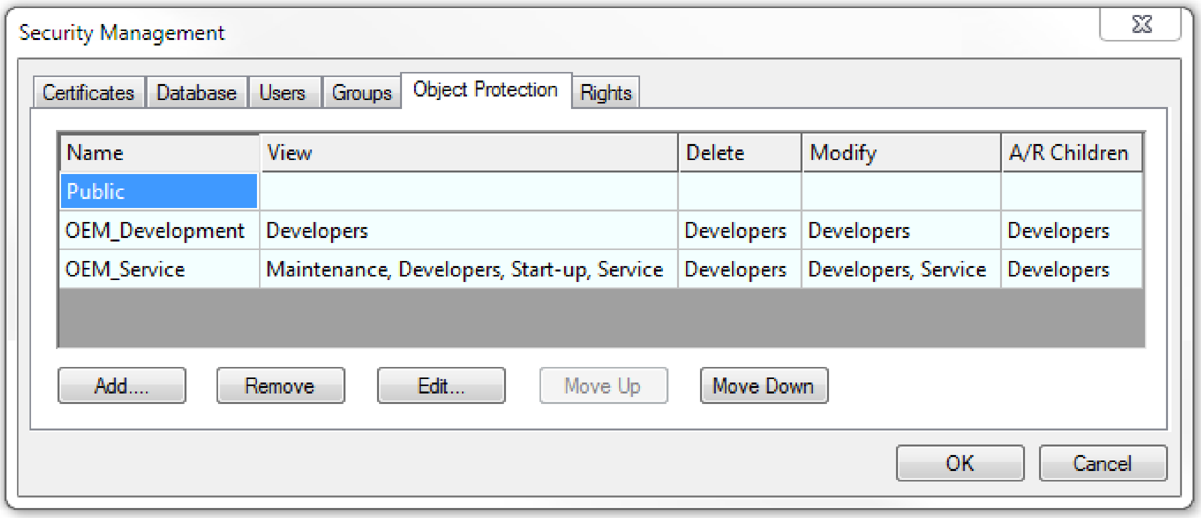
Name: OEM_Development

Comment:

GUID: {F18742DB-E293-4FA8-8E35-7DD09C6E17E5}

View:	Delete:	Modify:	Add/Remove Childs:
□ Guests	□ Guests	□ Guests	□ Guests
□ Users	□ Users	□ Users	□ Users
□ Maintenance	□ Maintenance	□ Maintenance	□ Maintenance
□ Service	□ Service	□ Service	□ Service
□ Start-up	□ Start-up	□ Start-up	□ Start-up
☑ Developers	☑ Developers	☑ Developers	☑ Developers
□ Administrators	□ Administrators	□ Administrators	□ Administrators

4. Confirm the dialog with **OK**.
   ⇨ The Object Protection Level with the user rights is created in the system and is displayed in the overview on the **Object Protection** tab in the software protection configurator.
5. Assign the required user rights for further user groups in an Object Protection Level accordingly.
6. To edit an Object Protection Level, select the corresponding column and click **Edit**.



Security Management

Name	View	Delete	Modify	A/R Children
Public				
OEM_Development	Developers	Developers	Developers	Developers
OEM_Service	Maintenance, Developers, Start-up, Service	Developers	Developers, Service	Developers

7. To remove an Object Protection Level, click **Remove**.
8. To change the position of the selected Object Protection Level in the overview, click on **Move up** or **Move down**.

All changes are only finally confirmed and valid on saving and signing the user database.

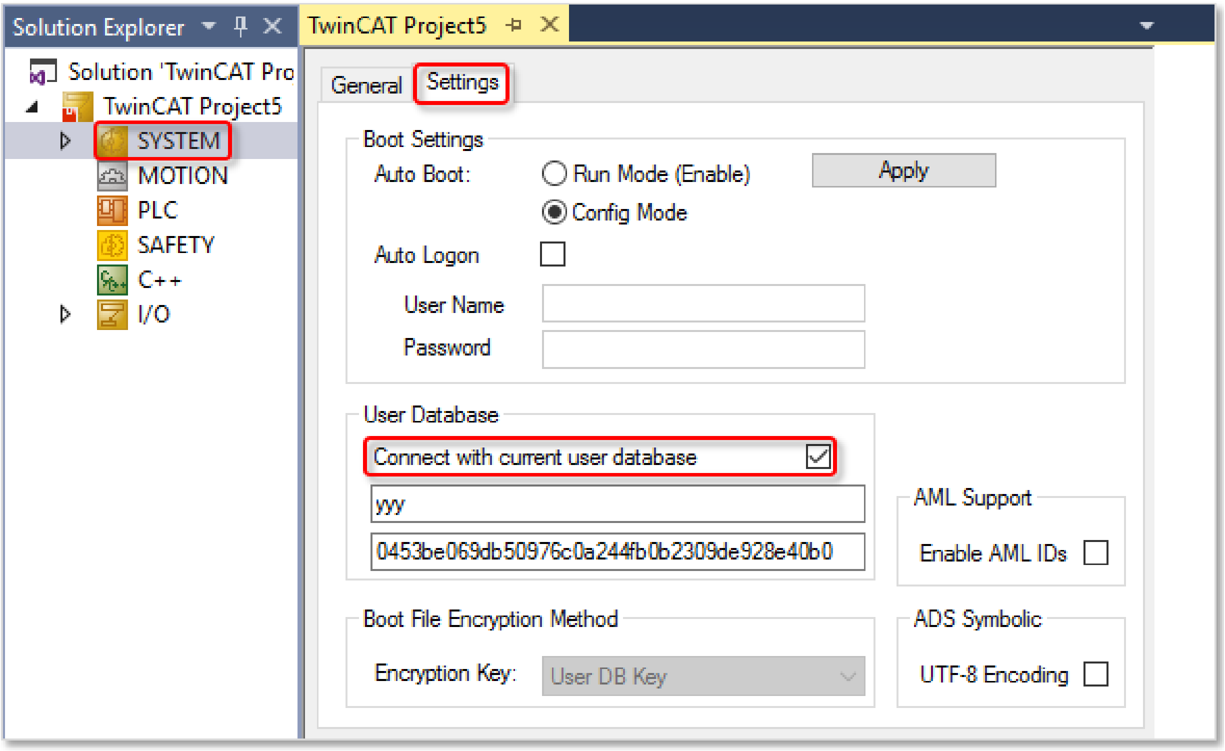
5.7 Linking the user database to a project

Initially, a project must be manually linked to a user database. The linking to the database is then stored in the project.

✓ Always make a backup copy of your project before connecting it with a user database.

✓ A user database has been created and activated. A TwinCAT project is opened.

1. In the TwinCAT project, double-click the SYSTEM node to open the system settings.
2. Open the **Settings** tab.
3. In the **User Database** area, check the **Connect with current user database** check box.



Boot Settings

Auto Boot: ○ Run Mode (Enable) ● Config Mode

Auto Logon □

User Name

Password

User Database

Connect with current user database ☑

yyy

0453be069db50976c0a244fb0b2309de928e40b0

AML Support

Enable AML IDs □

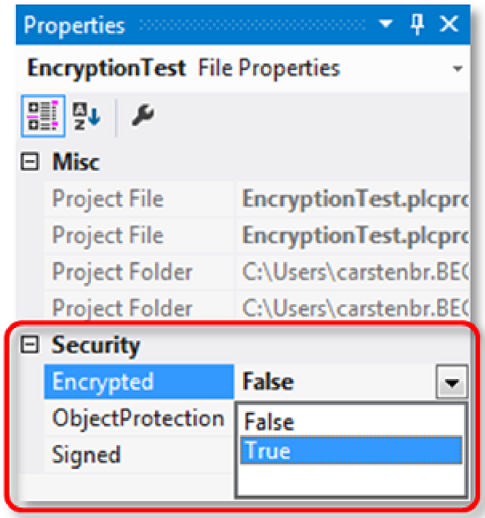
Boot File Encryption Method

Encryption Key: User DB Key

ADS Symbolic

UTF-8 Encoding □

⇨ The project is now linked to the user database. The **Security** area becomes visible in the **Properties** of a project component.



⊟ Misc	
Project File	EncryptionTest.plcprc
Project File	EncryptionTest.plcprc
Project Folder	C:\Users\carstenbr.BEC
Project Folder	C:\Users\carstenbr.BEC
⊟ Security	
Encrypted	False
ObjectProtection	False True
Signed	

5.8 Assigning user access rights in the project

Download link: Planning table for group rights and Object Protection Level

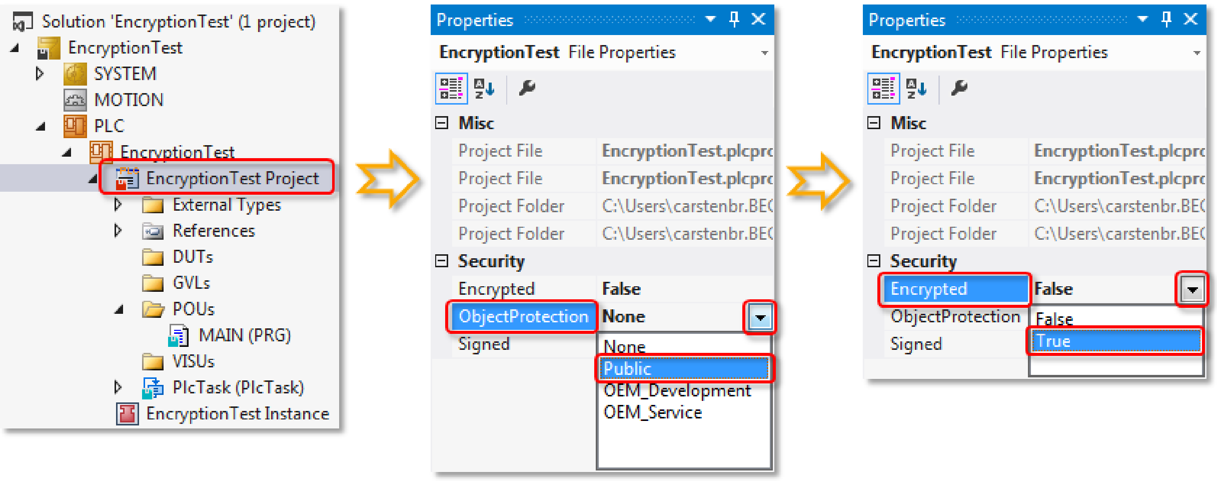
An Excel table for the simple planning of group rights and access rights group sets (Object Protection Level) can be downloaded https://infosys.beckhoff.com/content/1033/tc3_security_management/Resources/8882888971/.zip.

You can assign the Object Protection Levels [▶ 66] created to TwinCAT objects, e.g. to a PLC project.

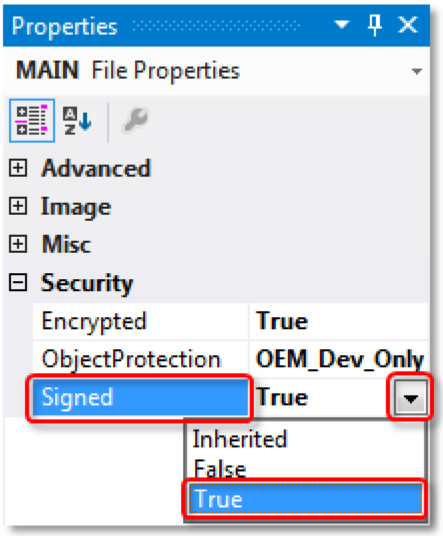
✓ The access authorization groups are defined.

✓ The project is linked to a user database.

1. Select the PLC object in the PLC project tree in the Solution Explorer.
   ⇨ The **Properties** view is updated. (If the **Properties** view is not open, select the **Properties Window** command in the **View** menu to open it).
2. Select the desired Object Protection Level from the drop-down list of the **ObjectProtection** property in the **Security** category.

