# Table of contents

1 Foreword .................................................................................................................................................. 5  
  1.1 Notes on the documentation ............................................................................................................ 5  
  1.2 Safety instructions ............................................................................................................................ 6  

2 Introduction ............................................................................................................................................. 7  
  2.1 General system requirements ............................................................................................................. 7  
  2.2 Three pillars of software access protection ....................................................................................... 7  
  2.3 User database as a central switching point ....................................................................................... 8  
  2.4 Software protection configurator ...................................................................................................... 10  

3 Quick start ............................................................................................................................................... 12  
  3.1 Control of access to the source code ................................................................................................. 12  
  3.2 Protection of the application against cloning .................................................................................. 14  

4 TwinCAT OEM certificates ....................................................................................................................... 18  
  4.1 Creating the "OEM Certificate Request File" ..................................................................................... 21  
      4.1.1 Using build 4022 to create a certificate .................................................................................... 26  
  4.2 Determining the file fingerprint of the OEM certificate file .............................................................. 28  
  4.3 Requesting the OEM certificate ......................................................................................................... 29  
  4.4 Installing the OEM certificate ........................................................................................................... 30  
  4.5 Extending an OEM certificate ............................................................................................................ 32  
  4.6 Updating an existing OEM certificate? .............................................................................................. 32  

5 User databases (user DBs) ....................................................................................................................... 34  
  5.1 Creating a user database .................................................................................................................... 34  
  5.2 Setting default settings for the user database in Visual Studio ......................................................... 39  
  5.3 Select the current user database in Visual Studio ............................................................................ 40  
  5.4 Default user in the user database ..................................................................................................... 42  
  5.5 Extensions for user databases .......................................................................................................... 42  
      5.5.1 Associated elements in the Software Protection configuration console .................................. 44  
      5.5.2 Manual procedure in the TwinCAT 3 Engineering ................................................................ 45  
      5.5.3 Automated procedure .............................................................................................................. 50  
  5.6 Expanding the user database ............................................................................................................. 50  
      5.6.1 Adding/changing database administrators .............................................................................. 50  
      5.6.2 Separating the database administrator and developer functions .......................................... 54  
      5.6.3 Adding users to a group ............................................................................................................. 55  
      5.6.4 Defining your own group access rights .................................................................................... 57  
  5.7 Linking the user database to a project .............................................................................................. 70  
  5.8 Assigning user access rights in the project ....................................................................................... 71  
  5.9 Distribution / exchange of user databases ....................................................................................... 73  

6 Logging in and selecting a user account ............................................................................................... 74  
  6.1 Build 4022 ......................................................................................................................................... 74  

7 Setting up basic protection of OEM application software .................................................................... 76  
  7.1 Encryption .......................................................................................................................................... 76  
      7.1.1 Source code encryption ............................................................................................................. 77  
      7.1.2 Project file encryption ............................................................................................................. 77  
      7.1.3 Encrypting the boot project ..................................................................................................... 77
# Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1.4 Displaying the object protection status</td>
<td>78</td>
</tr>
<tr>
<td>7.1.5 Display of the current encryption version</td>
<td>79</td>
</tr>
<tr>
<td>7.2 Signing files (protection against unauthorized changes)</td>
<td>80</td>
</tr>
<tr>
<td>7.3 Displaying the overview of the project software protection settings</td>
<td>81</td>
</tr>
<tr>
<td>8 Issuing and using your own OEM licenses</td>
<td>83</td>
</tr>
<tr>
<td>8.1 Creating OEM application licenses</td>
<td>84</td>
</tr>
<tr>
<td>8.1.1 Preparing TwinCAT 3 Engineering</td>
<td>85</td>
</tr>
<tr>
<td>8.1.2 Creating a license description file for an OEM application license</td>
<td>85</td>
</tr>
<tr>
<td>8.1.3 Creating License Request Files for an OEM application license</td>
<td>88</td>
</tr>
<tr>
<td>8.1.4 Creating License Response Files for an OEM application license</td>
<td>89</td>
</tr>
<tr>
<td>8.1.5 Importing License Response Files for an OEM application license</td>
<td>90</td>
</tr>
<tr>
<td>8.2 Storing OEM application licenses on a dongle</td>
<td>91</td>
</tr>
<tr>
<td>8.3 Querying the OEM application license in a PLC application</td>
<td>91</td>
</tr>
<tr>
<td>8.4 Providing OEM PLC libraries with license protection</td>
<td>95</td>
</tr>
<tr>
<td>9 Protecting an application against cloning</td>
<td>96</td>
</tr>
<tr>
<td>10 Support and Service</td>
<td>97</td>
</tr>
</tbody>
</table>
1 Foreword

