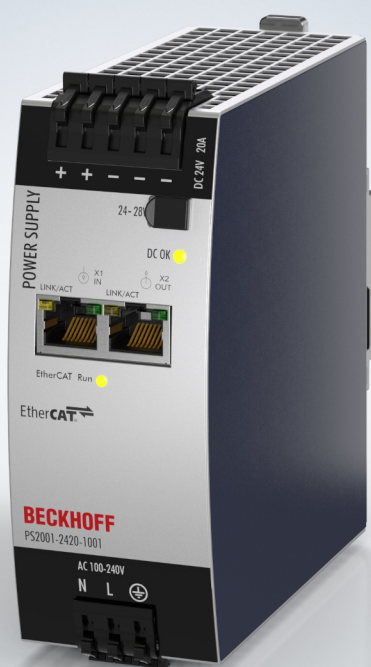


Documentation | EN

# PS2001-2420-1001

Power supply 24 V DC, 20 A, 1-phase, AC 100-240 V, EtherCAT interface





# Table of contents

<b>1 Overview</b> .....	<b>7</b>
<b>2 Foreword</b> .....	<b>8</b>
2.1 Notes on the documentation .....	8
2.2 Safety instructions .....	9
2.3 Terminology and abbreviations .....	12
2.4 Beckhoff Identification Code (BIC) .....	13
2.5 Electronic access to the BIC (eBIC) .....	15
<b>3 Technical data, mounting, wiring</b> .....	<b>17</b>
3.1 AC input .....	17
3.2 DC input .....	19
3.3 Input inrush current .....	20
3.4 Output .....	21
3.5 Hold-up time .....	23
3.6 DC-OK function (monitoring) .....	24
3.7 Efficiency and losses .....	25
3.8 Lifetime expectancy .....	26
3.9 MTBF .....	26
3.10 Terminals and wiring .....	27
3.11 Functional wiring diagram .....	28
3.12 Front side and operating elements .....	29
3.13 EMC .....	31
3.14 Environment .....	32
3.15 Protective functions .....	33
3.16 Safety features .....	33
3.17 Dielectric strength .....	34
3.18 Declaration of conformity and approvals .....	35
3.19 Dimensions and weight .....	36
<b>4 Application notes</b> .....	<b>37</b>
4.1 Peak current capability .....	37
4.2 Output circuit breakers .....	39
4.3 Charging batteries .....	40
4.4 Series connection .....	40
4.5 Parallel use to increase power .....	41
4.6 Parallel use for redundancy .....	42
4.6.1 1+1 Redundancy .....	43
4.6.2 N+1 Redundancy .....	44
4.7 Operation on two phases .....	45
4.8 Installation positions .....	45
<b>5 Basics communication</b> .....	<b>46</b>
5.1 EtherCAT basics .....	46
5.2 EtherCAT cabling – wire-bound .....	46
5.3 General notes for setting the watchdog .....	47
5.4 EtherCAT State Machine .....	49

5.5	CoE Interface .....	50
5.6	Distributed Clock .....	55
<b>6</b>	<b>Commissioning .....</b>	<b>56</b>
6.1	TwinCAT Quick Start.....	56
6.1.1	TwinCAT 3 .....	58
6.2	TwinCAT Development Environment .....	70
6.2.1	Installation of the TwinCAT real-time driver .....	71
6.2.2	Notes regarding ESI device description .....	76
6.2.3	TwinCAT ESI Updater .....	79
6.2.4	Distinction between Online and Offline .....	79
6.2.5	OFFLINE configuration creation.....	80
6.2.6	ONLINE configuration creation .....	85
6.2.7	EtherCAT subscriber configuration .....	93
6.2.8	Import/Export of EtherCAT devices with SCI and XTI.....	102
6.3	General Notes - EtherCAT Slave Application .....	108
6.4	First steps Power supplies with EtherCAT .....	115
6.4.1	Linking of process data .....	116
6.4.2	Selecting the info data.....	118
6.4.3	Setting the warning thresholds for the warning bit .....	121
6.4.4	Resetting counters and maximum/minimum values.....	122
6.4.5	Display of errors and warning messages .....	124
6.4.6	Writing and reading CoE objects using function blocks .....	125
6.4.7	Complete access to "PSU Diag Data" .....	129
6.4.8	DC-OK function (parameterization).....	131
6.4.9	Switching output via EtherCAT .....	132
6.4.10	Recognizing invalid data .....	134
6.4.11	Parallel use .....	135
6.5	Information on grid monitoring on the input side .....	136
6.6	Process data overview .....	137
6.6.1	Parameterization .....	137
6.6.2	Process data .....	138
6.7	Object description and parameterization.....	141
6.7.1	Restore object.....	141
6.7.2	Input data .....	142
6.7.3	Output data .....	143
6.7.4	Configuration data.....	143
6.7.5	Configuration data (vendor-specific) .....	144
6.7.6	Command object.....	145
6.7.7	Information - diagnostic data.....	146
6.7.8	Standard objects .....	147
6.8	JSON.....	154
<b>7</b>	<b>Disposal .....</b>	<b>157</b>
<b>8</b>	<b>Appendix.....</b>	<b>158</b>
8.1	EtherCAT AL Status Codes .....	158
8.2	Firmware update PSxxxx .....	158



8.2.1	Device description ESI file/XML .....	159
8.2.2	Firmware explanation .....	162
8.2.3	Updating controller firmware *.efw .....	163
8.2.4	Simultaneous updating of several EtherCAT devices .....	165
8.3	Restoring the delivery state .....	166
8.4	Accessories .....	167
8.4.1	Redundancy and Buffer Modules .....	167
8.4.2	USV component .....	168
8.4.3	Accessories for mounting .....	169
8.5	Documentation issue status .....	170
8.6	Support and Service .....	171



# 1 Overview

## PS2001-2420-1001 | Power supply 24 V, 20 A, 1-phase, Extra Power, with EtherCAT interface



- AC 100-240 V wide-range input
- Incl. EtherCAT interface
- Width only 48 mm
- Efficiency up to 95.5%
- Excellent part-load efficiency
- 120% peak power, 576 W
- Precise triggering of fuses through high overload / peak current
- Safe hiccup overload mode
- Active Power Factor Correction (PFC)
- Minimum inrush current surge
- Full output between -25°C and +60°C
- Current-sharing function for parallel use

The PS2001-2420-1001 is a 1-phase 24 V DC power supply with an integrated EtherCAT communication level suitable for applications with an output current of 20 A and an output power of 480 W. Thanks to the integration of EtherCAT technology, measurement data and internal device data can be processed directly by the controller. This makes it possible to influence the device directly via remote access. Device-internal parameter values, status bits and error messages can be read, configured and reset easily and conveniently via the TwinCAT automation software. In addition, the power supply provides extensive measured data for monitoring both the DC output range and the AC input range.

The power supplies with EtherCAT interface belong to the PS2000 family and have a width of just 48 mm. They have a wide-range input, Active Power Factor Correction (PFC) and inrush current limitation on the input side. The output operates according to the UI characteristic curve and switches to the safe hiccup mode in case of overload/short-circuit. The PS2001-xxxx-1001 features an Extra Power with a continuous maximum output power of 120% and is able to trigger fuses precisely with a short-term peak current. The status of the output voltage can be tracked via the process data and displayed via the DC-OK LED on the front of the housing.

Overview of technical data <sup>*)</sup>	PS2001-2420-1001	
Output voltage	DC 24 V (nominal factory setting 24.1 V)	
Adjustment range	24...28 V	
Output current	<b>AC 120 - 240 V, DC 150 - 300 V mains</b>	<b>AC 100 V, DC 110 - 150 V mains</b>
Extra Power at ambient temperature below +45 °C	24.0 A - 20.6 A (576 W)	21.0 A - 18.0 A (504 W)
Nominal load at ambient temperature up to +60 °C	20.0 - 17.1 A (480 W)	17.5 - 15.0 A (420 W)
Derating at ambient temperature above +60 °C	Linear derating 12.5 W/K	Linear derating 10.5 W/K
Input voltage AC	AC 100...240 V -15% / +10%	
Mains frequency	50...60 Hz ±6%	
Input current AC	4.26 / 2.23 A at 120 / 230 Vac	
AC input inrush current	10.0 / 4.5 A <sub>Peak</sub> at 120 / 230 Vac	
Power factor	0.99 / 0.98 at 120 / 230 Vac	
Input voltage DC	DC 110...300 V ±20%	
Input current DC	4.64 / 1.66 A at 110 / 300 Vdc	
Efficiency	94.1 / 95.5% at 120 / 230 Vac	
Losses	30.0 / 23.0 W at 120 / 230 Vac	
Hold-up time	32 / 32 ms at 120 / 230 Vac	
Temperature range	-25°C...+70°C	
Protocol / bus interface	EtherCAT / 2 x RJ45	
Data transfer rate	100 Mbit/s	
Dimensions (W x H x D)	48 mm x 124 mm x 127 mm (without DIN rail)	
Weight	835 g	
Approvals/markings	CE, cULus	
*) All values are typical figures specified at 24 V, 20 A, 230 Vac, 50 Hz, +25°C ambient temperature and after a 5 minutes run-in time, unless otherwise noted		

## 2 Foreword

### 2.1 Notes on the documentation

#### Intended audience

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

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

The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described 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: EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702 with corresponding applications or registrations in various other countries.



EtherCAT® is 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.

## 2.2 Safety instructions

### Signal words

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

#### Personal injury warnings

**⚠ DANGER**

Hazard with high risk of death or serious injury.

**⚠ WARNING**

Hazard with medium risk of death or serious injury.

**⚠ CAUTION**

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

#### Warning of damage to property or environment

**NOTICE**

The environment, equipment, or data may be damaged.

#### Information on handling the product



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

#### Intended use

This device is designed for installation in a housing and is intended for general professional use, for example in industrial control systems or office, communication and measuring equipment.

Do not use this power supply in installations where a malfunction could cause serious injury or danger to human life.

Without additional measures to reduce conducted emissions at the output (e.g. by using a filter), the device is not suitable for supplying a local DC network in residential, commercial and light-industrial environments. No restrictions apply for local DC power networks in industrial environments.

#### 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.

#### 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.

## Safety instructions and installation requirements for PS2001-2420-1001 power supply

### DANGER

#### Danger of electric shock, fire, injuries, injuries resulting in death!

- Do not use the power supply without proper grounding (protective conductor). Use the terminal at the input terminal strip for the ground connection, not one of the screws on the housing.
- Turn power off before working on the device. Provide protection against unintentional reconnection.
- Ensure proper wiring by following all local and national regulations.
- Do not modify or repair the device.
- Do not open the device, as high voltages are present inside.
- Avoid foreign objects entering the housing.
- Do not use in wet locations or in areas where moisture or condensation can be expected.
- Do not touch the device when it is switched on or immediately after it has been switched off. Hot surfaces may cause burns.

#### **i** Further notes on installation requirements

- If damage or malfunction should occur during installation or operation, immediately turn power off and send unit to the factory for inspection.
- Install the device in a housing that provides protection against electrical, mechanical and fire hazards.
- Install the device on a DIN rail according to EN 60715, with the input terminals on the bottom of the device.
- The maximum ambient air temperature is +70°C (+158°F). The operating temperature is the same as the ambient or surrounding air temperature and is defined 2 cm below the device.
- The device is designed for convection cooling and does not require an external fan. Do not obstruct the air flow or cover the ventilation grid!
- Observe the following installation clearances (see table Minimum installation clearances):
  - ⇒ 40 mm at the top, 20 mm at the bottom and 5 mm on the left and right are recommended if the device continuously runs at more than 50% of the nominal output.
  - ⇒ Increase this distance to 15 mm if the device is loaded with more than 90% of the nominal output or if the adjacent device is a heat source (e.g. another power supply).
  - ⇒ When the device is permanently loaded with less than 50%, the 5 mm can be reduced to zero.

#### Minimum installation clearances

Output power in relation to the nominal output	Minimum installation clearances (mm)		
	top	bottom	Left and right side
< 50%	40	20	0
≥ 50%	40	20	5
≥ 90%	40	20	15





**i****Further notes on installation requirements (connection and wiring)**

- Ensure that the wiring is correct by following all local and national regulations.
  - ⇒ Use suitable copper cables that are designed for a minimum operating temperature of:
    - 60°C at ambient temperatures up to 45°C
    - 75°C at ambient temperatures up to 60°C
    - 90°C at ambient temperatures up to 70°C
  - ⇒ Ensure that all strands of a stranded wire enter the terminal connection.
  - ⇒ Use ferrules for wires at the input terminals.
  - ⇒ Unused screw terminals should be securely tightened.
- The device is designed, tested and approved for branch-circuit currents up to 32 A (IEC) and 30 A (UL) without additional protection. If an external fuse is used, do not use circuit breakers smaller than 10 A (B or C characteristic) to avoid unwanted tripping of the circuit breaker.
- The input can also be supplied by batteries or similar DC sources. The continuous voltage between the supply voltage and the PE/ground potential must not exceed 375 Vdc.
- The device is designed to meet the requirements of protection class I according to IEC 61140.
  - ⇒ Do not use without a proper PE connection (protective earth).
- An isolating device must be provided for the device input.

**i****Further notes on operation**

- This device contains no parts that require maintenance. If an internal fuse trips, this is due to an internal defect.
- The device is designed for areas with a degree of pollution 2 in controlled environments. No condensation or frost is allowed.
- The device housing offers IP20 protection. The housing offers no protection against spilled liquids.
- The device is designed to operate in areas between 5% and 95% relative humidity.
- The device insulation is designed to withstand pulse voltages of overvoltage category III according to IEC 60664-1.
- The device is suitable for supply from TN, TT or IT networks. The continuous voltage between the input terminal and the PE potential must not exceed 300 Vac.
- The device is designed for altitudes up to 5000 m (16400 ft).
  - ⇒ Above 2000 m a reduction in output current and over voltage category is required.

## 2.3 Terminology and abbreviations

<b>PE and the earthing symbol</b> 	PE is the abbreviation for "protective earth" and has the same meaning as the earthing symbol 
<b>Earth, ground</b>	This document uses the term "earth" which is the same as the U.S. term "ground".
<b>T.b.d.</b>	Still to be defined, value or description will follow in due course.
<b>AC 230 V</b>	A value preceded by "AC" or "DC" represents a nominal voltage or a nominal voltage range. The nominal voltage or the nominal voltage range may be provided with tolerances. (e.g. AC 230 V $\pm$ 10%). The calculated total range then indicates the working range of the device.  Example: DC 12 V refers to a 12 V battery, regardless of whether it is fully charged (13.7 Vdc) or discharged (10 Vdc).
<b>230 Vac</b>	A value followed by the unit Vac or Vdc is an instantaneous value that does not contain any additional tolerances.
<b>50 Hz vs. 60 Hz</b>	Unless otherwise specified, AC 100 V and AC 230 V parameters are valid at a mains frequency of 50 Hz. AC 120 V parameters are valid for a mains frequency of 60 Hz.
<b>may</b>	A keyword indicating a choice without implied preference.
<b>shall</b>	A keyword indicating a mandatory requirement.
<b>should</b>	A keyword indicating a choice with a clearly preferred method of implementation.

## 2.4 Beckhoff Identification Code (BIC)

The Beckhoff Identification Code (BIC) is increasingly being applied to Beckhoff products to uniquely identify the product. The BIC is represented as a Data Matrix Code (DMC, code scheme ECC200), the content is based on the ANSI standard MH10.8.2-2016.

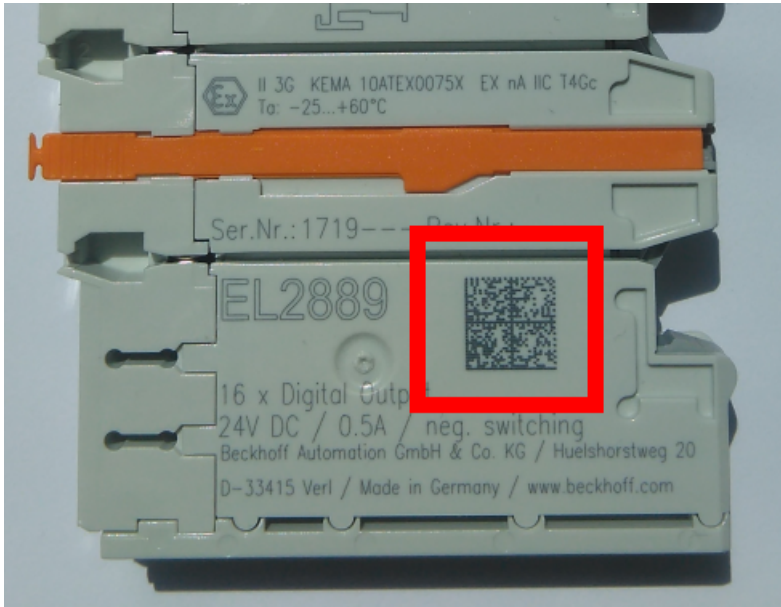


Fig. 1: BIC as data matrix code (DMC, code scheme ECC200)

The BIC will be introduced step by step across all product groups.

Depending on the product, it can be found in the following places:

- on the packaging unit
- directly on the product (if space suffices)
- on the packaging unit and the product

The BIC is machine-readable and contains information that can also be used by the customer for handling and product management.

Each piece of information can be uniquely identified using the so-called data identifier (ANSI MH10.8.2-2016). The data identifier is followed by a character string. Both together have a maximum length according to the table below. If the information is shorter, spaces are added to it.

Following information is possible, positions 1 to 4 are always present, the other according to need of production:

Position	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	<b>Beckhoff order number</b>	1P	8	<b>1P</b> 072222
2	Beckhoff Traceability Number (BTN)	<b>Unique serial number, see note below</b>	SBTN	12	<b>S</b> BTNk4p562d7
3	Article description	<b>Beckhoff article description, e.g. EL1008</b>	1K	32	<b>1K</b> EL1809
4	Quantity	<b>Quantity in packaging unit, e.g. 1, 10, etc.</b>	Q	6	<b>Q</b> 1
5	Batch number	Optional: Year and week of production	2P	14	<b>2P</b> 401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	<b>51S</b> 678294
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	<b>30P</b> F971, 2*K183
...					

Further types of information and data identifiers are used by Beckhoff and serve internal processes.

### Structure of the BIC

Example of composite information from positions 1 to 4 and with the above given example value on position 6. The data identifiers are highlighted in bold font:

**1P**072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

Accordingly as DMC:



Fig. 2: Example DMC **1P**072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

### BTN

An important component of the BIC is the Beckhoff Traceability Number (BTN, position 2). The BTN is a unique serial number consisting of eight characters that will replace all other serial number systems at Beckhoff in the long term (e.g. batch designations on IO components, previous serial number range for safety products, etc.). The BTN will also be introduced step by step, so it may happen that the BTN is not yet coded in the BIC.

### NOTICE

This information has been carefully prepared. However, the procedure described is constantly being further developed. We reserve the right to revise and change procedures and documentation at any time and without prior notice. No claims for changes can be made from the information, illustrations and descriptions in this information.

## 2.5 Electronic access to the BIC (eBIC)

### Electronic BIC (eBIC)

The Beckhoff Identification Code (BIC) is applied to the outside of Beckhoff products in a visible place. If possible, it should also be electronically readable.

The interface that the product can be electronically addressed by is crucial for the electronic readout.

### K-bus devices (IP20, IP67)

Currently, no electronic storage or readout is planned for these devices.

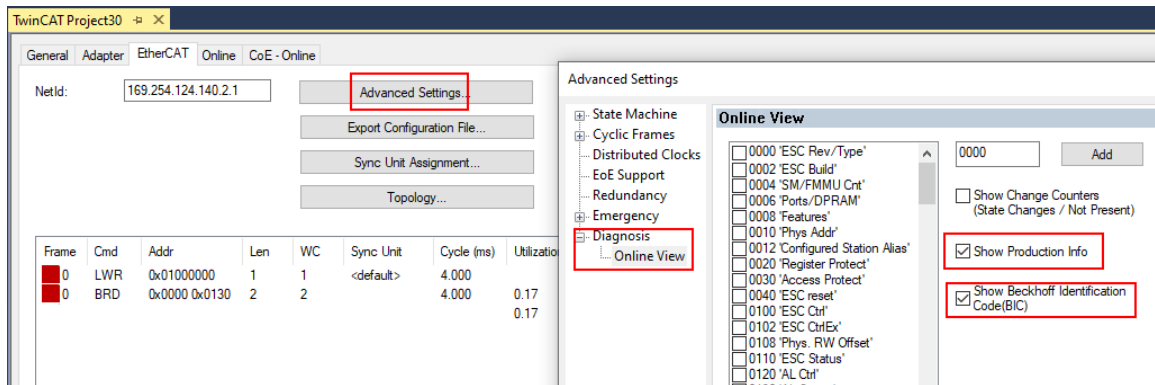
### EtherCAT devices (IP20, IP67)

All Beckhoff EtherCAT devices have an ESI-EEPROM which contains the EtherCAT identity with the revision number. The EtherCAT slave information, also colloquially known as the ESI/XML configuration file for the EtherCAT master, is stored in it. See the corresponding chapter in the EtherCAT system manual ([Link](#)) for the relationships.

Beckhoff also stores the eBIC in the ESI-EEPROM. The eBIC was introduced into Beckhoff IO production (terminals, box modules) in 2020; as of 2023, implementation is largely complete.

The user can electronically access the eBIC (if present) as follows:

- With all EtherCAT devices, the EtherCAT master (TwinCAT) can read the eBIC from the ESI-EEPROM
  - From TwinCAT 3.1 build 4024.11, the eBIC can be displayed in the online view.
  - To do this, check the "Show Beckhoff Identification Code (BIC)" checkbox under EtherCAT → Advanced Settings → Diagnostics:



- The BTN and its contents are then displayed:

No	Addr	Name	State	CRC	Fw	Hw	Production Data	ItemNo	BTN	Description	Quantity	BatchNo	SerialNo
1	1001	Term 1 (EK1100)	OP	0,0	0	0	---						
2	1002	Term 2 (EL1018)	OP	0,0	0	0	2020 KW36 Fr	072222	k4p562d7	EL1809	1		678294
3	1003	Term 3 (EL3204)	OP	0,0	7	6	2012 KW24 Sa						
4	1004	Term 4 (EL2004)	OP	0,0	0	0	---	072223	k4p562d7	EL2004	1		678295
5	1005	Term 5 (EL1008)	OP	0,0	0	0	---						
6	1006	Term 6 (EL2008)	OP	0,0	0	12	2014 KW14 Mo						
7	1007	Term 7 (EK1110)	OP	0	1	8	2012 KW25 Mo						

- Note: As shown in the figure, the production data HW version, FW version, and production date, which have been programmed since 2012, can also be displayed with "Show production info".
- Access from the PLC: From TwinCAT 3.1. build 4024.24, the functions *FB\_EcReadBIC* and *FB\_EcReadBTN* for reading into the PLC are available in the Tc2\_EtherCAT library from v3.3.19.0.
- EtherCAT devices with a CoE directory may also have the object 0x10E2:01 to display their own eBIC, which can also be easily accessed by the PLC:

- The device must be in PREOP/SAFEOP/OP for access:

Index	Name	Flags	Value
1000	Device type	RO	0x015E1389 (22942601)
1008	Device name	RO	ELM3704-0000
1009	Hardware version	RO	00
100A	Software version	RO	01
100B	Bootloader version	RO	J0.1.27.0
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
10E2:0	Manufacturer-specific Identification C...	RO	> 1 <
10E2:01	SubIndex 001	RO	1P158442SBTN0008jckp1KELM3704 Q1 2P482001000016
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170bfb277e

- The object 0x10E2 will be preferentially introduced into stock products in the course of necessary firmware revision.
- From TwinCAT 3.1. build 4024.24, the functions *FB\_EcCoEReadBIC* and *FB\_EcCoEReadBTN* for reading into the PLC are available in the *Tc2\_EtherCAT* library from v3.3.19.0
- The following auxiliary functions are available for processing the BIC/BTN data in the PLC in *Tc2\_Uilities* as of TwinCAT 3.1 build 4024.24
  - *F\_SplitBIC*: The function splits the Beckhoff Identification Code (BIC) *sBICValue* into its components using known identifiers and returns the recognized substrings in the *ST\_SplittedBIC* structure as a return value
  - *BIC\_TO\_BTN*: The function extracts the BTN from the BIC and returns it as a return value
- Note: If there is further electronic processing, the BTN is to be handled as a string(8); the identifier "SBTN" is not part of the BTN.
- Technical background  
The new BIC information is written as an additional category in the ESI-EEPROM during device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored using a category in accordance with the ETG.2010. ID 03 tells all EtherCAT masters that they may not overwrite these data in the event of an update or restore the data after an ESI update.  
The structure follows the content of the BIC, see here. The EEPROM therefore requires approx. 50..200 bytes of memory.
- Special cases
  - If multiple hierarchically arranged ESCs are installed in a device, only the top-level ESC carries the eBIC information.
  - If multiple non-hierarchically arranged ESCs are installed in a device, all ESCs carry the eBIC information.
  - If the device consists of several sub-devices which each have their own identity, but only the top-level device is accessible via EtherCAT, the eBIC of the top-level device is located in the CoE object directory 0x10E2:01 and the eBICs of the sub-devices follow in 0x10E2:nn.

## PROFIBUS; PROFINET, and DeviceNet devices

Currently, no electronic storage or readout is planned for these devices.



### 3 Technical data, mounting, wiring

#### 3.1 AC input

AC input			
Voltage range 1	Nom.	AC 120 - 240 V (-15% / +10%)	Suitable for TN, TT and IT networks
Voltage range 2		AC 100 - 240 V (-15% / +10%)	
AC input range	Min.	85 - 264 Vac	
	Min.	264 - 300 Vac	For up to 500 ms max.
Permissible voltage L or N to ground	Max.	300 Vac	Continuous according to IEC 60664-1
Input frequency	Nom.	50 - 60 Hz	±6%
Turn-on voltage	Typ.	82 Vac	Static, see Fig. <i>Input voltage range; turn-on behavior definitions</i>
Shut-down voltage	Typ.	72 Vac	Static, see Fig. <i>Input voltage range; turn-on behavior definitions</i>
External input protection	See notes on installation requirements in chapter " <a href="#">Safety instructions [▶ 11]</a> "		

AC input		AC 100 V	AC 120 V	AC 230 V	
Input current	Typ.	5.15 A	4.26 A	2.23 A	At 24 V, 20 A, see Fig. <i>Input current vs. output current; power factor vs. output current</i>
Power factor*)	Typ.	0.99	0.99	0.98	At 24 V, 20 A, see Fig. <i>Input current vs. output current; power factor vs. output current</i>
Peak factor**)	Typ.	1.65	1.63	1.63	At 24 V, 20 A
Inrush delay	Typ.	600 ms	550 ms	500 ms	See Fig. <i>Input voltage range; turn-on behavior definitions</i>
Rise time	Typ.	110 ms	110 ms	110 ms	At 24 V, 20 A constant current load, 0 mF load capacitance, see Fig. <i>Input voltage range; turn-on behavior definitions</i>
	Typ.	120 ms	120 ms	120 ms	at 24 V, 20 A constant current load, 20 mF load capacitance, see Fig. <i>Input voltage range; turn-on behavior definitions</i>
Turn-on overshoot	Max.	200 mV	200 mV	200 mV	See Fig. <i>Input voltage range; turn-on behavior definitions</i>

\*) The power factor is the ratio of real (or active) power to apparent power in an AC circuit.

\*\*\*) The peak factor is the mathematical ratio of the peak value to the RMS value of the input current waveform.

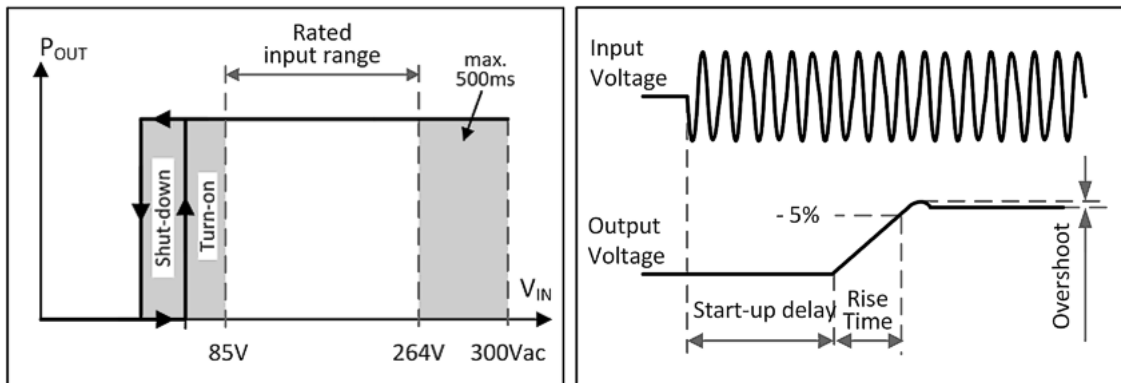


Fig. 3: Input voltage range, Turn-on behavior definitions

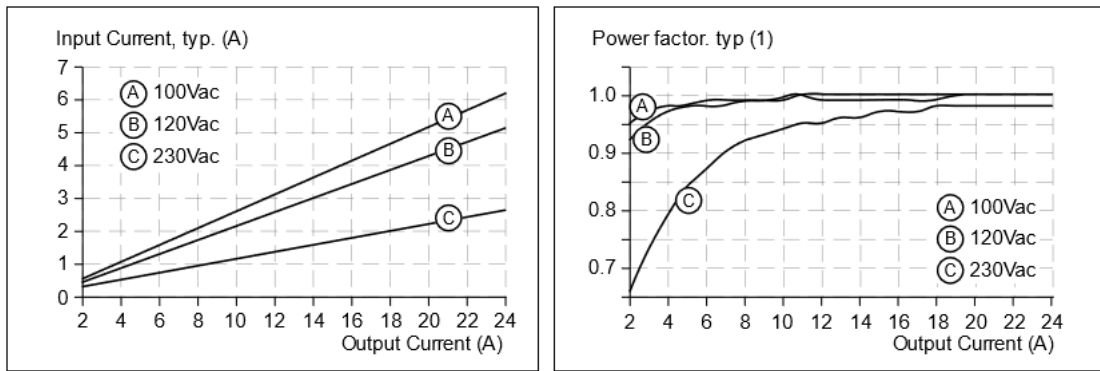


Fig. 4: Input current vs. output current; power factor vs. output current

### 3.2 DC input

DC input			
DC input	Nom.	DC 110 - 300 V	±20%
DC input range		88 - 360 Vdc	
DC input current	Typ.	4.64 A	at 110 Vdc, at 24 V, 20 A
		1.66 A	at 300 Vdc, at 24 V, 20 A
Voltage range 1	Nom.	DC 150 - 300 V	
Voltage range 2	Nom.	DC 110 - 150 V	
Permissible voltage L/N to ground	Max.	375 Vdc	Continuous according to IEC 62477-1
Turn-on voltage	Typ.	80 Vdc	Static
Shut-down voltage	Typ.	70 Vdc	Static

#### Instructions for DC operation

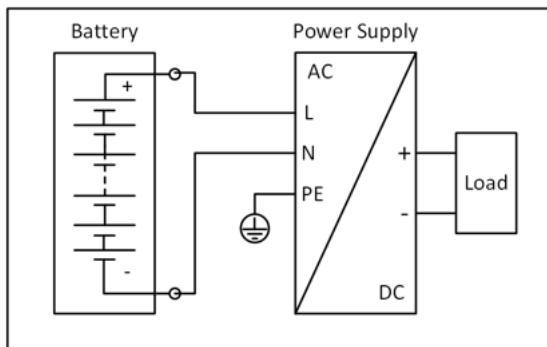


Fig. 5: Wiring for DC input

- The device is designed to be supplied from a DC input voltage. Use a battery or a comparable DC source.
- **Notice** Operation on the DC link of frequency converters is not recommended and may lead to defects or malfunctions.
- Connect the positive pole to L and the negative pole to N.
- Connect the PE terminal to the protective conductor or machine ground.

### 3.3 Input inrush current

Active inrush current limitation limits the inrush current surge after the input voltage is switched on.

The charging current of the interference suppression capacitors is disregarded in the first microseconds after switch-on.

		AC 100 V	AC 120 V	AC 230 V	
Input inrush current	Max.	15 A <sub>Peak</sub>	12 A <sub>Peak</sub>	5.5 A <sub>Peak</sub>	temperature-independent
	Typ.	12 A <sub>Peak</sub>	10 A <sub>Peak</sub>	4.5 A <sub>Peak</sub>	
Inrush energy	Max.	4 A <sup>2</sup> s	3 A <sup>2</sup> s	1 A <sup>2</sup> s	

### 3.4 Output

The output provides an SELV/PELV nominal voltage that is electrically isolated from the input voltage.

The device is designed to supply any type of load, including capacitive and inductive loads. If very large capacitors, such as EDLCs (electrical double-layer capacitors or "UltraCaps") with a capacitance > 1 F are connected to the output, the device can charge the capacitor intermittently.

The device is equipped with a "Single Use / Parallel Use" function, which can be selected via the CoE data.

In Parallel Use, the output voltage at no load is regulated so that it is 4% higher than at nominal load. This achieves load sharing between power supplies connected in parallel.

The output is electronically protected against overload, no-load and short circuit. In the event of a protection event, audible noises may occur.

Output voltage	Nom.	24 V	
Adjustment range	Min.	24...28 V	Guaranteed value
	Max.	30.0 V	This is the maximum output voltage which can occur at the clockwise end position of the potentiometer due to tolerances. It is not guaranteed value which can be achieved.
Factory settings	Typ.	24.1 V	±0.2%, "single use" mode at full load and cold unit ±0.2% in "parallel use" mode at 20 A, with cold unit (23.9 V ±0.2% at 24 A and 25.1 V ±0.2% at no load)
Line regulation	Max.	10 mV	Between 85 and 300 Vac
Load regulation	Max.	100 mV	Between 0 and 24 A, static value, "single use"
	Typ.	1000 mV	Between 0 and 20 A, static value, "parallel use", see Fig. <i>Output voltage vs. output current (parallel use)</i>
Ripple and noise voltage	Max.	50 mV <sub>pp</sub>	Bandwidth 20 Hz to 20 MHz, 50 Ohm
Output current for Input voltage range 1	Nom.	24 A <sup>1)</sup>	At 24 V and up to 45°C ambient temperature
	Nom.	20 A	At 24 V and up to 60°C ambient temperature
	Nom.	14.8 A	At 24 V and 70°C ambient temperature
	Nom.	20.6 A <sup>1)</sup>	At 28 V and up to 45°C ambient temperature
	Nom.	17.1 A	At 28 V and up to 60°C ambient temperature
	Nom.	12.7 A	At 28 V and 70°C ambient temperature
	Linear derating between +60°C and +70°C (see chapter <a href="#">Environment</a> [▶ 32])		
Output current Input voltage range 2	Nom.	21 A <sup>1)</sup>	At 24 V and up to 45°C ambient temperature
	Nom.	17.5 A	At 24 V and up to 60°C ambient temperature
	Nom.	13.1 A	At 24 V and 70°C ambient temperature
	Nom.	18 A <sup>1)</sup>	At 28 V and up to 45°C ambient temperature
	Nom.	15.0 A	At 28 V and up to 60°C ambient temperature
	Nom.	11.3 A	At 28 V and 70°C ambient temperature
	Linear derating between +60°C and +70°C		

<sup>1)</sup> This continuous power / current is permissible up to an ambient temperature of +45°C. It decreases linearly between 45 and 60°C up to the nominal power / nominal current s. chapter ["Environment"](#) [▶ 32]. Above +45°C, use this power / current for a maximum duty cycle of 10%, i.e. no more than 1 minute every 10 minutes.

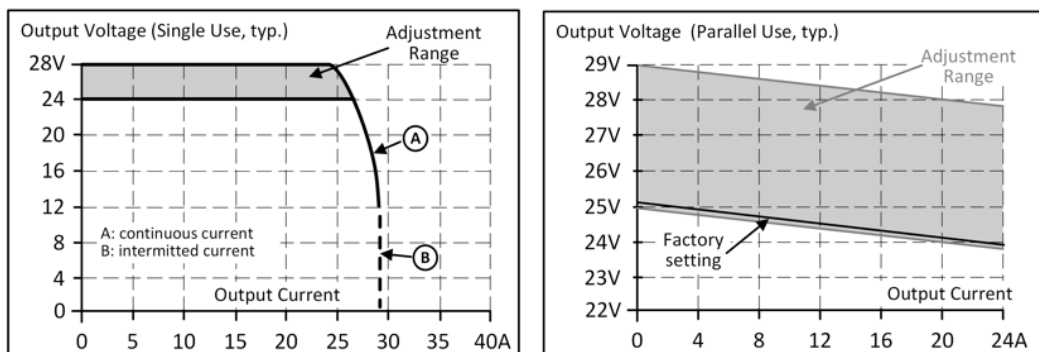


Fig. 6: Output voltage vs. output current (single use); output voltage vs. output current (parallel use)

Safety switch-off current	Typ.	60 A	Up to 12 ms once every five seconds, see Fig. <i>Dynamic output current carrying capacity, typ.</i> The fuse switch-off current is an increased transient current that contributes to the tripping of fuses on faulty output branches. The output voltage remains above 20 V.
Overload behavior		Continuous current	Output voltage > 13 Vdc, see fig. <i>Output voltage vs. output current (single use)</i>
		Hiccup mode <sup>2)</sup>	Output voltage < 13 Vdc, see fig. <i>Output voltage vs. output current (single use)</i>
Short-circuit current	Max.	29.8 A	Continuous current, see Fig. <i>Output voltage vs. output current, typ.</i>
	Typ.	29 A	Intermittent current peak value for typ. 2 s Load impedance 10 mOhm, see Fig. <i>Short circuit at output, hiccup mode, typ.</i> The discharge current of the output capacitors is not included.
	Max.	9.8 A	RMS value of current, load impedance 10 mOhm, see Fig. <i>Short circuit at output, hiccup mode, typ.</i>
Output capacitance	Typ.	8500 µF	Included in the power supply
Load feedback	Max.	35 V	The device is resistant to load feedback and does not show malfunctioning when a load feeds back voltage to the power supply. It does not matter whether the power supply is switched on or off. The absorbed energy can be determined by means of the built-in large sized output capacitor.