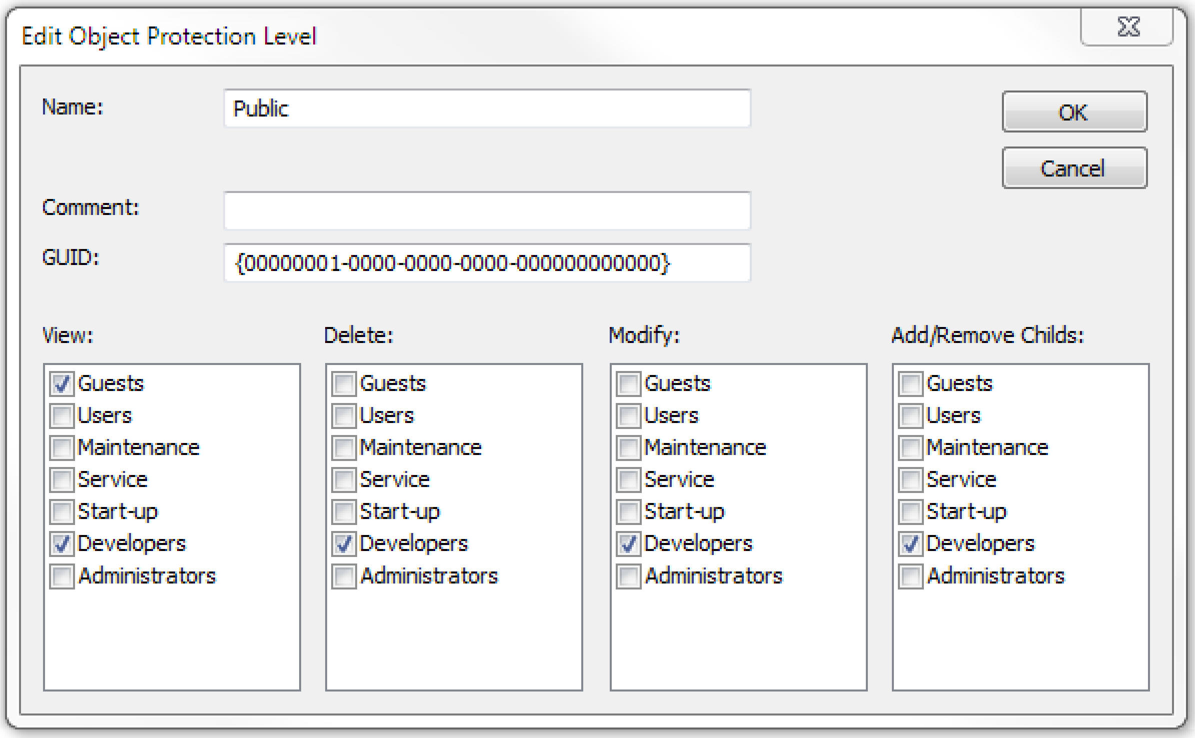

Solution 'EncryptionTest' (1 project)
EncryptionTest
SYSTEM
MOTION
PLC
EncryptionTest
EncryptionTest Project
External Types
References
DUTs
GVLs
POUs
MAIN (PRG)
VISUs
PlcTask (PlcTask)
EncryptionTest Instance
Properties
EncryptionTest File Properties
Misc
Project File EncryptionTest.plcprc
Project File EncryptionTest.plcprc
Project Folder C:\Users\carstenbr.BE(
Project Folder C:\Users\carstenbr.BE(
Security
Encrypted False
ObjectProtection None
Signed
None
Public
OEM_Development
OEM_Service
Properties
EncryptionTest File Properties
Misc
Project File EncryptionTest.plcprc
Project File EncryptionTest.plcprc
Project Folder C:\Users\carstenbr.BE(
Project Folder C:\Users\carstenbr.BE(
Security
Encrypted False
ObjectProtection
Signed
False
True

3. Then set the value of the property **Encrypted** to TRUE via the drop-down list. This setting is important in order to prevent access to the source code, e.g. via the operating system level.
4. Then set the value of the property **Signed** to TRUE via the drop-down list. This setting is important in order to prevent an unauthorized replacement of the object file at operating system level by another file of the same name.



⇨ The PLC project can now be accessed by the user groups, which were specified in the Object Protection Level. Save the PLC project to apply the settings.

In the example Object Protection Level "Public":



Edit Object Protection Level

Name: Public

Comment:

GUID: {00000001-0000-0000-0000-000000000000}

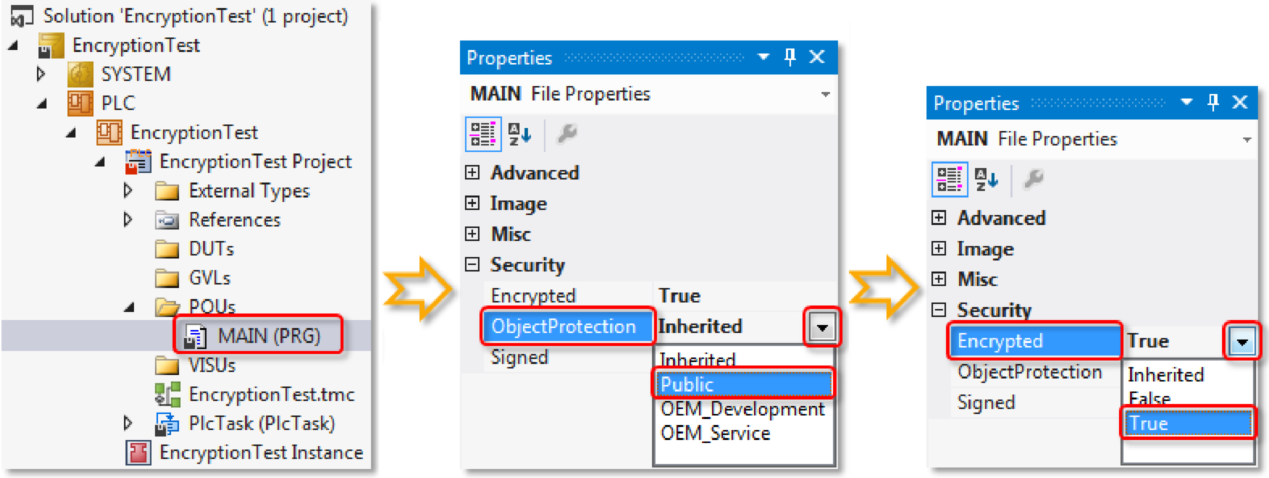
View:	Delete:	Modify:	Add/Remove Childs:
☑ Guests	□ Guests	□ Guests	□ Guests
□ Users	□ Users	□ Users	□ Users
□ Maintenance	□ Maintenance	□ Maintenance	□ Maintenance
□ Service	□ Service	□ Service	□ Service
□ Start-up	□ Start-up	□ Start-up	□ Start-up
☑ Developers	☑ Developers	☑ Developers	☑ Developers
□ Administrators	□ Administrators	□ Administrators	□ Administrators

- The "Guests" user group has read access to the PLC project.
- The "Developers" user group has full access.

  (No access rights were defined for the other user groups, since they are not used in the sample project.)

The access rights specified in the root of the PLC project are automatically passed on in the PLC project tree to all sub-elements of the SPS object, if they have the properties **Object Protection Level** and **Encryption**.

Alternatively, the Object Protection Level and the encryption can be assigned individually for each sub-element. This can be set in the sub-element properties.



Security	
Encrypted	True
ObjectProtection	Inherited
Signed	

Here, too, you must set the value of the **Encrypted** and **Signed** properties for the object to TRUE via the drop-down list. The purpose of this is firstly to prevent access to the source code, e.g. via the operating system level, and secondly to prevent an unauthorized replacement of the object file by another file of the same name.

5.9 Distribution / exchange of user databases

Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

No changes in settings of a user database when a project is open

No project may be open when changing the settings of a user database.

Note the following when working with user DBs:

- In the current TwinCAT 3 version the user DB must always be stored in directory *c:\TwinCAT\3.1\CustomConfig\UserDBs*.
- A user DB can be freely copied and pasted at the file level.
- When a user DB is created a one-to-one user DB key is generated, which identifies this database unambiguously.
- When a project is linked to a user DB, it can only be opened with a user DB with the same name and the same user DB key.
- Modifications of the content of a user DB do not affect the user DB key (this key is only generated once at the time when the user DB is created). In principle, you can therefore work with several different versions of a user DB. Example: The "in-house" version of a user DB contains other user accounts than the version supplied to the end customer on the control computer. The end customer can only see a specified selection of the available user accounts. You can severely restrict the available access options on the delivered machine, compared with the “in-house” development environment.
- Once a user DB has been created, the OEM certificate is no longer required for working with the user DB.
- Changes to the user DB must be signed by a (signing) administrator of the user DB. After changes to the user DB the corresponding query comes automatically when exiting the software protection configurator.

6 Logging in and selecting a user account

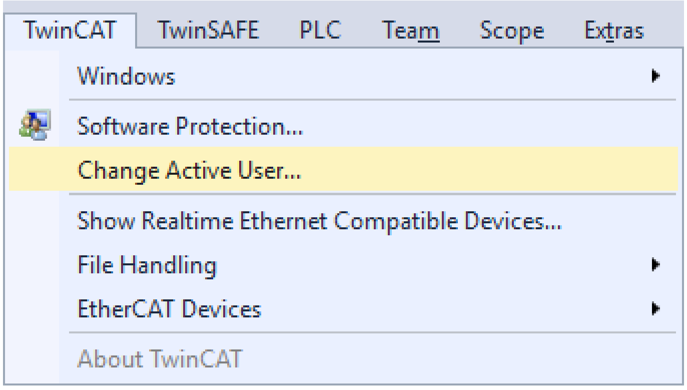
Allow operating system access for authorized users only

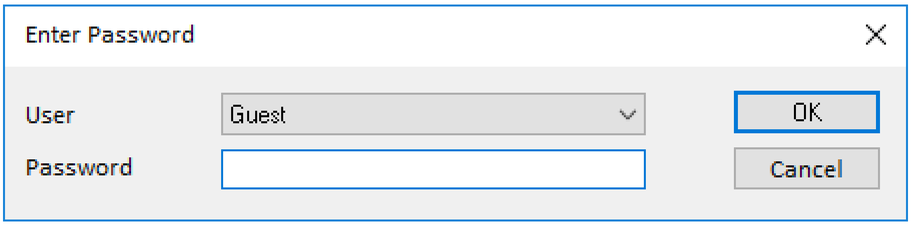
The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

No changes in settings of a user database when a project is open

No project may be open when changing the settings of a user database.

The user can be changed either via the toolbar or via the main menu item **TwinCAT -> Change Active User**:





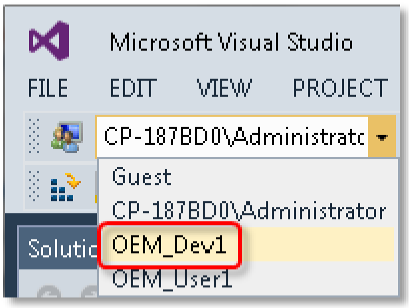
The user should only be changed when no project is loaded.

6.1 Build 4022

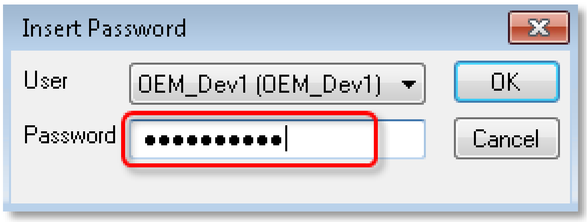
You can simply select a user account via the selection box in the Security Management toolbar.

✓ You have opened the Security Management toolbar [▶ 11].

1. Select the user account from the dropdown list.



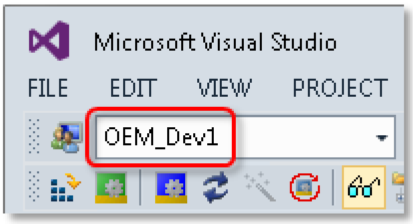
2. If the user login requires a password, a dialog for entering the password opens. Enter the password. If the authentication takes place via the Windows user account, no password is requested, since the authentication was already dealt with at the Windows login stage.



User OEM_Dev1 (OEM_Dev1)

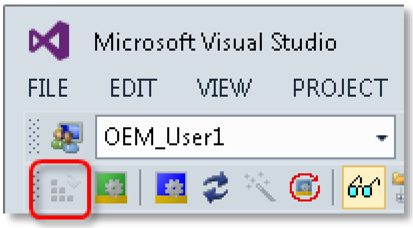
Password ●●●●●●●●●●

⇨ The selected user account is displayed in the Security Management toolbar.



OEM_Dev1

Depending on the user account rights, certain TwinCAT menu items may be grayed out and therefore disabled.



OEM_User1

7 Setting up basic protection of PLC application software

Allow operating system access for authorized users only

The content of the user database is protected against manipulation with a signature. The names of groups, object protection levels and users are not encrypted and could be read. Access to the IPC should be restricted to authorized users via the operating system.

7.1 Encryption

Make an unencrypted backup before encrypting!

Before encrypting a project: always make a backup of the project in its unencrypted state!

TwinCAT 3 uses 256-bit AES encryption and employs a private and public key procedure for the OEM certificate.