1.1 Notes on the documentation

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

It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.

It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

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

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement.

No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.

Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:


with corresponding applications or registrations in various other countries.

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.
1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of symbols

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

<table>
<thead>
<tr>
<th><strong>DANGER</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Serious risk of injury!</td>
</tr>
<tr>
<td>Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>WARNING</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk of injury!</td>
</tr>
<tr>
<td>Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>CAUTION</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal injuries!</td>
</tr>
<tr>
<td>Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>NOTE</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Damage to the environment or devices</td>
</tr>
<tr>
<td>Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.</td>
</tr>
</tbody>
</table>

Tip or pointer

This symbol indicates information that contributes to better understanding.
2 Introduction

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

- Configurable access restrictions to the source code through the definition of user groups and the assignment of access levels ("Object Protection Level")
- Know-how protection through encryption of 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).

An OEM certificate signed by Beckhoff is required in order to be able to use the functions to protect the application software. The central switching point of the access protection is a user database.

Contents

<table>
<thead>
<tr>
<th>Introduction [7]</th>
<th>This section provides general information on the system requirements, the software access protection, the user database and the software protection configurator.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick start [12]</td>
<td>This section enables you to quickly familiarize yourself with the topic of &quot;Protection of the application against cloning&quot;.</td>
</tr>
<tr>
<td>TwinCAT OEM certificates [18]</td>
<td>This section describes how you can apply for, install and extend the OEM certificate required to protect application software.</td>
</tr>
<tr>
<td>User databases (user DBs) [34]</td>
<td>This section describes how to create user databases and connect them with the project.</td>
</tr>
<tr>
<td>Setting up basic protection of OEM application software [76]</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.

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 4022.x. Do not use an older version on any account.

2.2 Three pillars of software access protection

The complete protection of a project component against unauthorized access includes the following measures:

- Regulation of the access rights to the project components
• Encryption of the associated project file
• Signing of the associated project file

Encryption without setting the correct access level will protect the corresponding project 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.

Also, without a signature, a project file could be exchanged for a different project file with the same name.

2.3 User database as a central switching point

Access to the project components is regulated via a user database (User DB) [p. 34].

The content of the database is protected against unauthorized changes through signature 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 access rights are 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 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 available 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.

Hints about configuration can be found in the following sections:

- TwinCAT OEM certificates [18]
- User databases (user DBs) [34]
- Setting up basic protection of OEM application software [76]
Quick start

3 Quick start

3.1 Control of access to the source code

From version build 4022, TwinCAT 3 offers the option to encrypt PLC source code and to control access to the 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).

System requirements

- TwinCAT 3 OEM Certificate TC0007
- Operating system: at least Windows 7 (also as Embedded version)
- TwinCAT Version: at least TwinCAT 3.1 Build 4022

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 4022.x. Do not use an older version on any account.

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
- Make a note the passwords of the OEM certificate and of the administrator of the User DB. 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
Quick start

1. Requesting the OEM certificate [29] (order)
2. Installing the OEM certificate [30]
3. Creating a user database [34]
   - **Note**: You only need to define the administrator’s name and password and make no further settings (e.g. no other users) in the User DB.
4. Linking the user database to a project [70]
5. Setting the encryption of the project file [77]
6. Setting access rights for PLC project root [71]
3.2 Protection of the application against cloning

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".