<sup>2)</sup> In the event of heavy overload (when the output voltage drops below 13 V), the power supply provides continuous output current for 2 s. The output is then switched off for about 18 seconds before a new switch-on attempt is automatically made. This cycle is repeated as long as the overload persists. After the overload has been rectified, the device will operate normally. See Fig. *Short circuit at output, hiccup mode, typ.*

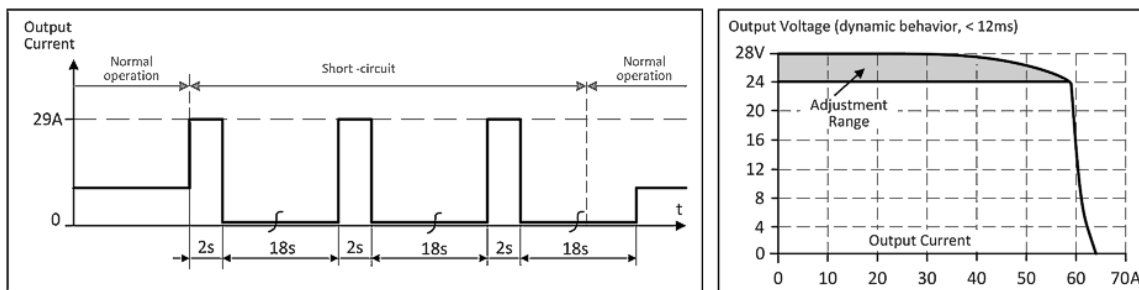


Fig. 7: Short circuit at output, hiccup mode, typ.; Dynamic output current capacity, typ.



### 3.5 Hold-up time

The hold-up time is the time during which the output voltage of a power supply remains within the specification after a loss of input power. The hold-up time depends on the output load. Without load the hold-up time can be up to several seconds. The green DC-OK LED is also on during this time.

		AC 100 V	AC 120 V	AC 230 V	
Hold-up time	Typ.	65 ms	65 ms	65 ms	At 24 V, 10 A, see Fig. <i>Hold-up time vs. input voltage</i>
	Min.	54 ms	54 ms	54 ms	At 24 V, 10 A, see Fig. <i>Hold-up time vs. input voltage</i>
	Typ.	32 ms	32 ms	32 ms	At 24 V, 20 A, see Fig. <i>Hold-up time vs. input voltage</i>
	Min.	24 ms	24 ms	24 ms	At 24 V, 20 A, see Fig. <i>Hold-up time vs. input voltage</i>

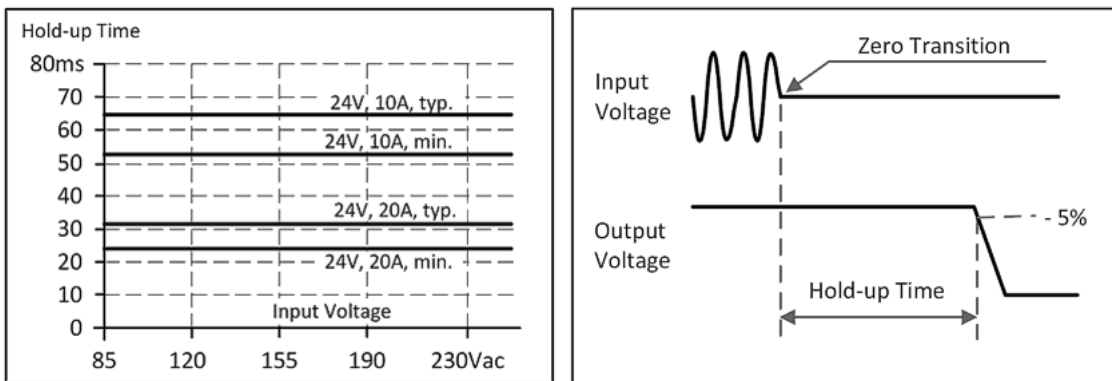


Fig. 8: Hold-up time vs. input voltage; Shut-down behavior, definitions

### 3.6 DC-OK function (monitoring)

This feature monitors the output voltage at the output terminal of an active power supply. The DC-OK function can be monitored via the DC-OK LED on the front of the device or via the EtherCAT interface.

LED (link with the DC-OK bit 0x6000:04)		
DC-OK	green	As soon as the output voltage has reached the setpoint (default 22 V)
DC-Not OK	off	As soon as the output voltage dips below a value, which is typically 0.5 V lower than the setpoint value (Hysteresis). (default: 21.5 V)
Switching hysteresis	0.5 V typ.	

Via EtherCAT interface		
DC-OK bit (index 0x6000:04)	TRUE	The output voltage is above the value defined in index 0x8000:16 "DC OK threshold".
	FALSE	As soon as the output voltage falls below the value defined in index 0x8000:16 "DC OK threshold".
"Output voltage" (Index 0x6000:11)	Measured output voltage	
"DC OK threshold" (Index 0x8000:16)	Parameterized value for the DC OK threshold	

**Also see about this**

- 📖 DC-OK function (parameterization) [▶ 131]

### 3.7 Efficiency and losses

		AC 100 V	AC 120 V	AC 230 V	
Efficiency	Typ.	93.6%	94.1%	95.5%	At 24 V, 20 A
	Typ.	93.3%	94.0%	95.4%	At 24 V, 24 A (Extra Power)
Average efficiency <sup>*)</sup>	Typ.	93.2%	93.8%	95.0%	25% at 5 A, 25% at 10A, 25% at 15 A. 25% at 20 A
Losses	Typ.	1.6 W	1.7 W	2.1 W	With activated shutdown (0x7000:01 [▶ 143] "Disable output")
	Typ.	3.5 W	3 W	3 W	At 24 V, 0 A
	Typ.	17.5 W	14.6 W	13.0 W	At 24 V, 10 A
	Typ.	33.0 W	30.0 W	23.0 W	At 24 V, 20 A
	Typ.	41.5 W	37.7 W	28.1 W	At 24 V, 24 A (Extra Power)

<sup>\*)</sup> The average efficiency is an assumption for a typical application where the power supply is loaded with 25% of the nominal load for 25% of the time, 50% of the nominal load for another 25% of the time, 75% of the nominal load for another 25% of the time and with 100% of the nominal load for the rest of the time.

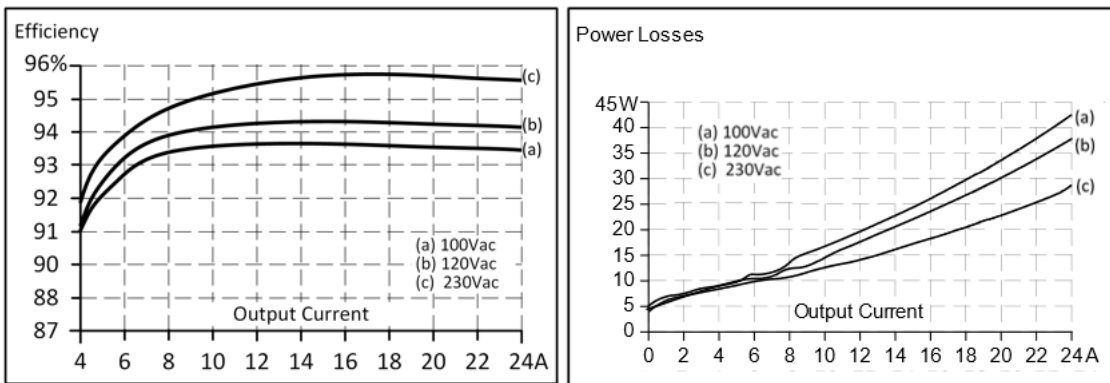


Fig. 9: Efficiency vs. output current; losses vs. output current

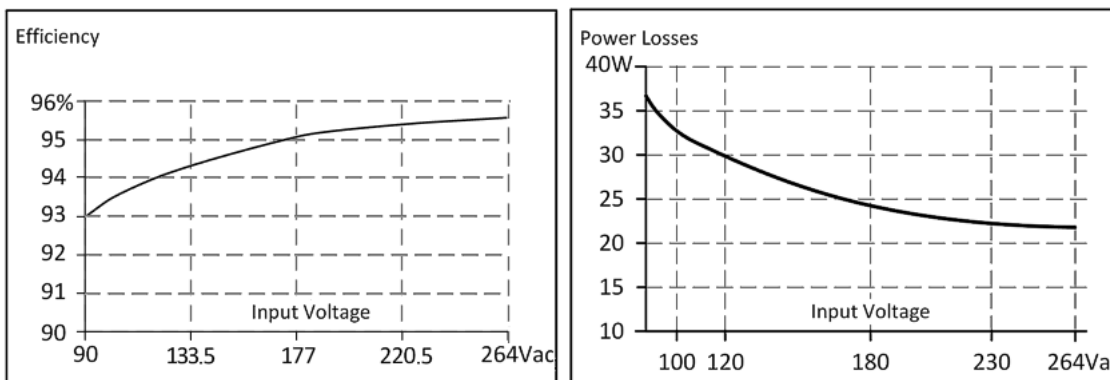


Fig. 10: Efficiency vs. input voltage; losses vs. input voltage

### 3.8 Lifetime expectancy

The lifetime expectancy shown in the table indicates the minimum number of operating hours (service life) and is determined by the lifetime expectancy of the built-in electrolytic capacitors. Lifetime expectancy is specified in operational hours and is calculated according to the capacitor's manufacturer specification. The manufacturer of the electrolytic capacitors only guarantees a maximum service life of up to 15 years (131,400 h). Any number exceeding this value represents a calculated theoretical service life which can be used to compare devices.

	AC 100 V	AC 120 V	AC 230 V	
<b>Lifetime expectancy</b>	159,000 h	185,000 h	223,000 h	At 24 V, 10 A and +40°C
	450,000 h	523,000 h	630,000 h	At 24 V, 10 A and +25°C
	48,000 h	63,000 h	121,000 h	At 24 V, 20 A and +40°C
	137,000 h	179,000 h	422,000 h	At 24 V, 20 A and +25°C
	23,000 h	38,000 h	73,000 h	At 24 V, 24 A and +40°C
	69,000 h	108,000 h	207,000 h	At 24 V, 24 A and +25°C

### 3.9 MTBF

MTBF stands for **Mean Time Between Failure**, which is calculated according to statistical device failures, and indicates reliability of a device. It is a statistical representation of the probability of equipment failure and does not necessarily represent the service life of a product.

The MTBF figure is a statistical representation of the likelihood of a device to fail. A MTBF figure of e.g., 1,000,000 h means that statistically one unit will fail every 100 hours if 10,000 units are installed in the field. However, it cannot be determined if the failed unit has been running for 50,000 h or only for 100 h.

For these types of units, the MTTF value (**Mean Time To Failure**) is the same value as the MTBF value.

	AC 100 V	AC 120 V	AC 230 V	
<b>MTBF SN 29500, IEC 61709</b>	287,000 h	298,000 h	356,000 h	At 24 V, 20 A and +40°C
	525,000 h	543,000 h	632,000 h	At 24 V, 20 A and +25°C
<b>MTBF MIL HDBK 217F</b>	170,000 h	174,000 h	202,000 h	At 24 V, 20 A and +40°C; Ground Benign GB40
	238,000 h	245,000 h	286,000 h	At 24 V, 20 A and +25°C; Ground Benign GB25

### 3.10 Terminals and wiring

The terminals are designed to be finger-safe according to IP20 and are suitable for field and factory wiring.

Technical data	Input	Output	EtherCAT
Connection cross-section	e*: max. 6 mm <sup>2</sup> f*: max. 4 mm <sup>2</sup> a*: max. 4 mm <sup>2</sup> (d < 2.8 mm)	e*: max. 6 mm <sup>2</sup> f*: max. 4 mm <sup>2</sup> a*: max 4 mm <sup>2</sup> (d < 2.8 mm)	Bus interface: 2 x RJ45  Data transfer medium: Ethernet/EtherCAT cable (min. Cat 5), shielded
Connection cross section (AWG)	e*: AWG 20-10 f*: AWG 20-10 a* AWG 20-10 (d <2.8 mm)	e*: AWG 20-10 f*: AWG 20-10 a* AWG 20-10 (d<2.8 mm)	
Strip length	10 mm / 0.28 inch	10 mm / 0.28 inch	

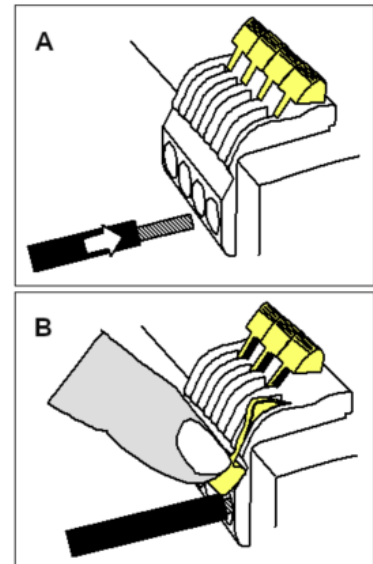
e\* = single-wire, solid wire

f\* = stranded wire

a\* = with ferrule

**Wiring instructions:**

- Use suitable copper cables that are designed for at least the following working temperatures:  
+60°C for ambient temperatures up to +45°C and  
+75°C for ambient temperatures up to +60°C and  
+90°C for ambient temperatures up to +70°C.
- Observe the national installation rules and regulations!
- Insert the wire into the opening (A).
- Turn the lever (B).
- To disconnect the cable, proceed in reverse order.
- Make sure that all single wires of a strand are connected to the terminal!
- Ferrules are permitted.
- Do not use the device without PE connection.
- Observe the information in the chapter "[EtherCAT cabling - wire-bound](#) [▶ 46]"



**Series connection of power supply units**

Series connection (looping from one power supply output to the next) is not allowed. Use a separate distribution terminal strip as shown in Fig. *Using distribution terminals*.

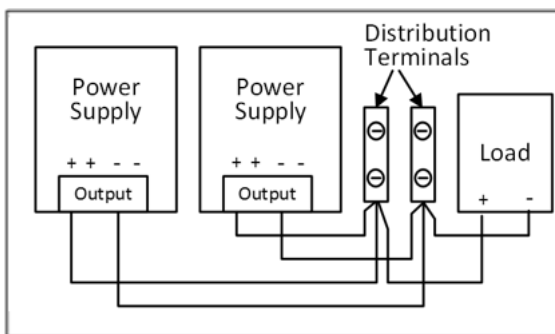


Fig. 11: Using distribution terminals

### 3.11 Functional wiring diagram

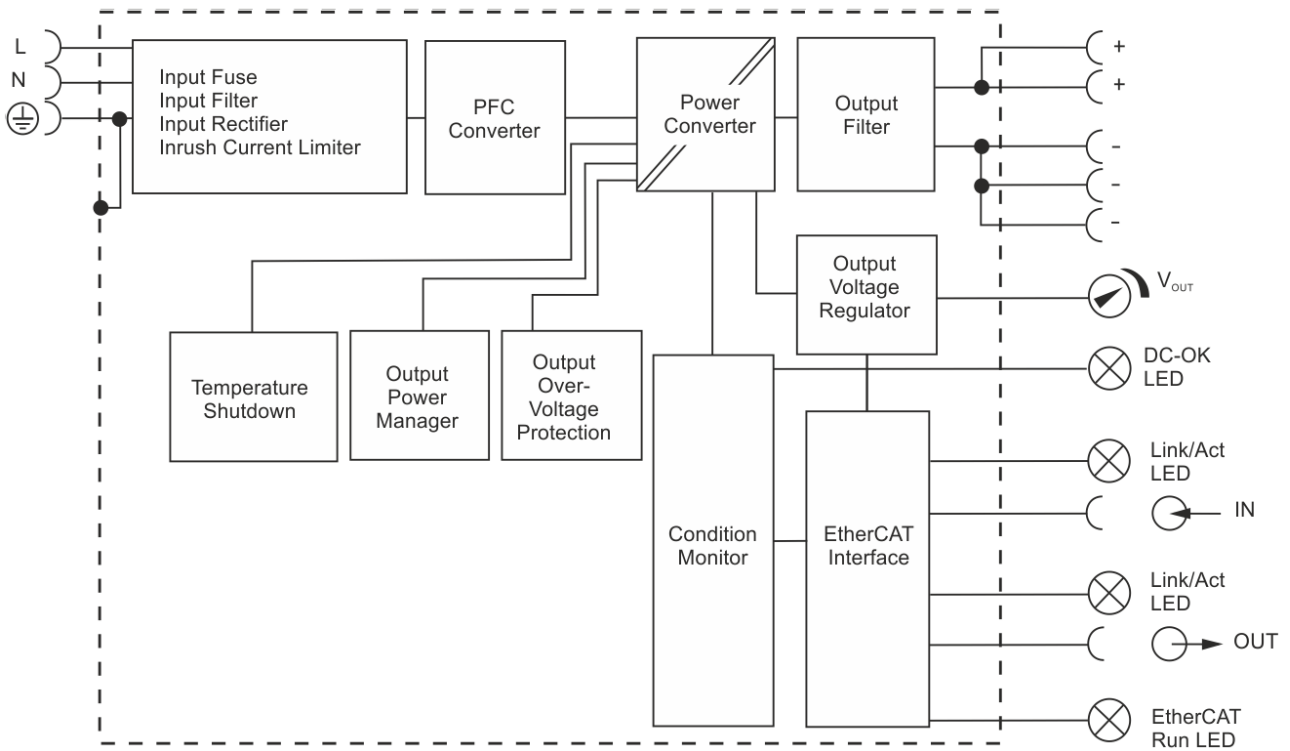



Fig. 12: Functional wiring diagram

### 3.12 Front side and operating elements



Fig. 13: Front PS2001-2420-1001

#### Input terminals (spring-loaded terminals)

Designation (A)	Description
N	Mains input N
L	Mains input L
	PE input (protective conductor)

#### Output terminals (spring-loaded terminals)

Designation (B)	Description
+	2 identical positive poles (internally connected), positive output
-	3 identical negative poles (internally connected), negative output

#### Output voltage potentiometer

Designation (C)	Description
Potentiometer	Set the output voltage. Factory setting: 24.1 V

#### DC-OK LED

Designation (D)	Description
LED green	Lights up when the set threshold value for the output voltage has been exceeded (index 0x8000:16 "DC OK threshold", factory setting: 22 V).

**EtherCAT signal input (RJ45), LED LINK/ACT**

Designation (E)	Description		
LINK / ACT X1 IN	<b>Connection:</b> RJ45 plug and Ethernet/EtherCAT cable (min. Cat.5), shielded (see chapter <a href="#">EtherCAT cabling - wire-bound</a> [ <a href="#">▶ 46</a> ])		
	<b>LED (green):</b>		
	<b>Display</b>	<b>State</b>	<b>Description</b>
	off	-	No connection on the incoming EtherCAT segment
	on	linked	Preceding EtherCAT device connected
flashes	active	Communication with preceding EtherCAT device	

**EtherCAT signal output (RJ45), LED LINK/ACT**

Designation (F)	Description		
LINK / ACT X2 OUT	<b>Connection:</b> RJ45 plug and Ethernet/EtherCAT cable (min. Cat.5), shielded (see chapter <a href="#">EtherCAT cabling - wire-bound</a> [ <a href="#">▶ 46</a> ])		
	<b>LED (green):</b>		
	<b>Display</b>	<b>State</b>	<b>Description</b>
	off	-	No connection on the outgoing EtherCAT segment
	on	linked	Following EtherCAT device connected
flashes	active	Communication with following EtherCAT device	

**EtherCAT - Run LED (green)**

Designation (G)	Description	
	Display	Description
EtherCAT RUN	off	State of the EtherCAT State Machine: <b>INIT</b> = initialization of the EtherCAT device
	flashing	State of the EtherCAT State Machine: <b>PREOP</b> = function for mailbox communication and different default settings set
	single flash	State of the EtherCAT State Machine: <b>SAFEOP</b> = verification of the Sync manager channels Outputs remain in safe state
	on	State of the EtherCAT State Machine: <b>OP</b> = normal operating state; mailbox and process data communication is possible
	flickering / rapidly flashing	State of the EtherCAT State Machine: <b>BOOTSTRAP</b> = function for <a href="#">Firmware updates</a> [ <a href="#">▶ 158</a> ]



### 3.13 EMC

The EMC behavior of the device is designed for applications in industrial environment as well as in residential, commercial, and light industry environments. The output may be earthed or non-earthed.

The device was tested according to EN 61000-6-1, EN 61000-6-2, EN 61000-6-3 and EN 61000-6-4.

Without additional measures to reduce the conducted emissions on the output (e.g. by using a filter), the device is not suited to supply a local DC power network in residential, commercial and light-industrial environments. No restrictions apply for local DC power networks in industrial environments.

EMC interference immunity			
Strong transients	VDE 0160	Over the entire load range	750 V, 0.3 ms Criterion A

EMC interference emission		
Conducted interference emission input lines	EN 55011, EN 55022, FCC Part 15, CISPR 11, CISPR 22	Class B
Interference emission	EN 55011, EN 55032, CISPR 11, CISPR 32	Class A
Harmonic input current	EN 61000-3-2	Class A fulfilled Class C fulfilled between 8 A and 24 A load
Voltage fluctuations, flicker	EN 61000-3-3	Fulfilled (Tested with constant current loads, non-pulsing)

This device complies with FCC Part 15.

Operation is subjected to following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Switching frequencies		
PFC converter	100 kHz	Fixed frequency
Main converter	80 kHz to 140 kHz	Output load dependent
Auxiliary converter	60 kHz	Fixed frequency
Mikrocontroller	24 MHz, 48 MHz, 64 MHz	The device uses a total of three microcontrollers

### 3.14 Environment

Environment		
Operating temperature <sup>1)</sup>	-25°C...+70°C	Reduction of the output power according to Fig. <i>Output power vs. ambient temperature</i>
Storage temperature	-40°C...+85°C	For storage and transport
Output load reduction (derating)	12.5 W / K 28 W / 1000 m or 5 K / 1000 m	+60°C...+70°C For altitudes > 2000 m, see figure <i>Output power vs. Altitude</i>
	The derating is not hardware-controlled. The user must take care by himself to stay below the de-rated current limits in order not to overload the unit.	
Humidity	5 to 95% r.h.	According to IEC 60068-2-30 Do not energize while condensation is present.
Atmospheric pressure	110 – 54 kPa	For details, see figure <i>Output power vs. Altitude</i>
Installation altitude	up to 5000 m	For details, see figure <i>Output power vs. Altitude</i>
Overvoltage category	II	According to IEC 60664 -1, installation altitudes up to 5000 m
Surge voltage strength	4 kV (according to overvoltage category III)	Input to PE According to IEC 60664-1, installation altitudes up to 2000 m
Degree of pollution	2	According to IEC 60664-1, not conductive
Oscillation, sinusoidal <sup>2)</sup>	2 - 17.8 Hz: ±1.6 mm; 17.8 - 500 Hz: 2 g 2 hours/axis	According to IEC 60068-2-6
Bumps <sup>2)</sup>	30 g 6 ms, 20 g 11 ms 3 bumps/direction, 18 bumps in total	According to IEC 60068-2-27

<sup>1)</sup> The operating temperature is the same as the ambient or surrounding temperature and is defined as the air temperature 2 cm below the unit.

<sup>2)</sup> Tested in combination with DIN rails according to EN 60715 with a height of 15 mm and a thickness of 1.3 mm and standard installation position.

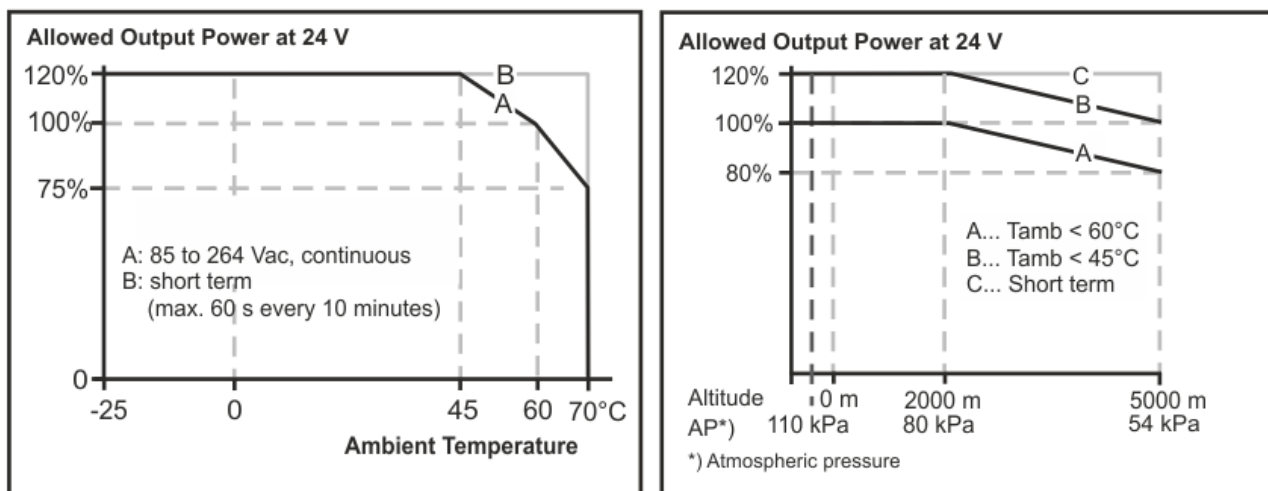


Fig. 14: Output power vs. ambient temperature; Output power vs. Altitude

### 3.15 Protective functions

Protective functions		
Output overvoltage protection	Typ. 30.5 Vdc Max. 32 Vdc	In the event of an internal fault in the power supply, a redundant circuit limits the maximum output voltage to 32 V. The output shuts down and tries to restart three times. If the error persists, the output shuts down. Switch off the input voltage to reset.
Protection rating	IP20	EN/IEC 60529
Overtemperature protection	Yes	Output shut-down with automatic restart. The temperature sensor is installed at critical components within the device and switches off the device in safety-critical situations (e.g. derating requirements not fulfilled, excessive ambient temperature, ventilation blocked). There is no correlation between the operating temperature and the switch-off temperature, since the latter depends on the input voltage, the load and the installation type.
Protection against input transients	MOV (Metal Oxide Varistor)	For further information see chapter <a href="#">EMC [▶ 31]</a>
Input fuse	Included	Non-user replaceable slow blow, high breaking capacity fuse

### 3.16 Safety features

Safety features		
Protection class	I	According to IEC 61140 PE (Protective Earth) connection required
Insulation resistance	> 500 MOhm	In the delivery state between input and output, measured at 500 Vdc
	> 500 MOhm	In the delivery state between input and protective conductor, measured at 500 Vdc
	> 500 MOhm	In the delivery state between output and protective conductor, measured at 500 Vdc
PE resistance	< 0.1 Ohm	Resistance between the protective conductor connection and the housing in the area of the DIN rail mounting bracket.
Leakage current	Typ. 0.12 mA / 0.31 mA	At 100 Vac, 50 Hz, TN, TT / IT network
	Typ. 0.18 mA / 0.45 mA	At 120 Vac, 60 Hz, TN, TT / IT network
	Typ. 0.30 mA / 0.76 mA	At 230 Vac, 50 Hz, TN, TT / IT network
	Max. 0.16 mA / 0.38 mA	At 110 Vac, 50 Hz, TN, TT / IT network
	Max. 0.23 mA / 0.55 mA	At 132 Vac, 60 Hz, TN, TT / IT network
	Max. 0.39 mA / 0.94 mA	At 264 Vac, 50 Hz, TN, TT / IT network

### 3.17 Dielectric strength

The output voltage is earth-free and has no ohmic connection to earth.  
The output is isolated from the input by double or reinforced insulation.

Type and routine tests are conducted by the manufacturer. Field tests may be conducted in the field using the appropriate test equipment which applies the voltage with a slow ramp (2 s up and 2 s down).

1. Connect all input terminals and all output poles to each other before performing the tests.
2. When testing, set the cut-off current settings to the value in the table below.
3. We recommend connecting either the positive pole, the negative pole or another part of the output circuit to the protective conductor system. This avoids situations in which the load starts unexpectedly or cannot be disconnected if an unnoticed earth leakage occurs.
4. To test EtherCAT interfaces, all eight lines of the respective RJ45 plug must also be short-circuited in order to apply the test voltage.

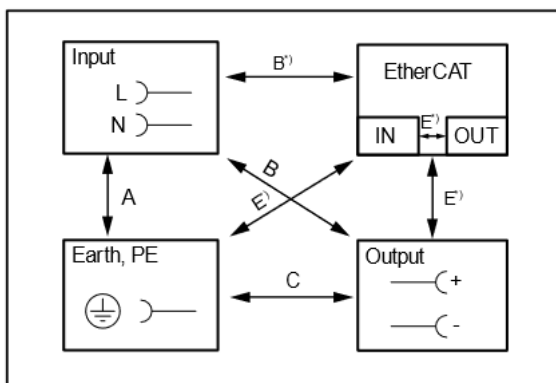





Fig. 15: Dielectric strength

B\*) When testing the EtherCAT inputs and outputs, ensure that the maximum voltage between the EtherCAT cables and the output is not exceeded (column "E-IN" and "E-OUT").

E\*) All tests are performed separately on EtherCAT inputs and outputs.

		<b>A</b>	<b>B</b>	<b>C</b>	<b>E-IN</b>	<b>E-OUT</b>
Type test	60 s	2500 Vac	3000 Vac	1000 Vac	1500 Vac	1500 Vac
Component test	5 s	2500 Vac	2500 Vac	500 Vac	1350 Vac	1350 Vac
Field test	5 s	2000 Vac	2000 Vac	500 Vac	1350 Vac	1350 Vac
Cut-off current setting		> 10 mA	> 10 mA	> 20 mA	> 1 mA	> 1 mA

### 3.18 Declaration of conformity and approvals

	EU declaration of conformity
	UL Certificate, UL 61010-1/2-201 Applicable for US and Canada
	Test Report Voltage Sag Immunity for Semiconductor Processing Equipment tested for AC 208 V L-L or L-N mains voltages, nominal output voltage and nominal output load

### 3.19 Dimensions and weight

Dimensions and weight	
Overall width	48 mm
Height	124 mm
Depth	127 mm The height of the DIN rails must be added to the depth of the device to calculate the total installation depth required
Weight	835 g
DIN rail	Use 35 mm DIN rails according to EN 60715 or EN 50022 with a height of 7.5 or 15 mm.
Housing material	Body: Aluminum alloy Cover: zinc-plated steel
Penetration protection	Small parts like screws, nuts, etc. with a diameter larger than 5 mm
Installation clearances	See chapter on <a href="#">Safety instructions and installation requirements</a> [▶ 11]

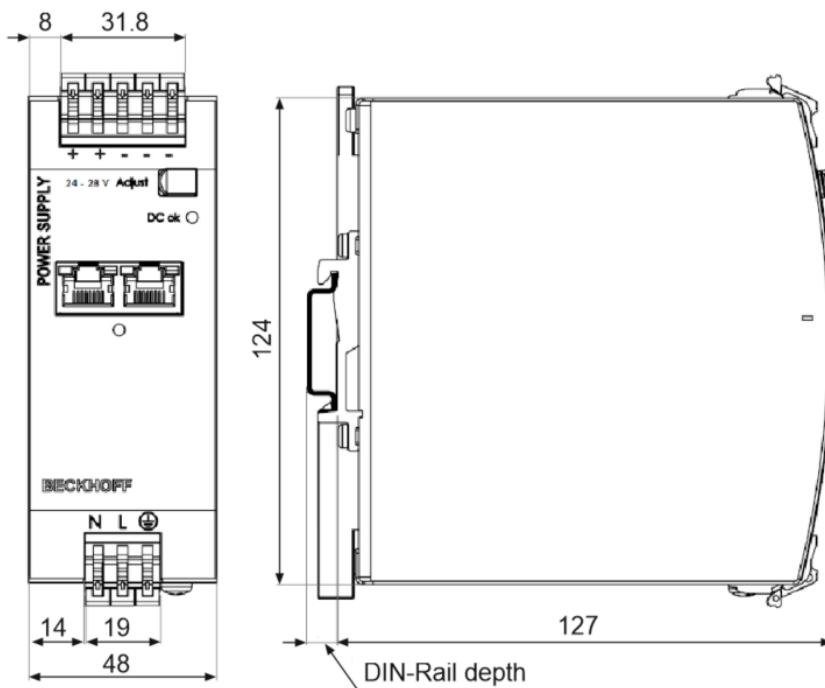


Fig. 16: Front/side view PS2001-2420-1001

## 4 Application notes

### 4.1 Peak current capability

The unit can deliver peak currents (up to several milliseconds) which are higher than the specified short-term currents.

This helps to start current demanding loads. Solenoids, contactors and pneumatic modules often have a steady state coil and a pick-up coil. The inrush current demand of the pick-up coil is several times higher than the steady-state current and usually exceeds the nominal output current (including the Extra Power). The same situation applies when starting a capacitive load.

The peak current capability also ensures the safe operation of subsequent circuit breakers of load circuits. The load circuits are often individually fused with circuit breakers or fuses. In case of a short or an overload in one branch circuit, the fuse or circuit breaker need a certain amount of over-current to open in a timely manner. This prevents a voltage drop in adjacent circuits.

The extra current (peak current) is supplied by the power converter and the built-in large sized output capacitors of the power supply. The capacitors are discharged during such an event, which leads to a voltage drop at the output. The following three examples show typical voltage drops for ohmic loads:

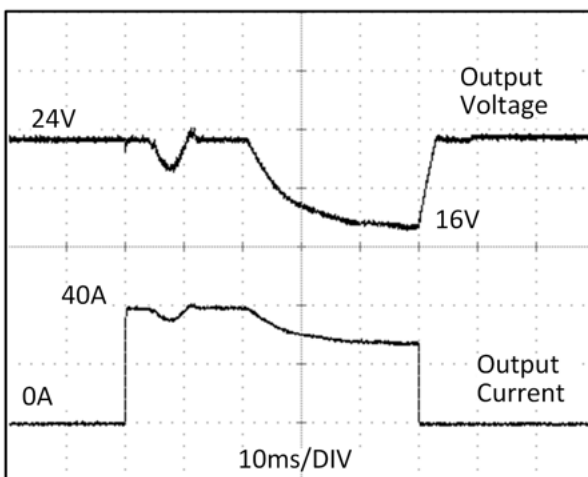


Fig. 17: 40 A peak current for 50 ms, typ. (2x nominal current)

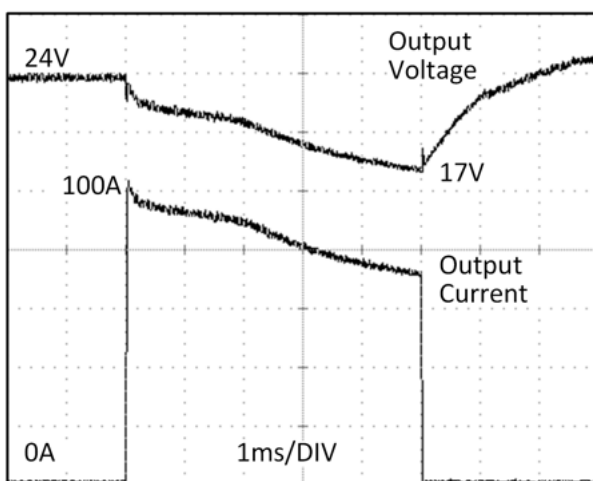


Fig. 18: 100 A peak current for 5 ms, typ. (5x nominal current)

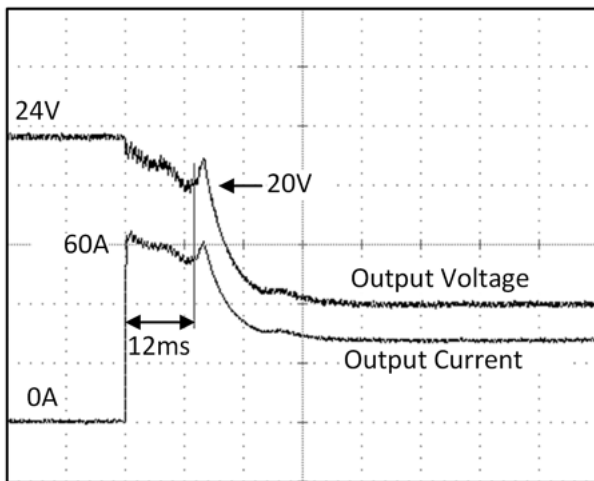


Fig. 19: 60 A peak current for 12 ms, typ. (3x nominal current)

Peak current voltage drops	
Typically from 24 V to 16 V	At 40 A for 50 ms, ohmic load
Typically from 24 V to 21 V	At 100 A for 2 ms, ohmic load
Typically from 24 V to 17 V	At 100 A for 5 ms, ohmic load



## 4.2 Output circuit breakers

Standard circuit breakers (or UL1077 circuit breakers) are generally used for AC supply systems and can also be used for 24 V branches.

Circuit breakers are used to protect wires and circuits. If the ampere value and the characteristics of the circuit breaker are matched to the wire thickness used, the wiring is considered thermally safe, regardless of whether the circuit breaker opens or not.

To avoid voltage drops and situations with undervoltage in adjacent 24 V branches fed from the same source, a fast (magnetic) trip of the circuit breaker is desirable. Fast shutdown within 10 ms is required, which approximately corresponds to the hold-up time of PLCs. For this purpose, power supplies with high current reserves and large output capacitors are advantageous. Furthermore, the impedance of the faulty branch must be sufficiently small in order for the current to actually flow.

The following table contains typical test results that show which circuit breakers with B and C characteristics trip magnetically, depending on the wire cross-section and the wire length.