Prerequisite for using this function: Issue of a TwinCAT OEM certificate [▶ 19]

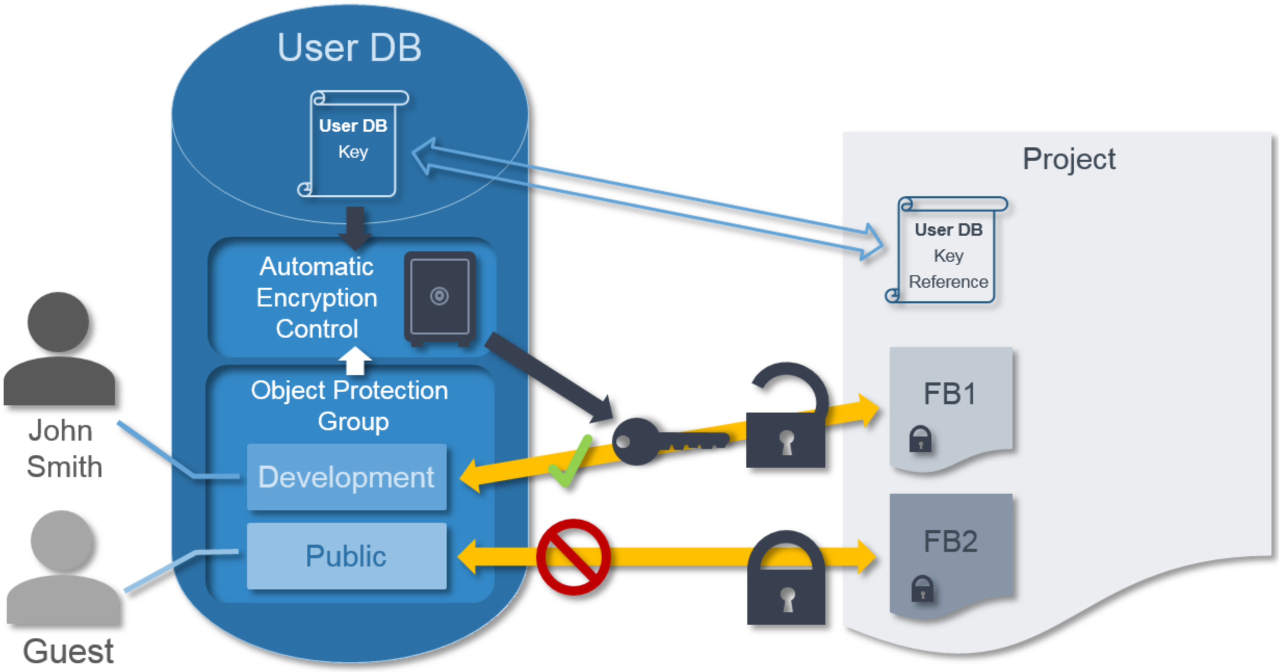
Following objects can be encrypted in TwinCAT:

- PLC source code
- Project file
- Boot project

Secure protection only with encryption of the project file

The project file must be encrypted in any case when using encryption, because it contains important information about the properties of the project. Manipulation of this information could prevent secure encryption of the source code.

The key used for the encryption is secured in the user database. The corresponding user database must therefore always be available on the Engineering computer. (Directory: *C:\Twincat\3.1\CustomConfig\userDBs*)



The user database is not necessary for the decryption of the boot project (= binary file).

System requirements

Operating system:

- At least Windows 7 (or its Embedded version) is required in order to be able to use all the functions for the protection of the application software. Windows XP and Windows CE (Windows Embedded Compact) do not support either the encryption of the boot file or OEM licenses.

TwinCAT version:

- The functionalities described require TwinCAT 3.1 build 4022 or higher.

Reliable protection only when using the latest TwinCAT 3 version

For reliable protection (e.g. secure encryption), always use the latest TwinCAT 3 version. This provides the maximum security.

Use at least TwinCAT 3.1 Build 4024.x.
For security reasons, do not use an older version!

7.1.1 PLC source code encryption

Make an unencrypted backup before encrypting!

Before encrypting a project: always make a backup of the project in its unencrypted state!

Access to encrypted objects is controlled via the Object Protection Level. In addition to the encryption, you therefore always have to set the required Object Protection Level for the TwinCAT 3 object. The Object Protection Level and encryption can be assigned conveniently in the properties of the respective TwinCAT object, e.g. a PLC project. The project must be linked to the user database. Encryption and the specification of the Object Protection Level are described in section Assigning user access rights in the project [▶ 70]. Save the project to apply the settings.

7.1.2 Project file encryption

Secure protection only with encryption of the project file

The project file must be encrypted in any case when using encryption, because it contains important information about the properties of the project. Manipulation of this information could prevent secure encryption of the source code.

Make an unencrypted backup before encrypting!

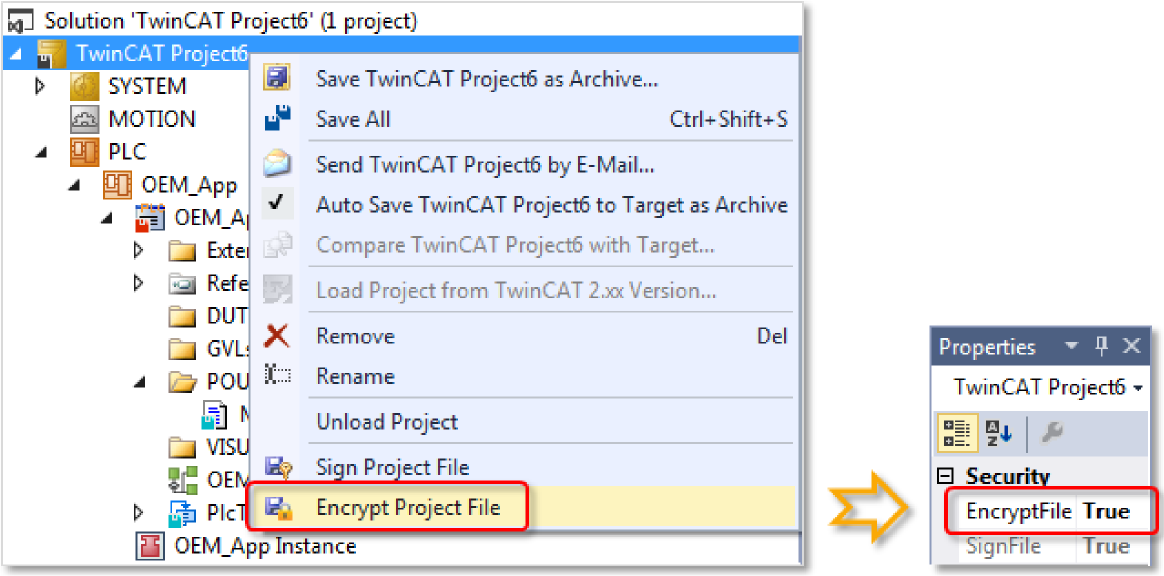
Before encrypting a project: always make a backup of the project in its unencrypted state!

The encryption of the project file is set via the TwinCAT project node.

✓ The project is linked to a user database.

1. Select the TwinCAT project node in the project tree in the Solution Explorer.
2. Select the command **Encrypt Project File** in the context menu.

⇨ In the **Properties** view, the value of the **EncryptFile** property in the **Security** category is set to TRUE.



Solution 'TwinCAT Project6' (1 project)
TwinCAT Project6
SYSTEM
MOTION
PLC
OEM_App
OEM_A
Exte
Refe
DUT
GVL
POU
VISU
OEM
PlcT
OEM_App Instance
Save TwinCAT Project6 as Archive...
Save All Ctrl+Shift+S
Send TwinCAT Project6 by E-Mail...
Auto Save TwinCAT Project6 to Target as Archive
Compare TwinCAT Project6 with Target...
Load Project from TwinCAT 2.xx Version...
Remove Del
Rename
Unload Project
Sign Project File
Encrypt Project File
Properties
TwinCAT Project6
Security

EncryptFile	True
SignFile	True

⇨ The project file is encrypted. It contains information on the components of the solution. On setting the encryption the project file itself is now encrypted. The encryption is not inherited to the components contained in the project. The encryption must be set individually for all (main) components of the project.

Only valid for TwinCAT 3.1 Build 4024.0: creation of a User DB requires Crypto Version 1

In the TwinCAT version Build **4024.0**, a user database [▶ 33] for the TwinCAT Software Protection may only be created with an OEM certificate with Crypto version 1!

7.1.3 Encrypting the boot project

Requirement: A current version of Windows or TwinCAT/BSD® on the target system

Encryption is not supported by older operating systems such as Windows NT, Windows CE / Windows Embedded Compact.

The encryption of the boot project is set (on the target-system) in the root node of the PLC project.

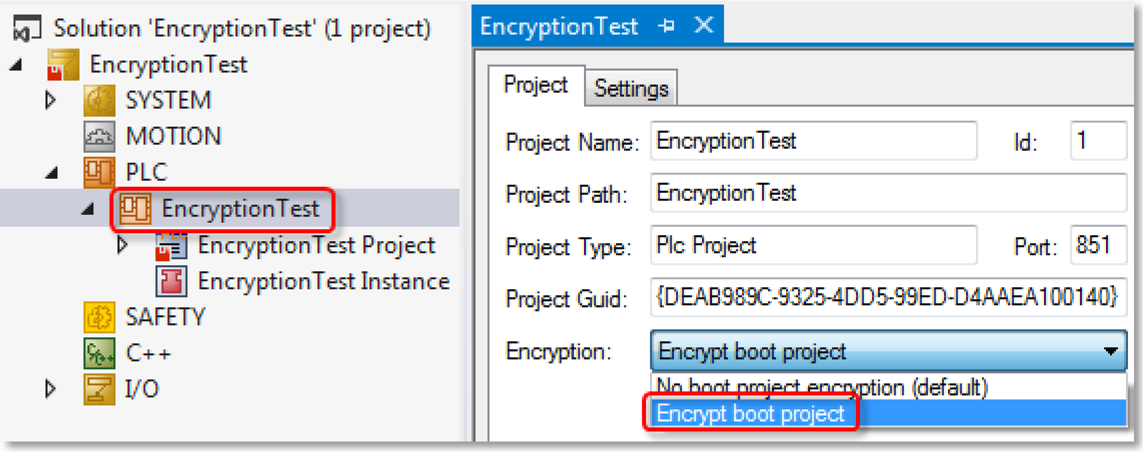
✓ A user database is selected [▶ 39] (and valid) in TwinCAT Engineering.

✓ The project must be linked with the user database [▶ 69] as information from the user database is used for encryption.

1. Double-click on the PLC project object in the PLC project tree in the Solution Explorer.

   ⇨ The PLC project settings are opened in an editor.

2. In the **Project** tab, select the **Encrypt boot project** entry in the drop-down list of the **Encryption** setting.



Project Name: EncryptionTest Id: 1

Project Path: EncryptionTest

Project Type: Plc Project Port: 851

Project Guid: {DEAB989C-9325-4DD5-99ED-D4AAEA100140}

Encryption: Encrypt boot project

No boot project encryption (default)

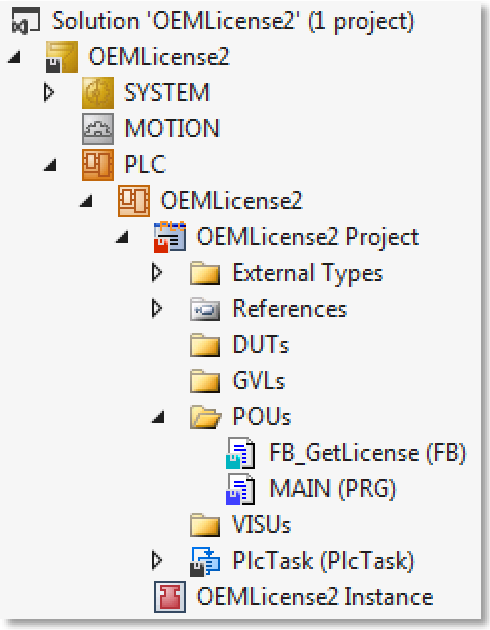
Encrypt boot project

⇨ The boot project is stored encrypted on the target system when it is activated for the target system.

Neither a user database nor an OEM certificate is required for **decryption** of the boot project on the **target system**.

7.1.4 Displaying the object protection status

The status of a TwinCAT object is indicated by the disk symbol in the object icon in the project tree.



To display the protection status of a TwinCAT object, the normal status display of the TwinCAT object is expanded. The following table shows the symbols and their meaning.