System requirements

- TwinCAT 3 OEM certificate
- Operating system: at least Windows 7 (also as Embedded version)
- 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)

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 4022.x. Do not use an older version on any account.

General notes

- If you use OEM licenses make sure you encrypt your boot project!
  
  Remember that the license ID [85] queried via FB_CheckLicense [91] 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"* [21].

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

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/zip/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 [p. 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.

In addition, TwinCAT build 4024 features an optional special TwinCAT OEM certificate version for signing TwinCAT 3 C++ executables.

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, which can be used to sign TwinCAT 3 C++ executables
- 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. In future, **all OEM certificates must ordered through the official channel**, so that the address and contact data can be validated. (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, in addition signing of TwinCAT 3 C++ executables)

**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 [34] 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** [21] is created (not when the order is placed):
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 the application software, and a second OEM certificate with the encryption version of TwinCAT 3.1 build 4024 for signing of TwinCAT 3 C++ code.

Storage instructions for the application area: Protection of the 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
- Creating OEM application license description files (Basis for issuing OEM application licenses)
- Issuing (signing) of OEM application licenses

The OEM Certificate Standard (TC0007) is only required for these three purposes.
On which computer does the OEM Certificate Standard have 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 Standard (TC0007) is neither required for UserDBs, nor for the program flow, nor 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 of TwinCAT 3 C++ code
The OEM key included in certificate version TC0008 (TwinCAT OEM Extended Validation certificate) can be used to sign additional TwinCAT 3 C++ executables.

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

On which computer does the OEM Certificate Extended Validation have to be stored?
The OEM certificate should only be stored on the computer on which TwinCAT 3 C++ executables are 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 Extended Validation (TC0008) is not required for running the TwinCAT 3 C++ executables 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 [32])

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
If you have any questions, please contact the TwinCAT OEM certificate section (tccertificate@beckhoff.com).

The following documentation refers to TwinCAT 3.1 build 4024.

(Link to the relevant documentation for TwinCAT 3.1 build 4022 [26])

Order numbers for TwinCAT OEM certificates

TC0007: TwinCAT OEM Certificate Standard (TwinCAT software protection)
TC0008: TwinCAT OEM Certificate Extended Validation (like TC0007, in addition signing of TwinCAT 3 C++ executables)

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 [34] for the TwinCAT Software Protection may only be created with an OEM certificate with Crypto version 1!

✓ The Software protection configurator [10] is opened.
1. Select the Certificates tab.
2. Click **Create New**…

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

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

   - **OEM Name:**
   - **Unique Name (domain or email address):**
   - **Certificate for:**
     - Sign user DB
     - Sign OEM license request
     - Sign TwinCAT C++ executable (*.tmx)
   - **Crypto Version:** 2
• Enter your company name in the **OEM Name** text box. The name must have a clear reference to your company or your business unit.

• 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:
• 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. Note: Crypto version "1" can only be selected for a Standard certificate (TC0007).

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 [34] 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:

- 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!)

5. 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.

6. Issue a password for the OEM Private Key.

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

- 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 OEM certificates [29].

### 4.1.1 Using build 4022 to create a certificate

- The **Software protection configurator** [10] is opened.
- 1. Select the **Certificates** tab.
- 2. Click on **Create New OEM Certificate**.
3. Enter the required data for an "OEM Certificate Request File":
   - Enter your company name in the **OEM Name** text box. The name must have a clear reference to your company or your business unit.
   - 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.
   - No further information is required. If you are working with an older TwinCAT version, ignore any other fields in the input mask that may be present. When the certificate is issued, default values are used for these fields, and changes are ignored.

4. Once you have entered the data, click **Start** and select a directory to save the file.
   - A dialog for selecting a password for the OEM Private Key opens.