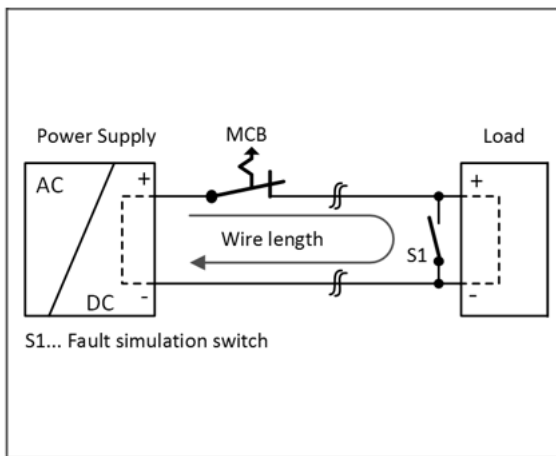


Fig. 20: Test circuit

Maximal wire length<sup>1)</sup> for a fast (magnetic) tripping:

	0.75 mm <sup>2</sup>	1.0 mm <sup>2</sup>	1.5 mm <sup>2</sup>	2.5 mm <sup>2</sup>
<b>C-2A</b>	31 m	37 m	63 m	98 m
<b>C-3A</b>	28 m	34 m	51 m	78 m
<b>C-4A</b>	18 m	25 m	38 m	58 m
<b>C-6A</b>	9 m	11 m	18 m	26 m
<b>C-8A</b>	6 m	7 m	12 m	14 m
<b>C-10A</b>	4 m	6 m	11 m	13 m
<b>C-13A</b>	2 m	2 m	4 m	7 m

	0.75 mm <sup>2</sup>	1.0 mm <sup>2</sup>	1.5 mm <sup>2</sup>	2.5 mm <sup>2</sup>
<b>B-6A</b>	23 m	28 m	46 m	66 m
<b>B-10A</b>	11 m	14 m	19 m	32 m
<b>B-13A</b>	7 m	11 m	16 m	29 m
<b>B-16A</b>	5 m	6 m	8 m	15 m
<b>B-20A</b>	1 m	1 m	2 m	4 m
<b>B-25A</b>				1 m

<sup>1)</sup> Don't forget to double the distance to the load (or the cable length) when calculating the total cable length (plus and minus cable).

## 4.3 Charging batteries

The power supply can be used to charge lead-acid batteries or maintenance-free batteries (SLA or VRLA batteries). Two 12 V batteries connected in series are required.

### Instructions for charging batteries:

- Use only matched batteries when putting 12 V types in series.
- Ensure that the ambient temperature of the power supply remains below 40°C.
- Use a 30 A or 32 A circuit breaker or a decoupling diode between the power supply and the battery.
- Ensure that the output current of the power supply is below the allowed charging current of the battery.
- The reverse current to the power supply is typically 3.5 mA. This reverse current can discharge the battery when the power supply is switched off, unless a decoupling diode is used.
- Switch the device to "Parallel Use" mode and adjust the output voltage, measured at no load and at the battery end of the cable, very precisely to the end-of-charge voltage.

<b>End-of-charge voltage</b>	27.8 V	27.5 V	27.15 V	26.8 V
<b>Battery temperature</b>	10°C	20°C	30°C	40°C

## 4.4 Series connection

Power supplies of the same type can be connected in series for higher output voltages. As many devices can be connected in series as necessary, as long as the sum of the output voltages does not exceed 150 Vdc. Voltages with a potential higher than 60 Vdc are no longer regarded as safety extra-low voltage and can be dangerous. Such voltages must be installed with a protection against touching.

Avoid return voltage (e.g., from a decelerating motor or battery) which is applied to the output terminals.

### Restrictions:

- Keep an installation clearance of 15 mm (left / right) between two power supplies.
- Do not install the power supplies on top of each other.
- Pay attention that leakage current, EMI, inrush current, harmonics will increase when using multiple power supplies.

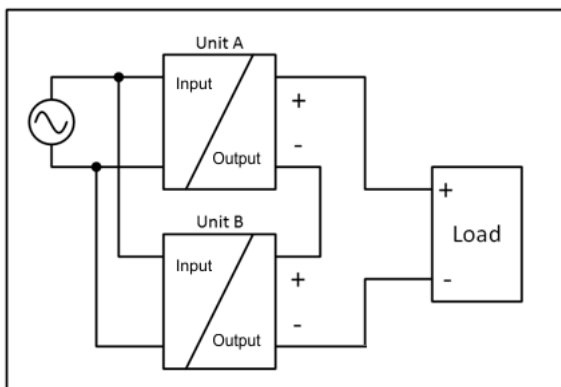


Fig. 21: Series connection

## 4.5 Parallel use to increase power

PS2001-2420-1001 power supplies can be connected in parallel to increase the output power.

1. The output voltage of all power supplies must be set to the same value ( $\pm 100$  mV) in "Single Use" mode and with the same load conditions on all units, or the units can be left with the factory settings.
2. After the adjustments, set the unit to "Parallel Use" mode (see chapter [Parallel Use](#) [► 135]), in order to achieve load sharing. The "Parallel Use" mode regulates the output voltage in such a manner that the voltage at no load is approx. 4% higher than at nominal load, see also chapter [Output](#) [► 21].
3. The ambient temperature is not allowed to exceed  $+60^{\circ}\text{C}$ .
4. If more than three units are connected in parallel, a fuse or circuit breaker with a rating of 30 A or 32 A is required on each output. Alternatively a diode can be used.
5. Energize all devices simultaneously. It may also be necessary to ramp up and shut down the input power (turn off for at least five seconds) if the output was overloaded or short-circuited and the required output current is higher than the current of a device.

### Restrictions:

- Keep an installation clearance of 15 mm (left / right) between two power supplies.
- Do not install the power supplies on top of each other.
- Pay attention that leakage current, EMI, inrush current, harmonics will increase when using multiple power supplies.

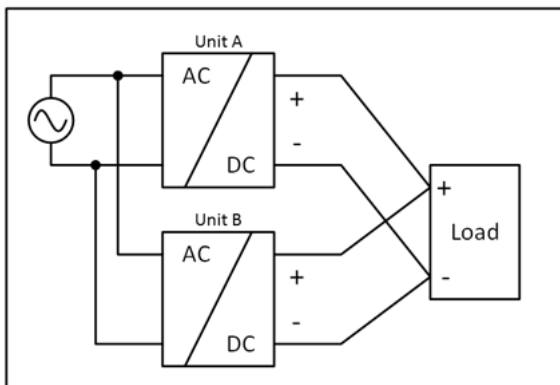


Fig. 22: Parallel connection

## 4.6 Parallel use for redundancy

### Recommendations for the construction of redundant power supply systems:

- Power supplies that contain a DC-OK signal contact are best suited for the construction of redundant power supply systems.
  - Use this DC-OK signal contact to monitor the individual power supplies.
  - For the power supplies with EtherCAT (e.g. PS2001-2420-1001) the DC-OK bit can be used for monitoring.
- Use separate grids for each power supply whenever possible.
- It is recommended to set the output voltages of all devices to the same value ( $\pm 100$  mV) or to leave them at the factory setting.
- In addition, parallel use must be enabled via the "PSU Features" object in index 0x8001:01 "Parallel use", see chapter "[Parallel Use \[► 135\]](#)".

### 4.6.1 1+1 Redundancy

It is possible to connect power supplies in parallel for redundancy operation in order to achieve better system availability. Redundant systems provide a certain amount of additional power to operate the load when a power supply unit fails. The simplest method is to connect two power supplies in parallel. This is called 1+1 redundancy. If one power supply fails, the other can automatically supply the load current without interruption.

The 1+1 redundancy allows ambient temperatures up to +70°C.

**i** **Notes on parallel use for 1+1 redundancy**

- Be sure to use a redundancy module in order to decouple devices from one another. This prevents the defective device from becoming a load for the other device and the output voltage no longer being maintainable.
- Note that leakage current, EMI, input inrush current and harmonics increase when using multiple devices.

**Wiring example:**

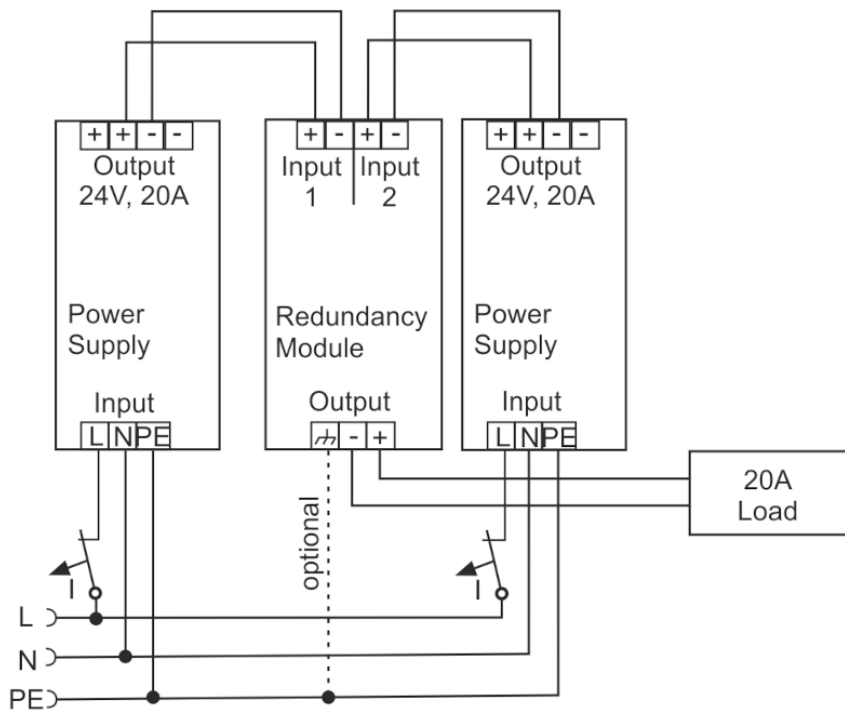


Fig. 23: Wiring for 1+1 redundancy with one redundancy module PS9401-2440-0000

## 4.6.2 N+1 Redundancy

### Notes on parallel use for N + 1 redundancy



- Note that leakage current, EMI, inrush current and harmonics increase when using multiple devices.
- For N + 1 redundancy the ambient temperature is not allowed to exceed +60°C.

Redundant systems for a higher power demand are usually built in a N + 1 method. E.g. four devices, each rated for 20 A are paralleled to build a 60 A redundant system.

- Keep an installation clearance of 15 mm (left / right) between two power supplies.
- Do not install the power supplies on top of each other.

### Wiring example:

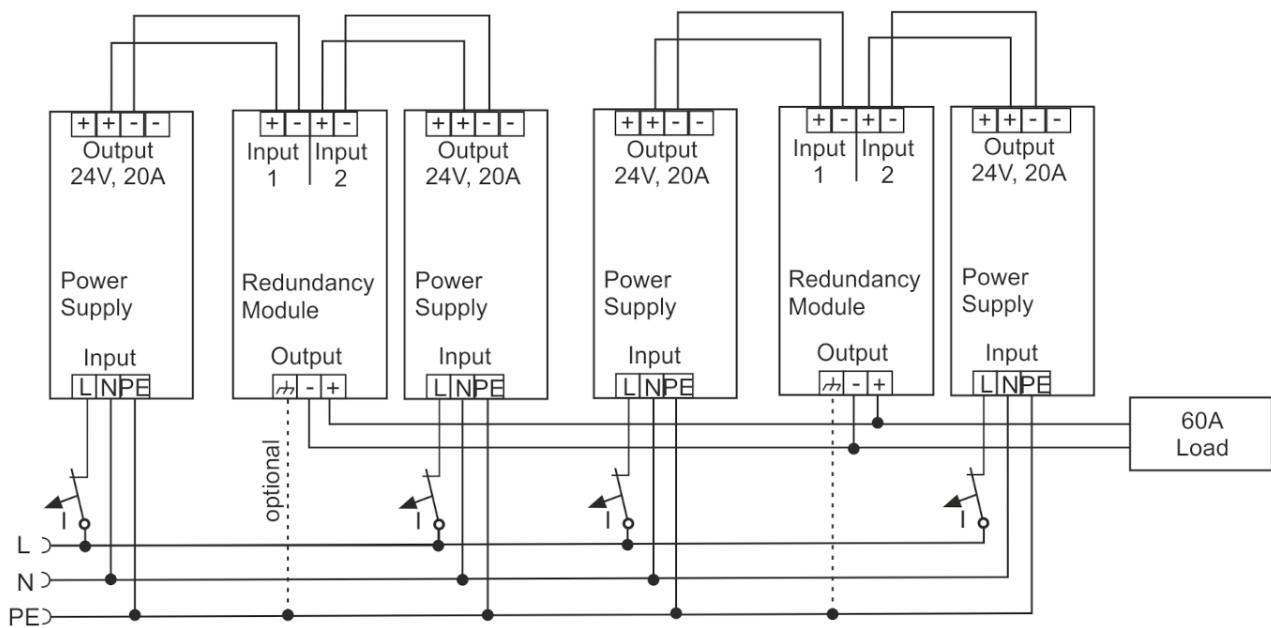


Fig. 24: Wiring for n+1 redundancy with two redundancy modules PS9401-2440-0000

## 4.7 Operation on two phases

The power supply can also be used on two phases of a three-phase system. Such a phase-to-phase connection is permissible as long as the supply voltage is below 240 V+10%.

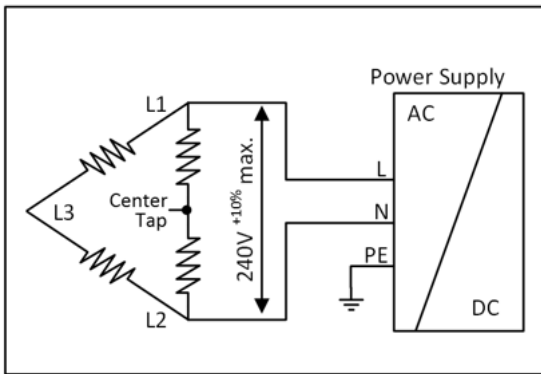


Fig. 25: Operation on two phases

Make sure that the connection cable connected to the N terminal is fused appropriately.

## 4.8 Installation positions

Install the device onto a DIN rail with the input terminals on the bottom of the device (see following fig. *Mounting position A (Standard mounting position)*)

The test environment is set up installing wiring ducts with a depth of 80 mm with clearances seen in the following table.

### Minimum installation clearances

Output power in relation to the nominal output	Minimum installation clearances (mm)		
	top	bottom	Left and right side
< 50%	40	20	0
≥ 50%	40	20	5
≥ 90%	40	20	15

The following curve gives an indication for the allowed output power for altitudes up to 2000 m at an input voltage range of 100 - 240 Vac.

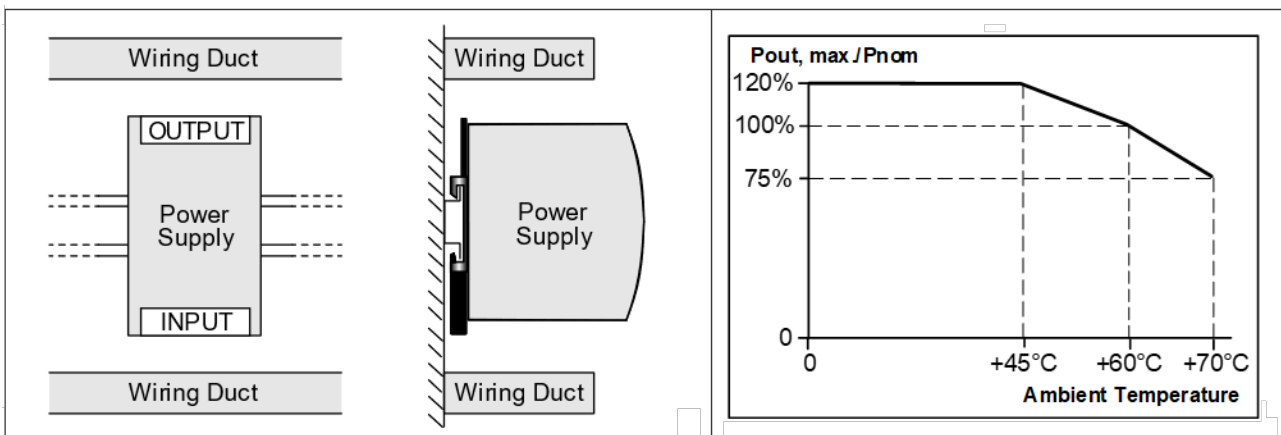


Fig. 26: Mounting position A (Standard mounting position)

## 5 Basics communication

### 5.1 EtherCAT basics

Please refer to the [EtherCAT System Documentation](#) for the EtherCAT fieldbus basics.

### 5.2 EtherCAT cabling – wire-bound

The maximum permissible cable length between two EtherCAT devices is 100 m. This results from the FastEthernet technology, which, above all for reasons of signal attenuation over the length of the cable, allows a maximum link length of 5 + 90 + 5 m if cables with appropriate properties are used. See also the [Design recommendations for EtherCAT/Ethernet infrastructure](#).

#### Cables and connectors

For connecting EtherCAT devices only Ethernet connections (cables + plugs) that meet the requirements of at least category 5 (Cat5) according to EN 50173 or ISO/IEC 11801 should be used. EtherCAT uses 4 wires for signal transfer.

EtherCAT uses RJ45 plug connectors, for example. The pin assignment is compatible with the Ethernet standard (ISO/IEC 8802-3).

Pin	Color of conductor	Signal	Description
1	yellow	TD +	Transmission Data +
2	orange	TD -	Transmission Data -
3	white	RD +	Receiver Data +
6	blue	RD -	Receiver Data -

Due to automatic cable detection (auto-crossing) symmetric (1:1) or cross-over cables can be used between EtherCAT devices from Beckhoff.

#### ● Recommended cables

- i** It is recommended to use the appropriate Beckhoff components e.g.
- cable sets ZK1090-9191-xxxx respectively
  - RJ45 connector, field assembly ZS1090-0005
  - EtherCAT cable, field assembly ZB9010, ZB9020

Suitable cables for the connection of EtherCAT devices can be found on the [Beckhoff website!](#)



### 5.3 General notes for setting the watchdog

The EtherCAT terminals are equipped with a safety device (watchdog) which, e. g. in the event of interrupted process data traffic, switches the outputs (if present) to a presettable state after a presettable time, depending on the device and setting, e. g. to FALSE (off) or an output value.

The EtherCAT slave controller (ESC) features two watchdogs:

- SM watchdog (default: 100 ms)
- PDI watchdog (default: 100 ms)

Their times are individually parameterized in TwinCAT as follows:

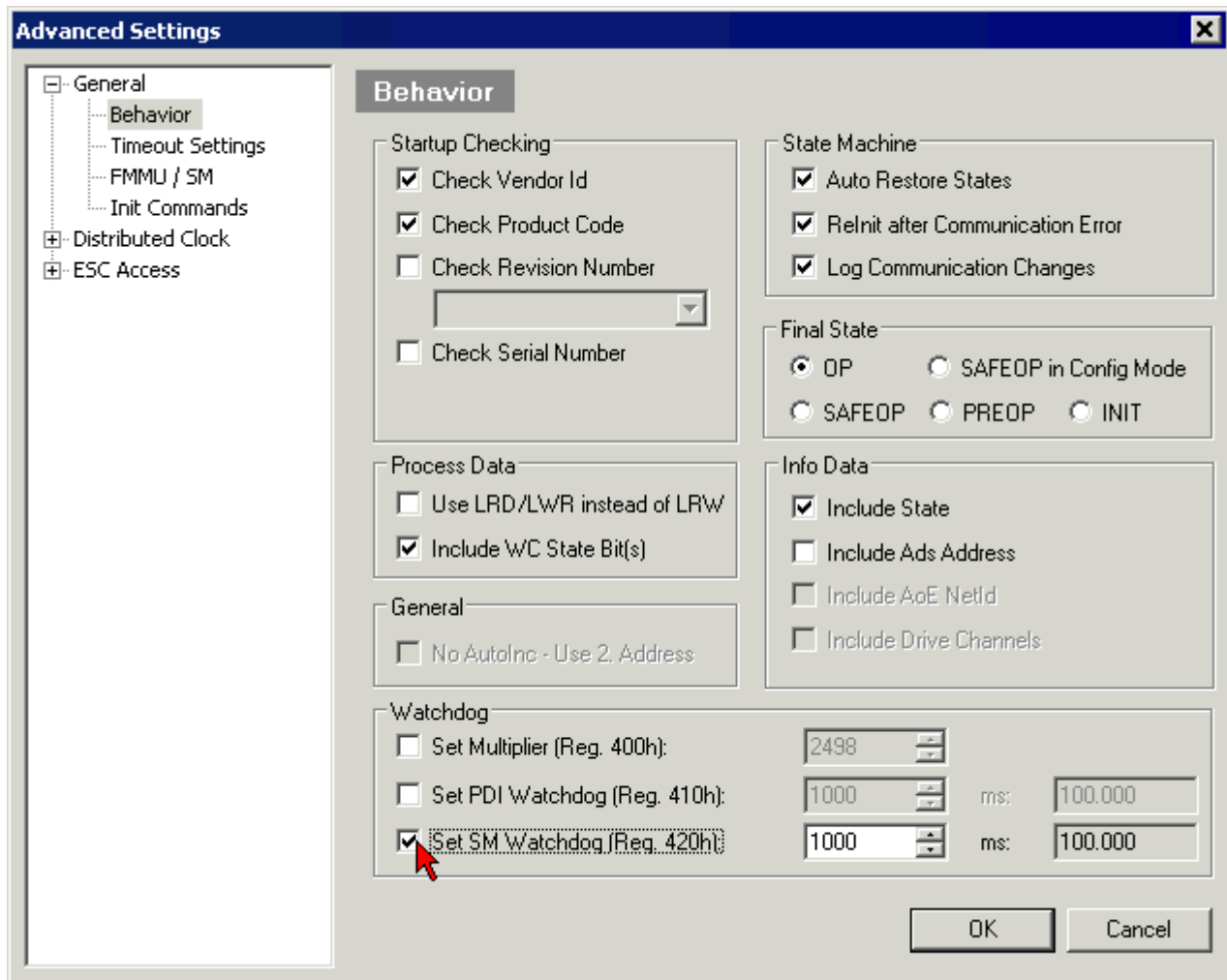


Fig. 27: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the Multiplier Register 400h (hexadecimal, i. e. x0400) is valid for both watchdogs.
- each watchdog has its own timer setting 410h or 420h, which together with the Multiplier results in a resulting time.
- important: the Multiplier/Timer setting is only loaded into the slave at EtherCAT startup if the checkbox in front of it is activated.
- if it is not checked, nothing is downloaded and the setting located in the ESC remains unchanged.
- the downloaded values can be seen in the ESC registers x0400/0410/0420: ESC Access -> Memory

### SM watchdog (SyncManager Watchdog)

The SyncManager watchdog is reset with each successful EtherCAT process data communication with the terminal. If, for example, no EtherCAT process data communication with the terminal takes place for longer than the set and activated SM watchdog time due to a line interruption, the watchdog is triggered. The status of the terminal (usually OP) remains unaffected. The watchdog is only reset again by a successful EtherCAT process data access.

The SyncManager watchdog is therefore a monitoring for correct and timely process data communication with the ESC from the EtherCAT side.

The maximum possible watchdog time depends on the device. For example, for "simple" EtherCAT slaves (without firmware) with watchdog execution in the ESC it is usually up to 170 seconds. For complex EtherCAT slaves (with firmware) the SM watchdog function is usually parameterized via Reg. 400/420 but executed by the  $\mu$ C and can be significantly lower. In addition, the execution may then be subject to a certain time uncertainty. Since the TwinCAT dialog may allow inputs up to 65535, a test of the desired watchdog time is recommended.

### PDI watchdog (Process Data Watchdog)

If there is no PDI communication with the EtherCAT slave controller (ESC) for longer than the set and activated PDI watchdog time, this watchdog is triggered.

PDI (Process Data Interface) is the internal interface of the ESC, e.g. to local processors in the EtherCAT slave. With the PDI watchdog this communication can be monitored for failure.

The PDI watchdog is therefore a monitoring for correct and timely process data communication with the ESC, but viewed from the application side.

### Calculation

Watchdog time =  $[1/25 \text{ MHz} * (\text{Watchdog multiplier} + 2)] * \text{PDI/SM watchdog}$

Example: default setting Multiplier=2498, SM watchdog=1000 -> 100 ms

The value in Multiplier + 2 corresponds to the number of 40ns base ticks representing one watchdog tick.

#### CAUTION

##### Undefined state possible!

The function for switching off the SM watchdog via SM watchdog = 0 is only implemented in terminals from version -0016. In previous versions this operating mode should not be used.

#### CAUTION

##### Damage of devices and undefined state possible!

If the SM watchdog is activated and a value of 0 is entered the watchdog switches off completely. This is the deactivation of the watchdog! Set outputs are NOT set in a safe state if the communication is interrupted.

## 5.4 EtherCAT State Machine

The state of the EtherCAT slave is controlled via the EtherCAT State Machine (ESM). Depending upon the state, different functions are accessible or executable in the EtherCAT slave. Specific commands must be sent by the EtherCAT master to the device in each state, particularly during the bootup of the slave.

A distinction is made between the following states:

- Init
- Pre-Operational
- Safe-Operational and
- Operational
- Boot

The regular state of each EtherCAT slave after bootup is the OP state.

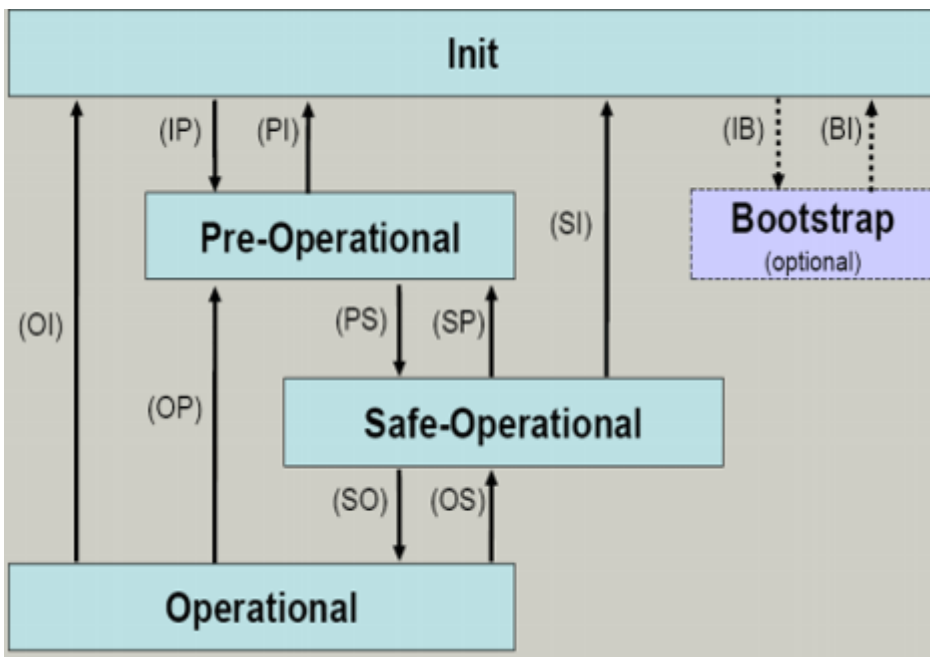


Fig. 28: States of the EtherCAT State Machine

### Init

After switch-on the EtherCAT slave in the *Init* state. No mailbox or process data communication is possible. The EtherCAT master initializes sync manager channels 0 and 1 for mailbox communication.

### Pre-Operational (Pre-Op)

During the transition between *Init* and *Pre-Op* the EtherCAT slave checks whether the mailbox was initialized correctly.

In *Pre-Op* state mailbox communication is possible, but not process data communication. The EtherCAT master initializes the sync manager channels for process data (from sync manager channel 2), the FMMU channels and, if the slave supports configurable mapping, PDO mapping or the sync manager PDO assignment. In this state the settings for the process data transfer and perhaps terminal-specific parameters that may differ from the default settings are also transferred.

### Safe-Operational (Safe-Op)

During transition between *Pre-Op* and *Safe-Op* the EtherCAT slave checks whether the sync manager channels for process data communication and, if required, the distributed clocks settings are correct. Before it acknowledges the change of state, the EtherCAT slave copies current input data into the associated DP-RAM areas of the EtherCAT slave controller (ECSC).

In *Safe-Op* state mailbox and process data communication is possible, although the slave keeps its outputs in a safe state, while the input data are updated cyclically.

### ● Outputs in SAFEOP state

**I** The default set watchdog monitoring sets the outputs of the module in a safe state - depending on the settings in SAFEOP and OP - e.g. in OFF state. If this is prevented by deactivation of the watchdog monitoring in the module, the outputs can be switched or set also in the SAFEOP state.

### Operational (Op)

Before the EtherCAT master switches the EtherCAT slave from *Safe-Op* to *Op* it must transfer valid output data.

In the *Op* state the slave copies the output data of the masters to its outputs. Process data and mailbox communication is possible.

### Boot

In the *Boot* state the slave firmware can be updated. The *Boot* state can only be reached via the *Init* state.

In the *Boot* state mailbox communication via the *file access over EtherCAT* (FoE) protocol is possible, but no other mailbox communication and no process data communication.

## 5.5 CoE Interface

### General description

The CoE interface (CAN application protocol over EtherCAT) is used for parameter management of EtherCAT devices. EtherCAT slaves or the EtherCAT master manage fixed (read only) or variable parameters which they require for operation, diagnostics or commissioning.

CoE parameters are arranged in a table hierarchy. In principle, the user has read access via the fieldbus. The EtherCAT master (TwinCAT System Manager) can access the local CoE lists of the slaves via EtherCAT in read or write mode, depending on the attributes.

Different CoE parameter types are possible, including string (text), integer numbers, Boolean values or larger byte fields. They can be used to describe a wide range of features. Examples of such parameters include manufacturer ID, serial number, process data settings, device name, calibration values for analog measurement or passwords.

The order is specified in two levels via hexadecimal numbering: (main)index, followed by subindex. The value ranges are

- Index: 0x0000 ...0xFFFF (0...65535<sub>dec</sub>)
- SubIndex: 0x00...0xFF (0...255<sub>dec</sub>)

A parameter localized in this way is normally written as 0x8010:07, with preceding "0x" to identify the hexadecimal numerical range and a colon between index and subindex.

The relevant ranges for EtherCAT fieldbus users are:

- 0x1000: This is where fixed identity information for the device is stored, including name, manufacturer, serial number etc., plus information about the current and available process data configurations.
- 0x8000: This is where the operational and functional parameters for all channels are stored, such as filter settings or output frequency.

Other important ranges are:

- 0x4000: here are the channel parameters for some EtherCAT devices. Historically, this was the first parameter area before the 0x8000 area was introduced. EtherCAT devices that were previously equipped with parameters in 0x4000 and changed to 0x8000 support both ranges for compatibility reasons and mirror internally.
- 0x6000: Input PDOs ("input" from the perspective of the EtherCAT master)

- 0x7000: Output PDOs (“output” from the perspective of the EtherCAT master)

**i Availability**

Not every EtherCAT device must have a CoE list. Simple I/O modules without dedicated processor usually have no variable parameters and therefore no CoE list.

If a device has a CoE list, it is shown in the TwinCAT System Manager as a separate tab with a listing of the elements:

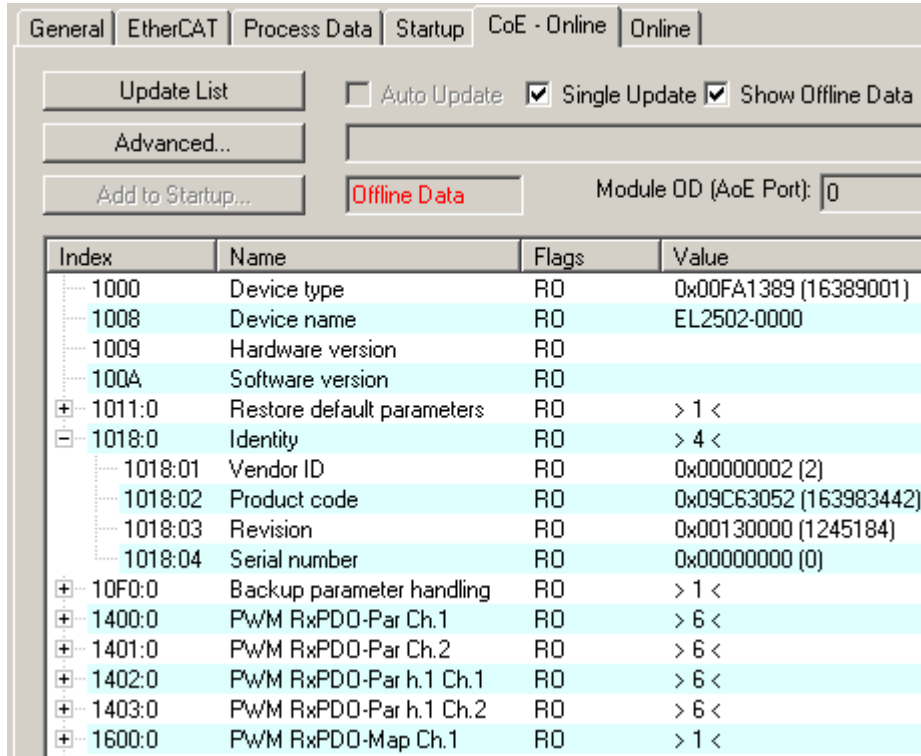


Fig. 29: “CoE Online” tab

The figure above shows the CoE objects available in device “EL2502”, ranging from 0x1000 to 0x1600. The subindices for 0x1018 are expanded.

**NOTICE**

**Changes in the CoE directory (CAN over EtherCAT), program access**

When using/manipulating the CoE parameters observe the general CoE notes in chapter “CoE interface” of the EtherCAT system documentation:

- Keep a startup list if components have to be replaced,
- Distinction between online/offline dictionary,
- Existence of current XML description (download from the [Beckhoff website](#)),
- “CoE-Reload” for resetting the changes
- Program access during operation via PLC (see [TwinCAT3 | PLC Library: Tc2 EtherCAT](#) and [Example program R/W CoE](#))

**Data management and function “NoCoeStorage”**

Some parameters, particularly the setting parameters of the slave, are configurable and writeable. This can be done in write or read mode

- via the System Manager (Fig. “CoE Online” tab) by clicking  
This is useful for commissioning of the system/slaves. Click on the row of the index to be parameterized and enter a value in the “SetValue” dialog.

- from the control system/PLC via ADS, e.g. through blocks from the TcEtherCAT.lib library  
This is recommended for modifications while the system is running or if no System Manager or operating staff are available.

**i Data management**

If slave CoE parameters are modified online, Beckhoff devices store any changes in a fail-safe manner in the EEPROM, i.e. the modified CoE parameters are still available after a restart. The situation may be different with other manufacturers.

An EEPROM is subject to a limited lifetime with respect to write operations. From typically 100,000 write operations onwards it can no longer be guaranteed that new (changed) data are reliably saved or are still readable. This is irrelevant for normal commissioning. However, if CoE parameters are continuously changed via ADS at machine runtime, it is quite possible for the lifetime limit to be reached. Support for the NoCoeStorage function, which suppresses the saving of changed CoE values, depends on the firmware version.

Please refer to the technical data in this documentation as to whether this applies to the respective device.

- If the function is supported: the function is activated by entering the code word 0x12345678 once in CoE 0xF008 and remains active as long as the code word is not changed. After switching the device on it is then inactive. Changed CoE values are not saved in the EEPROM and can thus be changed any number of times.
- Function is not supported: continuous changing of CoE values is not permissible in view of the lifetime limit.

**i Startup list**

Changes in the local CoE list of the terminal are lost if the terminal is replaced. If a terminal is replaced with a new Beckhoff terminal, it will have the default settings. It is therefore advisable to link all changes in the CoE list of an EtherCAT slave with the Startup list of the slave, which is processed whenever the EtherCAT fieldbus is started. In this way a replacement EtherCAT slave can automatically be parameterized with the specifications of the user.

If EtherCAT slaves are used which are unable to store local CoE values permanently, the Startup list must be used.

**Recommended approach for manual modification of CoE parameters**

- Make the required change in the System Manager  
The values are stored locally in the EtherCAT slave
- If the value is to be stored permanently, enter it in the Startup list.  
The order of the Startup entries is usually irrelevant.

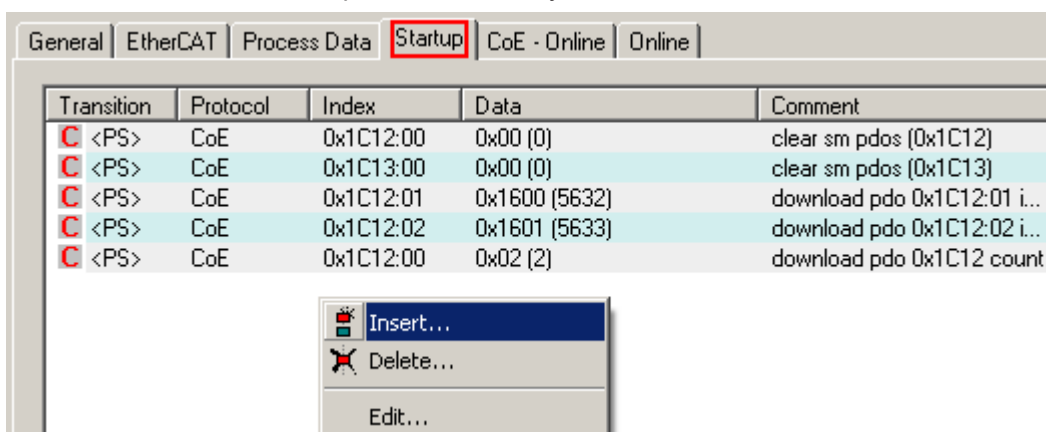


Fig. 30: Startup list in the TwinCAT System Manager

The Startup list may already contain values that were configured by the System Manager based on the ESI specifications. Additional application-specific entries can be created.

**Online/offline list**

While working with the TwinCAT System Manager, a distinction has to be made whether the EtherCAT device is “available”, i.e. switched on and linked via EtherCAT and therefore **online**, or whether a configuration is created **offline** without connected slaves.

In both cases a CoE list as shown in Fig. “CoE online tab” is displayed. The connectivity is shown as offline/online.

- If the slave is offline
  - The offline list from the ESI file is displayed. In this case modifications are not meaningful or possible.
  - The configured status is shown under Identity.
  - No firmware or hardware version is displayed, since these are features of the physical device.
  - **Offline** is shown in red.