TwinCAT object status symbols

Symbol	Meaning
💾	No changes
💾	Unsaved changes
💾	Signed
💾	Encrypted

Rules:

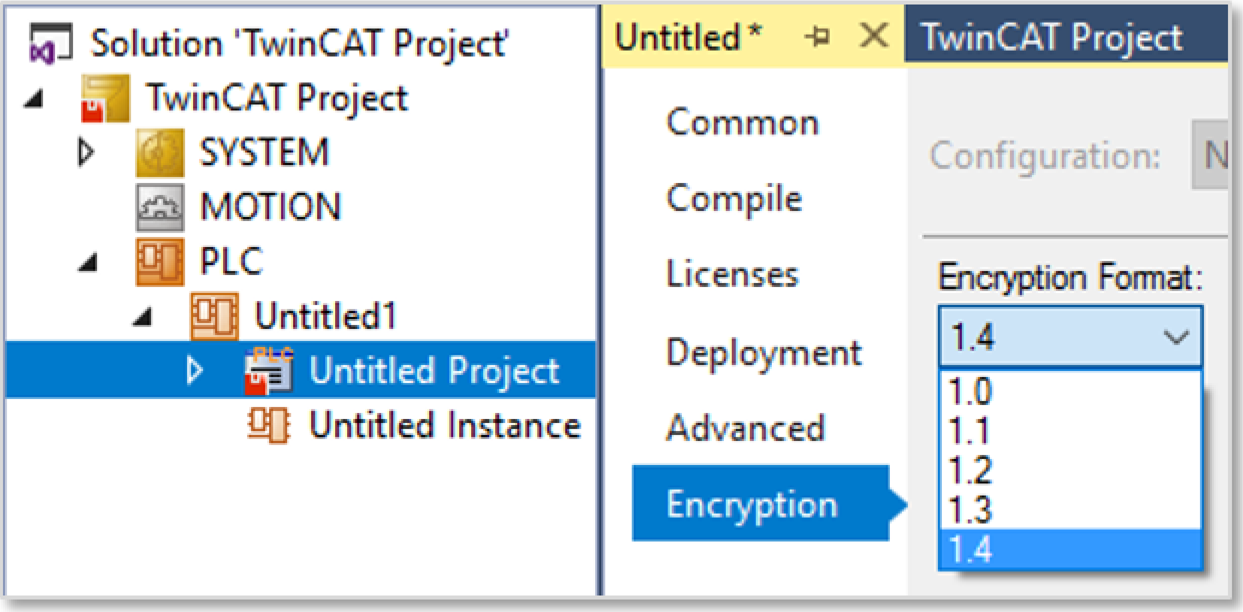
1. Turquoise overrides blue
2. Red overrides all other colours

7.1.5 Display of the current encryption version

The current TwinCAT version uses the current encryption version. For TwinCAT build 4022.x, this was version 1.4 at the time this documentation was created.

Previously, versions 1.0-1.3 were used (build 4020.x). The use of these encryption versions is strongly discouraged. Always use the latest available version.

The currently used encryption version can be viewed in the properties of a project:



Encryption Format: 1.4

If it is an older project (created with build 4020.x), a newer encryption version can be set here.

The current encryption version is only available in the current TwinCAT 3 build. The TwinCAT 3 build 4020.x versions, do not support encryption version 1.4, for example.

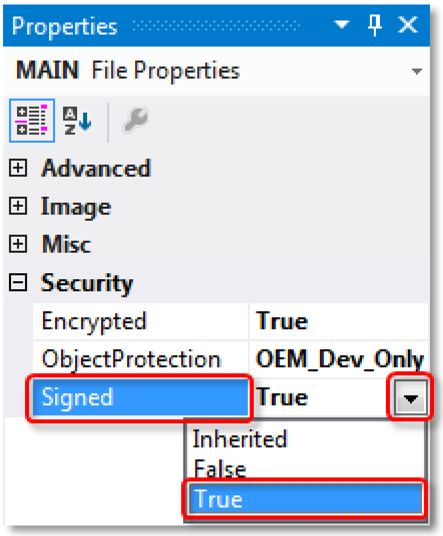
A prerequisite for secure encryption is the use of a current TwinCAT version with the current encryption version!

7.2 Signing files (protection against unauthorized changes)

By signing project components (files), you ensure that individual components cannot be replaced without authorization.

You should also sign the project file itself, since the information on which components must be signed is stored there.

If the project is linked to a user database, you can set the signing in the properties of the respective project component. Mark the project components in the Solution Explorer and set the value of the **Signed** property in the **Properties** view to TRUE.



Encrypted	True
ObjectProtection	OEM_Dev_Only
Signed	True

System requirements

Operating system:

- At least Windows 7 (or its Embedded version) is required in order to be able to use all the functions for the protection of the application software. Windows XP and Windows CE (Windows Embedded Compact) do not support either the encryption of the boot file or OEM licenses.

TwinCAT version:

- The functionalities described require TwinCAT 3.1 build 4022 or higher.

Reliable protection only when using the latest TwinCAT 3 version

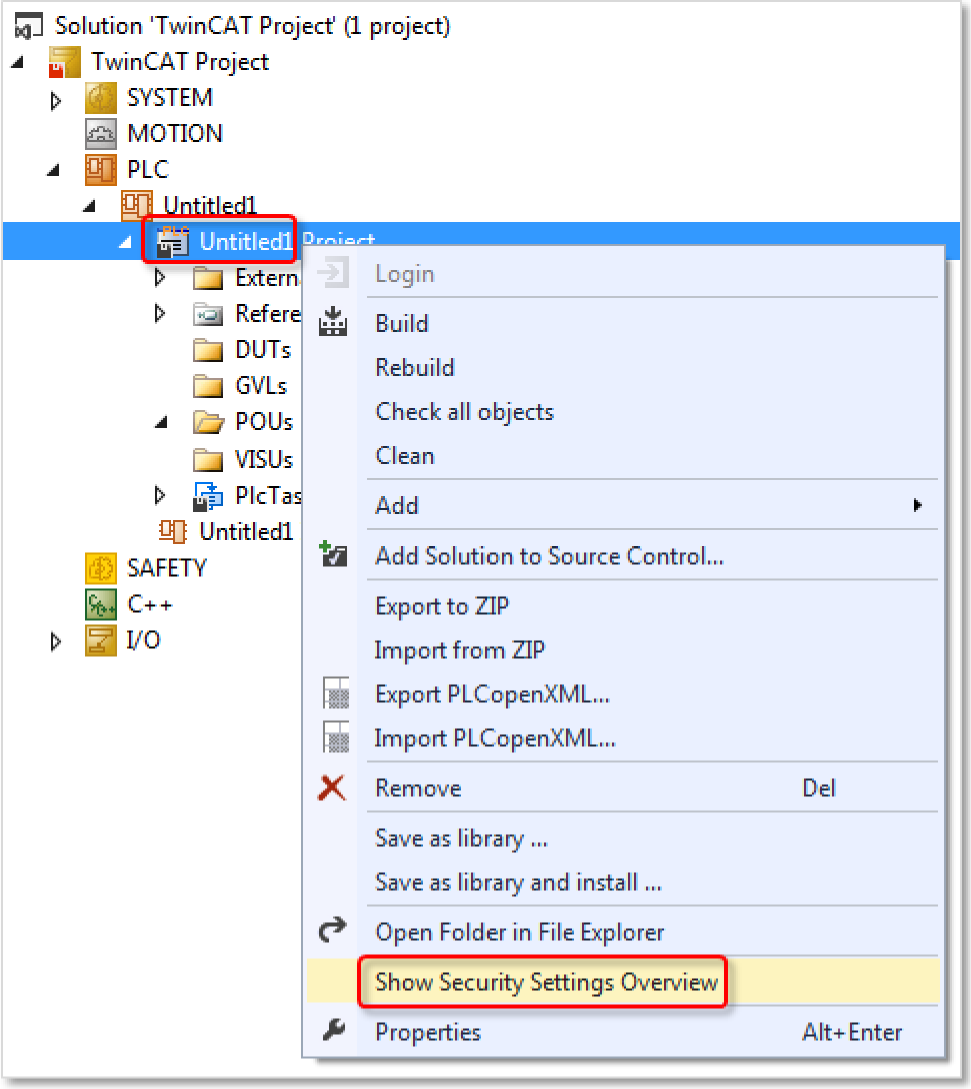
For reliable protection (e.g. secure encryption), always use the latest TwinCAT 3 version. This provides the maximum security.

Use at least TwinCAT 3.1 Build 4024.x.
For security reasons, do not use an older version!

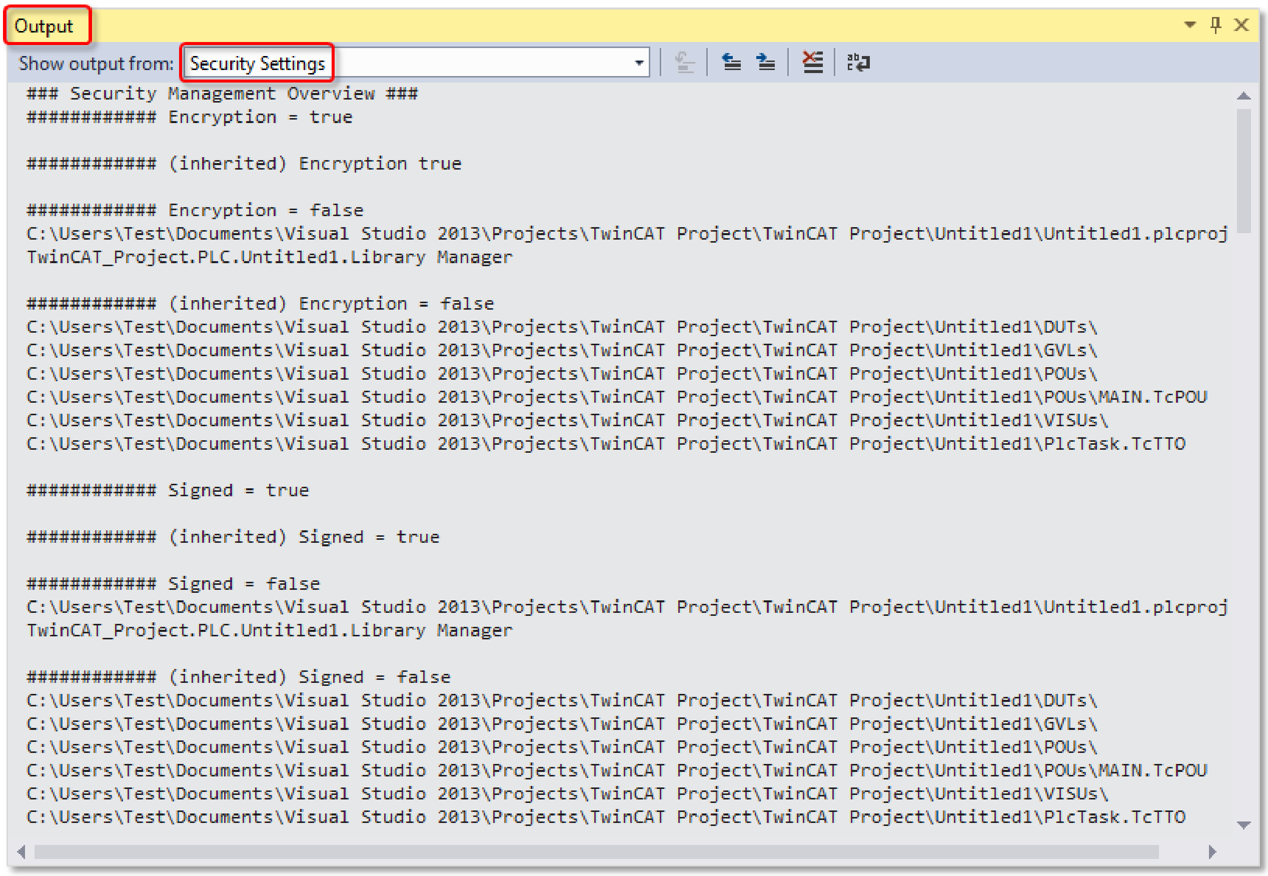
7.3 Displaying the overview of the software protection settings of the PLC project

You can display the settings for the protection of the PLC application software in the output window of the TwinCAT 3 development environment.

Mark the root node of the PLC project in the Solution Explorer and select the **Show Security Settings Overview** command in the context menu.