5. Issue a password for the OEM Private Key. Be sure to use a strong password for your OEM certificate!

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

   - 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 OEM certificates [29].

4.2 Determining the file fingerprint of the OEM certificate file

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

<table>
<thead>
<tr>
<th>System requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>This functionality requires TwinCAT 3.1 build 4024 of higher.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Note for OEM Certificate Request File</th>
</tr>
</thead>
<tbody>
<tr>
<td>The &quot;OEM Certificate Request File&quot; 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 &quot;TwinCAT OEM certificate file&quot; is used for both file versions in the following sections.</td>
</tr>
</tbody>
</table>

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".

Notes:

- 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" [21], the file is already in this directory.

Procedure:

1. Call up the TwinCAT 3 Software Protection Configurator

2. Select the "Certificates" tab
3. Check the "Extended Info" box
4. In the window scroll to the right until you see the Fingerprint column

Note: As an alternative to points 3 + 4, you can simply double-click the certificate line. The file fingerprint is then displayed in a pop-up window:

![Image showing the fingerprint pop-up window]

Note: 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 [18]
- Creating the "OEM Certificate Request File" [21]
- Extending an OEM certificate [32]

**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.
- 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 are different.
- 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, in addition signing of TwinCAT 3 C++ executables)

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 [34] for the TwinCAT Software Protection may only be created with an OEM certificate with Crypto version 1!

Overview of the ordering and validation process

1. Order "TC0007" or "TC0008" from your Beckhoff sales contact.
2. Important: Enter your contact details (= contact details of the inquirer) of the certificate (company name, address, contact name and email address of the contact) as the delivery address.
3. The email address of the inquirer must be a company email account (freemailers such as GMail or similar are not permitted) and correspond with the company name of the inquirer.
4. The Beckhoff certification team will contact the inquirer named in the delivery address to verify the contact details given in the order.
6. Use TwinCAT Engineering to determine the "File Fingerprint" of the OEM certificate file (see Determining the file fingerprint of the OEM certificate file [28]). Communicate this file fingerprint to the Beckhoff certification team as part of your contact data verification - they will legitimize your OEM certificate file for later signing.
7. The inquirer receives the necessary information and requirements for sending his "OEM certificate file" to the certification team.
8. The inquirer now sends the "OEM certificate file" with an informal request by e-mail to the contact person at the local Beckhoff certification body.
9. After signing your 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

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 `c:\twincat\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 the 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
- Creating OEM application license description files (Basis for issuing OEM application licenses)
- Issuing (signing) of OEM application licenses

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

**On which computer does the OEM Certificate Standard have 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 Standard (TC0007) is neither required for UserDBs, nor for the program flow, nor 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 of TwinCAT 3 C++ code

The OEM key included in certificate version TC0008 (TwinCAT OEM Extended Validation certificate) can be used to sign additional TwinCAT 3 C++ executables.

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

On which computer does the OEM Certificate Extended Validation have to be stored?
The OEM certificate should only be stored on the computer on which TwinCAT 3 C++ executables are 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 Extended Validation (TC0008) is not required for running the TwinCAT 3 C++ executables 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 [29]. 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 [34], OEM license description files [85] or OEM application licenses [89]?

TwinCAT User DB

• Application 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).

• Application 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 [70]. 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)

- 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 here [42].

5.1 Creating a user database

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 [34] 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 [10] 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...

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...
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/zip/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 dialog box]

- **Database File**: C:\TwinCAT\3.1\CustomConfig\UserDBs\StdUserDB.tcwd
- **Database Name**: StdUserDB
- **Database Unique Name**: StdUserDB/1.0
- **Database Admin**: Admin
- **Database Template**: C:\TwinCAT\3.1\Components\Base\UserDBTemplate\TemplateOEM.tcwd
- **Expire Time**: 22.11.2020
- **OEM Certificate File**

**Note**: 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 dialog box]

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:

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.