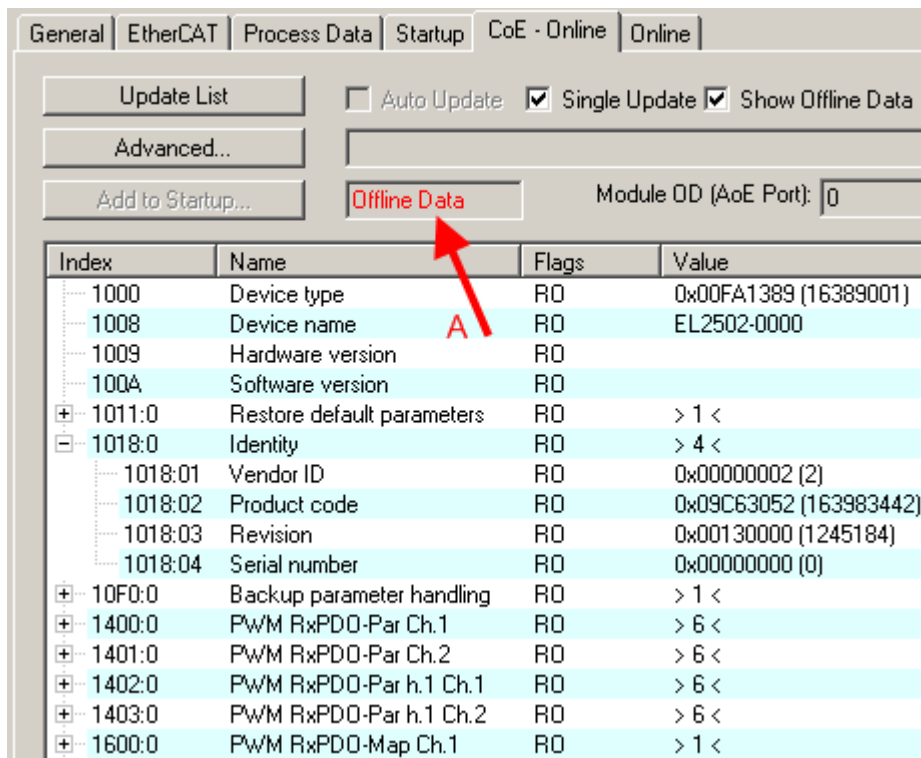


Fig. 31: Offline list

- If the slave is online
  - The actual current slave list is read. This may take several seconds, depending on the size and cycle time.
  - The actual identity is displayed
  - The firmware and hardware version of the equipment according to the electronic information is displayed
  - **Online** is shown in green.

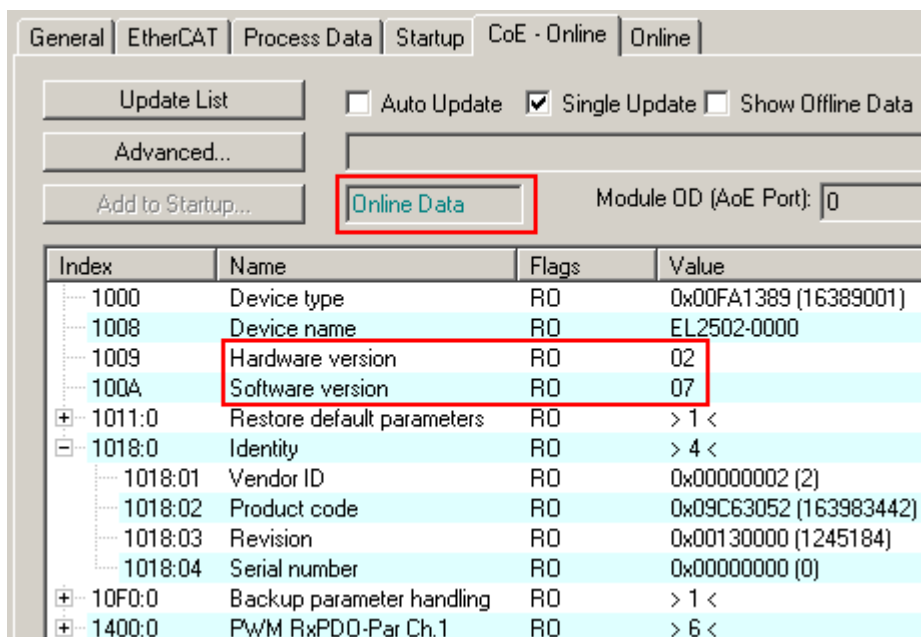


Fig. 32: Online list

### Channel-based order

The CoE list is available in EtherCAT devices that usually feature several functionally equivalent channels. For example, a 4-channel analog 0...10 V input terminal also has four logical channels and therefore four identical sets of parameter data for the channels. In order to avoid having to list each channel in the documentation, the placeholder “n” tends to be used for the individual channel numbers.

In the CoE system 16 indices, each with 255 subindices, are generally sufficient for representing all channel parameters. The channel-based order is therefore arranged in  $16_{dec}/10_{hex}$  steps. The parameter range 0x8000 exemplifies this:

- Channel 0: parameter range 0x8000:00 ... 0x800F:255
- Channel 1: parameter range 0x8010:00 ... 0x801F:255
- Channel 2: parameter range 0x8020:00 ... 0x802F:255
- ...

This is generally written as 0x80n0.

Detailed information on the CoE interface can be found in the [EtherCAT system documentation](#) on the Beckhoff website.



## 5.6 Distributed Clock

The distributed clock represents a local clock in the EtherCAT slave controller (ESC) with the following characteristics:

- Unit *1 ns*
- Zero point *1.1.2000 00:00*
- Size *64 bit* (sufficient for the next 584 years; however, some EtherCAT slaves only offer 32-bit support, i.e. the variable overflows after approx. 4.2 seconds)
- The EtherCAT master automatically synchronizes the local clock with the master clock in the EtherCAT bus with a precision of < 100 ns.

For detailed information please refer to the [EtherCAT system description](#).

## 6 Commissioning

### 6.1 TwinCAT Quick Start

TwinCAT is a development environment for real-time control including a multi PLC system, NC axis control, programming and operation. The whole system is mapped through this environment and enables access to a programming environment (including compilation) for the controller. Individual digital or analog inputs or outputs can also be read or written directly, in order to verify their functionality, for example.

For further information, please refer to <http://infosys.beckhoff.com>:

- **EtherCAT System Manual:**  
Fieldbus Components → EtherCAT Terminals → EtherCAT System Documentation → Setup in the TwinCAT System Manager
- In particular, for TwinCAT – driver installation:  
**Fieldbus components** → Fieldbus Cards and Switches → FC900x – PCI Cards for Ethernet → Installation

Devices contain the relevant terminals for the actual configuration. All configuration data can be entered directly via editor functions (offline) or via the `scan` function (online):

- **"offline"**: The configuration can be customized by adding and positioning individual components. These can be selected from a directory and configured.
- **"online"**: The existing hardware configuration is read
  - See also <http://infosys.beckhoff.com>:  
**Fieldbus components** → Fieldbus Cards and Switches → FC900x – PCI Cards for Ethernet → Installation → Searching for devices

The following relationship is envisaged between the user PC and individual control elements:

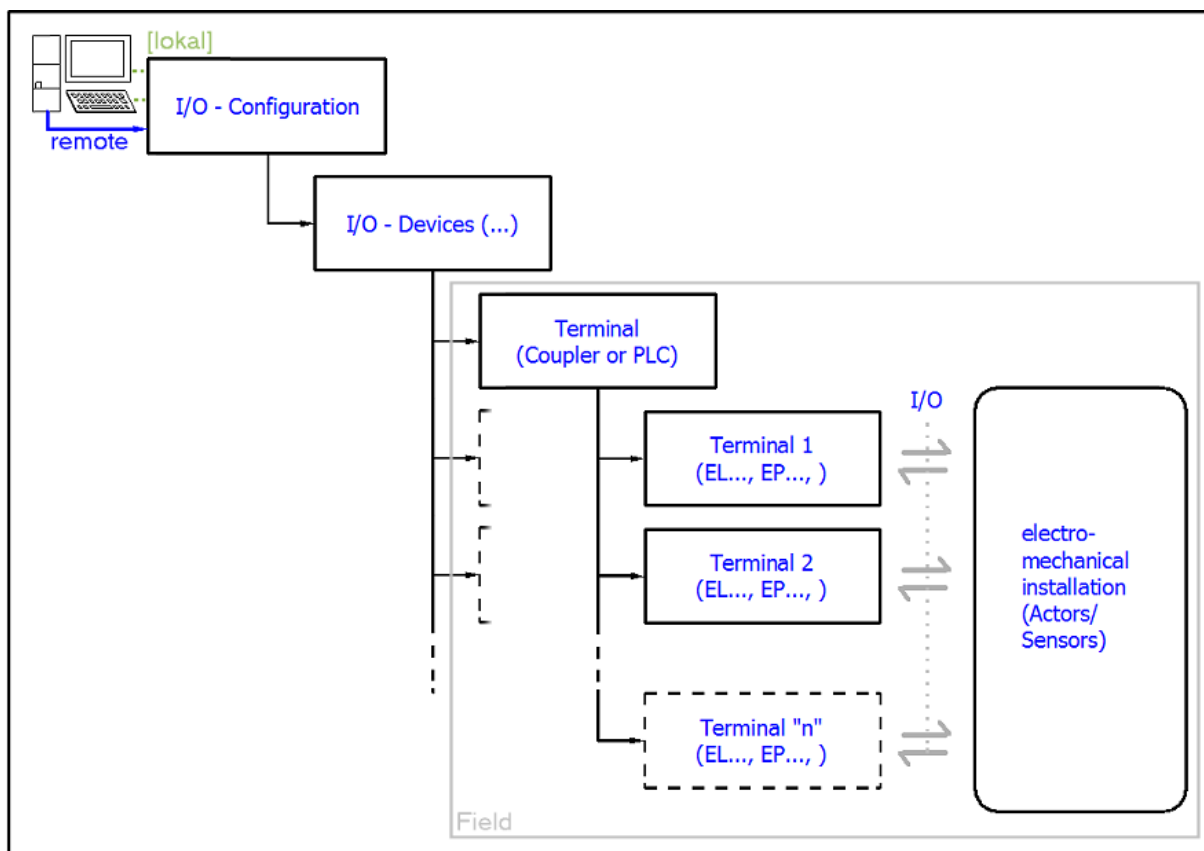


Fig. 33: Relationship between user side (commissioning) and installation

The user inserting of certain components (I/O device, terminal, box, ...) is the same in TwinCAT 2 and TwinCAT 3. In the following descriptions, only the "online" process is used.



## 6.1.1 TwinCAT 3

### Startup

TwinCAT 3 makes the development environment areas available all together, with Microsoft Visual Studio: after startup, the project folder explorer appears on the left in the general window area (see “TwinCAT System Manager” of TwinCAT 2) for communication with the electromechanical components.

After successful installation of the TwinCAT system on the PC to be used for development, TwinCAT 3 (shell) displays the following user interface after startup:

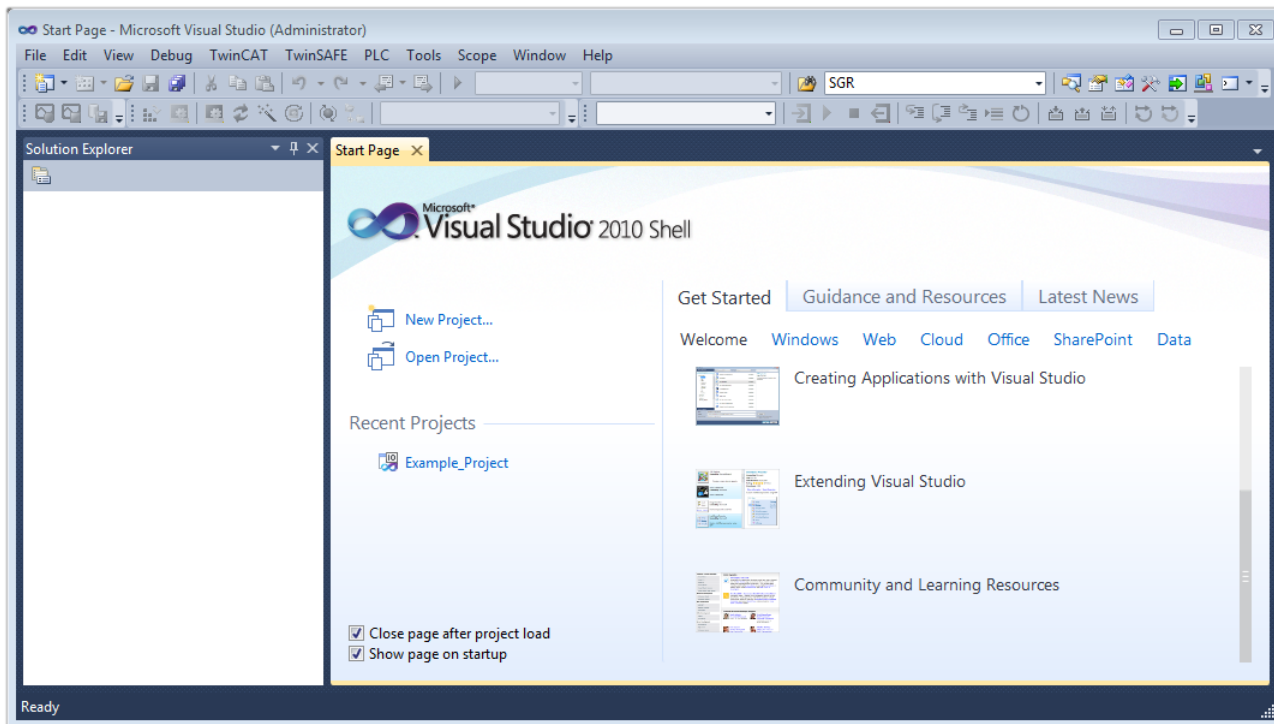



Fig. 35: Initial TwinCAT 3 user interface

First create a new project via  **New TwinCAT Project...** (or under “File”→“New”→“Project...”). In the following dialog, make the corresponding entries as required (as shown in the diagram):

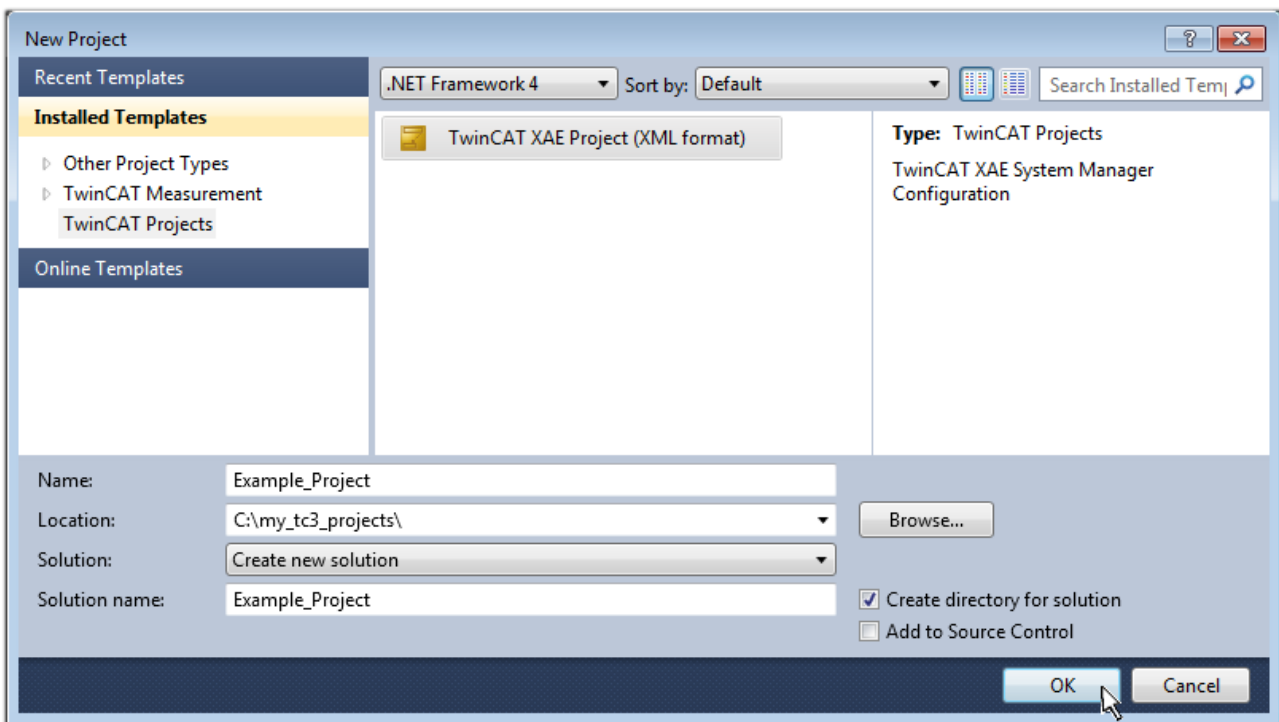


Fig. 36: Create new TwinCAT 3 project

The new project is then available in the project folder explorer:

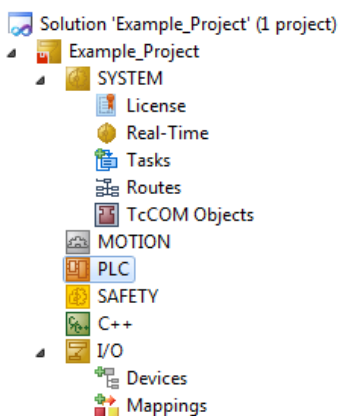
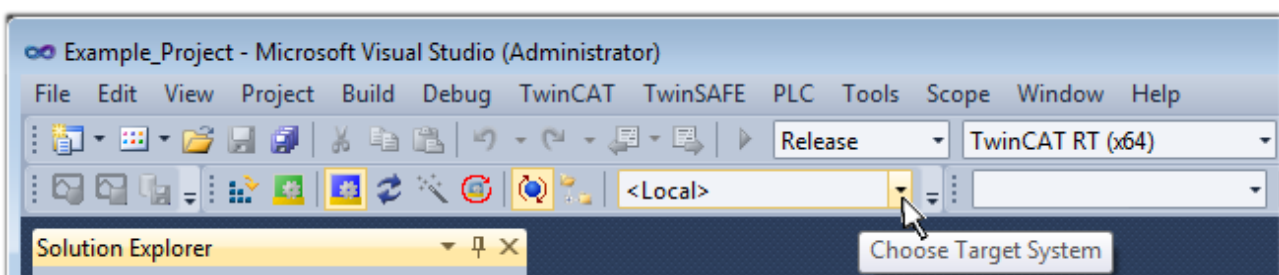


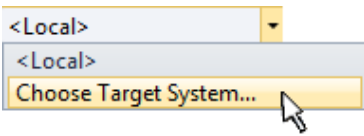
Fig. 37: New TwinCAT 3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC (locally), TwinCAT can be used in local mode and the process can be continued with the next step, “Insert Device |▶ 61|”.

If the intention is to address the TwinCAT runtime environment installed on a PLC remotely from another system used as a development environment, the target system must be made known first. Via the symbol in the menu bar:



expand the pull-down menu:



and open the following window:

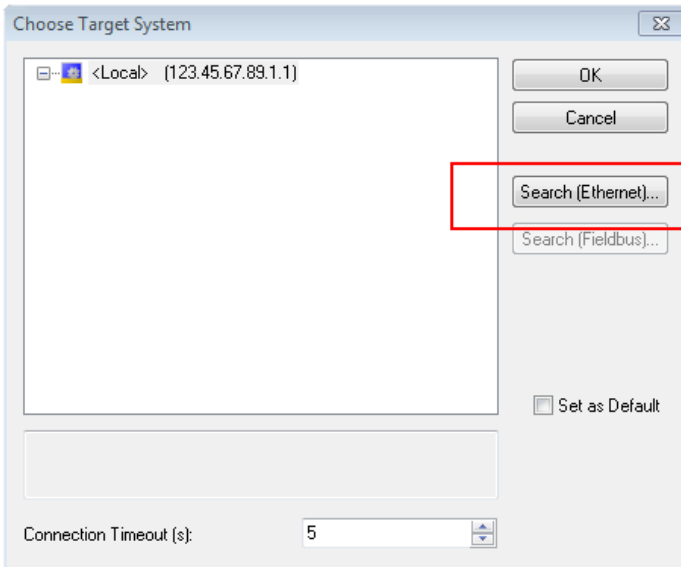


Fig. 38: Selection dialog: Choose the target system

Use “Search (Ethernet)...” to enter the target system. Thus another dialog opens to either:

- enter the known computer name after “Enter Host Name / IP:” (as shown in red)
- perform a “Broadcast Search” (if the exact computer name is not known)
- enter the known computer – IP or AmsNetID

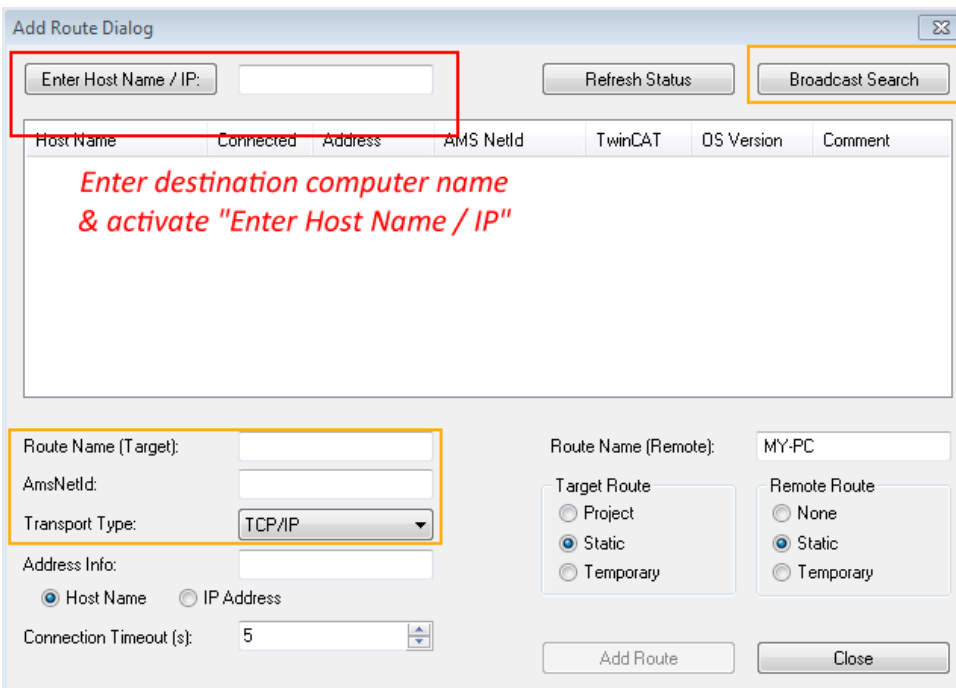
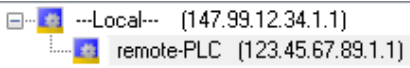




Fig. 39: specify the PLC for access by the TwinCAT System Manager: selection of the target system

Once the target system has been entered, it is available for selection as follows (the correct password may have to be entered beforehand):



After confirmation with “OK” the target system can be accessed via the Visual Studio shell.

### Adding devices

In the project folder explorer on the left of the Visual Studio shell user interface, select “Devices” within the element “I/O”, then right-click to open a context menu and select “Scan” or start the action via  in the menu bar. The TwinCAT System Manager may first have to be set to “Config mode” via  or via the menu “TwinCAT” → “Restart TwinCAT (Config Mode)”.

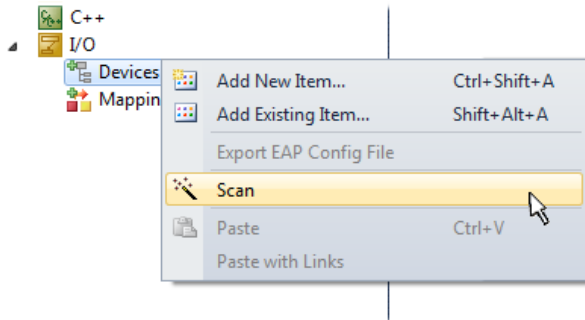


Fig. 40: Select “Scan”

Confirm the warning message, which follows, and select the “EtherCAT” devices in the dialog:

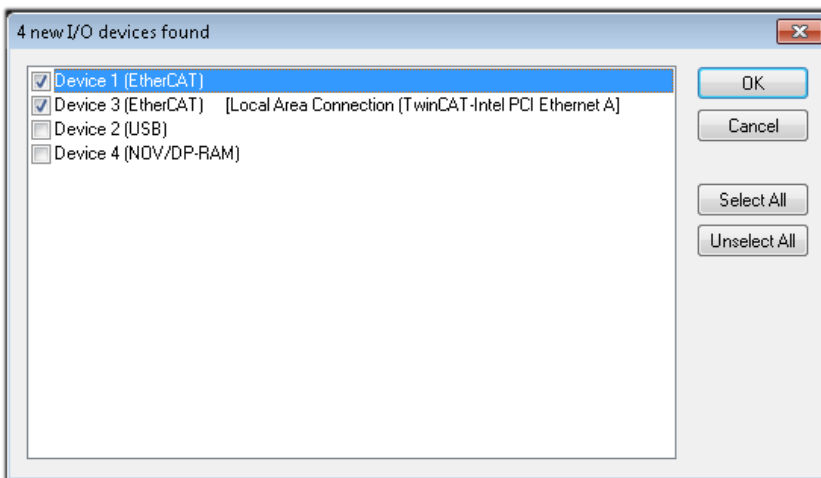


Fig. 41: Automatic detection of I/O devices: selection of the devices to be integrated

Confirm the message “Find new boxes”, in order to determine the terminals connected to the devices. “Free Run” enables manipulation of input and output values in “Config Mode” and should also be acknowledged.

Based on the [example configuration](#) [▶ 57] described at the beginning of this section, the result is as follows:

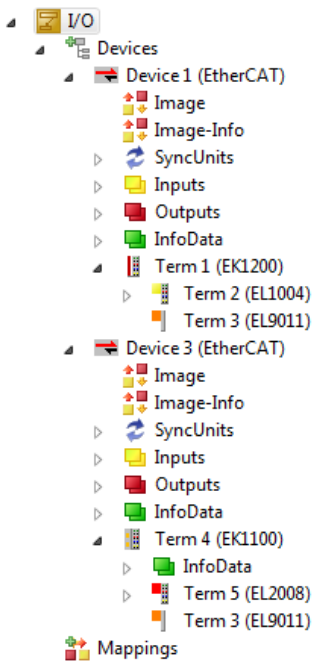


Fig. 42: Mapping of the configuration in VS shell of the TwinCAT 3 environment

The whole process consists of two stages, which can also be performed separately (first determine the devices, then determine the connected elements such as boxes, terminals, etc.). A scan (search function) can also be initiated by selecting “Device ...” from the context menu, which then only reads the elements below which are present in the configuration:

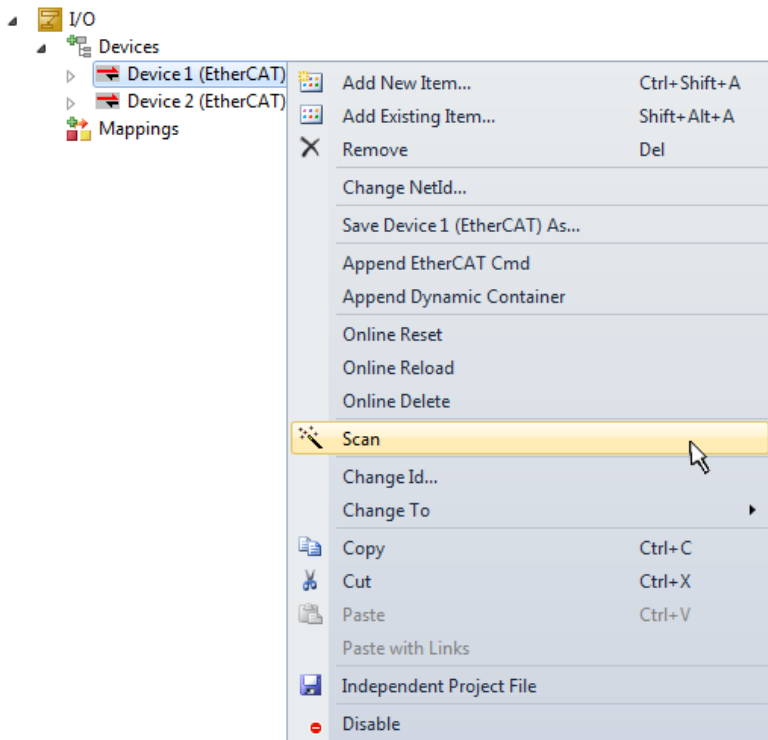


Fig. 43: Reading of individual terminals connected to a device

This functionality is useful if the actual configuration is modified at short notice.

**Programming the PLC**

TwinCAT PLC Control is the development environment for generating the controller in different program environments: TwinCAT PLC Control supports all languages described in IEC 61131-3. There are two text-based languages and three graphical languages.



- **Text-based languages**
  - Instruction List (IL)
  - Structured Text (ST)
- **Graphical languages**
  - Function Block Diagram (FBD)
  - Ladder Diagram (LD)
  - The Continuous Function Chart Editor (CFC)
  - Sequential Function Chart (SFC)

The following section refers solely to Structured Text (ST).

In order to create a programming environment, a PLC subproject is added to the example project via the context menu of the “PLC” in the project folder explorer by selecting “Add New Item...”:

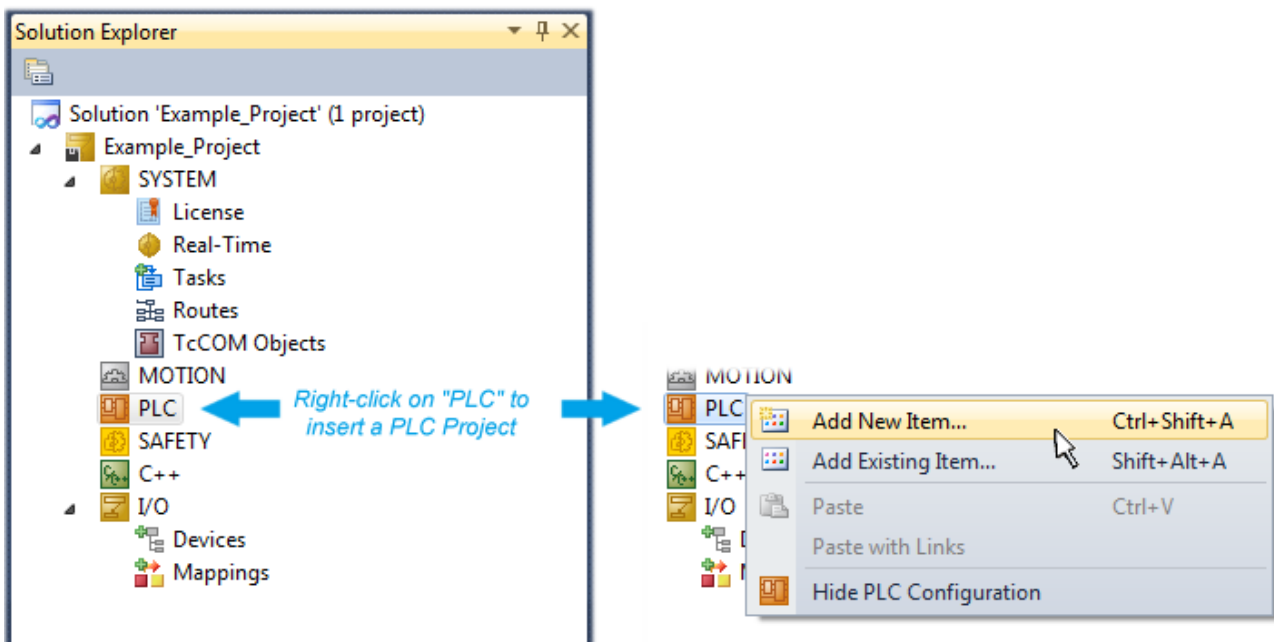


Fig. 44: Adding the programming environment in “PLC”

In the dialog that opens, select “Standard PLC project” and enter “PLC\_example” as project name, for example, and select a corresponding directory:

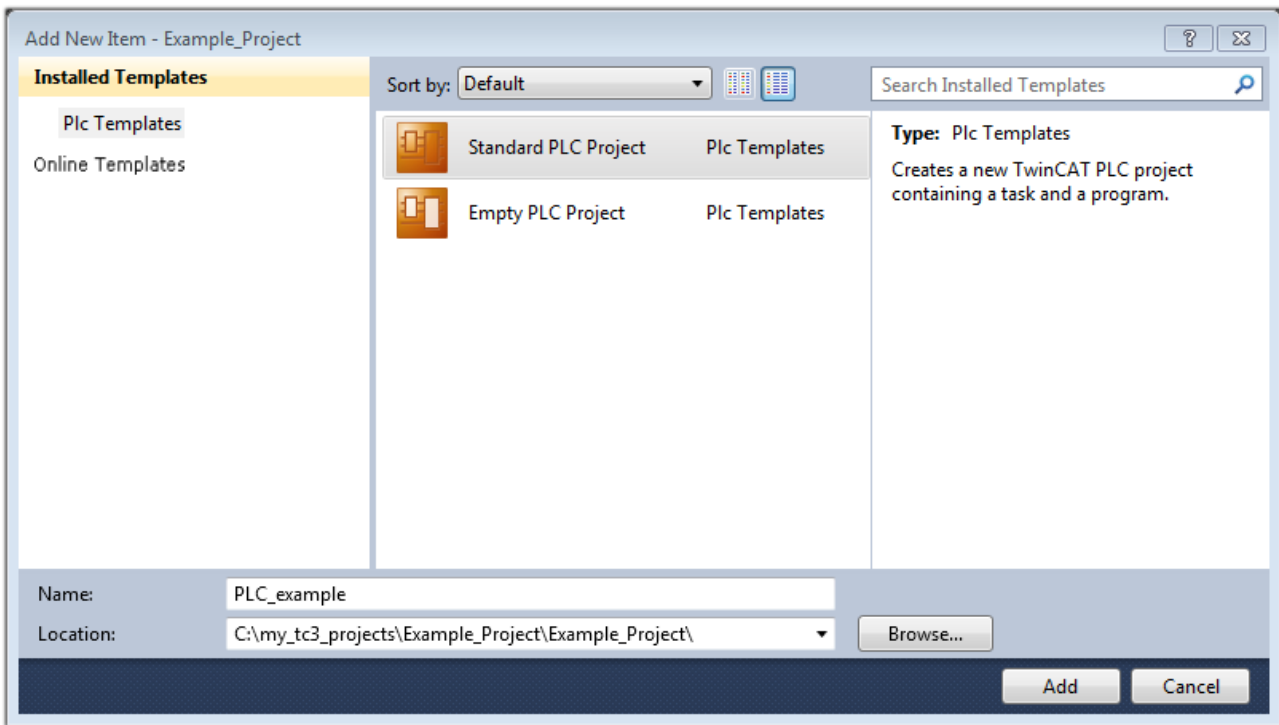


Fig. 45: Specifying the name and directory for the PLC programming environment

The “Main” program, which already exists due to selecting “Standard PLC project”, can be opened by double-clicking on “PLC\_example\_project” in “POUs”. The following user interface is shown for an initial project:

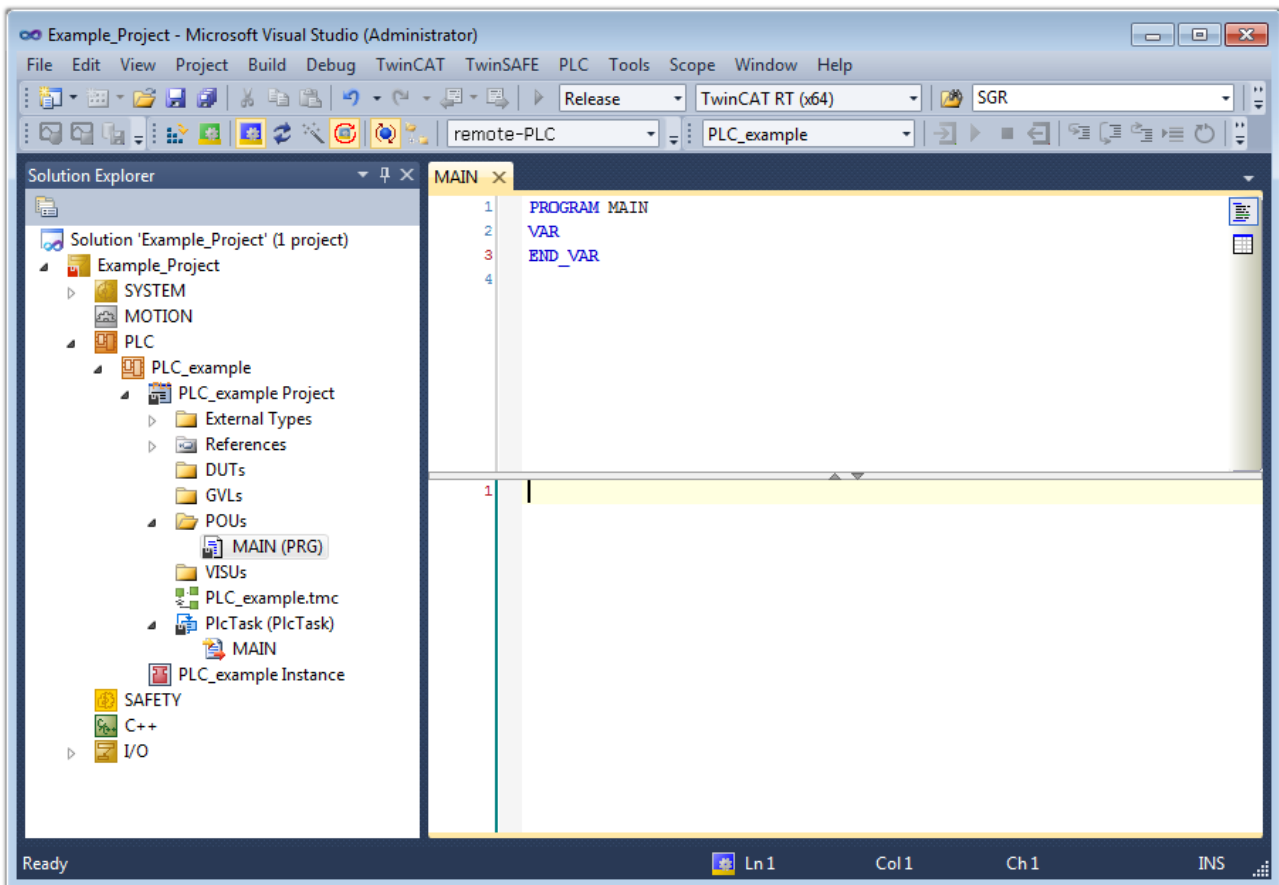


Fig. 46: Initial “Main” program for the standard PLC project

Now example variables and an example program have been created for the next stage of the process:

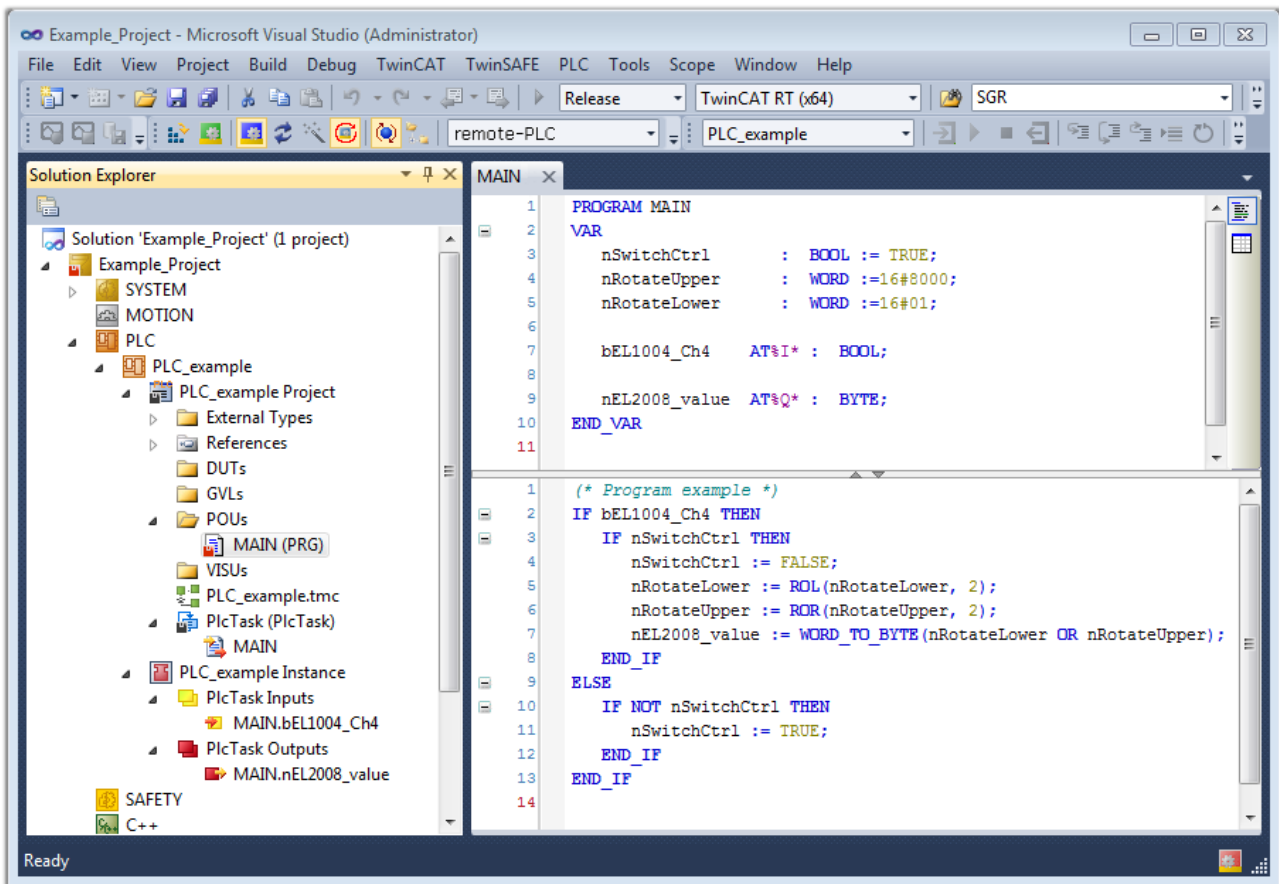


Fig. 47: Example program with variables after a compile process (without variable integration)

The control program is now created as a project folder, followed by the compile process:

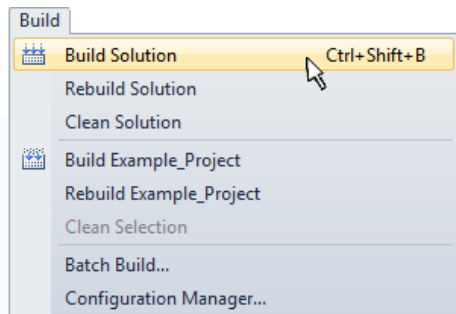
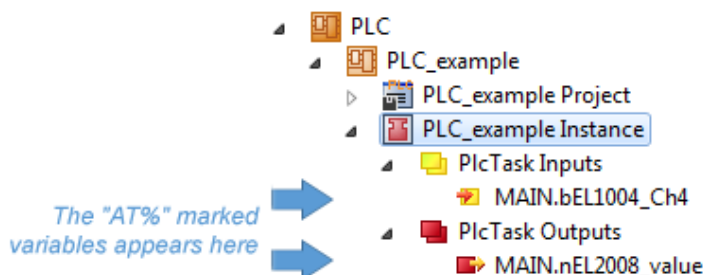


Fig. 48: Start program compilation

The following variables, identified in the ST/PLC program with “AT%”, are then available under “Assignments” in the project folder explorer:



**Assigning variables**

Via the menu of an instance – variables in the “PLC” context, use the “Modify Link...” option to open a window to select a suitable process object (PDO) for linking:

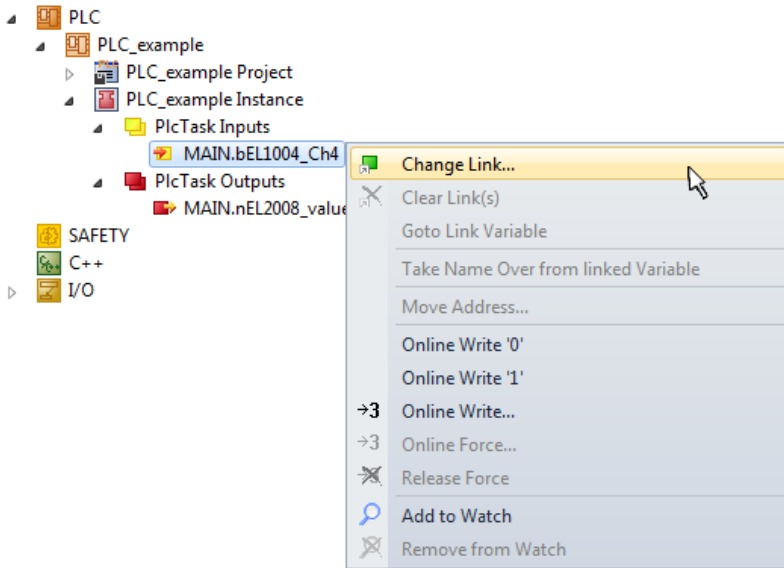


Fig. 49: Creating the links between PLC variables and process objects

In the window that opens, the process object for the “bEL1004\_Ch4” BOOL-type variable can be selected from the PLC configuration tree:

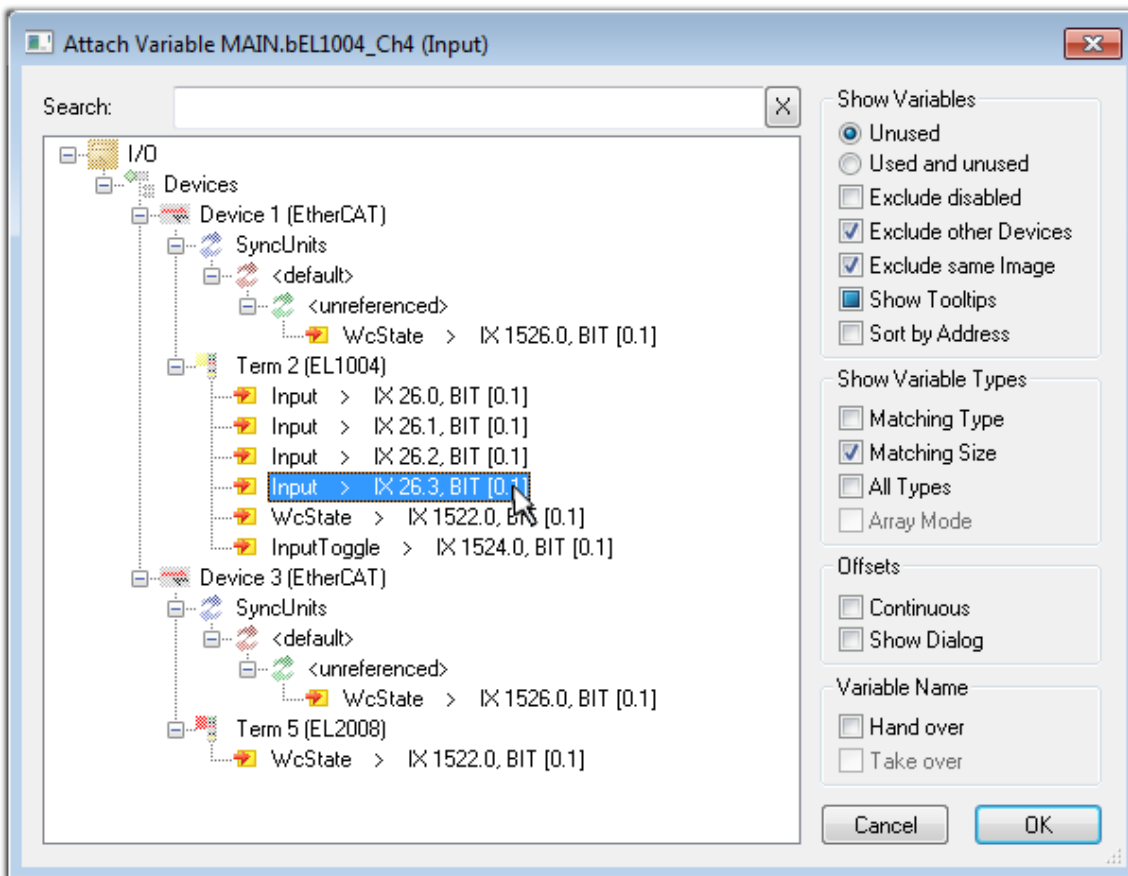


Fig. 50: Selecting BOOL-type PDO

According to the default setting, only certain PDO objects are now available for selection. In this example, the input of channel 4 of the EL1004 terminal is selected for linking. In contrast, the checkbox “All types” must be ticked to create the link for the output variables, in order to allocate a set of eight separate output bits to a byte variable in this case. The following diagram shows the whole process:

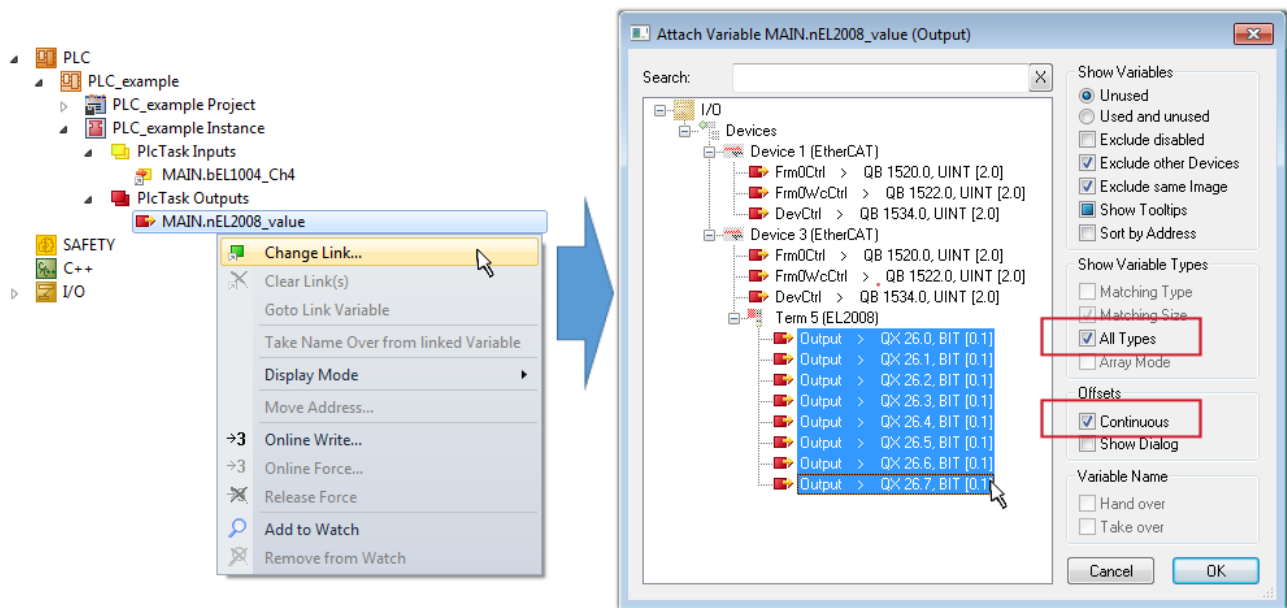



Fig. 51: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the “nEL2008\_value” variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus possible to subsequently address all eight outputs of the terminal in the program with a byte corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol (  ) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting “Goto Link Variable” from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:

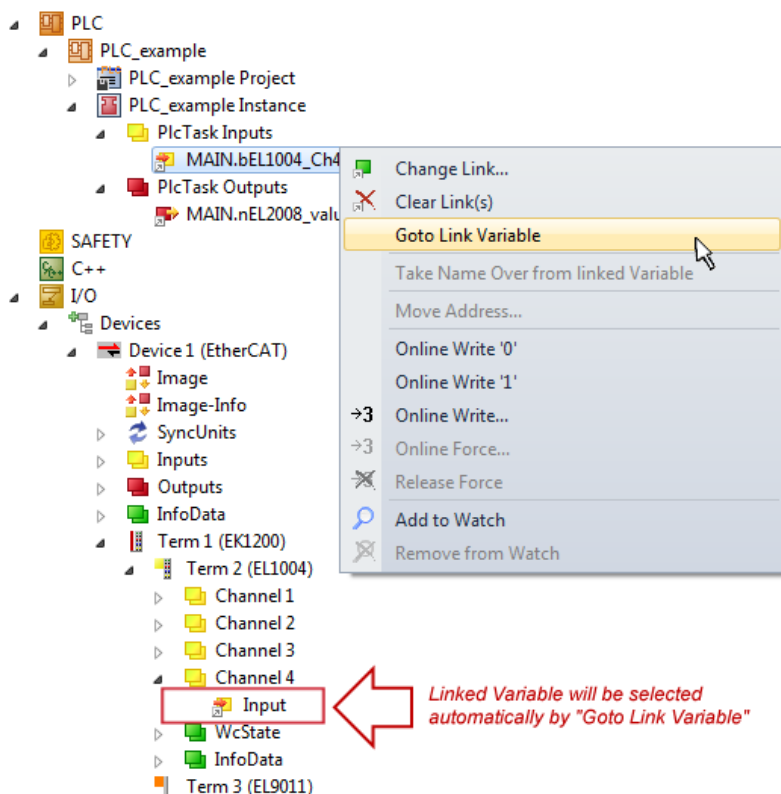


Fig. 52: Application of a “Goto Link Variable”, using “MAIN.bEL1004\_Ch4” as an example

The process of creating links can also be performed in the opposite direction, i.e. starting with individual PDOs to a variable. However, in this example, it would not be possible to select all output bits for the EL2008, since the terminal only makes individual digital outputs available. If a terminal has a byte, word,

integer or similar PDO, it is also possible to allocate this to a set of bit-standardized variables. Here, too, a “Goto Link Variable” can be executed in the other direction, so that the respective PLC instance can then be selected.

### ● Note on type of variable assignment



The following type of variable assignment can only be used from TwinCAT version V3.1.4024.4 onwards and is only available for terminals with a microcontroller.

In TwinCAT, a structure can be created from the mapped process data of a terminal. An instance of this structure can then be created in the PLC, so it is possible to access the process data directly from the PLC without having to declare own variables.

The procedure for the EL3001 1-channel analog input terminal -10...+10 V is shown as an example.

1. First, the required process data must be selected in the “Process data” tab in TwinCAT.
2. After that, the PLC data type must be generated in the “PLC” tab via the check box.
3. The data type in the “Data Type” field can then be copied using the “Copy” button.

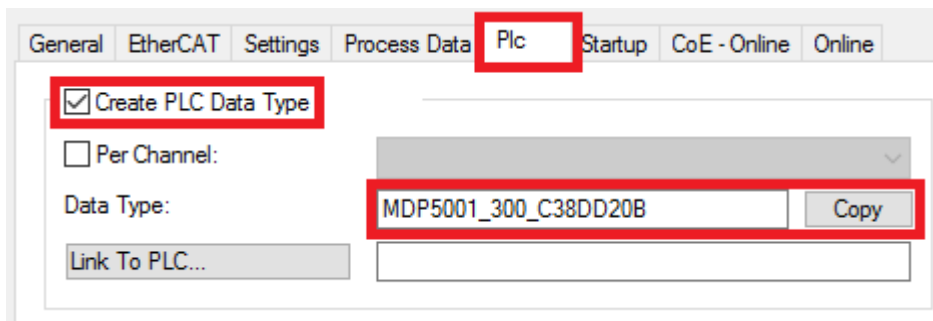


Fig. 53: Creating a PLC data type

4. An instance of the data structure of the copied data type must then be created in the PLC.

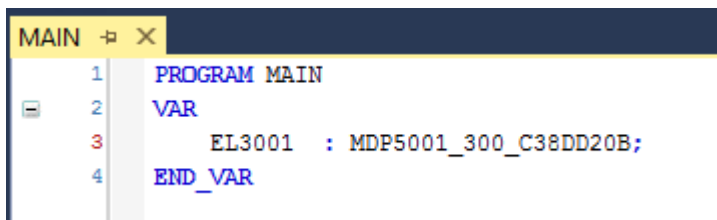


Fig. 54: Instance\_of\_struct

5. Then the project folder must be created. This can be done either via the key combination “CTRL + Shift + B” or via the “Build” tab in TwinCAT.
6. The structure in the “PLC” tab of the terminal must then be linked to the created instance.

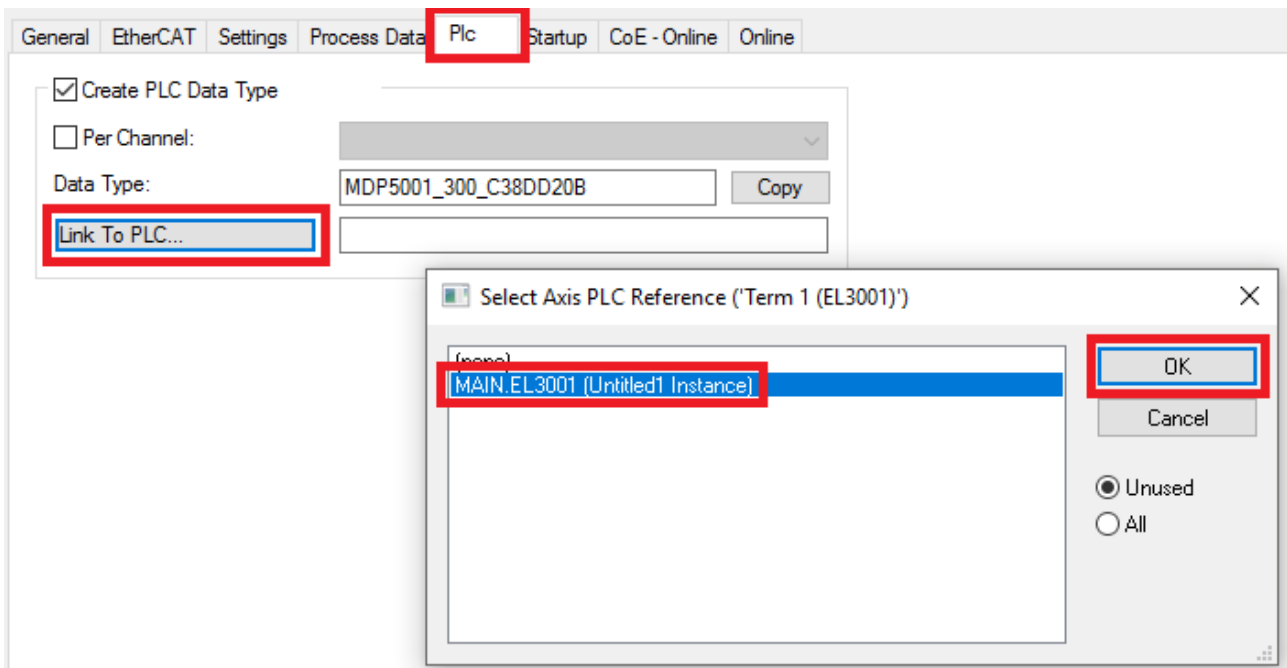


Fig. 55: Linking the structure

7. In the PLC, the process data can then be read or written via the structure in the program code.

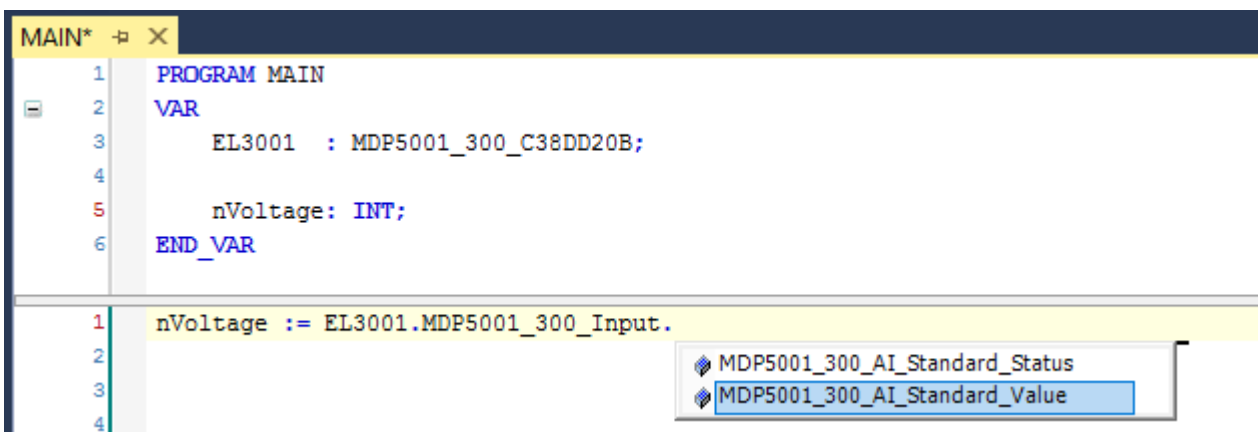

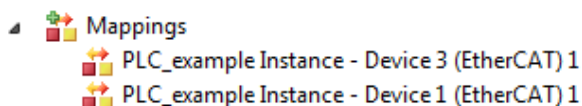


Fig. 56: Reading a variable from the structure of the process data

**Activation of the configuration**

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs

and outputs of the terminals. The configuration can now be activated with  or via the menu under “TwinCAT” in order to transfer the settings of the development environment to the runtime system. Confirm the messages “Old configurations will be overwritten!” and “Restart TwinCAT system in Run mode” with “OK”. The corresponding assignments can be seen in the project folder explorer:





A few seconds later, the corresponding status of the Run mode is displayed in the form of a rotating symbol



at the bottom right of the VS shell development environment. The PLC system can then be started as described below.

## Starting the controller

Select the menu option “PLC” → “Login” or click on  to link the PLC with the real-time system and load the control program for execution. This results in the message “No program on the controller! Should the new program be loaded?”, which should be acknowledged with “Yes”. The runtime environment is ready for

the program to be started by clicking on symbol , the “F5” key or via “PLC” in the menu, by selecting “Start”. The started programming environment shows the runtime values of individual variables:

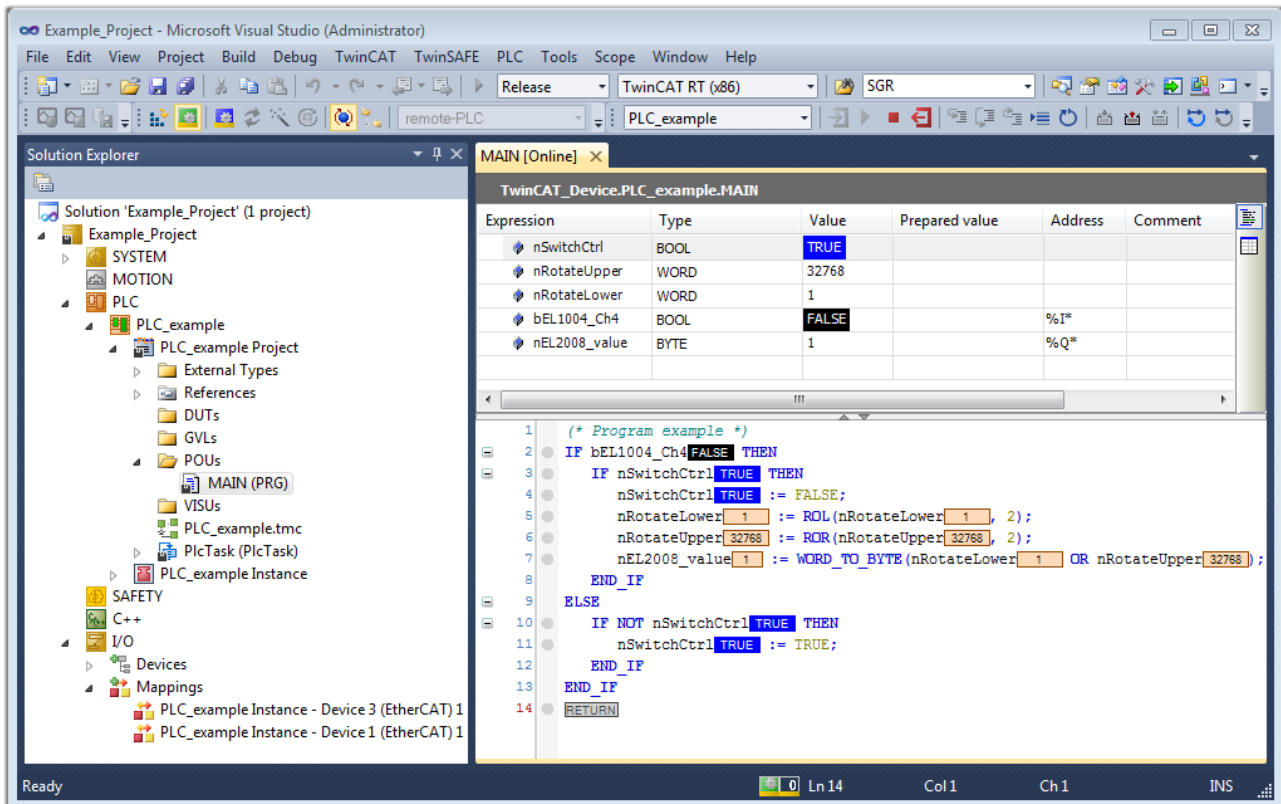




Fig. 57: TwinCAT 3 development environment (VS shell): logged-in, after program startup

The two operator control elements for stopping  and logout  result in the required action (also, “Shift + F5” can be used for stop, or both actions can be selected via the PLC menu).

## 6.2 TwinCAT Development Environment

The TwinCAT (The Windows Control and Automation Technology) automation software is divided into:

- TwinCAT 2: System Manager (configuration) and PLC Control (programming)
- TwinCAT 3: further development of TwinCAT 2 (programming and configuration via a common development environment)

### Details:

- **TwinCAT 2:**
  - Provides variable-oriented linkage of I/O devices and tasks
  - Provides variable-oriented linkage of tasks to other tasks
  - Supports units at bit level
  - Supports both synchronous and asynchronous relationships
  - Consistent exchange of data areas and process images



- Data connection to NT programs using open Microsoft standards (OLE, OCX, ActiveX, DCOM+, etc.).
- Embedding of IEC 61131-3 software PLC, software NC and software CNC in Windows NT/2000/XP/Vista, Windows 7, NT/XP Embedded, CE
- Connection to all common fieldbuses
- More...

#### Additionally offers:

- **TwinCAT 3 (eXtended Automation):**
  - Visual Studio® integration
  - Choice of programming language
  - Support for the object-oriented extension of IEC 61131-3
  - Use of C/C++ as a programming language for real-time applications
  - Link to Matlab®/Simulink®
  - Open interfaces for extensibility
  - Flexible runtime environment
  - Active support for multi-core and 64-bit operating systems
  - Automatic code generation and project creation with the TwinCAT Automation Interface
  - More...

The following chapters explain the commissioning of the TwinCAT development environment on a PC system of the control system and the most important functions of individual control elements.

Please see further information on TwinCAT 2 and TwinCAT 3 at <http://infosys.beckhoff.de/>.

## 6.2.1 Installation of the TwinCAT real-time driver

In order to assign real-time capability to a standard Ethernet port of an IPC controller, the Beckhoff real-time driver has to be installed on this port under Windows.

This can be done in several ways. One option is described here.

In TwinCAT 3, the TwinCAT overview of the local network interfaces can be called up via the menu "TwinCAT" → Show real-time Ethernet compatible devices:

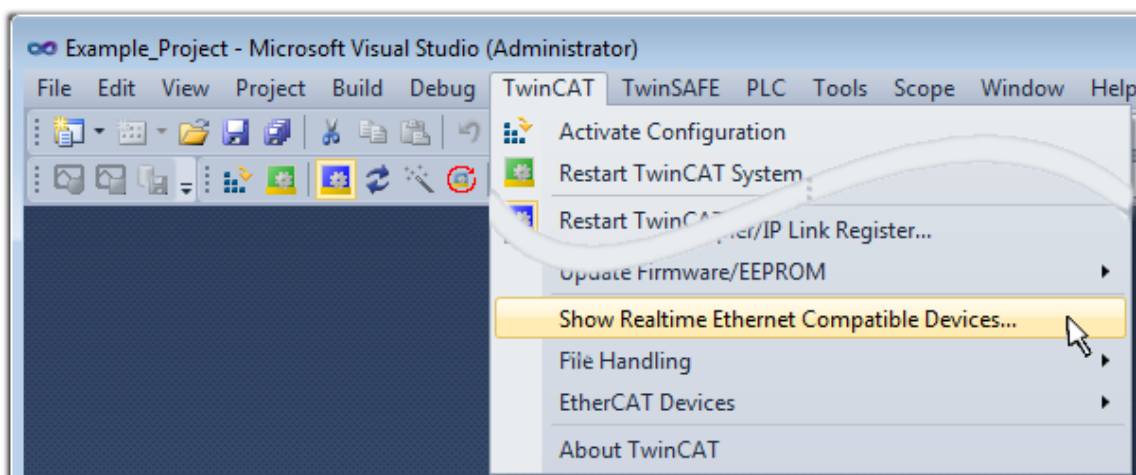


Fig. 58: Call in VS Shell (TwinCAT 3)

The following dialog appears:

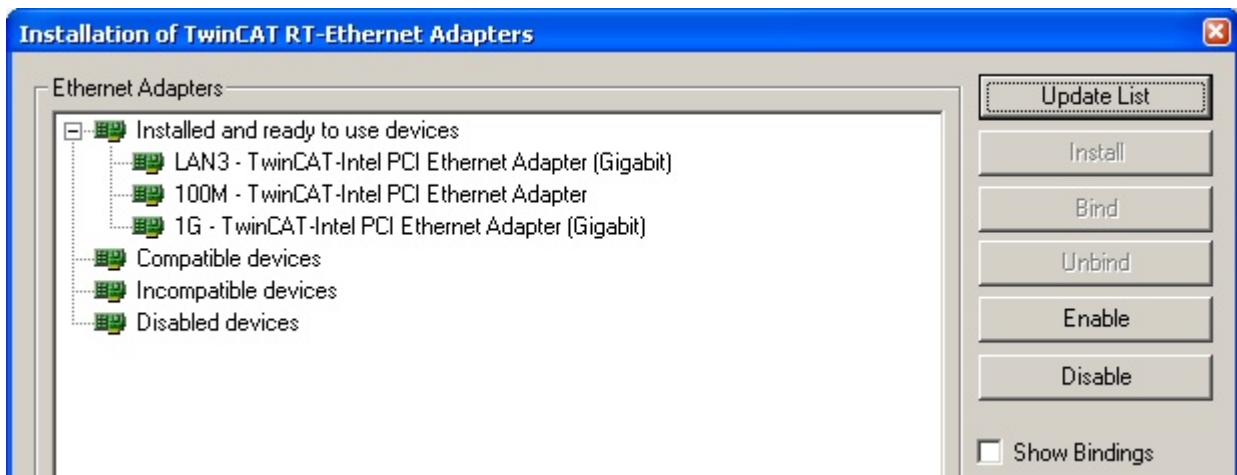


Fig. 59: Overview of network interfaces

Interfaces listed under “Compatible devices” can be assigned a driver via the “Install” button. A driver should only be installed on compatible devices.

A Windows warning regarding the unsigned driver can be ignored.

**Alternatively**, as described in the chapter Offline configuration creation, section “Creating the EtherCAT device” [▶ 80], you can first enter an EtherCAT device and then view the compatible Ethernet ports via its properties (“Adapter” tab, “Compatible Devices...” button):

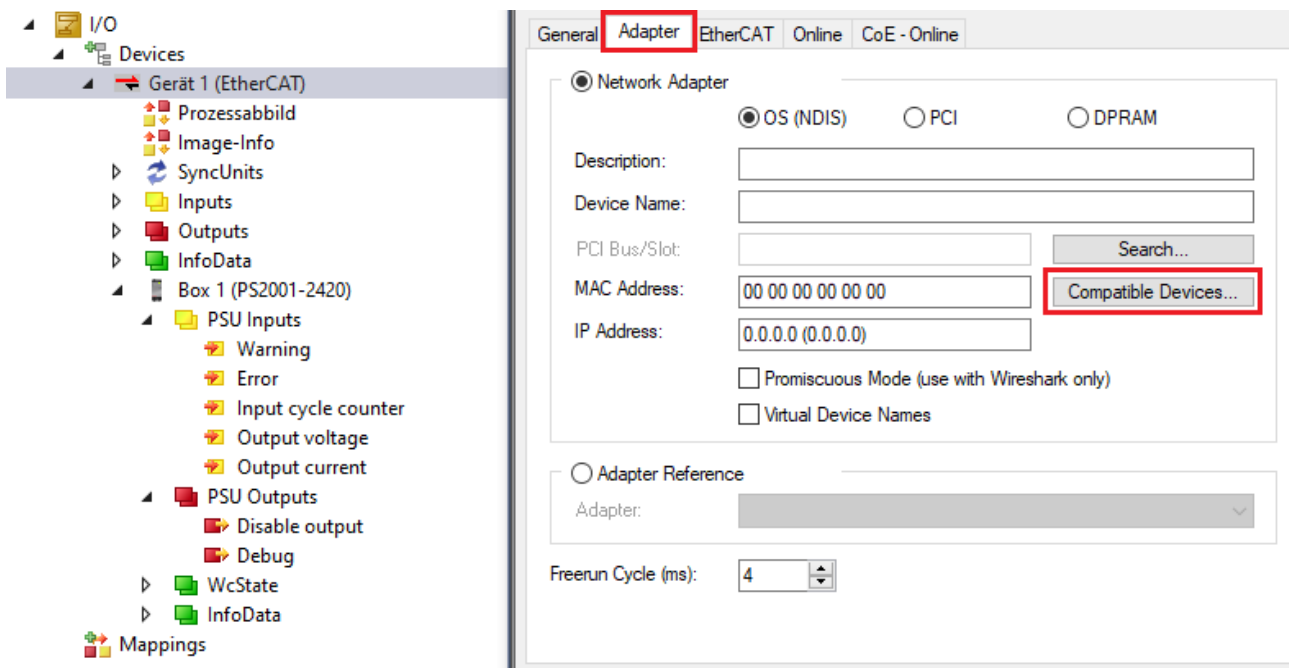
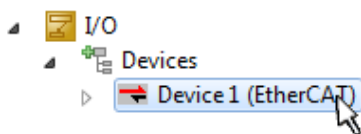


Fig. 60: Properties of EtherCAT device (TwinCAT 3): Click on “Compatible Devices...” from “Adapter”

TwinCAT 3: the properties of the EtherCAT device can be opened by double click on “Device .. (EtherCAT)” within the Solution Explorer under “I/O”:



After installation, the driver appears enabled in the Windows overview of the individual network interface (Windows Start → Control Panel → Network)

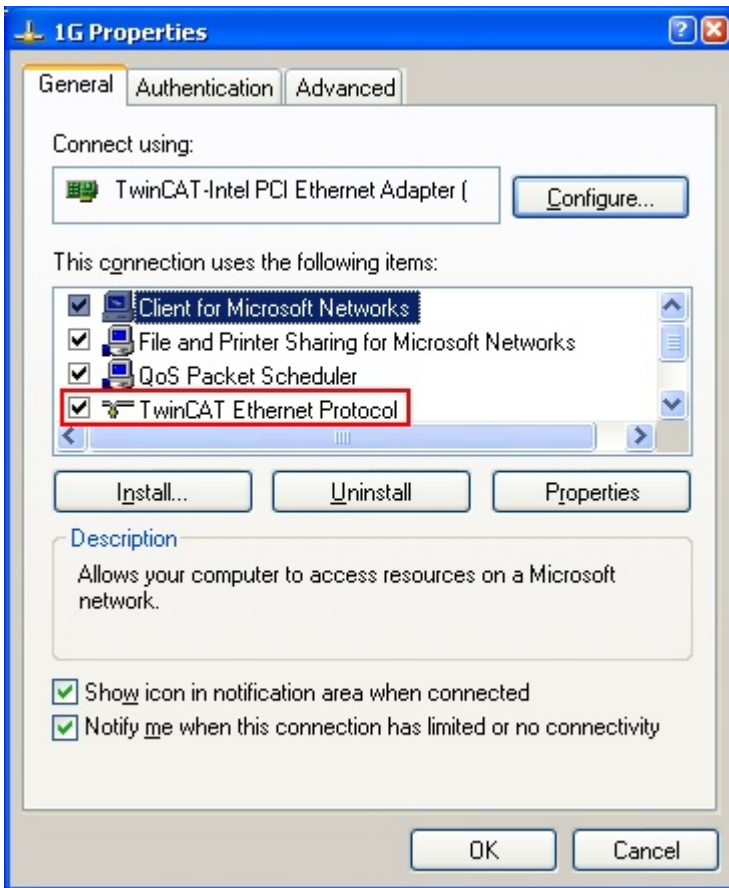


Fig. 61: Windows properties of the network interface

A correct setting of the driver could look like this:

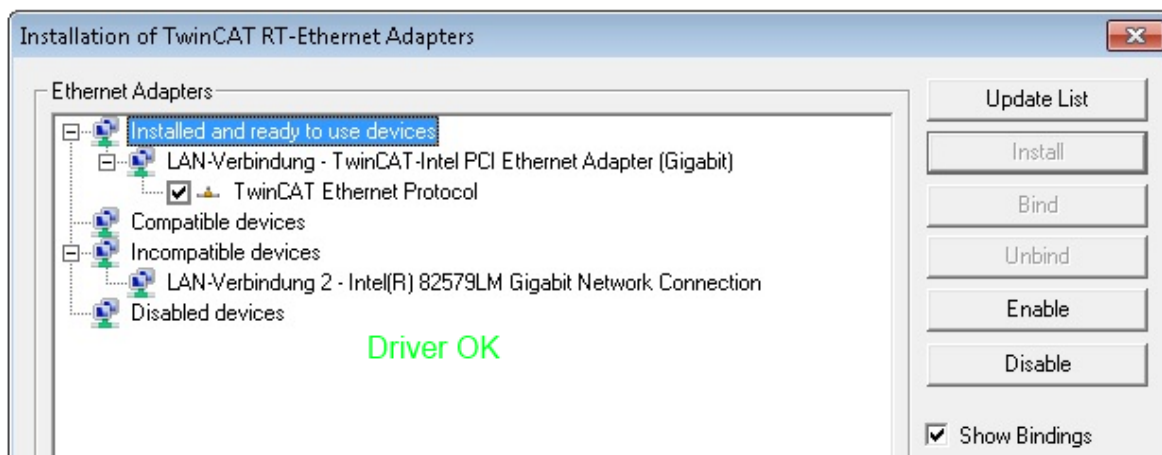


Fig. 62: Example of the correct driver setting for the Ethernet port

Other possible settings are to be avoided:

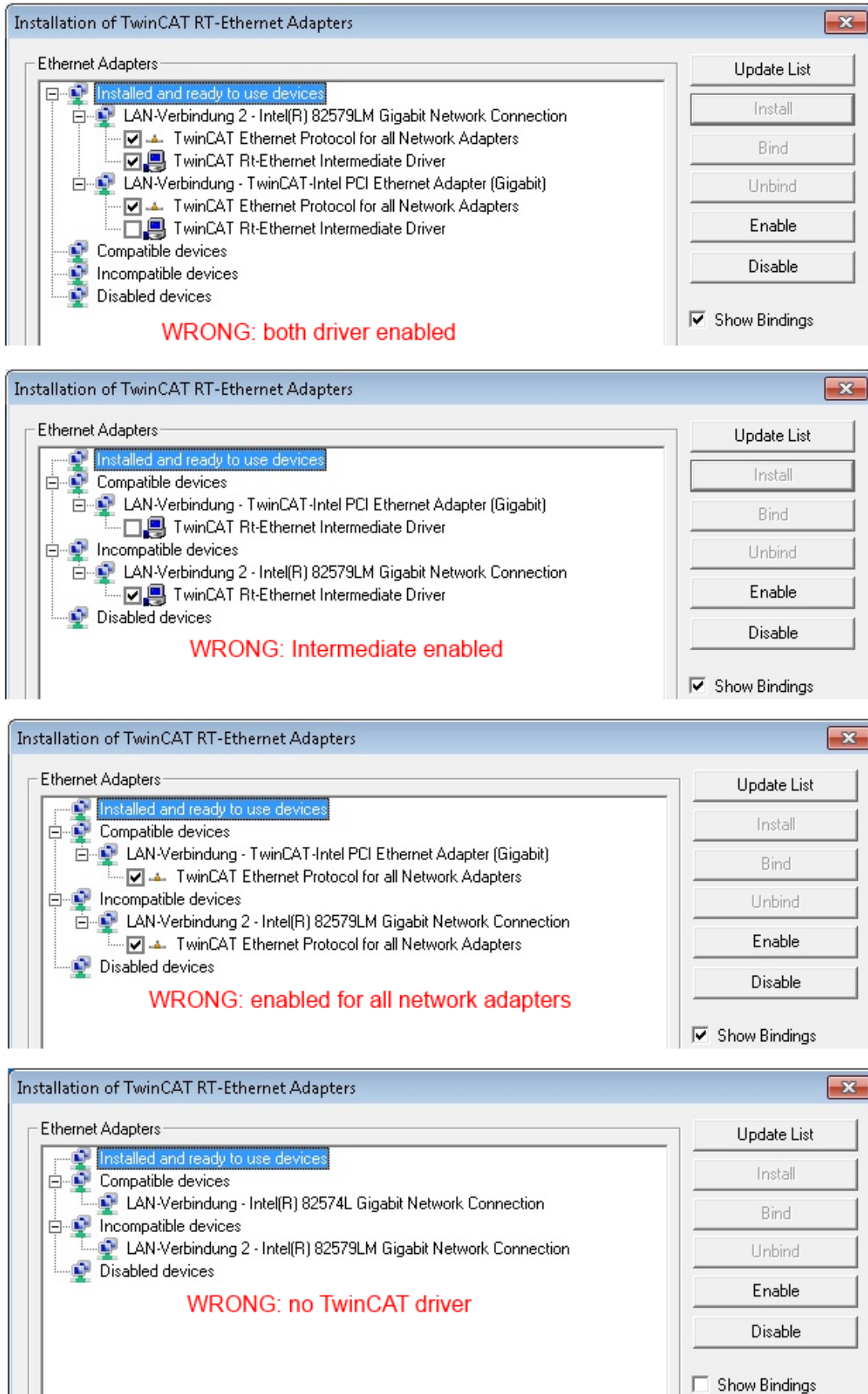


Fig. 63: Wrong driver settings for the Ethernet port

IP address of the port used

**i** IP address/DHCP

In most cases an Ethernet port that is configured as an EtherCAT device will not transport general IP packets. For this reason and in cases where an EL6601 or similar devices are used it is useful to specify a fixed IP address for this port via the “Internet Protocol TCP/IP” driver setting and to disable DHCP. In this way the delay associated with the DHCP client for the Ethernet port assigning itself a default IP address in the absence of a DHCP server is avoided. A suitable address space is 192.168.x.x, for example.

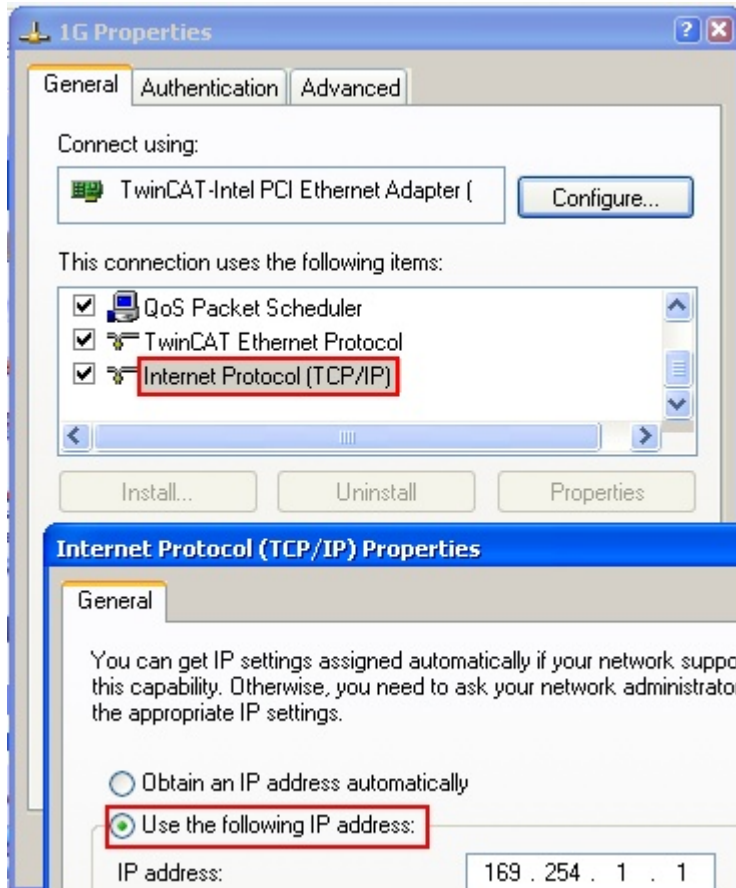


Fig. 64: TCP/IP setting for the Ethernet port

## 6.2.2 Notes regarding ESI device description

### Installation of the latest ESI device description

The TwinCAT EtherCAT master/System Manager needs the device description files for the devices to be used in order to create the configuration in online or offline mode. The device descriptions are contained in the so-called ESI files (EtherCAT Slave Information) in XML format. These files can be requested from the respective vendor and are made available for download. An \*.xml file may contain several device descriptions.

The ESIs for Beckhoff EtherCAT devices are provided on the [Beckhoff website](#).

The ESI files must be stored in the TwinCAT installation directory.

Default settings:

- **TwinCAT 3:** C:\TwinCAT\3.1\Config\lo\EtherCAT

The files are read (once) when a new System Manager window is opened, if they have changed since the last time the System Manager window was opened.

A TwinCAT installation includes the set of Beckhoff ESI files that was current at the time when the TwinCAT build was created.

From TwinCAT 3 on the ESI directory can be updated from the System Manager if the programming PC is connected to the internet; at

**TwinCAT 3:** TwinCAT → EtherCAT Devices → “Update Device Descriptions (via ETG Website)...”

The [TwinCAT ESI Updater \[► 79\]](#) is available for this purpose.



#### ESI

The \*.xml files are associated with \*.xsd files, which describe the structure of the ESI XML files. To update the ESI device descriptions, both file types should therefore be updated.

### Device differentiation

EtherCAT devices/slaves are differentiated by four properties, from which the complete device designation is composed. For example, the device designation "EL2521-0025-1018" consists of:

- family key “EL”
- name "2521"
- type "0025"
- and revision “1018”

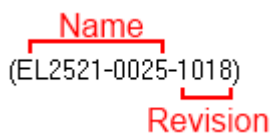


Fig. 65: Device designation: Structure

The order identifier consisting of name + version (here: EL2521-0010) describes the device function. The revision indicates the technical progress and is managed by Beckhoff. In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation. Each revision has its own ESI description. See further [notes \[► 158\]](#).

### Faulty ESI file

If there is a faulty ESI file that cannot be read by the System Manager, the System Manager displays an information window.



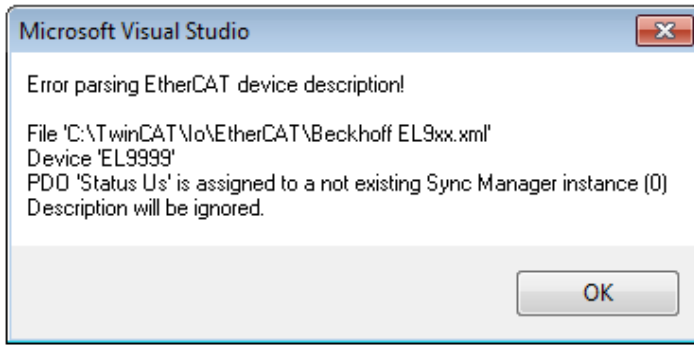


Fig. 66: Information window faulty ESI file (TwinCAT 3)

This can be caused by

- Structure of the \*.xml does not correspond to the associated \*.xsd file → check your schematics
- Contents cannot be compiled into a device description → contact the file manufacturer

**Online description**

If the EtherCAT configuration is created online by scanning real existing devices (see chapter Online creation) and there is no ESI description available for a slave found (identified by name and revision), the System Manager asks whether it should use the description available in the device. The System Manager always requires this information in order to set the cyclical and acyclical communication with the slave correctly.

A window appears in TwinCAT 3 that also offers the web update:

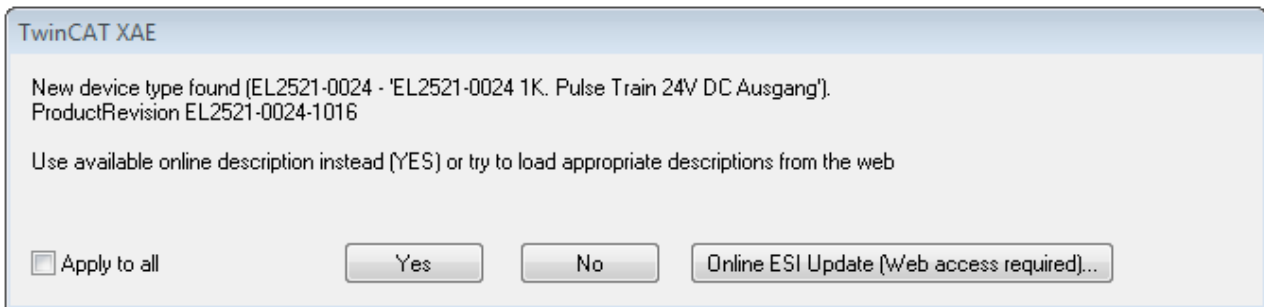


Fig. 67: OnlineDescription information window (TwinCAT 3)

If possible, the Yes is to be rejected and the required ESI is to be requested from the device vendor. After installation of the XML/XSD file the configuration process should be repeated.

**NOTICE**

**Changing the ‘usual’ configuration through a scan**

- ✓ in the case of a device detected by a scan but still unknown to TwinCAT, two cases must be distinguished. Taking the example here of the EL2521-0000 in the revision 1019:
  - a) no ESI is present for the EL2521-0000 device at all, either for the revision 1019 or for an older revision. The ESI must then be requested from the vendor (in this case Beckhoff).
  - b) for the EL2521-0000 device, an ESI is only available in an older revision, e.g. 1018 or 1017. In this case, you should first check internally whether the spare parts inventory allows the higher revision to be integrated into the configuration. A new/larger revision usually also brings new features. If these are not to be used, work can continue without reservations with the previous revision 1018 in the configuration. This is also stated by the Beckhoff compatibility rule.

Refer in particular to the chapter "[General notes on the use of Beckhoff EtherCAT I/O components](#)" and for manual configuration to the chapter "[Offline configuration creation \[► 80\]](#)".

If the online description is nevertheless used, the System Manager reads a copy of the device description from the EEPROM in the EtherCAT slave. In complex slaves the size of the EEPROM may not be sufficient for the complete ESI, in which case the ESI would be *incomplete* in the configurator. For this reason, the use of an offline ESI file is recommended as a matter of priority.

The System Manager creates a new file "OnlineDescription0000...xml" in its ESI directory for device descriptions recorded "online", which contains all ESI descriptions read online.

OnlineDescriptionCache00000002.xml

Fig. 68: Created by the System Manager OnlineDescription.xml

If a slave is then to be added manually to the configuration, slaves created "online" are identified by a preceding ">" symbol in the selection list (see figure *Identification of an ESI recorded online using the EL2521 as an example*).

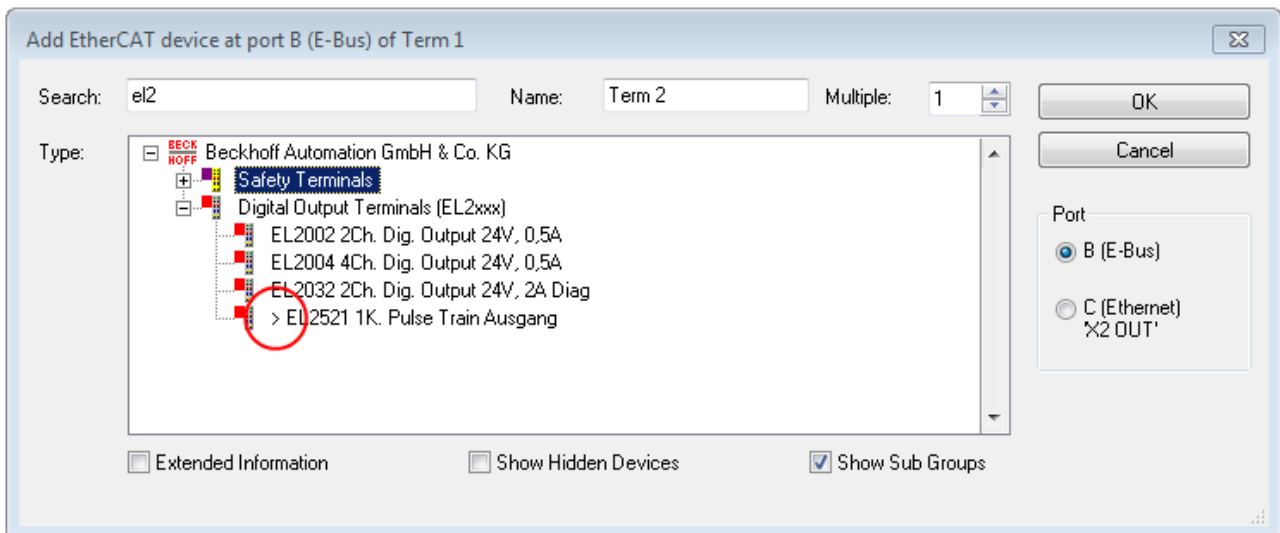


Fig. 69: Identification of an ESI recorded online using the EL2521 as an example

If such ESI files are used and the vendor's files become available later, the file OnlineDescription....xml should be deleted as follows:

- close all System Manager windows
- restart TwinCAT in Config mode
- delete "OnlineDescription0000...xml"
- open TwinCAT System Manager again

This file must then no longer be visible, update folder with <F5> if necessary.

### ● OnlineDescription under TwinCAT 3.x

**I** In addition to the "OnlineDescription0000...xml" file mentioned above, TwinCAT 3.x also creates a so-called EtherCAT cache with newly discovered devices, e.g. under Windows 7 at

*C:\User\[USERNAME]\AppData\Roaming\Beckhoff\TwinCAT3\Components\Base\EtherCATCache.xml*

(note the language settings of the operating system!)

This file must be deleted at the same time as the other file.



### 6.2.3 TwinCAT ESI Updater

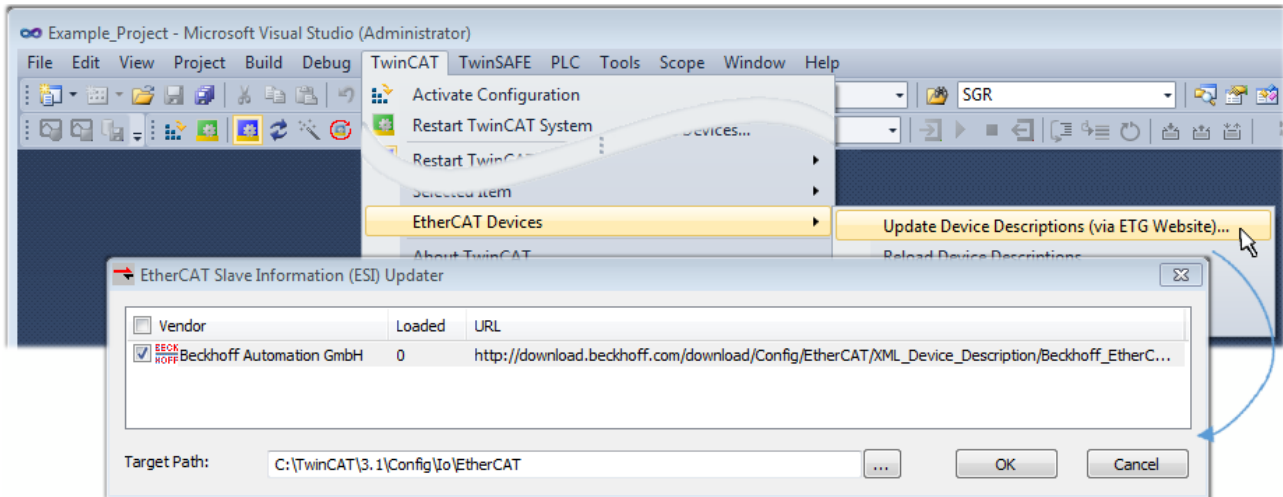


Fig. 70: Using the ESI Updater (TwinCAT 3)

The ESI Updater is a convenient option for automatic downloading of ESI data provided by EtherCAT manufacturers via the Internet into the TwinCAT directory (ESI = EtherCAT slave information). TwinCAT accesses the central ESI ULR directory list stored at ETG; the entries can then be viewed in the Updater dialog, although they cannot be changed there.

The call is made at:

"TwinCAT" → "EtherCAT Devices" → "Update Device Description (via ETG website)...".

### 6.2.4 Distinction between Online and Offline

The distinction between online and offline refers to the existence of the actual I/O environment (drives, terminals, EJ modules). If the configuration is to be prepared in advance of the system configuration as a programming system, e.g. on a laptop, this is only possible in "Offline configuration" mode. In this case all components have to be entered manually in the configuration, e.g. based on the electrical design.

If the designed control system is already connected to the EtherCAT system and all components are energized and the infrastructure is ready for operation, the TwinCAT configuration can simply be generated through "scanning" from the runtime system. This is referred to as online configuration.

In any case, during each startup the EtherCAT master checks whether the slaves it finds match the configuration. This test can be parameterized in the extended slave settings. See the note "[Installing the latest ESI-XML-Device-Description \[▶ 76\]](#)".

#### For the creation of the configuration

- the real EtherCAT hardware (devices, couplers, drives) must be present and installed.
- the devices/modules must be connected via EtherCAT cables or in the terminal segment in the same way as they are intended to be used later
- the devices/modules must be connected to the power supply and ready for communication
- TwinCAT must be in CONFIG mode on the target system.

#### The online scan process consists of:

- [Detecting the EtherCAT device \[▶ 85\]](#) (Ethernet port at the IPC)
- [Detecting the connected EtherCAT devices \[▶ 86\]](#). This step can be carried out independent of the preceding step.
- [Troubleshooting \[▶ 89\]](#)

The [scan with existing configuration \[▶ 90\]](#) can also be carried out for comparison.

## 6.2.5 OFFLINE configuration creation

### Creating the EtherCAT device

An EtherCAT device must first be created in an empty System Manager window.

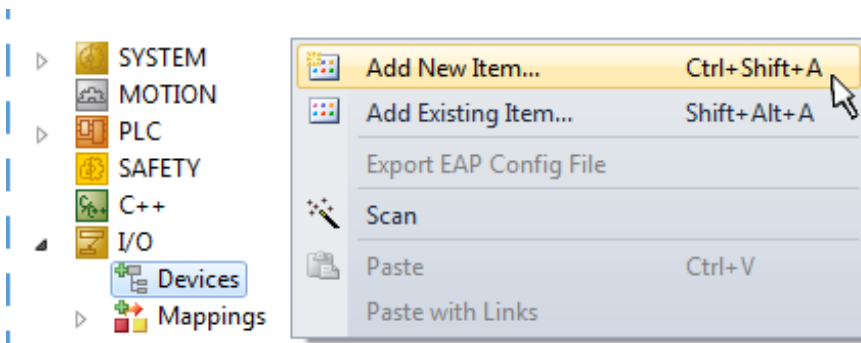


Fig. 71: Adding an EtherCAT device (TwinCAT 3)

The "EtherCAT" type must be selected for an EtherCAT I/O application with EtherCAT slaves. "EtherCAT Automation Protocol via EL6601" must be selected for the previous Publisher/Subscriber service in combination with an EL6601/EL6614 terminal.

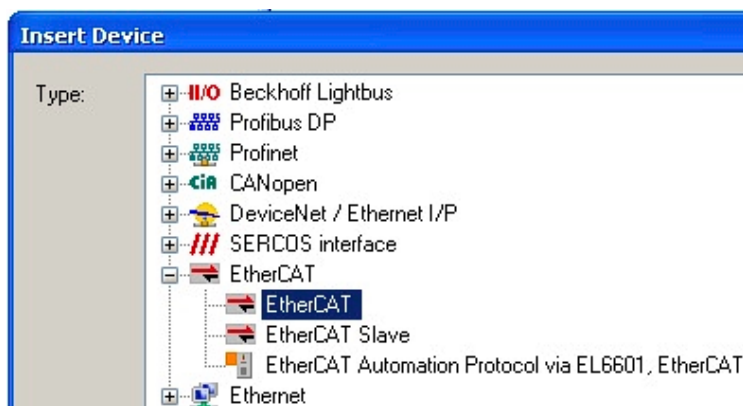


Fig. 72: Selection EtherCAT connection (TwinCAT 3)

A real Ethernet port on the runtime system must then be assigned to this virtual device.

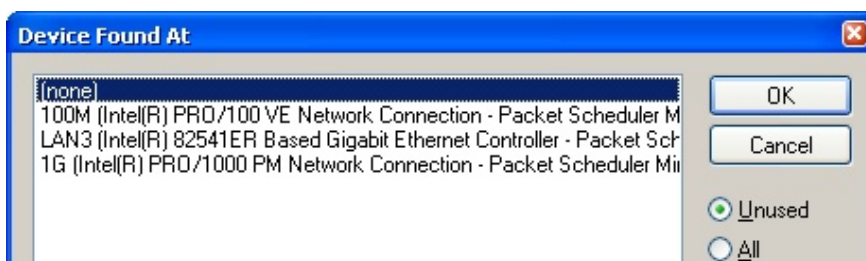


Fig. 73: Selection Ethernet port

This query can appear automatically when the EtherCAT device is created, or the assignment can be set/changed later in the properties dialog; see Fig. "EtherCAT device properties (TwinCAT 3)".

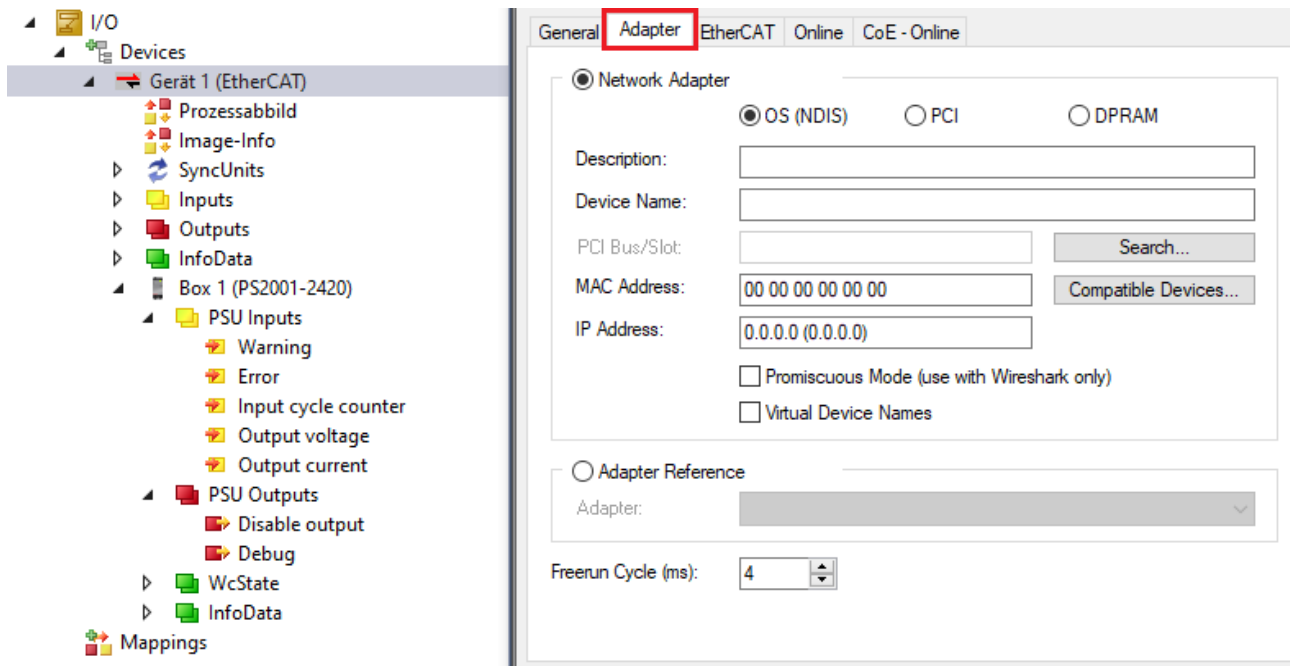
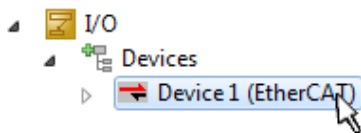


Fig. 74: EtherCAT device properties (TwinCAT 3)

TwinCAT 3: the properties of the EtherCAT device can be opened by double click on "Device .. (EtherCAT)" within the Solution Explorer under "I/O":



**i Selection Ethernet port**

Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective [installation page \[p 71\]](#).

**Defining EtherCAT slaves**

Further devices can be appended by right-clicking on a device in the configuration tree.

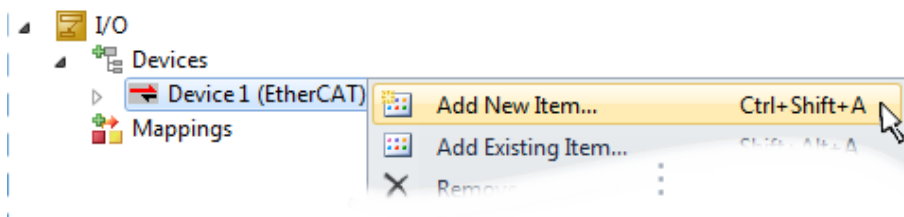


Fig. 75: Appending EtherCAT devices (TwinCAT 3)

The dialog for selecting a new device opens. Only devices for which ESI files are available are displayed.

The selection also only offers devices that are to be added to the previously clicked device - the physical layer possible on this port is displayed (Fig. "New EtherCAT device selection dialog", A). It can be wired FastEthernet Ethernet physics with PHY transmission, in which case only wired devices can be selected, as shown in the "New EtherCAT device selection dialog". If the previous device has several free ports (e.g. EK1122 or EK1100), the desired port can be selected on the right-hand side (A).

**Overview of physical layer**

- "Ethernet": Wired 100BASE-TX: EK couplers, EP box modules, devices with RJ45/M8/M12 connector
- "E-bus": LVDS "terminal bus", "EJ modules": EL/ES terminals, various modular modules

The search field facilitates finding a specific device (from TwinCAT 2.11 or TwinCAT 3).

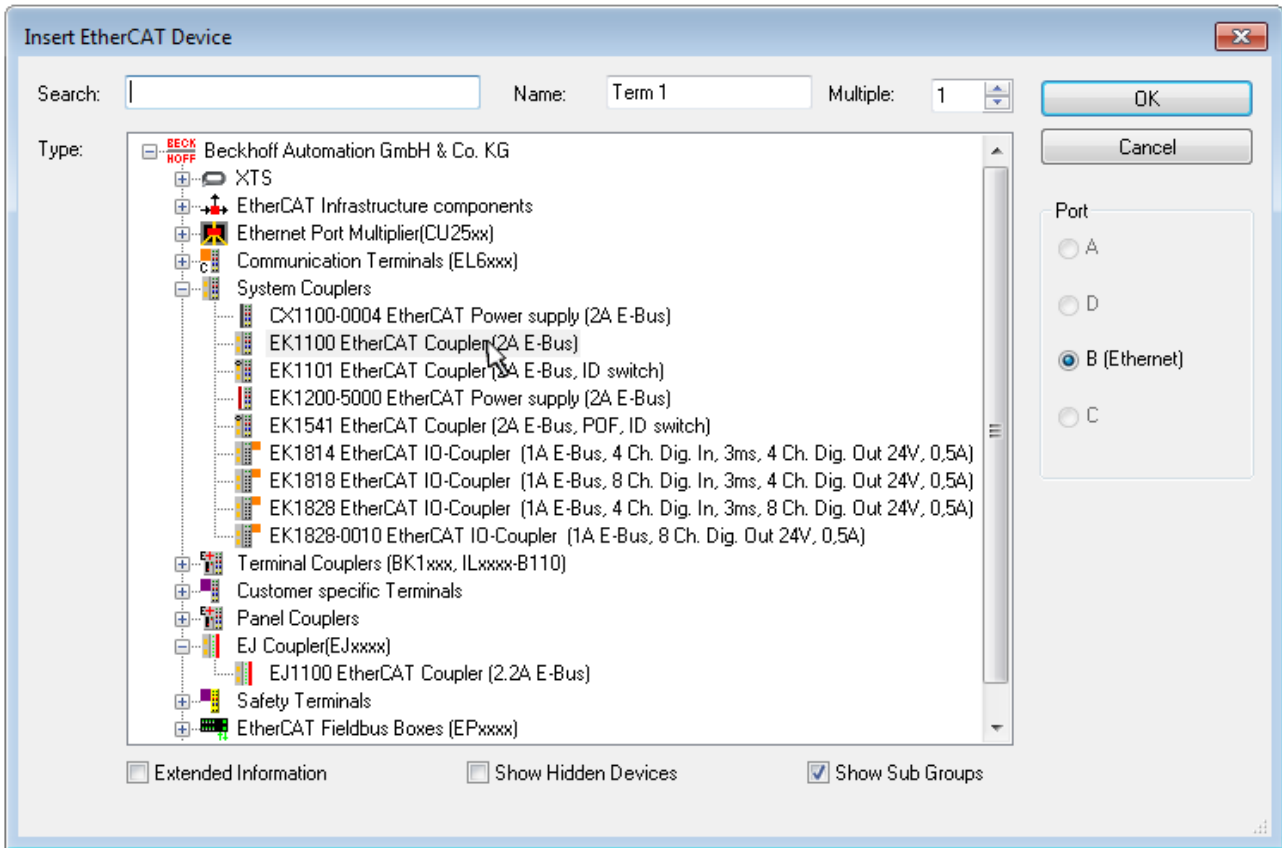


Fig. 76: Selection dialog for new EtherCAT device

By default only the name/type of the device is used as selection criterion. For selecting a specific revision of the device the revision can be displayed as “Extended Information”.

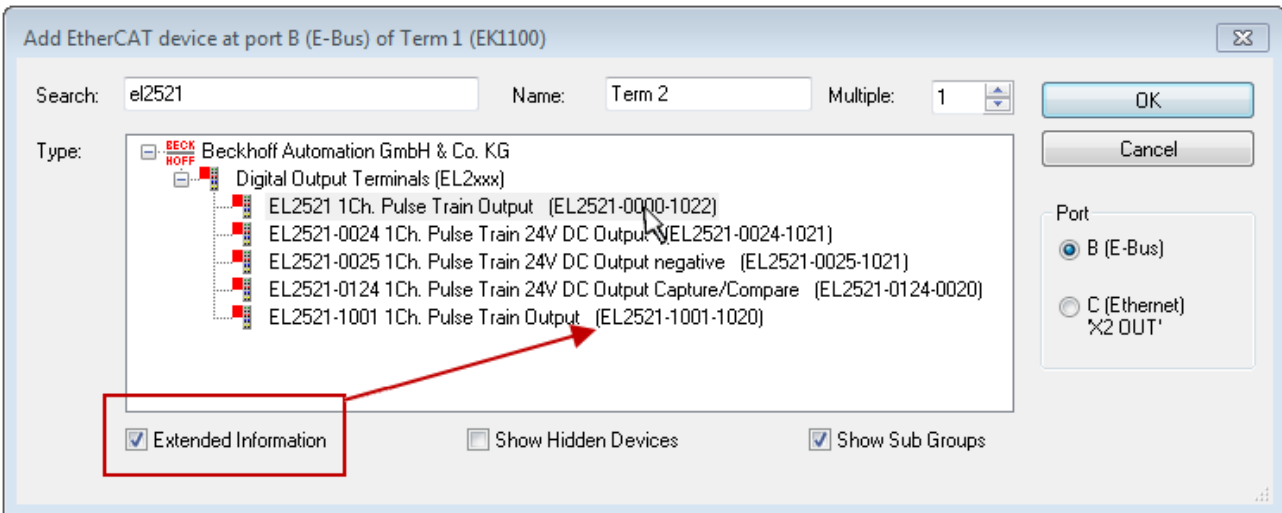


Fig. 77: Display of device revision

In many cases several device revisions were created for historic or functional reasons, e.g. through technological advancement. For a simplified display (see Fig. "New EtherCAT device selection dialog"), only the last (=highest) revision and thus the last production status is displayed in the selection dialog for Beckhoff devices. If all revisions of a device available in the system as ESI descriptions are to be displayed, the "Show Hidden Devices" checkbox must be selected, see Fig. "Display of previous revisions".

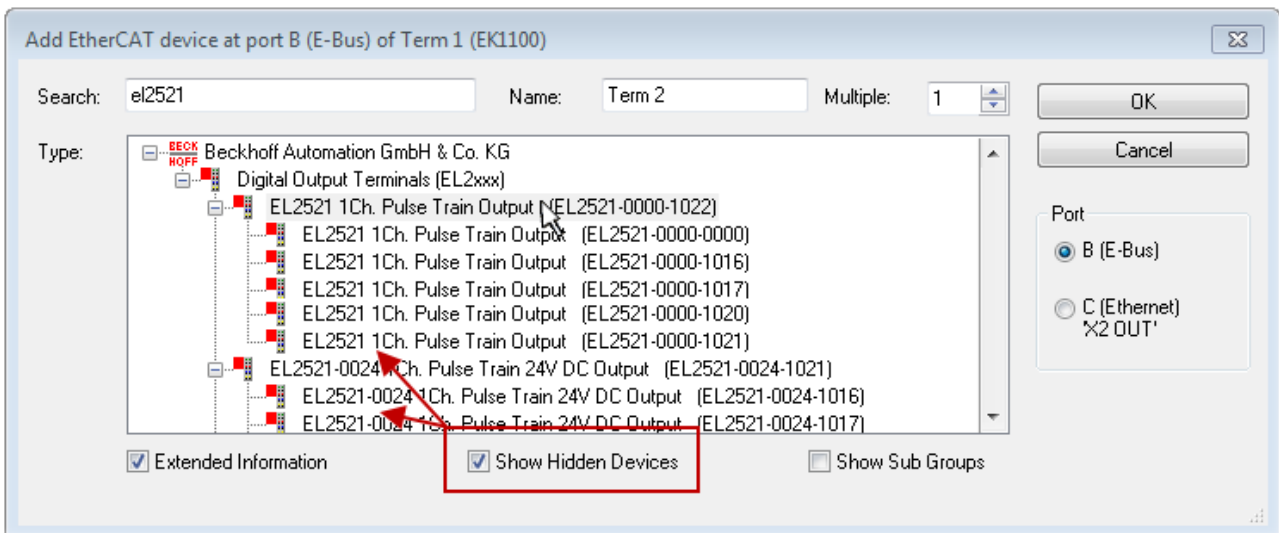


Fig. 78: Display of previous revisions

**i Device selection based on revision, compatibility**

The ESI description also defines the process image, the communication type between master and slave/device and the device functions, if applicable. The physical device (firmware, if available) has to support the communication queries/settings of the master. This is backward compatible, i.e. newer devices (higher revision) should be supported if the EtherCAT master addresses them as an older revision. The Beckhoff compatibility rule for EtherCAT Terminals/ Box modules/ EJ modules is to be assumed:

**Device revision in the system >= Device revision in the configuration**

This also allows devices to be replaced later without changing the configuration (deviating specifications possible for drives).