A summary of the current project security settings is shown in the output window.



```
### Security Management Overview ###
############ Encryption = true

############ (inherited) Encryption true

############ Encryption = false
C:\Users\Test\Documents\Visual Studio 2013\Projects\TwinCAT Project\TwinCAT Project\Untitled1\Untitled1.plcproj
TwinCAT_Project.PLC.Untitled1.Library Manager

############ (inherited) Encryption = false
C:\Users\Test\Documents\Visual Studio 2013\Projects\TwinCAT Project\TwinCAT Project\Untitled1\DUTs\
C:\Users\Test\Documents\Visual Studio 2013\Projects\TwinCAT Project\TwinCAT Project\Untitled1\GVLs\
C:\Users\Test\Documents\Visual Studio 2013\Projects\TwinCAT Project\TwinCAT Project\Untitled1\POUs\
C:\Users\Test\Documents\Visual Studio 2013\Projects\TwinCAT Project\TwinCAT Project\Untitled1\POUs\MAIN.TcPOU
C:\Users\Test\Documents\Visual Studio 2013\Projects\TwinCAT Project\TwinCAT Project\Untitled1\VISUs\
C:\Users\Test\Documents\Visual Studio 2013\Projects\TwinCAT Project\TwinCAT Project\Untitled1\PlcTask.TcTTO

############ Signed = true

############ (inherited) Signed = true

############ Signed = false
C:\Users\Test\Documents\Visual Studio 2013\Projects\TwinCAT Project\TwinCAT Project\Untitled1\Untitled1.plcproj
TwinCAT_Project.PLC.Untitled1.Library Manager

############ (inherited) Signed = false
C:\Users\Test\Documents\Visual Studio 2013\Projects\TwinCAT Project\TwinCAT Project\Untitled1\DUTs\
C:\Users\Test\Documents\Visual Studio 2013\Projects\TwinCAT Project\TwinCAT Project\Untitled1\GVLs\
C:\Users\Test\Documents\Visual Studio 2013\Projects\TwinCAT Project\TwinCAT Project\Untitled1\POUs\
C:\Users\Test\Documents\Visual Studio 2013\Projects\TwinCAT Project\TwinCAT Project\Untitled1\POUs\MAIN.TcPOU
C:\Users\Test\Documents\Visual Studio 2013\Projects\TwinCAT Project\TwinCAT Project\Untitled1\VISUs\
C:\Users\Test\Documents\Visual Studio 2013\Projects\TwinCAT Project\TwinCAT Project\Untitled1\PlcTask.TcTTO
```

8 Issuing and using your own OEM licenses

With the help of TwinCAT 3 license technology a PLC application can be protected against cloning through binding to hardware (Beckhoff IPC or TwinCAT dongle). Also, additional functions of the application can be licensed to end users through the creation of so-called "feature licenses".

Here you can find the Quick start.

System requirements

Operating system:

- At least Windows 7 (or its Embedded version) is required in order to be able to use all the functions for the protection of the application software. Windows XP and Windows CE (Windows Embedded Compact) do not support either the encryption of the boot file or OEM licenses.

TC3 PLC Lib Tc2_Utiltities:

- Use at least version 3/3/24 of the TC3 PLC Lib Tc2_Utilities, as they provide various functions for the comfortable handling of TwinCAT 3 licenses. It is mandatory for the use of TwinCAT 3 dongles for OEM application licenses. The TC3 PLC Lib is included from TwinCAT 3.1 Build 4022.16.

TwinCAT Version:

- The functionalities described above require TwinCAT 3.1 build 4024 or higher.

Reliable protection only when using the latest TwinCAT 3 version

For reliable protection (e.g. secure encryption), always use the latest TwinCAT 3 version. This provides the maximum security.

Use at least TwinCAT 3.1 Build 4024.x.
For security reasons, do not use an older version!

General notes

If you use OEM licenses make sure you encrypt your boot project!

Remember that the license ID [▸ 85] queried via FB_CheckLicense [▸ 92] in the binary code can easily be found and (with a little effort) manipulated with a hex editor. Therefore, be sure to encrypt your boot project [▸ 77] (safest), or at least disguise the queried license ID in the source code as best as possible.

- A user database is not required for the application licensing.
- The license is validated by the TwinCAT 3 runtime (XAR). The TwinCAT 3 runtime must therefore be installed on the IPC.
- The validity of the application license is independent of the validity period of the OEM certificate. The application license thus remains valid even after the validity of the OEM certificate has expired.
- The use of OEM application licenses always requires a TwinCAT 3 dongle or a Beckhoff IPC.
- For IPCs with a platform level >= 90 (non-Beckhoff IPCs) a TwinCAT-3 dongle must always be used as a "License Device" for security reasons!

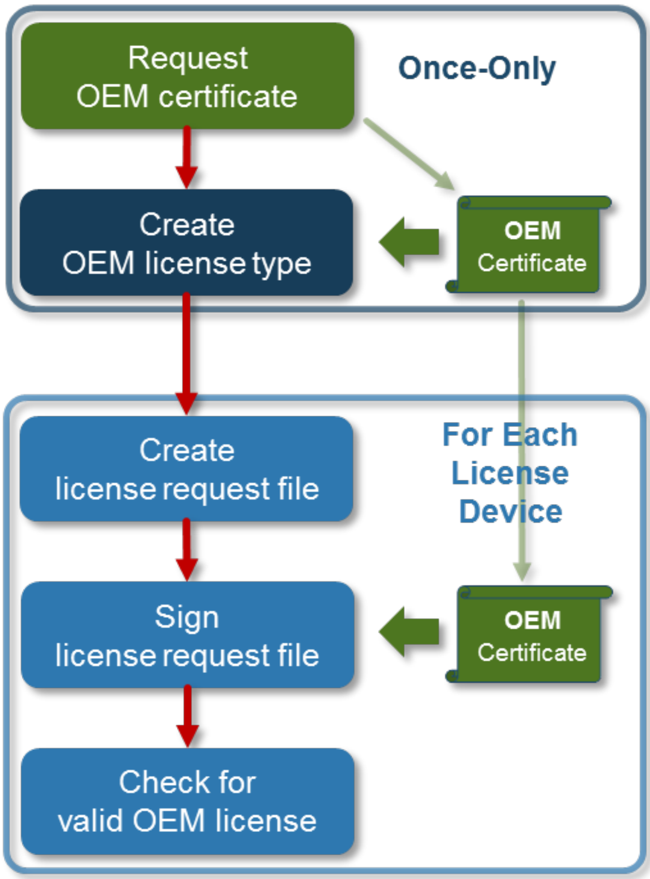
Prerequisite for using this function: Issue of a TwinCAT OEM certificate [▸ 19]

Licensing process

The licensing process is subdivided into the following steps:

1. Creating a general license description file.
   The license description file is used for describing and selecting a specific license type during the licensing process. Among other information it contains a unique license ID, which is used to unambiguously identify the license type.
2. Creating a License Request File for the required target system.

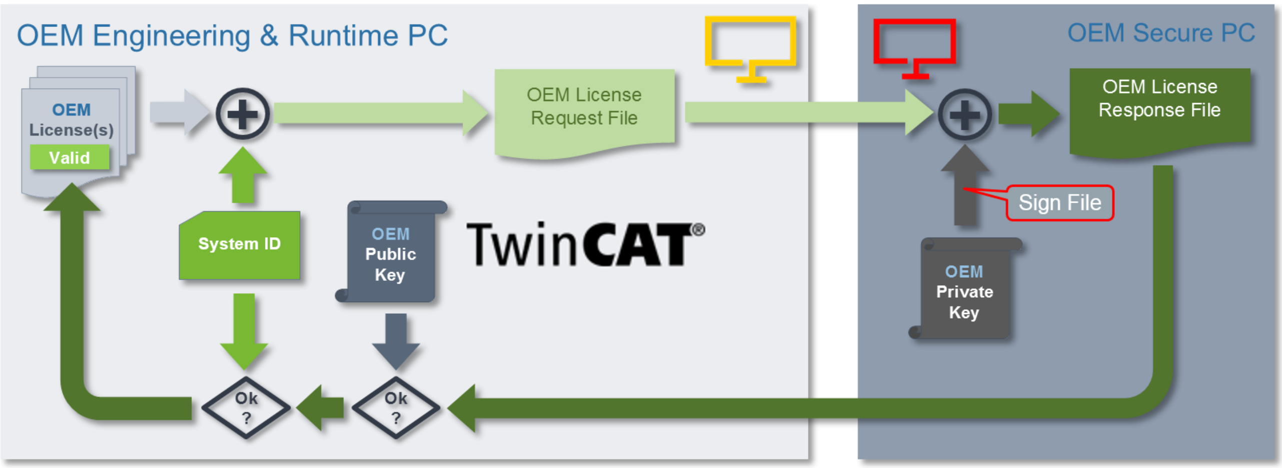
3. Signing the License Request File with the OEM certificate, thereby creating a License Response File for the specified target system. This activates the corresponding OEM application license on the respective target system.



The details of the licensing process are described in the following sections.

8.1 Creating OEM application licenses

The following diagram provides a general overview of the licensing process:



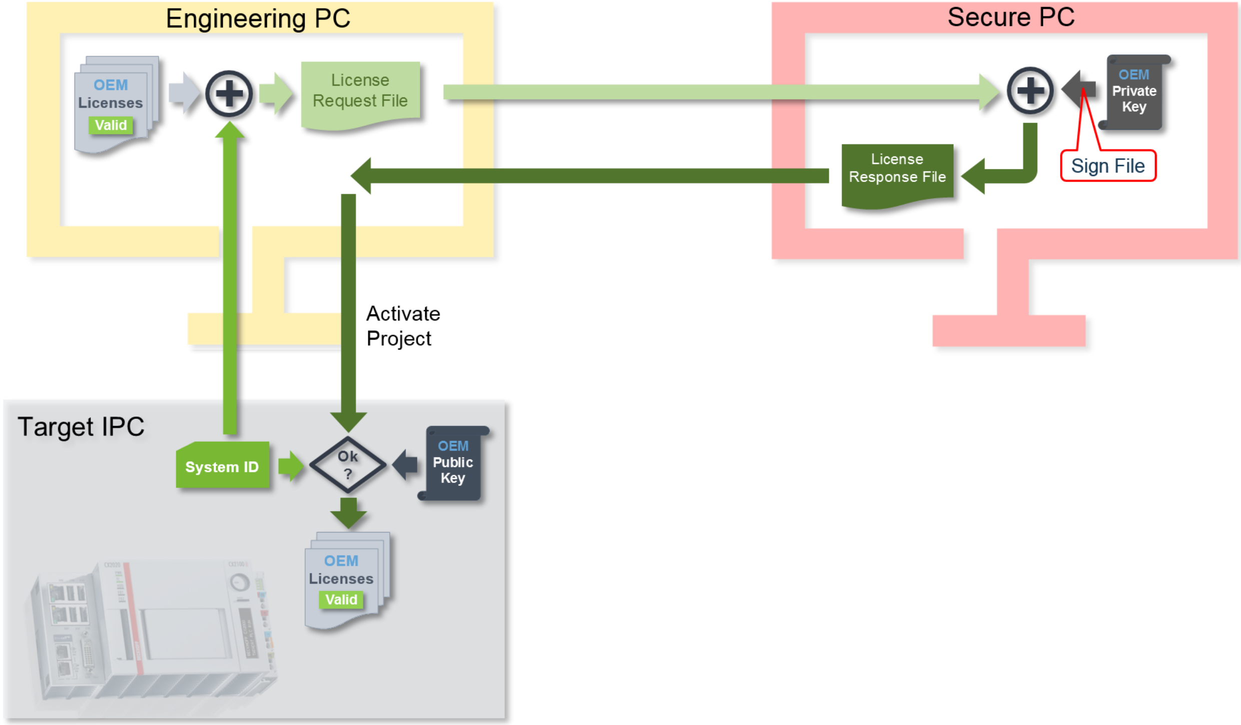
The left part of the diagram (light grey box) illustrates the creation of a License Request Files for a TwinCAT 3 license and its verification in the TwinCAT 3 Runtime.

The right part of the diagram (dark grey box) illustrates the licensing processes, which are handled by the Beckhoff license server.
The issuing process of an OEM application license is handled by the OEM through signing with the OEM private key. In other words, the Beckhoff license server is not integrated in the OEM application license generation process.

OEM certificates should only be used in a secure environment

Since generating an OEM application license requires handling of the OEM certificate and its password, the process should only be carried out in an environment that is protected against malicious software (protected PC), in order to prevent the password for the OEM private key being accessed by malicious software.

If the control computer and the Engineering computer are separate devices, the process flow looks as follows:



8.1.1 Preparing TwinCAT 3 Engineering

By default, TwinCAT Engineering is not preconfigured for generating OEM licenses.

1. Create the following directories:
   - <TwinCAT_ROOT>*\3.1\CustomConfig\Licenses*
   - <TwinCAT_ROOT>*\3.1\Components\Base\License*
2. Copy the tool “CreateLicense.exe*” into the directory *c:\TwinCAT\3.1\Components\Base\License*. This tool can be requested by sending an email to tccertificate@beckhoff.com.