**Note:** You can create additional administrators or make changes later, after the database has been created.

**Note:** 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**.

   ![Password dialog](image)

**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. 

![Dialog box](image)

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

- **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.
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**.

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**

- **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.
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 [34].

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 [70].

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 [50].

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.
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.

Note 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.

Note 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 :**

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:

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

- 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.

5.5.2.1 Creating Extensions

- 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.

  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!

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

   ![Create new Extension dialog](image)

   **Note**: 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:

2. 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) ...
3. ... and thus validated:

![User databases (user DBs)](image)

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

### 5.5.2.2 Creating users in extensions

- **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.

- **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:

![Creating users in extensions](image)

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:

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) ...
2. ... and thus validated:

![Image]

### 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

<table>
<thead>
<tr>
<th><strong>No changes in settings of a user database when a project is open</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>No project may be open when changing the settings of a user database.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Changes in the database must be signed when saving</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

**Database template “TemplateOEM”**

The "TemplateOEM" 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 [▶ 42]
- Adding/changing database administrators [▶ 50]
- Separating the database administrator and developer functions [▶ 54]
- Adding more users with the “Developer” group assignment [▶ 55]

**Separate group rights [▶ 57]** 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/zip/9007208137629963.zip.

### 5.6.1 Adding/changing database administrators

<table>
<thead>
<tr>
<th><strong>No changes in settings of a user database when a project is open</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>No project may be open when changing the settings of a user database.</td>
</tr>
</tbody>
</table>