**Example**

An EL2521-0025-1018 is specified in the configuration, then an EL2521-0025-1018 or higher (-1019, -1020) can be used in practice.

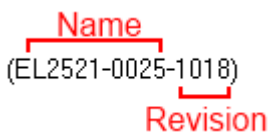


Fig. 79: Name/Revision Terminal

If current ESI descriptions are available in the TwinCAT system, the last revision offered in the selection dialog matches the Beckhoff state of production. It is recommended to use the last device revision when creating a new configuration, if current Beckhoff devices are used in the real application. Older revisions should only be used if older devices from stock are to be used in the application.

The device is then displayed with its process image in the configuration tree and can only be parameterized: Linking with the task, CoE/DC settings, PlugIn definition, StartUp settings, ...

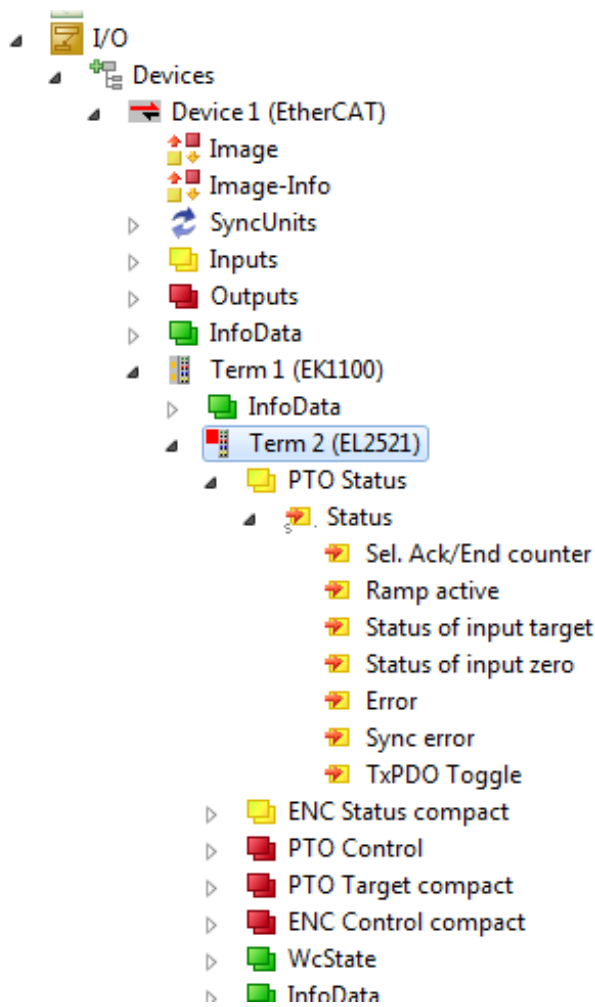


Fig. 80: EtherCAT Terminal in the TwinCAT tree (TwinCAT 3)


## 6.2.6 ONLINE configuration creation

### Detecting/scanning of the EtherCAT device

If the TwinCAT system is in CONFIG mode, you can search for devices online. This can be recognized by a symbol at the bottom right of the information bar:


- in the user interface of the TwinCAT 3 development environment by a symbol .

TwinCAT can be set in this mode:

- TwinCAT 3: by selecting  from the menu bar or via "TWINCAT" → "Restart TwinCAT (Config Mode)"

### **i** Online scanning in Config mode

The online search is not available in RUN mode (production operation). Note the differentiation between TwinCAT programming system and TwinCAT target system.

The TwinCAT 3 icon () in the Windows taskbar always represents the TwinCAT mode of the local IPC. In contrast, the TwinCAT state of the target system is displayed in the System Manager window in the TwinCAT 3 user interface.

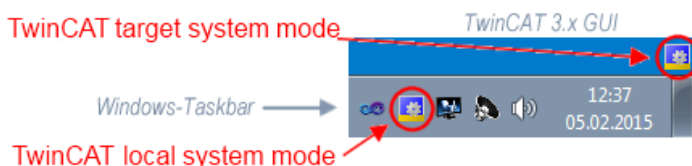


Fig. 81: Differentiation between local system/target system (TwinCAT 3)

Right-clicking on "I/O Devices" in the configuration tree opens the search dialog.

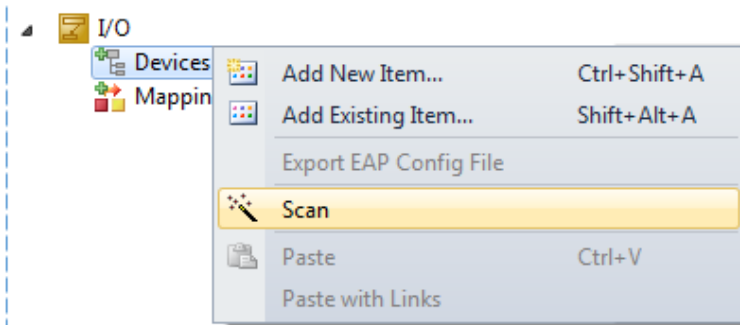


Fig. 82: Scan Devices (TwinCAT 3)

This scan mode not only tries to find EtherCAT devices (or Ethernet ports that can be used as such), but also NOVRAM, fieldbus cards, SMB etc. Not all devices can be found automatically.

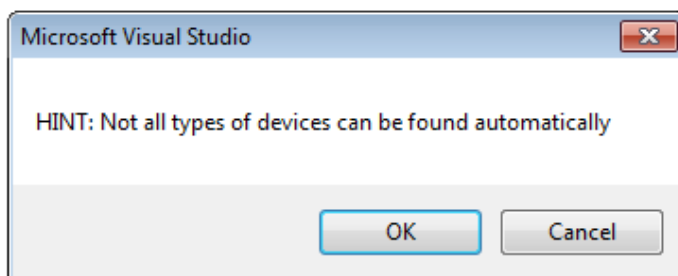


Fig. 83: Note automatic device scan (TwinCAT 3)

Ethernet ports with installed TwinCAT real-time driver are shown as "RT Ethernet" devices. An EtherCAT frame is sent to these ports for testing purposes. If the scan agent detects from the response that an EtherCAT slave is connected, the port is immediately shown as an "EtherCAT Device".

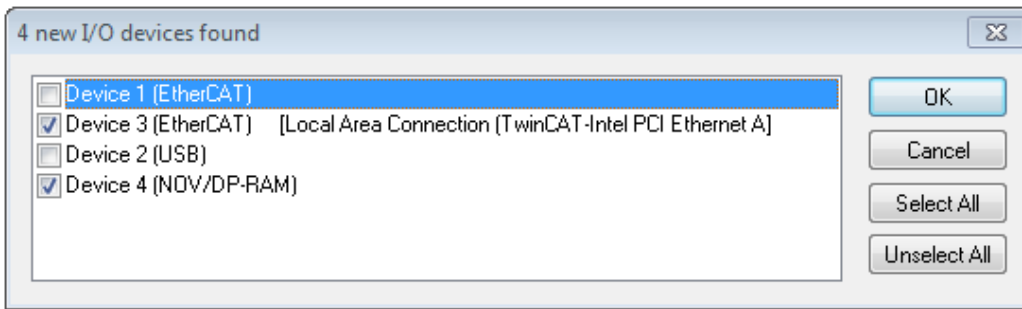


Fig. 84: Detected Ethernet devices

Devices can be selected using the corresponding checkboxes (e.g. device 3 and device 4 are selected as shown in the figure "Detected Ethernet devices"). After confirming with "OK", a device scan is suggested for all selected devices, see Fig. "Scan query after automatic creation of an EtherCAT device".

### ● Selection Ethernet port



Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective [installation page](#) [▶ 71].

## Detecting/Scanning the EtherCAT devices

### ● Online scan functionality



During a scan the master queries the identity information of the EtherCAT slaves from the slave EEPROM. The name and revision are used for determining the type. The respective devices are located in the stored ESI data and integrated in the configuration tree in the default state defined there.

**Name**  
(EL2521-0025-1018)  
**Revision**

Fig. 85: Example Default state

## NOTICE

### Slave scanning in practice in standard machine production

The scanning function should be used with care. It is a practical and fast tool for creating an initial configuration as a basis for commissioning. In standard machine construction or when reproducing the system, however, the function should no longer be used to create the configuration but, if necessary, for [comparison](#) [▶ 90] with the defined initial configuration.

Background: as Beckhoff occasionally increases the revision status of the delivered products for reasons of product maintenance, a configuration can be created by such a scan which (with an identical machine structure) is identical in terms of the device list, but the respective device revision may differ from the initial configuration.

### Example:

Company A builds the prototype of a machine B, which is to be produced in series later on. To do this the prototype is built, a scan of the I/O devices is performed in TwinCAT and the initial configuration 'B.tsm' is created. The EL2521-0025 EtherCAT terminal with the revision 1018 is located at any position. It is thus built into the TwinCAT configuration in this way:



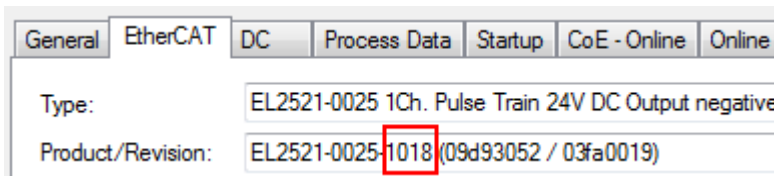


Fig. 86: Installation EtherCAT Terminal with revision -1018

Likewise, during the prototype test phase, the functions and properties of this terminal are tested by the programmers/commissioning engineers and used if necessary, i.e. addressed from the PLC 'B.pro' or the NC. (the same applies analogously to the TwinCAT 3 solution files).

The prototype development is now completed and series production of machine B starts, for which Beckhoff continues to supply the EL2521-0025-0018. If the commissioning engineers of the standard machine production department always carry out a scan, a B configuration with the identical contents results again for each machine. Likewise, A might create spare parts stores worldwide for the coming standard machines with EL2521-0025-1018 terminals.

After some time Beckhoff extends the EL2521-0025 by a new feature C. Therefore the FW is changed, outwardly recognizable by a higher FW version **and a new revision -1019**. Nevertheless the new device naturally supports functions and interfaces of the predecessor version(s); an adaptation of 'B.tsm' or even 'B.pro' is therefore unnecessary. The standard machines can continue to be built with 'B.tsm' and 'B.pro'; it makes sense to perform a [comparative scan](#) [► 90] against the initial configuration 'B.tsm' in order to check the built machine.

However, if the series machine production department now doesn't use 'B.tsm', but instead carries out a scan to create the productive configuration, the revision **-1019** is automatically detected and built into the configuration:

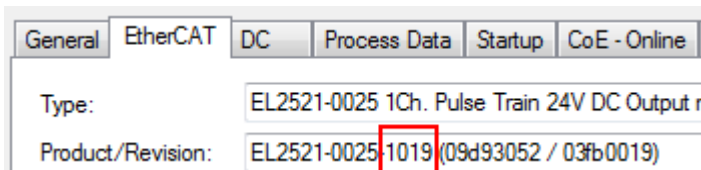


Fig. 87: Recognizing EtherCAT Terminal with revision -1019

This is usually not noticed by the commissioning engineers. TwinCAT cannot signal anything either, since virtually a new configuration is created. According to the compatibility rule, however, this means that no EL2521-0025-**1018** should be built into this machine as a spare part (even if this nevertheless works in the vast majority of cases).

In addition, it could be the case that, due to the development accompanying production in company A, the new feature C of the EL2521-0025-1019 (for example, an improved analog filter or an additional process data for the diagnosis) is discovered and used without in-house consultation. The previous stock of spare part devices are then no longer to be used for the new configuration 'B2.tsm' created in this way.

If standard machine production is established, the scan should only be performed for informative purposes for comparison with a defined initial configuration. Changes are to be made with care!

If an EtherCAT device was created in the configuration (manually or through a scan), the I/O field can be scanned for devices/slaves.

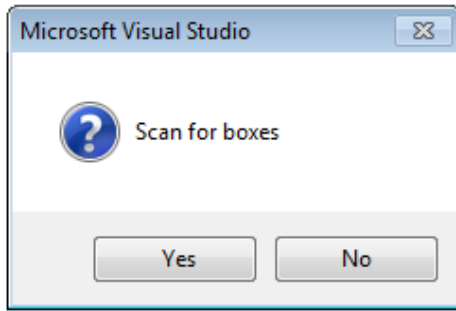


Fig. 88: Scan query after automatic creation of an EtherCAT device (TwinCAT 3)

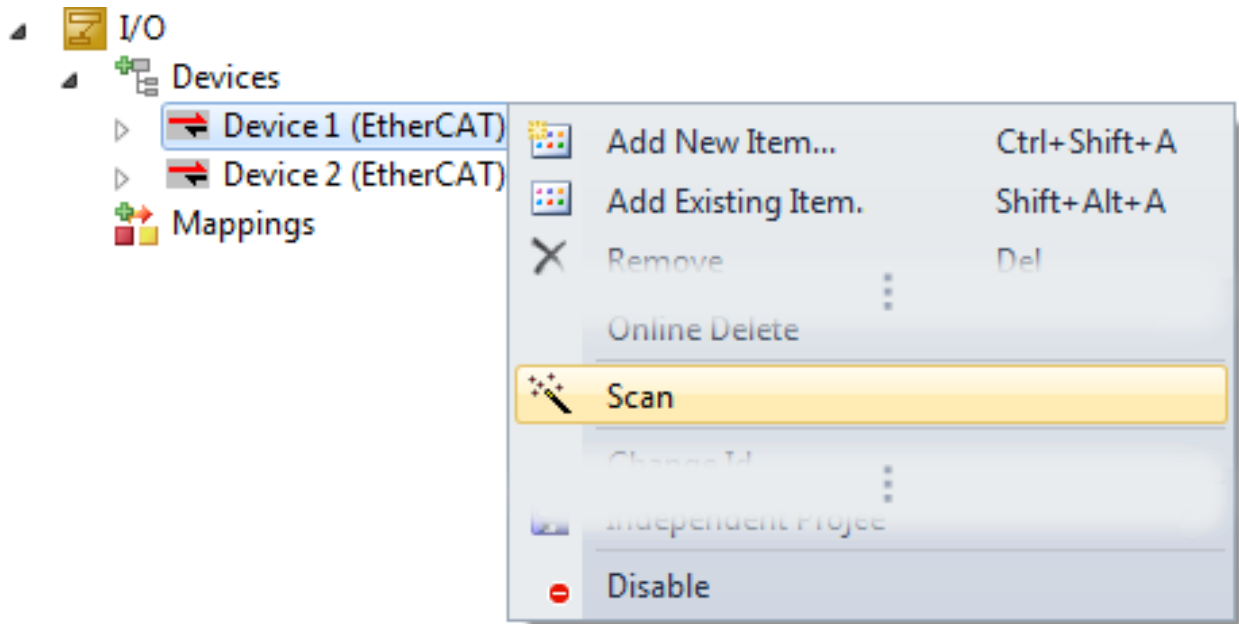


Fig. 89: Manual triggering of a device scan on a specified EtherCAT device (TwinCAT 3)

In the user interface (TwinCAT 3), the scan process can be followed via the loading bar at the bottom of the status bar.

The configuration is established and can then be switched to online state (OPERATIONAL).

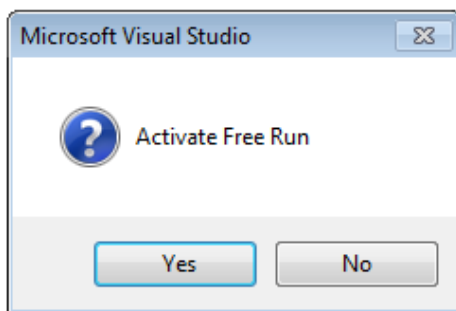


Fig. 90: Config/FreeRun query (TwinCAT 3)

In Config/FreeRun mode the System Manager display alternates between blue and red, and the EtherCAT device continues to operate with the idling cycle time of 4 ms (default setting), even without active task (NC, PLC).



Fig. 91: Display of the change between "Free Run" and "Config Mode" at the bottom right of the status bar



Fig. 92: TwinCAT can also be switched to this state by using a button (TwinCAT 3)

The EtherCAT system should then be in functional cyclical operation as shown in Fig. *Example online display*.



Fig. 93: Example online display

Please note:

- all slaves should be in OP state
- the EtherCAT master should be in "Actual State" OP
- "frames/sec" should match the cycle time taking into account the sent number of frames
- no excessive "LostFrames" or CRC errors should occur

The configuration is now complete. It can be modified as described under [manual procedure](#) [► 80].

### Troubleshooting

Various effects may occur during scanning.

- An **unknown device** is detected, i.e. an EtherCAT slave for which no ESI XML description is available. In this case the System Manager offers to read any ESI that may be stored in the device. This case is described in the chapter "Notes regarding ESI device description".
- **Device are not detected properly**  
Possible reasons include:
  - faulty data links, resulting in data loss during the scan
  - slave has invalid device description

The connections and devices should be checked in a targeted manner, e.g. via the emergency scan. Then re-run the scan.

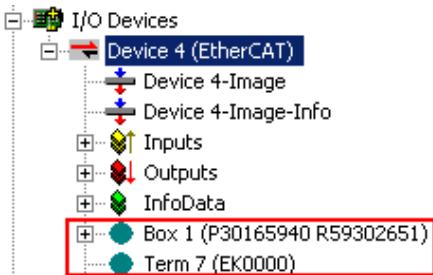


Fig. 94: Faulty identification

In the System Manager such devices may be set up as EK0000 or unknown devices. Operation is not possible or meaningful.

## Scan over existing Configuration

**NOTICE****Change of the configuration after comparison**

With this scan (TwinCAT 2.11 or 3.1) only the device properties vendor (manufacturer), device name and revision are compared at present! A 'ChangeTo' or 'Copy' should only be carried out with care, taking into consideration the Beckhoff I/O compatibility rule (see above). The device configuration is then replaced by the revision found; this can affect the supported process data and functions.

If a scan is initiated for an existing configuration, the actual I/O environment may match the configuration exactly or it may differ. This enables the configuration to be compared.

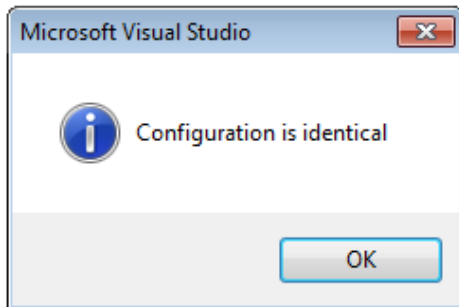


Fig. 95: Identical configuration (TwinCAT 3)

If differences are detected, these are displayed in the correction dialog and the configuration can be adjusted immediately.

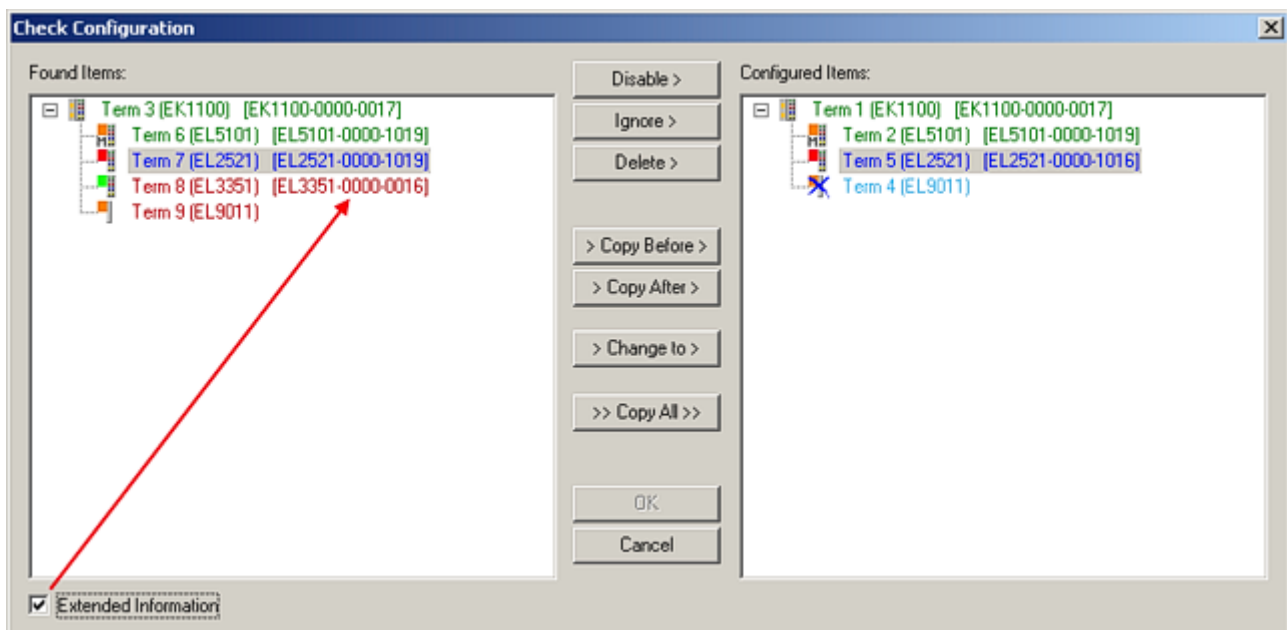


Fig. 96: Correction dialog

The display of "Extended Information" is recommended because it makes differences in the revision visible.

Color	Explanation
green	This EtherCAT slave matches the entry on the other side. Both type and revision match.
blue	This EtherCAT slave is present on the other side, but in a different revision. This other revision may have different default settings for the process data and different/additional functions. If the revision found is > than the configured revision, it can be used, taking compatibility into account.  If the found revision is lower than the configured revision, it is likely that the use is not possible. The found devices may not support all functions that the master expects based on the higher revision number.
light blue	This EtherCAT slave is ignored ("Ignore" button)
red	<ul style="list-style-type: none"> <li>This EtherCAT slave is not present on the other side.</li> <li>It is available, but in a different revision, which also differs in its properties from the one specified. The compatibility principle also applies here: If the revision found is &gt; than the configured revision, it can be used, taking compatibility into account, as successor devices should support the functions of the predecessor devices.  If the found revision is lower than the configured revision, it is likely that the use is not possible. The found devices may not support all functions that the master expects based on the higher revision number.</li> </ul>

**i Device selection based on revision, compatibility**

The ESI description also defines the process image, the communication type between master and slave/device and the device functions, if applicable. The physical device (firmware, if available) has to support the communication queries/settings of the master. This is backward compatible, i.e. newer devices (higher revision) should be supported if the EtherCAT master addresses them as an older revision. The Beckhoff compatibility rule for EtherCAT Terminals/ Box modules/ EJ modules is to be assumed:

**Device revision in the system >= Device revision in the configuration**

This also allows devices to be replaced later without changing the configuration (deviating specifications possible for drives).

**Example**

An EL2521-0025-**1018** is specified in the configuration, then an EL2521-0025-**1018** or higher (**-1019**, **-1020**) can be used in practice.

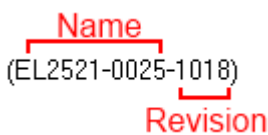


Fig. 97: Name/Revision Terminal

If current ESI descriptions are available in the TwinCAT system, the last revision offered in the selection dialog matches the Beckhoff state of production. It is recommended to use the last device revision when creating a new configuration, if current Beckhoff devices are used in the real application. Older revisions should only be used if older devices from stock are to be used in the application.

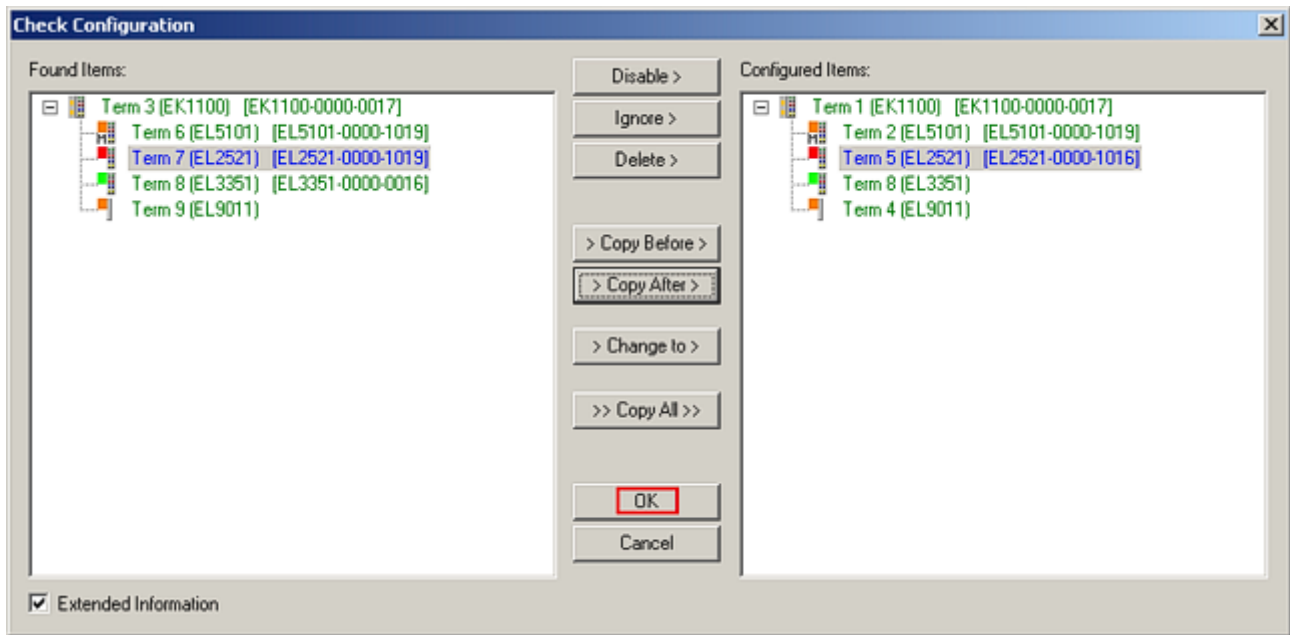


Fig. 98: Correction dialog with modifications

Once all modifications have been saved or accepted, click "OK" to transfer them to the real \*.tsm configuration.

### Change to Compatible Type

With "Change to Compatible Type...", TwinCAT offers a function for replacing a device while retaining the links in the task.

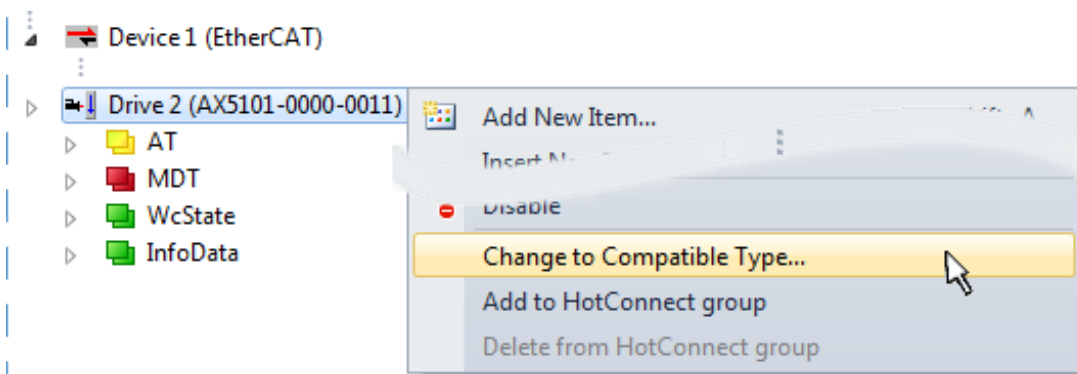


Fig. 99: Dialog "Change to Compatible Type" (TwinCAT 3)

The following elements in the ESI of an EtherCAT device are compared by TwinCAT and assumed to be the same in order to decide whether a device is displayed as "compatible":

- Physics (e.g. RJ45, Ebus...)
- FMMU (additional ones are permitted)
- SyncManager (SM, additional ones are permitted)
- EoE (attributes MAC, IP)
- CoE (attributes SdoInfo, PdoAssign, PdoConfig, PdoUpload, CompleteAccess)
- FoE
- PDO (process data: sequence, SyncUnit SU, SyncManager SM, EntryCount, Entry.Datatype)

This function is used intensively in devices of the AX5000 family.

### Change to Alternative Type

The TwinCAT System Manager offers a function for the exchange of a device: Change to Alternative Type

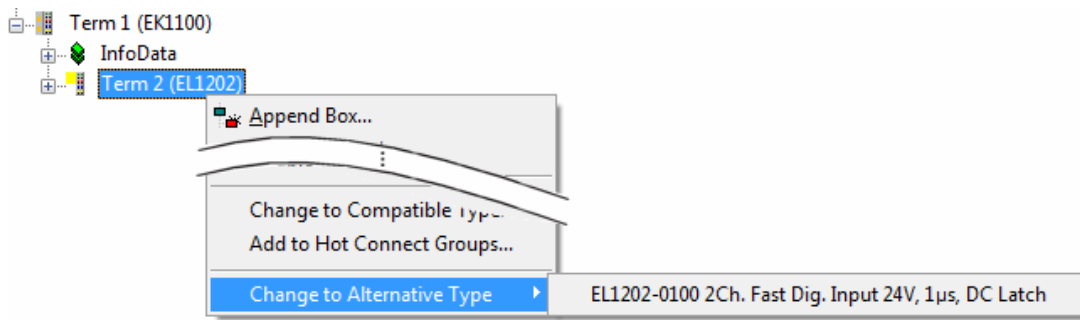


Fig. 100: TwinCAT 2 Dialog Change to Alternative Type

If called, the System Manager searches in the procured device ESI (in this example: EL1202-0000) for details of compatible devices contained there. The configuration is changed and the ESI-EEPROM is overwritten at the same time – therefore this process is possible only in the online state (ConfigMode).

### 6.2.7 EtherCAT subscriber configuration

In the TwinCAT 3 development environment, click in the Solution Explorer on the element of the terminal in the tree that you want to configure (in the example: Terminal 3: EL3751).

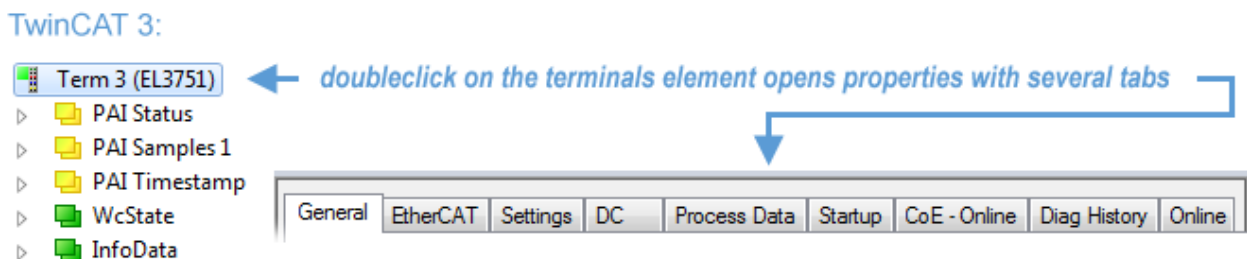


Fig. 101: "Tree branch" element as terminal EL3751

Various tabs for configuring the terminal are now available in the development environment (TwinCAT 3). The level of complexity of a device determines which tabs are available. As can be seen in the example above, the EL3751 terminal offers many setting options and provides a corresponding number of tabs. In contrast, only the "General", "EtherCAT", "Process Data" and "Online" tabs are available for the EL1004 terminal, for example. Some terminals, such as the EL6695, offer special functions via a tab with its own terminal name, in this case "EL6695". There is also a special "Settings" tab for terminals with extensive setting options (e.g. EL3751).

#### “General” tab

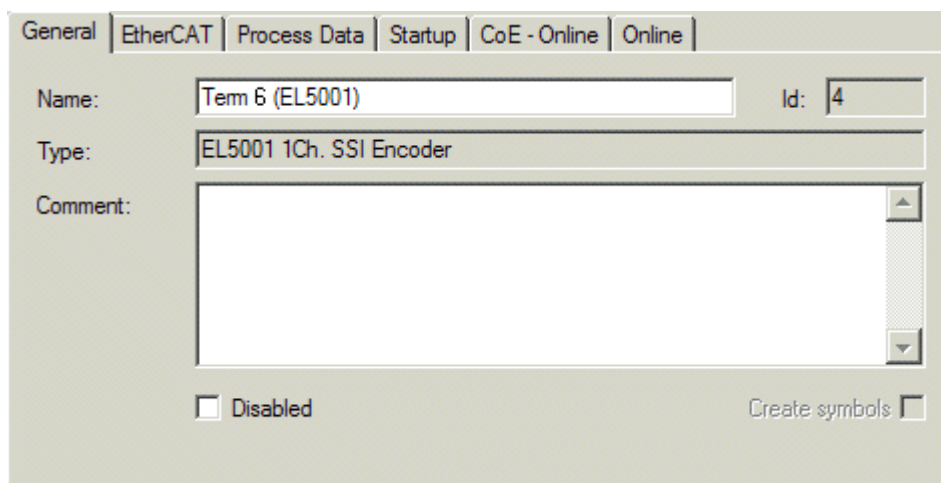


Fig. 102: “General” tab



<b>Name</b>	Name of the EtherCAT device
<b>Id</b>	Number of the EtherCAT device
<b>Type</b>	EtherCAT device type
<b>Comment</b>	Here you can add a comment (e.g. regarding the system).
<b>Disabled</b>	Here you can deactivate the EtherCAT device.
<b>Create symbols</b>	Access to this EtherCAT slave via ADS is only available if this control box is activated.

#### “EtherCAT” tab

Fig. 103: “EtherCAT” tab

<b>Type</b>	EtherCAT device type
<b>Product/Revision</b>	Product and revision number of the EtherCAT device
<b>Auto Inc Addr.</b>	Auto increment address of the EtherCAT device. The auto increment address can be used for addressing each EtherCAT device in the communication ring through its physical position. Auto increment addressing is used during the start-up phase when the EtherCAT master allocates addresses to the EtherCAT devices. With auto increment addressing the first EtherCAT slave in the ring has the address 0000 <sub>hex</sub> . For each further slave the address is decremented by 1 (FFFF <sub>hex</sub> , FFFE <sub>hex</sub> etc.).
<b>EtherCAT Addr.</b>	Fixed address of an EtherCAT slave. This address is allocated by the EtherCAT master during the start-up phase. Tick the control box to the left of the input field in order to modify the default value.
<b>Previous Port</b>	Name and port of the EtherCAT device to which this device is connected. If it is possible to connect this device with another one without changing the order of the EtherCAT devices in the communication ring, then this combination field is activated and the EtherCAT device to which this device is to be connected can be selected.
<b>Advanced Settings</b>	This button opens the dialogs for advanced settings.

The link at the bottom of the tab points to the product page for this EtherCAT device on the web.

#### “Process Data” tab

Indicates the configuration of the process data. The input and output data of the EtherCAT slave are represented as CANopen process data objects (**Process Data Objects**, PDOs). The user can select a PDO via PDO assignment and modify the content of the individual PDO via this dialog, if the EtherCAT slave supports this function.



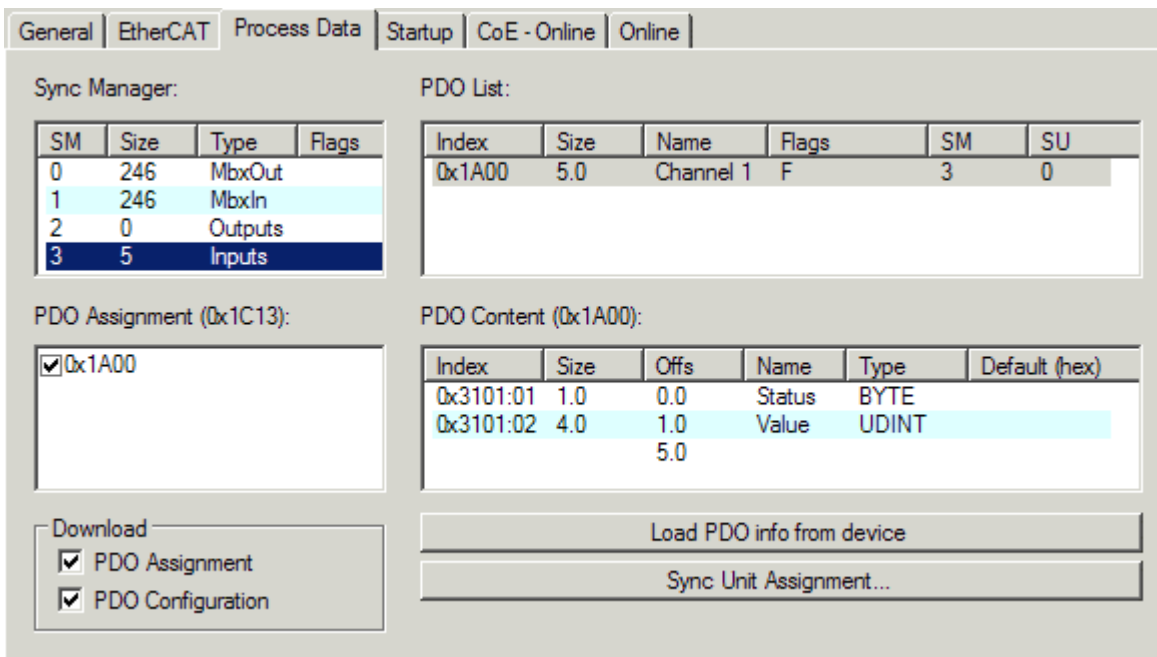


Fig. 104: "Process Data" tab

The process data (PDOs) transferred by an EtherCAT slave during each cycle are user data which the application expects to be updated cyclically or which are sent to the slave. To this end the EtherCAT master (Beckhoff TwinCAT) parameterizes each EtherCAT slave during the start-up phase to define which process data (size in bits/bytes, source location, transmission type) it wants to transfer to or from this slave. Incorrect configuration can prevent successful start-up of the slave.