TwinCAT root directory <TwinCAT_ROOT>

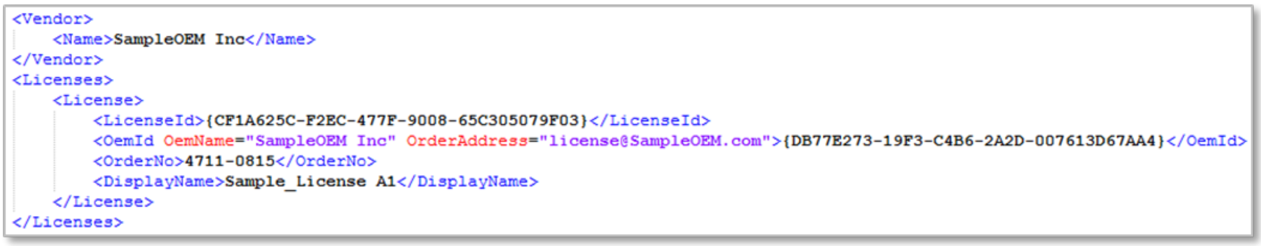
Up to and including TwinCAT 3.1.4024: **C:\TwinCAT**
From TwinCAT 3.1.4026: **C:\ProgramData\Beckhoff\TwinCAT**

8.1.2 Creating a license description file for an OEM application license

The type parameters for a TwinCAT 3 license are specified in a license description file in XML format with the extension .tmc.

A license description file contains:

- a one-to-one “License ID”, which makes the license type reliably identifiable
- the one-to-one OEM ID (from the OEM certificate)
- the OEM name
- the name of the license type
- the order number
- optionally an email address for receiving the License Request File



```
<Vendor>
    <Name>SampleOEM Inc</Name>
</Vendor>
<Licenses>
    <License>
        <LicenseId>{CF1A625C-F2EC-477F-9008-65C305079F03}</LicenseId>
        <OemId OemName="SampleOEM Inc" OrderAddress="license@SampleOEM.com">{DB77E273-19F3-C4B6-2A2D-007613D67AA4}</OemId>
        <OrderNo>4711-0815</OrderNo>
        <DisplayName>Sample_License A1</DisplayName>
    </License>
</Licenses>
```

The OEM ID can be used to assign the license to a specific OEM. Only this OEM (with this OEM ID in its OEM certificate) can sign the license with its OEM certificate, thereby making it valid.

An OEM license description file can be opened and modified with a suitable editor. Ensure that the XML structure is not damaged.

Creating a new OEM license description file

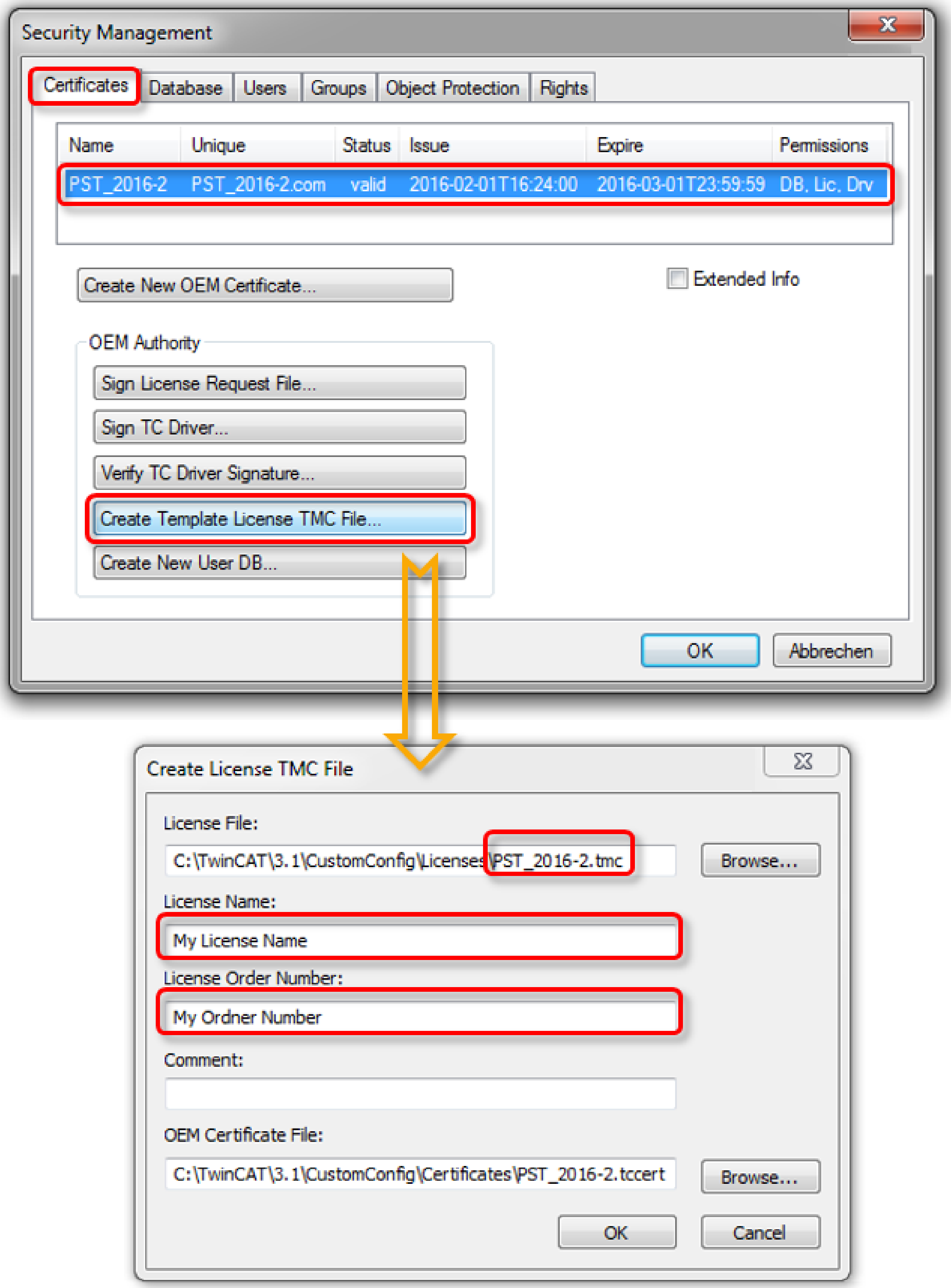
TwinCAT root directory <TwinCAT_ROOT>

Up to and including TwinCAT 3.1.4024: **C:\TwinCAT**
From TwinCAT 3.1.4026: **C:\ProgramData\Beckhoff\TwinCAT**

✓ The Software protection configurator [▶ 11] is opened.

1. In the **Certificates** tab, select the OEM certificate on the basis of which the OEM license description file is to be created.
2. Click on **Create Template License TMC File**.

   ⇨ The **Create Licenses TMC File** dialog opens.
3. Enter the parameters for the OEM license description file:
   - Save the license description file in the folder *<TwinCAT_ROOT>\3.1\CustomConfig\Licenses* and restart TwinCAT 3 Engineering. Only then will the license description file be recognized by TwinCAT 3.
   - Enter a license name and license order number.



Security Management

Name	Unique	Status	Issue	Expire	Permissions
PST_2016-2	PST_2016-2.com	valid	2016-02-01T16:24:00	2016-03-01T23:59:59	DB, Lic, Drv

□ Extended Info

Create License TMC File

License File: C:\TwinCAT\3.1\CustomConfig\Licenses\PST_2016-2.tmc

License Name: My License Name

License Order Number: My Ordner Number

Comment:

OEM Certificate File: C:\TwinCAT\3.1\CustomConfig\Certificates\PST_2016-2.tccert

4. Restart TwinCAT 3 Engineering, so that the new license type is detected.

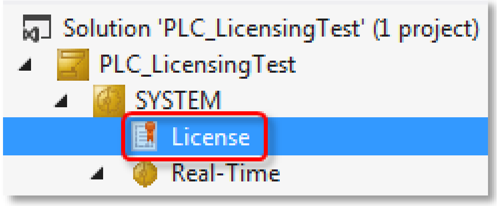
⇨ The license description file has been created.

8.1.3 Creating License Request Files for an OEM application license

TwinCAT 3 licenses for non-Beckhoff IPCs

If you use an IPC from a manufacturer other than Beckhoff (TwinCAT 3 platform level >= 90), a TwinCAT 3 license dongle is always required for licensing TwinCAT 3.

1. Double-click on the **License** SYSTEM sub-node in the TwinCAT project tree to open the TwinCAT 3 license manager.

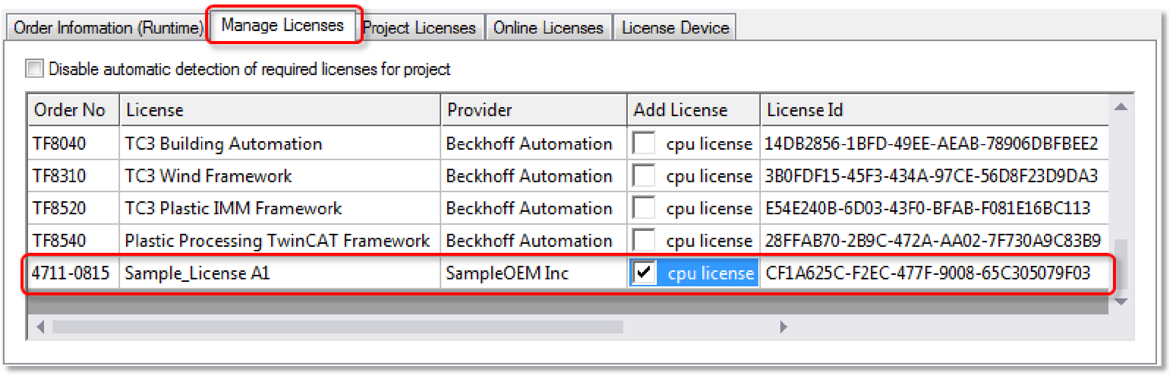


Solution 'PLC_LicensingTest' (1 project)
- PLC_LicensingTest
  - SYSTEM
    - License
    - Real-Time

⇨ The license settings open in an editor.

2. Open the **Manage Licenses** tab and scroll down.

⇨ The newly generated OEM license can be found at the end of the list.

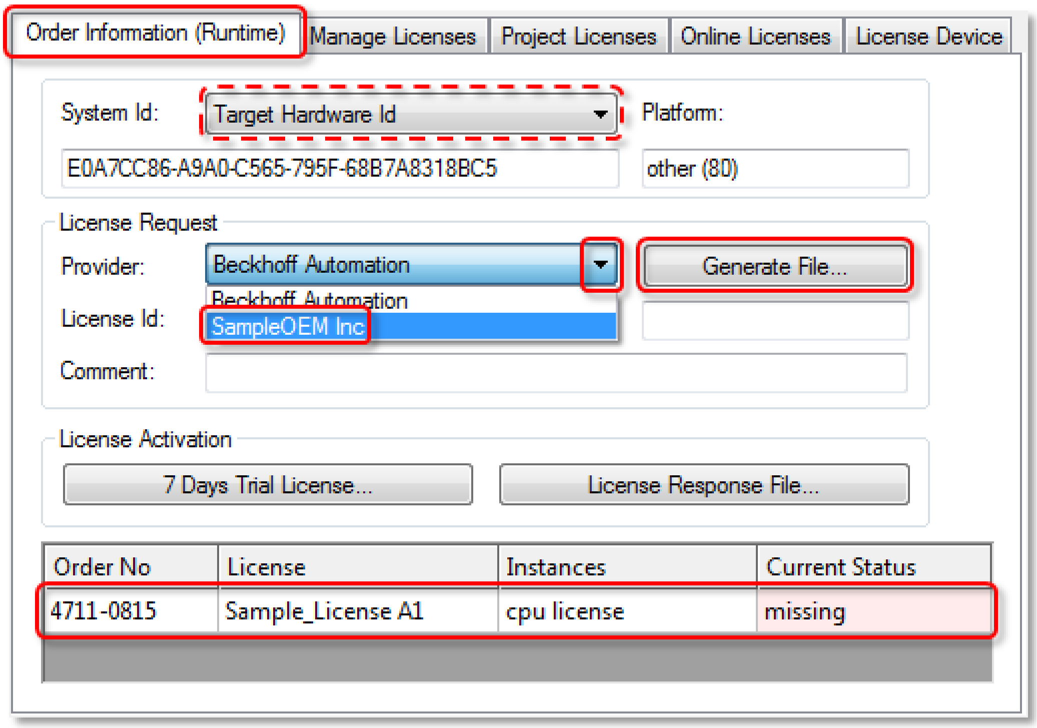


□ Disable automatic detection of required licenses for project

Order No	License	Provider	Add License	License Id
TF8040	TC3 Building Automation	Beckhoff Automation	□ cpu license	14DB2856-1BFD-49EE-AEAB-78906DBFBEE2
TF8310	TC3 Wind Framework	Beckhoff Automation	□ cpu license	3B0FDF15-45F3-434A-97CE-56D8F23D9DA3
TF8520	TC3 Plastic IMM Framework	Beckhoff Automation	□ cpu license	E54E240B-6D03-43F0-BFAB-F081E16BC113
TF8540	Plastic Processing TwinCAT Framework	Beckhoff Automation	□ cpu license	28FFAB70-2B9C-472A-AA02-7F730A9C83B9
4711-0815	Sample_License A1	SampleOEM Inc	☑ cpu license	CF1A625C-F2EC-477F-9008-65C305079F03