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:
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:

```plaintext
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:
Creating **new editing database administrators**

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

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.

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

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

- **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.

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

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 [48]" of user databases. This is described [here](#42).

The creation of new user groups and the adaptation of existing ones is explained in the chapter "Creating and editing user groups [62]".
Defining your own group access rights

- 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.

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/zip/9007208137629963.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 4022.x. Do not use an older version on any account.

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/zip/9007208137629963.zip.

TwinCAT user access rights

- Access rights are assigned to groups in the TwinCAT 3 Engineering.
- Users can be assigned to one or more groups.
- Groups can also be members of another group.

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.
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 [62]).

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).
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](#).)

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

<table>
<thead>
<tr>
<th>Groups</th>
<th>OPL_OEMDev</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRP_Guest</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRP_OEMDev</td>
<td>x x x x x x</td>
<td>x x x x</td>
</tr>
<tr>
<td>GRP_Administrators</td>
<td>x x x x x</td>
<td></td>
</tr>
<tr>
<td>GRP_OEMService</td>
<td>x x x x</td>
<td>x</td>
</tr>
</tbody>
</table>

[Table: Group Rights and Object Protection Levels]
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":

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](#71)).
**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":

![Object Protection Levels](image)

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

- **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.

- **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** is opened.
1. Select the **Users** tab.

   ![User Management Screen]

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**.

   ![Password Setting Dialog]

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 [48]" of user databases. This is described [here](#).

---

All changes are only finally confirmed and valid on saving and signing the user database.
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 [2].

1. The Software protection configurator is opened.
2. Select the Groups tab.
3. Click on Add to create a new group.
   - The Edit User Group dialog opens.
4. Give the group a name (Name).
5. If the group is to inherit the rights of another group, select the corresponding group in the Groups section.
6. Close the dialog with OK.
   - The new group appears in the overview.
7. To edit an entry, select the user group in the list and click Edit.
8. 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 [65].

5.6.4.4 Adjusting the access rights of user groups

- 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 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/zip/9007208137629963.zip.

- User database can only be created or edited if no project is open. Close any open projects.
- The Software protection configurator [10] 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.

![User Rights Management](image)

- The **Grant Right to User Groups** dialog opens.
3. Use the check boxes to select which user groups should have this right.

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)

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.

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/zip/9007208137629963.zip.

✓ User database can only be created or edited if no project is open. Close any open projects.
✓ The Software protection configurator [10] 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:

- 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.

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**.

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.

The project is now linked to the user database. The Security area becomes visible in the Properties of a project component.
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/zip/9007208137629963.zip.

You can assign the Object Protection Levels 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.

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":

- 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.

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

- **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

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:

![Security Management toolbar]

The user may only be changed when no project is open.

6.1 Build 4022

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

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.

![Password dialog](image)

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

![User account displayed](image)

Depending on the user account rights, certain TwinCAT menu items may be grayed out and therefore disabled.
7 Setting up basic protection of OEM application software

7.1 Encryption

TwinCAT 3 uses 256-bit AES-encryption with private and public key.

Following objects can be encrypted in TwinCAT:

- Source code
- Project file
- Boot project

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 4022.x. Do not use an older version on any account.
7.1.1 Source code encryption

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 is described in section Assigning user access rights in the project [71]. Save the project to apply the settings.

7.1.2 Project file encryption

In the TwinCAT version Build 4024.0, a user database [34] for the TwinCAT Software Protection may only be created with an OEM certificate with Crypto version 1!

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

1. The project is linked to a user database.
2. Select the TwinCAT project node in the project tree in the Solution Explorer.
3. 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.

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.

7.1.3 Encrypting the boot project

Requirement: An up-to-date version of Windows or TC/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.

1. A user database is selected [40] (and valid) in the TwinCAT Engineering.
   - Reason: Information from the user database is used for the encryption.
   - (However, the project doesn’t need to be connected with the user database).
   - 1. Double-click the PLC project object in the PLC project tree in the Solution Explorer.
The PLC project settings open in an editor.

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

   ![Encryption Project Settings](image1.png)

   - The boot project is encrypted.
   - No user database is required for the **decryption** of the boot project (= binary file) 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.

![TwinCAT Project Tree](image2.png)

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.
Setting up basic protection of OEM application software

TwinCAT object status symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>📁</td>
<td>No changes</td>
</tr>
<tr>
<td>📁</td>
<td>Unsaved changes</td>
</tr>
<tr>
<td>📁</td>
<td>Signed</td>
</tr>
<tr>
<td>📁</td>
<td>Encrypted</td>
</tr>
</tbody>
</table>

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:

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.

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 4022.x. Do not use an older version on any account.
7.3 Displaying the overview of the project software protection settings

You can display the settings for the protection of the application software in the display 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 display window.
Setting up basic protection of OEM application software

---

### 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\VISUAL\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\VISUAL\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\VISUAL\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\VISUAL\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".

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 Tc2Utilities:

- Use at least version 3.3.24 of the TC3 PLC Lib Tc2Utilities, 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 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 4022.x. Do not use an older version on any account.

General notes

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

Remember that the license ID \[85\] queried via FB_CheckLicense[91] 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!

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 File 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.
8.1.1 Preparing TwinCAT 3 Engineering

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

1. Create the following directories:
   • `c:\TwinCAT\3.1\CustomConfig\Licenses`
   • `c:\TwinCAT\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`.

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

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

✓ The Software protection configurator [10] 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 C:\TwinCAT\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.
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.

   - 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.

3. Tick the check box for the license.

4. Open the Order Information tab.
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** [10].

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

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.

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 C:\TwinCAT\3.1\sdk\Bin.

**Call parameters**

```
```

- **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.
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 `C:\twincat\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 [91] 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.

When the TwinCAT 3 runtime starts, TwinCAT 3 automatically checks for valid licenses. The result can be retrieved with function block FB_CheckLicense.

**FB_CheckLicense**

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

**VAR_INPUT**

- **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**
Issuing and using your own OEM licenses

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 :
        ST
            stLicenseId : GUID;
            tExpirationTime : TIMESTRUCT;
            sExpirationTime : STRING(80);
            eResult : E_LicenseHResult;
            nCount : UDINT;
        END_STRIP
    END_TYPE

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stLicenseId</td>
<td>License ID</td>
</tr>
<tr>
<td>tExpirationTime</td>
<td>Expiry date</td>
</tr>
<tr>
<td>sExpirationTime</td>
<td>Expiry date</td>
</tr>
<tr>
<td>eResult</td>
<td>License status (see E_LicenseHResult [92])</td>
</tr>
<tr>
<td>nCount</td>
<td>Number of instances for this license (0=unlimited)</td>
</tr>
</tbody>
</table>

ENUM E_LicenseHResult

TYPE E_LicenseHResult :
    DINT
    /
    /success
    E_LHR_LicenseOK : DINT := 0,
    E_LHR_LicenseOK_Pending : DINT := #203,
    E_LHR_LicenseOK_Demo : DINT := #254,
    E_LHR_LicenseOK_OEM : DINT := #255,
    /error
    E_LHR_LicenseNoFound : DINT := DWORD_TO_DINT(#98110700 + #24),
    E_LHR_LicenseExpired : DINT := DWORD_TO_DINT(#98110700 + #25),
    E_LHR_LicenseExceeded : DINT := DWORD_TO_DINT(#98110700 + #26),
    E_LHR_LicenseInvalid : DINT := DWORD_TO_DINT(#98110700 + #27),
    E_LHR_LicenseSystemIdInvalid : DINT := DWORD_TO_DINT(#98110700 + #28),
    E_LHR_LicenseNoTimeLimit : DINT := DWORD_TO_DINT(#98110700 + #29),
    E_LHR_LicenseTimeInFuture : DINT := DWORD_TO_DINT(#98110700 + #2A),
    E_LHR_LicenseTimePeriodToLong : DINT := DWORD_TO_DINT(#98110700 + #2B),
    E_LHR_DeviceException : DINT := DWORD_TO_DINT(#98110700 + #2C),
    E_LHR_LicenseDuplicated : DINT := DWORD_TO_DINT(#98110700 + #2D),
    E_LHR_LicenseRestricted : DINT := DWORD_TO_DINT(#98110700 + #31),
    E_LHR_LicenseDemoDenied : DINT := DWORD_TO_DINT(#98110700 + #32),
    E_LHR_LicensePlatformLevelInv : DINT := DWORD_TO_DINT(#98110700 + #33)
) DINT;
END_TYPE
Issuing and using your own OEM licenses

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

**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:

```
<License>
  <LicenseId>[CPlA625C-F2EC-477F-9008-65C305079F03]</LicenseId>
  <OemId>SampleOEM Inc</OemId>
  <OemName>SampleOEM Inc</OemName>
  <OrderAddress>OrderAddress@SampleOEM</OrderAddress>
  <OrderNo>4711-0815</OrderNo>
  <DisplayName>Sample_License A1</DisplayName>
</License>
```

"Manage Licenses" tab of the license manager:
Issuing and using your own OEM licenses

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

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_Utilsitites:

- Use at least version 3.3.24 of the TC3 PLC Lib Tc2_Utilsitites, 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 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 4022.x. Do not use an older version on any account.

See also: Documentation on the PLC library Tc2_Utilities, Licensing functions section

8.4 Providing OEM PLC libraries with license protection

In order to provide a self-made PLC library with license protection (i.e. to prevent use without a valid license), there is an option in the properties of the library to connect this library directly with a License GUID.

The difference between this and the method of querying the license via FB_CheckLicense is that the TwinCAT runtime has to check the program reaction:

- FB_CheckLicense merely returns the status of a certain license. The user program defines the reaction to it. -> The OEM has 100% control over what should happen depending on the license status.
- If the PLC is provided with a License GUID, conversely, the TwinCAT runtime controls what should happen.
9 Protecting an application against cloning

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

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 her internet pages: https://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
Fax: +49 5246 963 9157
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
Fax: +49 5246 963 479
e-mail: service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG
Huelshorstweg 20
33415 Verl
Germany
Phone: +49 5246 963 0
Fax: +49 5246 963 198
e-mail: info@beckhoff.com
web: https://www.beckhoff.com