3. Tick the check box for the license.

4. Open the **Order Information** tab.



System Id: Target Hardware Id

E0A7CC86-A9A0-C565-795F-68B7A8318BC5

Platform: other (80)

License Request

Provider: Beckhoff Automation

License Id:

Comment:

License Activation

Order No	License	Instances	Current Status
4711-0815	Sample_License A1	cpu license	missing

5. You can optionally also choose a TwinCAT 3 dongle as the license hardware in **System ID** (dashed line).
6. Set the respective OEM as **Provider**. You may not select the “Beckhoff” entry – this applies only to TwinCAT 3 licenses from Beckhoff.
   ⇨ The selected OEM license must show up as active (i.e. not grayed out) in the list at the bottom of the window. If the license is grayed out, an incorrect “Provider” was selected. Only the licenses shown as “active” are transferred to the License Request File.
7. Click on **Generate File** to generate the License Request File (extension: *.tclrq).
   ⇨ The standard dialog for saving a file opens.
8. Select a storage location and confirm the dialog.

⇨ The License Request File for an OEM application license has been created.

8.1.4 Creating License Response Files for an OEM application license

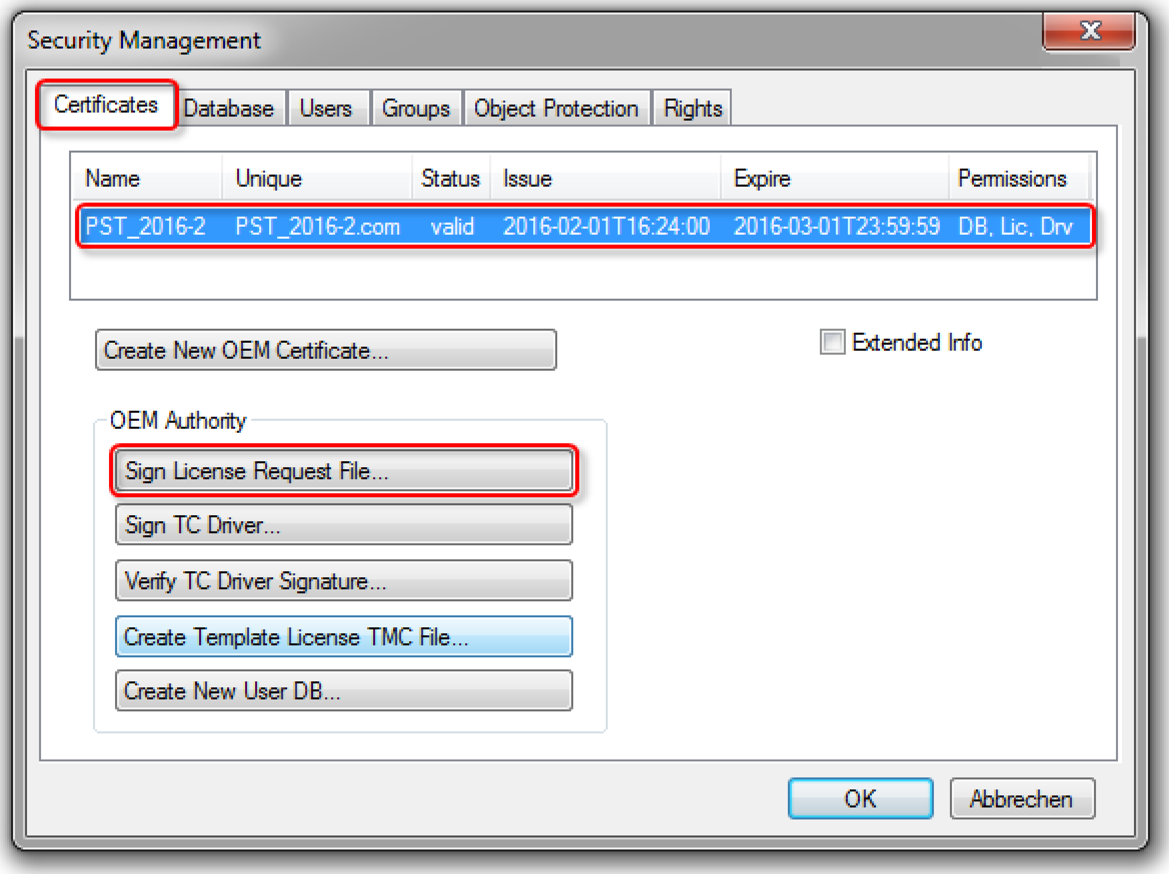
8.1.4.1 Manual creation via the TwinCAT Engineering

OEM certificates should only be used in a secure environment

Since generating an OEM application license requires handling of the OEM certificate and its password, the process should only be carried out in an environment that is protected against malicious software (protected PC), in order to prevent the password for the OEM private key being accessed by malicious software.

The manual signing of a License Request File, and thus the creation of a License Response File, is done in the TwinCAT Engineering in the Software protection configurator [▸ 11].

1. Select the OEM certificate in the **Certificates** tab.



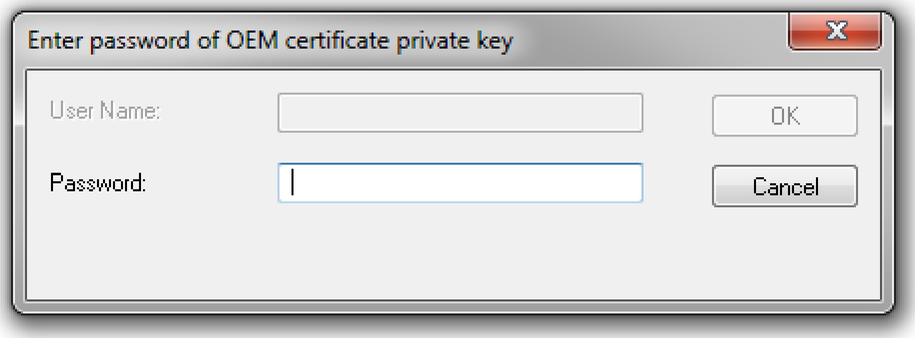
Name	Unique	Status	Issue	Expire	Permissions
PST_2016-2	PST_2016-2.com	valid	2016-02-01T16:24:00	2016-03-01T23:59:59	DB, Lic, Drv

2. Click on **Sign License Request File**.

⇨ An Explorer window opens.

3. Select the License Request File to be signed (extension: *.tclrq).

⇨ A password dialog opens.



Enter password of OEM certificate private key

User Name:

Password:

4. Enter the password and click on **OK**.

⇨ The License Request File is signed, and the result is stored as a License Response File (extension: *.tclrs). The License Response File now has to be transferred back to the Engineering PC or the control computer.

8.1.4.2 Automated creation via a command line tool

OEM certificates should only be used in a secure environment

Since generating an OEM application license requires handling of the OEM certificate and its password, the process should only be carried out in an environment that is protected against malicious software (protected PC), in order to prevent the password for the OEM private key being accessed by malicious software.

TwinCAT root directory <TwinCAT_ROOT>

Up to and including TwinCAT 3.1.4024: **C:\TwinCAT**
From TwinCAT 3.1.4026: **C:\ProgramData\Beckhoff\TwinCAT**

The TwinCAT 3 Engineering uses a command line tool (TcSignTool.exe) to issue (sign) OEM licenses. This tool can also be called from a user program for the automated issuing of OEM licenses.

TcSignTool.exe is located in a TwinCAT 3 installation in the path *<TwinCAT_ROOT>\3.1\sdk\Bin*.

Call parameters

tcsigntool licsign /f certificatefile [/p password] [/i issueTime] [/d validDays] [/q] licfile1 [licfile2]

- certificatefile: OEM certificate file
- password: password for the OEM certificate
- issueTime: Format yyyy-mm-ddThh:mm:ss (default value = current time)
- validDays: Default value = Unlimited
- licfile<n>: License Request or Response file with extension '.tclrq' or '.tclrs'. License Request files with the extension '.tclrq' will be renamed '.tclrs'.
- /q Quiet Mode
- Return values: 0 = Succeeded, 1 = Failed

8.1.5 Importing License Response Files for an OEM application license

TwinCAT 3 licenses for non-Beckhoff IPCs

If you use an IPC from a manufacturer other than Beckhoff (TwinCAT 3 platform level >= 90), a TwinCAT 3 license dongle is always required for licensing TwinCAT 3.

TwinCAT root directory <TwinCAT_ROOT>

Up to and including TwinCAT 3.1.4024: **C:\TwinCAT**
From TwinCAT 3.1.4026: **C:\ProgramData\Beckhoff\TwinCAT**

The OEM application license is activated in the same way as a standard TwinCAT 3 license. The simplest way to activate a TwinCAT 3 License Response file in TwinCAT 3 is to import it via the TwinCAT 3 license manager. Further information can be found in the “Licensing” documentation in the section Importing and activating a License Response File.

However, you can also save the license file directly on the target system in the directory *<TwinCAT_ROOT>3.1\target\license.*

The procedure to save the license file on a TwinCAT 3 dongle is described in the documentation "Licensing" in the section Saving license files manually on the dongle.

8.2 Storing OEM application licenses on a dongle

There are two options for storing an OEM application license on a license dongle, which are described in the TwinCAT 3 licensing section:

- Saving license files manually on the dongle
- PLC function blocks relating to the storage function of the license dongles

8.3 Querying the OEM application license in a PLC application

If you use OEM licenses make sure you encrypt your boot project!

Remember that the license ID [▶ 85] queried via FB_CheckLicense [▶ 92] in the binary code can easily be found and (with a little effort) manipulated with a hex editor. Therefore, be sure to encrypt your boot project [▶ 77] (safest), or at least disguise the queried license ID in the source code as best as possible.

The license check in the TwinCAT Runtime startup process takes place in two steps:

1. TwinCAT 3 first reads the license files stored in the system (on the hard disk), checks their contents and creates an internal list of the licenses found.
2. The final check of the licenses takes place after commissioning of the EtherCAT bus, because only then all necessary information is available. (Before this, the presence of an EL6070 license terminal, for example, cannot be verified)

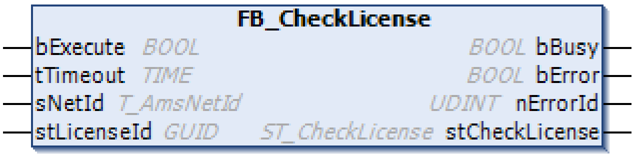
You can retrieve the result after the completed startup with the function block FB_CheckLicense.

During operation (after startup and final license check) the status of the licenses is checked again by the TwinCAT Runtime approx. **every two minutes**. This should be taken into account accordingly in the PLC program (e.g. call FB_CheckLicense only every 10 seconds).

Notes:

- FB_CheckLicense only reads the currently stored license status in the internal table, but does not trigger a new license check. Removing a dongle while the system is running can therefore take up to two minutes before the license status of the associated license becomes noticeable.
- Tip: if required, dongles currently connected to the system can be determined using the function block FB_GetLicenseDongles.
- The license check is part of the TwinCAT Runtime startup process. Means: no running runtime = no current license information!

FB_CheckLicense



The function block determines the TwinCAT 3 license status for a given license ID.

VAR_INPUT

```
VAR_INPUT
    bExecute    : BOOL;
    tTimeout    : TIME;
    sNetId      : T_AmsNetId;
    stLicenseId : GUID;
END_VAR
```

bExecute: The function block is activated by a positive edge at this input.

tTimeout: Timeout time that must not be exceeded when the command is executed.

sNetId: AmsNetId (AMS network identifier) of the TwinCAT computer whose license status is to be read (type: T_AmsNetId). If it is to be run on the local computer, an empty string can be entered.

stLicenseId: License ID (type: GUID)

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy          : BOOL;
    bError         : BOOL;
    nErrorId       : UDINT;
    stCheckLicense : ST_CheckLicense
END_VAR
```

bBusy: TRUE, as long as the function block is active.

bError: TRUE if an error occurs during command execution.

nErrorId: Supplies the ADS error number when the bError output is set.

stCheckLicense: Structure with license data (type: ST_CheckLicense [▶ 92])

STRUCT ST_CheckLicense

```
TYPE ST_CheckLicense :
STRUCT
    stLicenseId     : GUID;
    tExpirationTime : TIMESTRUCT;
    sExpirationTime : STRING(80);
    eResult         : E_LicenseHResult;
    nCount          : UDINT;
END_STRUCT
END_TYPE
```

Name	Description
stLicenseId	License ID
tExpirationTime	Expiry date
sExpirationTime	Expiry date
eResult	License status (see E_LicenseHResult [▶ 92])
nCount	Number of instances for this license (0=unlimited)

ENUM E_LicenseHResult

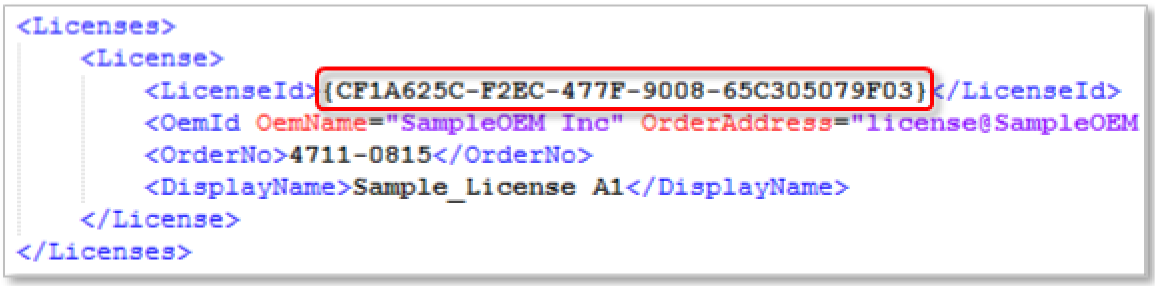
```
TYPE E_LicenseHResult :
(
    //success
    E_LHR_LicenseOK                : DINT := 0,
    E_LHR_LicenseOK_Pending        : DINT := 16#203,
    E_LHR_LicenseOK_Demo           : DINT := 16#254,
    E_LHR_LicenseOK_OEM            : DINT := 16#255,
    //error
    E_LHR_LicenseNoFound           : DINT := DWORD_TO_DINT(16#98110700+16#24),
    E_LHR_LicenseExpired           : DINT := DWORD_TO_DINT(16#98110700+16#25),
    E_LHR_LicenseExceeded          : DINT := DWORD_TO_DINT(16#98110700+16#26),
    E_LHR_LicenseInvalid           : DINT := DWORD_TO_DINT(16#98110700+16#27),
    E_LHR_LicenseSystemIdInvalid   : DINT := DWORD_TO_DINT(16#98110700+16#28),
    E_LHR_LicenseNoTimeLimit       : DINT := DWORD_TO_DINT(16#98110700+16#29),
    E_LHR_LicenseTimeInFuture      : DINT := DWORD_TO_DINT(16#98110700+16#2A),
    E_LHR_LicenseTimePeriodToLong  : DINT := DWORD_TO_DINT(16#98110700+16#2B),
    E_LHR_DeviceException          : DINT := DWORD_TO_DINT(16#98110700+16#2C),
    E_LHR_LicenseDuplicated        : DINT := DWORD_TO_DINT(16#98110700+16#2D),
    E_LHR_SignatureInvalid         : DINT := DWORD_TO_DINT(16#98110700+16#2E),
    E_LHR_CertificateInvalid       : DINT := DWORD_TO_DINT(16#98110700+16#2F),
    E_LHR_LicenseOemNotFound       : DINT := DWORD_TO_DINT(16#98110700+16#30),
    E_LHR_LicenseRestricted        : DINT := DWORD_TO_DINT(16#98110700+16#31),
    E_LHR_LicenseDemoDenied        : DINT := DWORD_TO_DINT(16#98110700+16#32),
    E_LHR_LicensePlatformLevelInv  : DINT := DWORD_TO_DINT(16#98110700+16#33)
)DINT;
END_TYPE
```

Value	Meaning
E_LHR_LicenseOK	License is valid
E_LHR_LicenseOK_Pending	Validation of the licensing device (e.g. License Key Terminal) required
E_LHR_LicenseOK_Demo	Trial license is valid
E_LHR_LicenseOK_OEM	OEM license is valid
E_LHR_LicenseNoFound	Missing license
E_LHR_LicenseExpired	License expired
E_LHR_LicenseExceeded	License has too few instances
E_LHR_LicenseInvalid	License is invalid
E_LHR_LicenseSystemIdInvalid	Incorrect system ID for the license
E_LHR_LicenseNoTimeLimit	License not limited in time
E_LHR_LicenseTimeInFuture	License problem: Time of issue is in the future
E_LHR_LicenseTimePeriodToLong	License period too long
E_LHR_DeviceException	Exception at system startup
E_LHR_LicenseDuplicated	License data read multiple times
E_LHR_SignatureInvalid	Invalid signature
E_LHR_CertificateInvalid	Invalid certificate
E_LHR_LicenseOemNotFound	OEM license for unknown OEM
E_LHR_LicenseRestricted	License invalid for the system
E_LHR_LicenseDemoDenied	Trial license not allowed
E_LHR_LicensePlatformLevelInv	Invalid platform level for the license

Determining the license ID of the OEM license

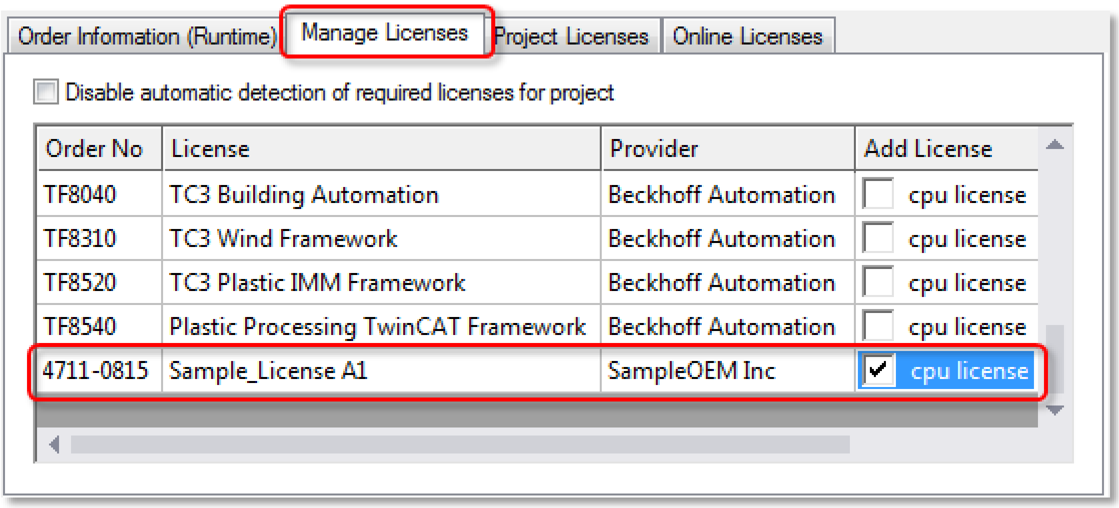
The license ID of the OEM license can be obtained from the corresponding license description file or the license manager.

License description file:



```
<Licenses>
    <License>
        <LicenseId>{CF1A625C-F2EC-477F-9008-65C305079F03}</LicenseId>
        <OemId OemName="SampleOEM Inc" OrderAddress="license@SampleOEM
        <OrderNo>4711-0815</OrderNo>
        <DisplayName>Sample_License A1</DisplayName>
    </License>
</Licenses>
```

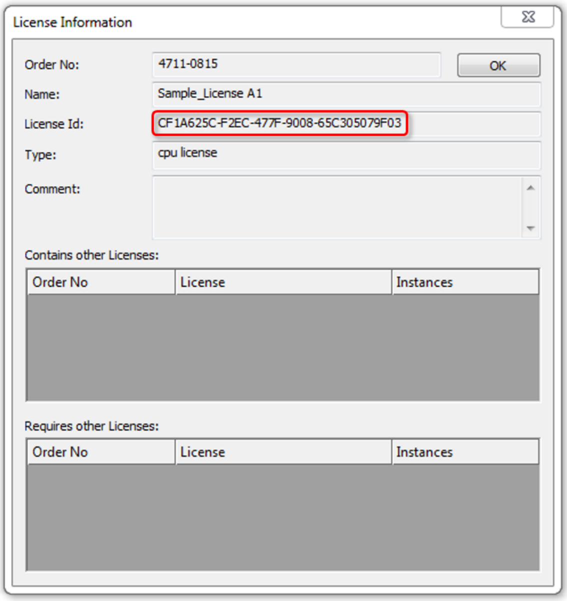
"Manage Licenses" tab of the license manager:



□ Disable automatic detection of required licenses for project

Order No	License	Provider	Add License
TF8040	TC3 Building Automation	Beckhoff Automation	□ cpu license
TF8310	TC3 Wind Framework	Beckhoff Automation	□ cpu license
TF8520	TC3 Plastic IMM Framework	Beckhoff Automation	□ cpu license
TF8540	Plastic Processing TwinCAT Framework	Beckhoff Automation	□ cpu license
4711-0815	Sample_License A1	SampleOEM Inc	☑ cpu license

Double-clicking on the row containing the license line opens a window showing the license properties, including the license ID:



Order No: 4711-0815

Name: Sample_License A1

License Id: CF1A625C-F2EC-477F-9008-65C305079F03

Type: cpu license

Comment:

Contains other Licenses:

Order No	License	Instances

Requires other Licenses:

Order No	License	Instances

The OEM can specify in their PLC application how the system should respond to the presence or absence of the OEM application license. Options include program termination or activation of an additional feature.

System requirements

Operating system:

- At least Windows 7 (or its Embedded version) is required in order to be able to use all the functions for the protection of the application software. Windows XP and Windows CE (Windows Embedded Compact) do not support either the encryption of the boot file or OEM licenses.

TC3 PLC Lib Tc2_Utiltities:

- Use at least version 3/3/24 of the TC3 PLC Lib Tc2_Utilities, as they provide various functions for the comfortable handling of TwinCAT 3 licenses. It is mandatory for the use of TwinCAT 3 dongles for OEM application licenses. The TC3 PLC Lib is included from TwinCAT 3.1 Build 4022.16.

TwinCAT Version:

- The functionalities described above require TwinCAT 3.1 build 4024 or higher.

Reliable protection only when using the latest TwinCAT 3 version

For reliable protection (e.g. secure encryption), always use the latest TwinCAT 3 version. This provides the maximum security.

Use at least TwinCAT 3.1 Build 4024.x.
For security reasons, do not use an older version!

See also: documentation for the PLC library Tc2_Utilities , section Licensing functions

8.4 Providing OEM PLC libraries with license protection

Always query OEM license with FB_CheckLicense!

The method described below can be used as a supplement to the query with FB_CheckLicense (not as an alternative method).

The license status must always be queried with FB_CheckLicense [▶ 91], as this is the only way to determine a secure current license status.

This license check with FB_CheckLicense is completely sufficient; it is not necessary (and therefore not recommended) to additionally enter the License GUID in the properties of the self-created library.

With the entry of the License GUID additionally in the properties of the self-created library, the TwinCAT 3 Runtime knows that this license is required for the project, and a **first** check of this license is performed when the runtime is started.

This first check takes place very early in the start-up phase of the TwinCAT Runtime. The EtherCAT bus is not started until later in the startup process, for example; the presence of an EL6070 License Key Terminal can therefore only be verified afterwards.

It is therefore very important to perform a license check with FB_CheckLicense in every case, **after** the complete system has been started (and thus the EtherCAT bus is in operation).

The status of all licenses is checked by the TwinCAT Runtime (after startup) during operation approx. **every two minutes**. This should be taken into account accordingly in the PLC program (e.g. no call of FB_CheckLicense in each PLC cycle).

9 Protecting an application against cloning

See Issuing and using your own OEM licenses [▶ 83]

10 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Download finder

Our download finder contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963-157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963-460
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963-0
e-mail:	info@beckhoff.com
web:	www.beckhoff.com

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