For Beckhoff EtherCAT EL, ES, EM, EJ and EP slaves the following applies in general:

- The input/output process data supported by the device are defined by the manufacturer in the ESI/XML description. The TwinCAT EtherCAT Master uses the ESI description to configure the slave correctly.
- The process data can be modified in the System Manager. See the device documentation. Examples of modifications include: mask out a channel, displaying additional cyclic information, 16-bit display instead of 8-bit data size, etc.
- In so-called "intelligent" EtherCAT devices the process data information is also stored in the CoE directory. Any changes in the CoE directory that lead to different PDO settings prevent successful startup of the slave. It is not advisable to deviate from the designated process data, because the device firmware (if available) is adapted to these PDO combinations.

If the device documentation allows modification of process data, proceed as follows (see Figure *Configuring the process data*).

- A: select the device to configure
- B: in the "Process Data" tab select Input or Output under SyncManager (C)
- D: the PDOs can be selected or deselected
- H: the new process data are visible as linkable variables in the System Manager  
The new process data are active once the configuration has been activated and TwinCAT has been restarted (or the EtherCAT master has been restarted)
- E: if a slave supports this, Input and Output PDO can be modified simultaneously by selecting a so-called PDO record ("predefined PDO settings").

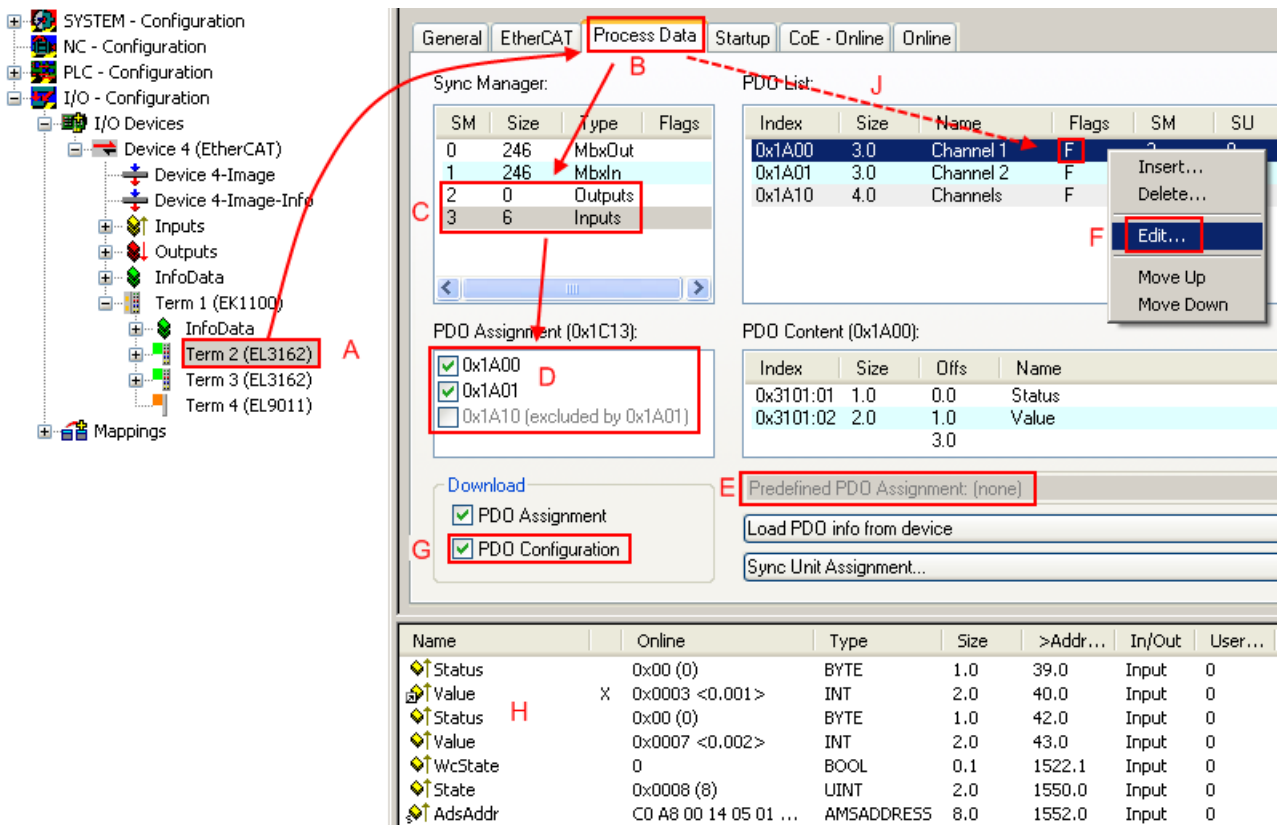


Fig. 105: Configuring the process data

**i Manual modification of the process data**

According to the ESI description, a PDO can be identified as “fixed” with the flag “F” in the PDO overview (Fig. *Configuring the process data*, J). The configuration of such PDOs cannot be changed, even if TwinCAT offers the associated dialog (“Edit”). In particular, CoE content cannot be displayed as cyclic process data. This generally also applies in cases where a device supports download of the PDO configuration, “G”. In case of incorrect configuration the EtherCAT slave usually refuses to start and change to OP state. The System Manager displays an “invalid SM cfg” logger message: This error message (“invalid SM IN cfg” or “invalid SM OUT cfg”) also indicates the reason for the failed start.

A detailed description [► 101] can be found at the end of this section.

**“Startup” tab**

The *Startup* tab is displayed if the EtherCAT slave has a mailbox and supports the *CANopen over EtherCAT* (CoE) or *Servo drive over EtherCAT* protocol. This tab indicates which download requests are sent to the mailbox during startup. It is also possible to add new mailbox requests to the list display. The download requests are sent to the slave in the same order as they are shown in the list.

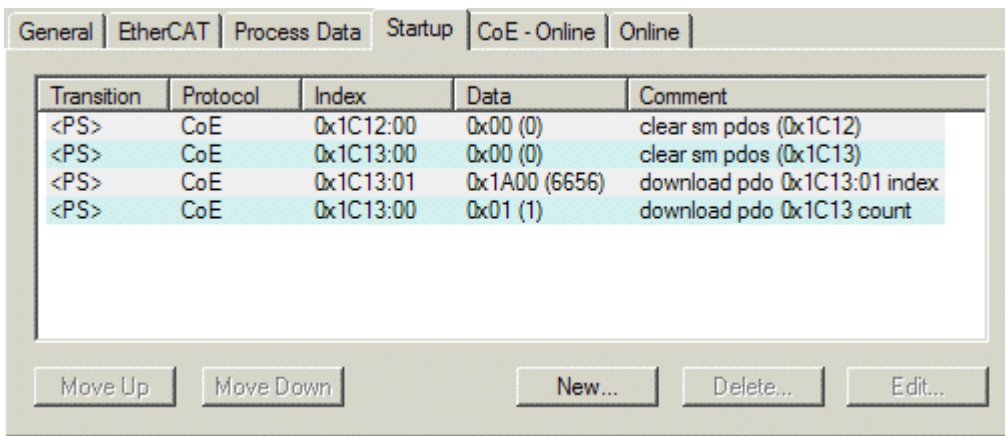


Fig. 106: "Startup" tab

Column	Description
Transition	Transition to which the request is sent. This can either be <ul style="list-style-type: none"> <li>the transition from pre-operational to safe-operational (PS), or</li> <li>the transition from safe-operational to operational (SO).</li> </ul> If the transition is enclosed in "<>" (e.g. <PS>), the mailbox request is fixed and cannot be modified or deleted by the user.
Protocol	Type of mailbox protocol
Index	Index of the object
Data	Date on which this object is to be downloaded.
Comment	Description of the request to be sent to the mailbox

- Move Up**      This button moves the selected request up by one position in the list.
- Move Down**      This button moves the selected request down by one position in the list.
- New**      This button adds a new mailbox download request to be sent during startup.
- Delete**      This button deletes the selected entry.
- Edit**      This button edits an existing request.

**"CoE - Online" tab**

The additional *CoE - Online* tab is displayed if the EtherCAT slave supports the *CANopen over EtherCAT* (CoE) protocol. This dialog lists the content of the object list of the slave (SDO upload) and enables the user to modify the content of an object from this list. Details for the objects of the individual EtherCAT devices can be found in the device-specific object descriptions.

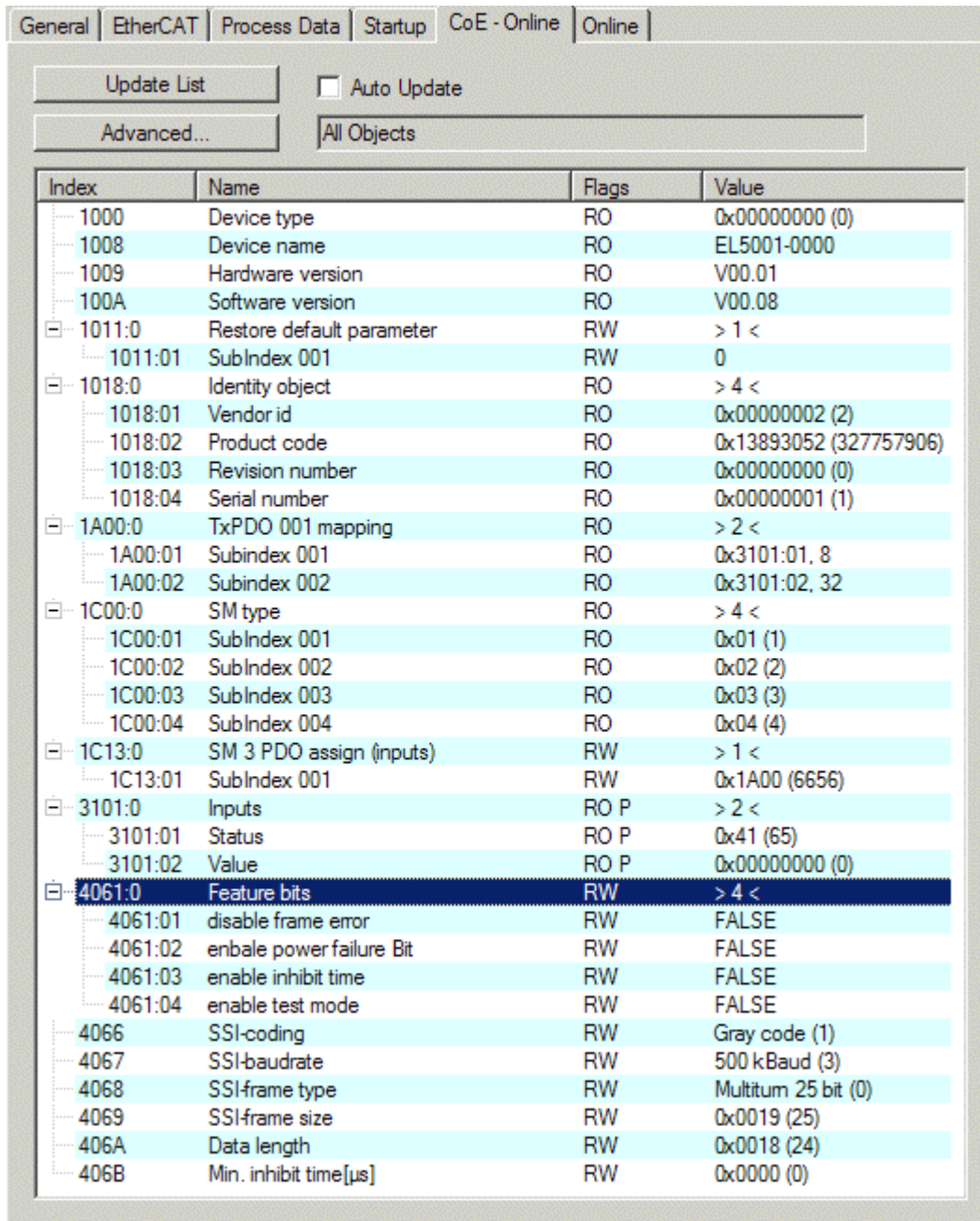


Fig. 107: "CoE - Online" tab

**Object list display**

Column	Description
Index	Index and sub-index of the object
Name	Name of the object
Flags	RW The object can be read, and data can be written to the object (read/write)
	RO The object can be read, but no data can be written to the object (read only)
	P An additional P identifies the object as a process data object.
Value	Value of the object

- Update List** The *Update list* button updates all objects in the displayed list
- Auto Update** If this check box is selected, the content of the objects is updated automatically.
- Advanced** The *Advanced* button opens the *Advanced Settings* dialog. Here you can specify which objects are displayed in the list.

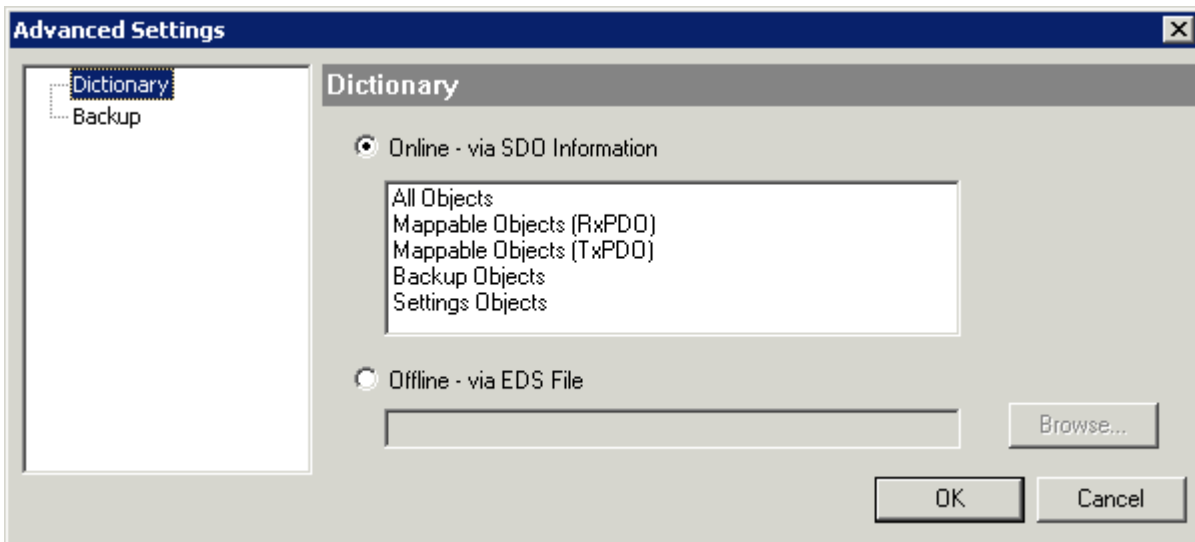


Fig. 108: Dialog “Advanced settings”

**Online - via SDO Information** If this option button is selected, the list of the objects included in the object list of the slave is uploaded from the slave via SDO information. The list below can be used to specify which object types are to be uploaded.

**Offline - via EDS File** If this option button is selected, the list of the objects included in the object list is read from an EDS file provided by the user.

“Online” tab

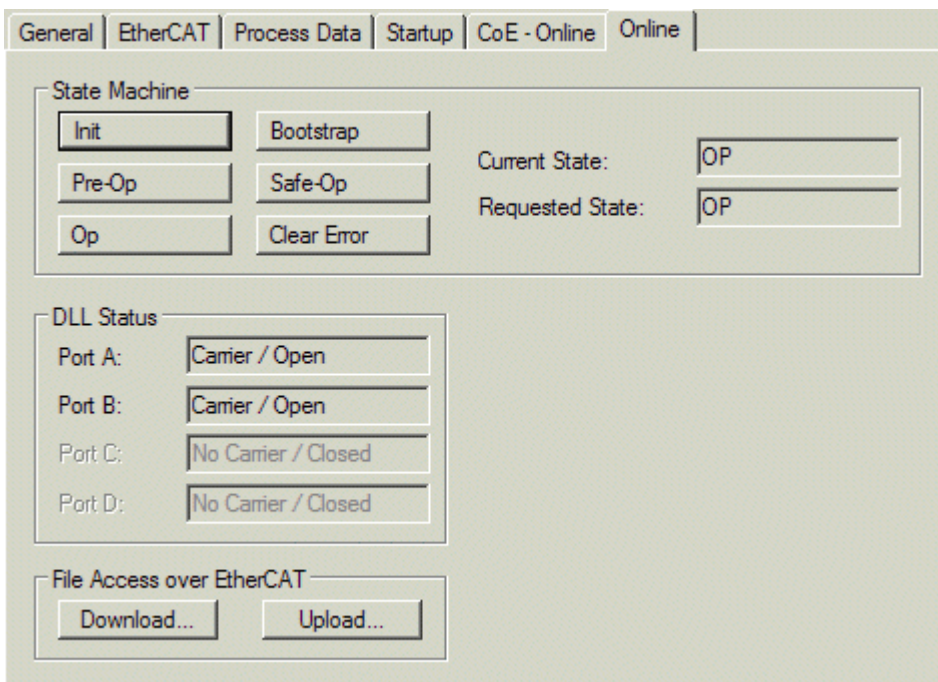


Fig. 109: “Online” tab

**State Machine**

- Init** This button attempts to set the EtherCAT device to the *Init* state.
- Pre-Op** This button attempts to set the EtherCAT device to the *pre-operational* state.
- Op** This button attempts to set the EtherCAT device to the *operational* state.
- Bootstrap** This button attempts to set the EtherCAT device to the *Bootstrap* state.
- Safe-Op** This button attempts to set the EtherCAT device to the *safe-operational* state.
- Clear Error** This button attempts to delete the fault display. If an EtherCAT slave fails during change of state it sets an error flag.  
  
Example: An EtherCAT slave is in PREOP state (pre-operational). The master now requests the SAFEOP state (safe-operational). If the slave fails during change of state it sets the error flag. The current state is now displayed as ERR PREOP. When the *Clear Error* button is pressed the error flag is cleared, and the current state is displayed as PREOP again.
- Current State** Indicates the current state of the EtherCAT device.
- Requested State** Indicates the state requested for the EtherCAT device.

**DLL Status**

Indicates the DLL status (data link layer status) of the individual ports of the EtherCAT slave. The DLL status can have four different states:

Status	Description
No Carrier / Open	No carrier signal is available at the port, but the port is open.
No Carrier / Closed	No carrier signal is available at the port, and the port is closed.
Carrier / Open	A carrier signal is available at the port, and the port is open.
Carrier / Closed	A carrier signal is available at the port, but the port is closed.

**File Access over EtherCAT**

- Download** With this button a file can be written to the EtherCAT device.
- Upload** With this button a file can be read from the EtherCAT device.

**“DC” tab (Distributed Clocks)**

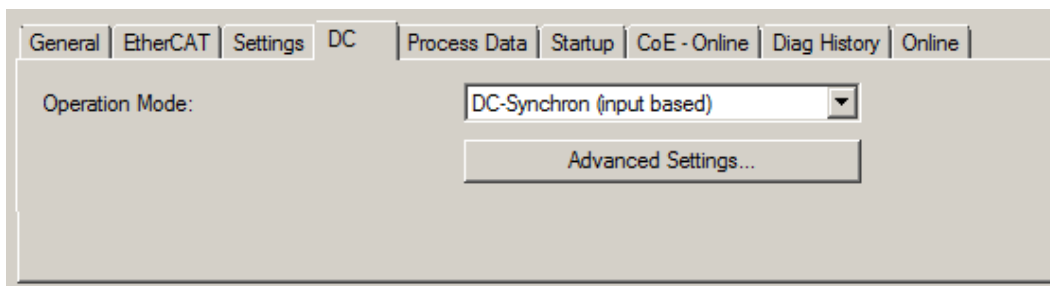


Fig. 110: “DC” tab (Distributed Clocks)

- Operation Mode** Options (optional):
  - FreeRun
  - SM-Synchron
  - DC-Synchron (Input based)
  - DC-Synchron
- Advanced Settings...** Advanced settings for readjustment of the real time determinant TwinCAT-clock

Detailed information to Distributed Clocks is specified on <http://infosys.beckhoff.com>:

**Fieldbus Components** → EtherCAT Terminals → EtherCAT System documentation → EtherCAT basics → Distributed Clocks



### 6.2.7.1 Detailed description of Process Data tab

#### Sync Manager

Lists the configuration of the Sync Manager (SM).

If the EtherCAT device has a mailbox, SM0 is used for the mailbox output (MbxOut) and SM1 for the mailbox input (MbxIn).

SM2 is used for the output process data (outputs) and SM3 (inputs) for the input process data.

If an input is selected, the corresponding PDO assignment is displayed in the *PDO Assignment* list below.

#### PDO assignment


PDO assignment of the selected Sync Manager. All PDOs defined for this Sync Manager type are listed here:

- If the output Sync Manager (outputs) is selected in the Sync Manager list, all RxPDOs are displayed.
- If the input Sync Manager (inputs) is selected in the Sync Manager list, all TxPDOs are displayed.

The selected entries are the PDOs involved in the process data transfer. In the tree diagram of the System Manager these PDOs are displayed as variables of the EtherCAT device. The name of the variable is identical to the *Name* parameter of the PDO, as displayed in the PDO list. If an entry in the PDO assignment list is disabled (not selected and grayed out), this indicates that the input is excluded from the PDO assignment. In order to be able to select a grayed out PDO, the currently selected PDO has to be deselected first.

#### **i** Activation of the PDO assignment

- ✓ If you have changed the PDO assignment, in order to activate the new PDO assignment,
  - a) the EtherCAT slave has to run through the PS state transition cycle (from pre-operational to safe-operational) once (see [Online \[▶ 99\]](#) tab),

- b) and the System Manager has to reload the EtherCAT slaves 

#### PDO list

List of all PDOs supported by this EtherCAT device. The content of the selected PDOs is displayed in the *PDO Content* list. The PDO configuration can be modified by double-clicking on an entry.

Column	Description	
Index	PDO index.	
Size	Size of the PDO in bytes.	
Name	Name of the PDO. If this PDO is assigned to a Sync Manager, it appears as a variable of the slave with this parameter as the name.	
Flags	F	Fixed content: The content of this PDO is fixed and cannot be changed by the System Manager.
	M	Mandatory PDO. This PDO is mandatory and must therefore be assigned to a Sync Manager! Consequently, this PDO cannot be deleted from the <i>PDO Assignment</i> list
SM	Sync Manager to which this PDO is assigned. If this entry is empty, this PDO does not take part in the process data traffic.	
SU	Sync unit to which this PDO is assigned.	

#### PDO Content

Indicates the content of the PDO. If flag F (fixed content) of the PDO is not set the content can be modified.

## Download

If the device is intelligent and has a mailbox, the configuration of the PDO and the PDO assignments can be downloaded to the device. This is an optional feature that is not supported by all EtherCAT slaves.

## PDO Assignment

If this check box is selected, the PDO assignment that is configured in the PDO Assignment list is downloaded to the device on startup. The required commands to be sent to the device can be viewed in the Startup [► 96] tab.

## PDO Configuration

If this check box is selected, the configuration of the respective PDOs (as shown in the PDO list and the PDO Content display) is downloaded to the EtherCAT slave.

## 6.2.8 Import/Export of EtherCAT devices with SCI and XTI

### SCI and XTI Export/Import – Handling of user-defined modified EtherCAT slaves

#### 6.2.8.1 Basic principles

An EtherCAT slave is basically parameterized through the following elements:

- Cyclic process data (PDO)
- Synchronization (Distributed Clocks, FreeRun, SM-Synchron)
- CoE parameters (acyclic object dictionary)

Note: Not all three elements may be present, depending on the slave.

For a better understanding of the export/import function, let's consider the usual procedure for IO configuration:

- The user/programmer processes the IO configuration in the TwinCAT system environment. This involves all input/output devices such as drives that are connected to the fieldbuses used.  
Note: In the following sections, only EtherCAT configurations in the TwinCAT system environment are considered.
- For example, the user manually adds devices to a configuration or performs a scan on the online system.
- This results in the IO system configuration.
- On insertion, the slave appears in the system configuration in the default configuration provided by the vendor, consisting of default PDO, default synchronization method and CoE StartUp parameter as defined in the ESI (XML device description).
- If necessary, elements of the slave configuration can be changed, e.g. the PDO configuration or the synchronization method, based on the respective device documentation.

It may become necessary to reuse the modified slave in other projects in this way, without having to make equivalent configuration changes to the slave again. To accomplish this, proceed as follows:

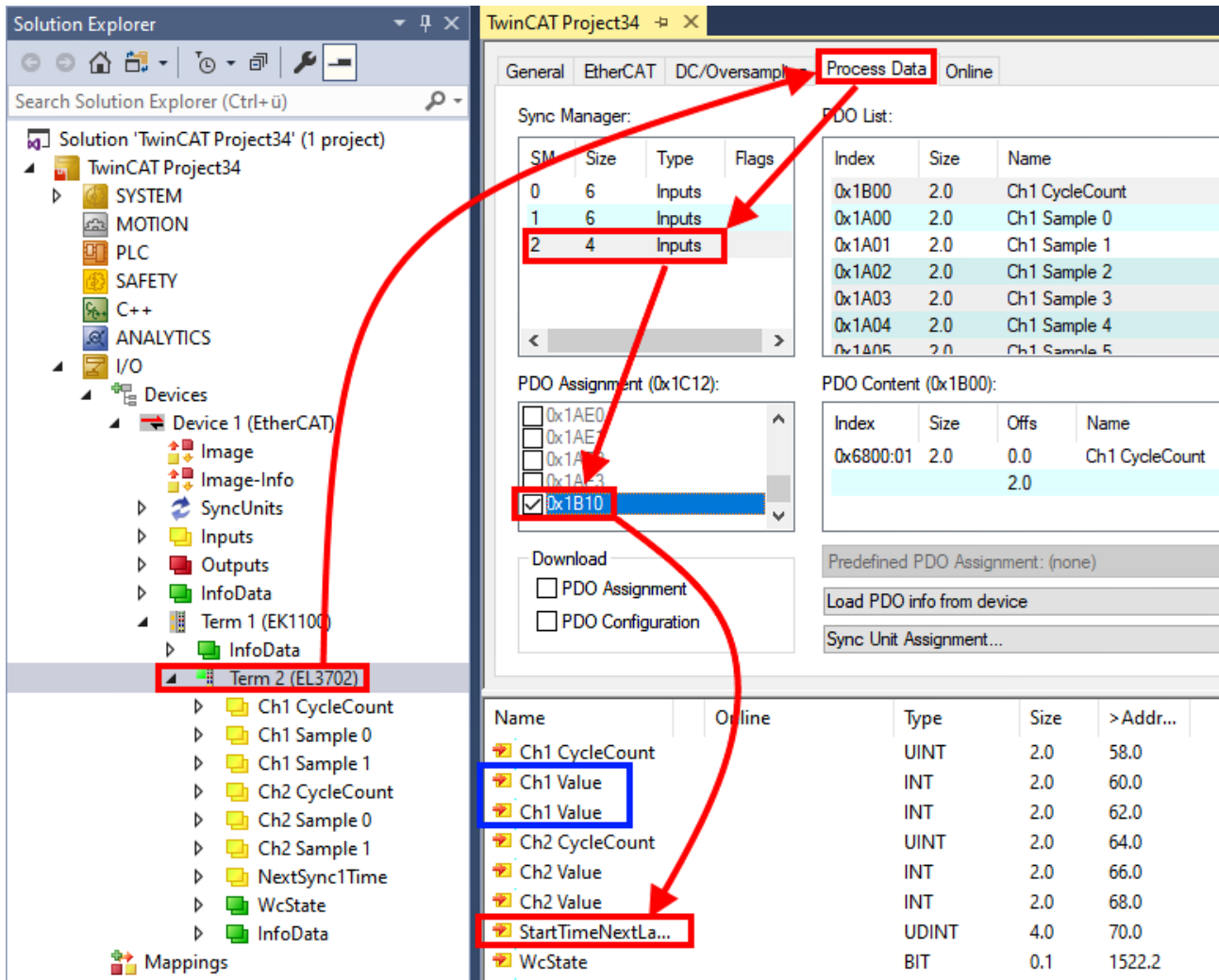
- Export the slave configuration from the project,
- Store and transport as a file,
- Import into another EtherCAT project.

TwinCAT offers two methods for this purpose:

- within the TwinCAT environment: Export/Import as **x**ti file or
- outside, i.e. beyond the TwinCAT limits: Export/Import as **s**ci file.

An example is provided below for illustration purposes: an EL3702 terminal with standard setting is switched to 2-fold oversampling (blue) and the optional PDO "StartTimeNextLatch" is added (red):

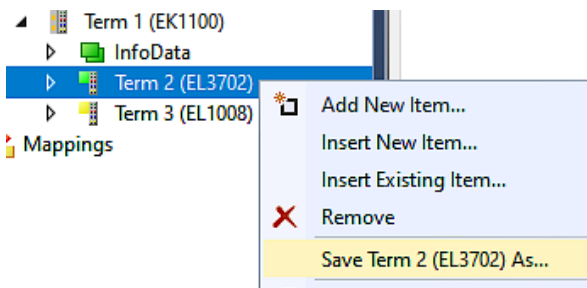




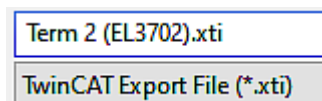
The two methods for exporting and importing the modified terminal referred to above are demonstrated below.

### 6.2.8.2 Procedure within TwinCAT with xti files

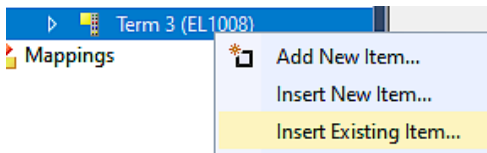
Each IO device can be exported/saved individually:



The xti file can be stored:



and imported again in another TwinCAT system via "Insert Existing item":



### 6.2.8.3 Procedure within and outside TwinCAT with sci file

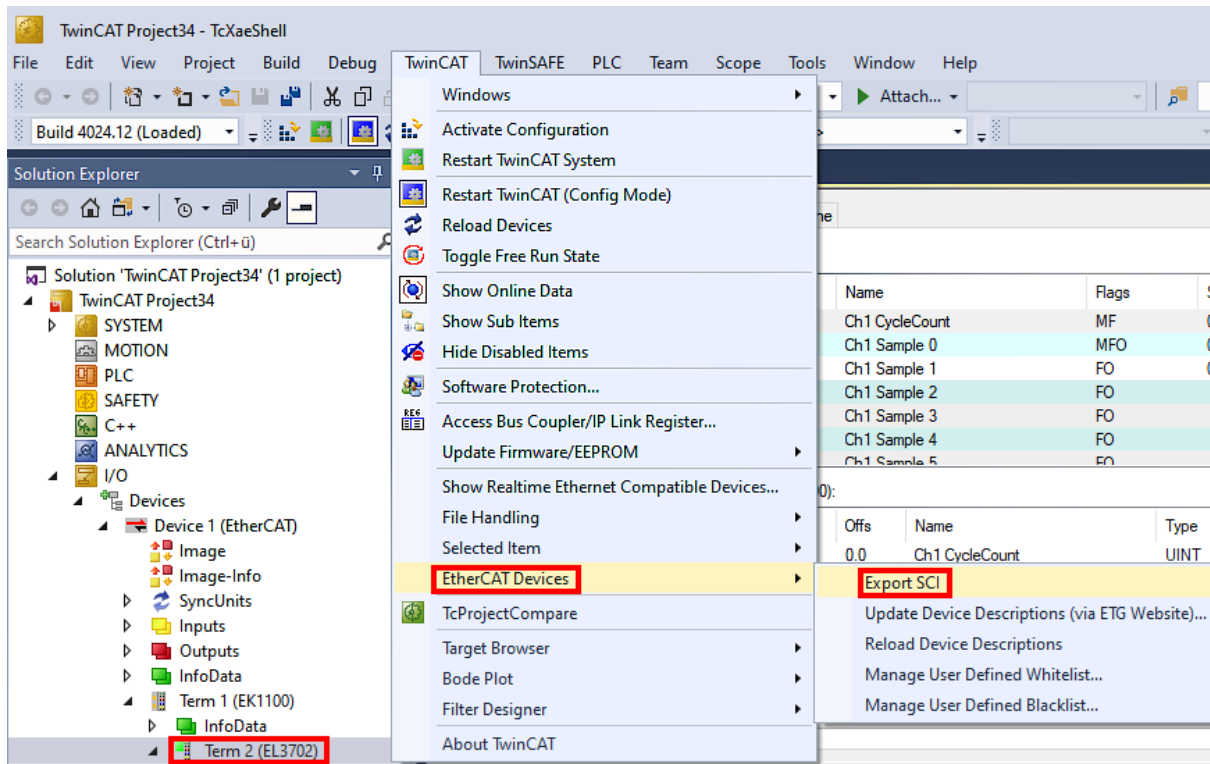
Note regarding availability (2021/01)

The SCI method is available from TwinCAT 3.1 build 4024.14.

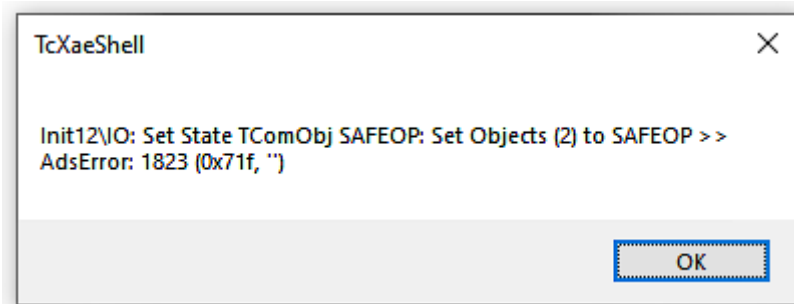
The Slave Configuration Information (SCI) describes a specific complete configuration for an EtherCAT slave (terminal, box, drive...) based on the setting options of the device description file (ESI, EtherCAT Slave Information). That is, it includes PDO, CoE, synchronization.

**Export:**

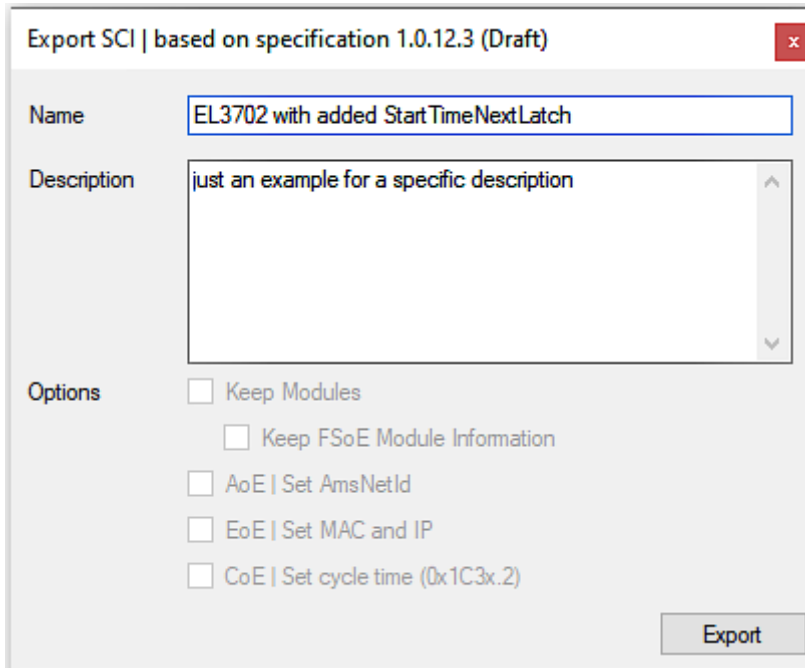
- select a single device via the menu (multiple selection is also possible):  
TwinCAT → EtherCAT Devices → Export SCI.



- If TwinCAT is offline (i.e. if there is no connection to an actual running controller) a warning message may appear, because after executing the function the system attempts to reload the EtherCAT segment. However, in this case this is not relevant for the result and can be acknowledged by clicking OK:



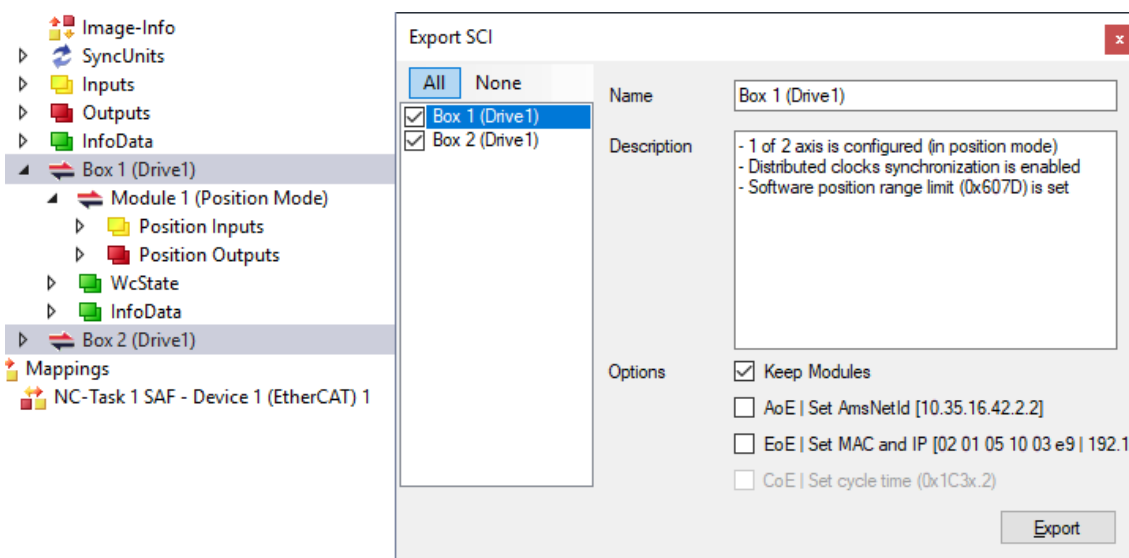
- A description may also be provided:



- Explanation of the dialog box:

Name	Name of the SCI, assigned by the user.	
Description	Description of the slave configuration for the use case, assigned by the user.	
Options	Keep modules	If a slave supports modules/slots, the user can decide whether these are to be exported or whether the module and device data are to be combined during export.
	AoE   Set AmsNetId	The configured AmsNetId is exported. Usually this is network-dependent and cannot always be determined in advance.
	EoE   Set MAC and IP	The configured virtual MAC and IP addresses are stored in the SCI. Usually these are network-dependent and cannot always be determined in advance.
	CoE   Set cycle time(0x1C3x.2)	The configured cycle time is exported. Usually this is network-dependent and cannot always be determined in advance.
ESI	Reference to the original ESI file.	
Export	Save SCI file.	

- A list view is available for multiple selections (*Export multiple SCI files*):

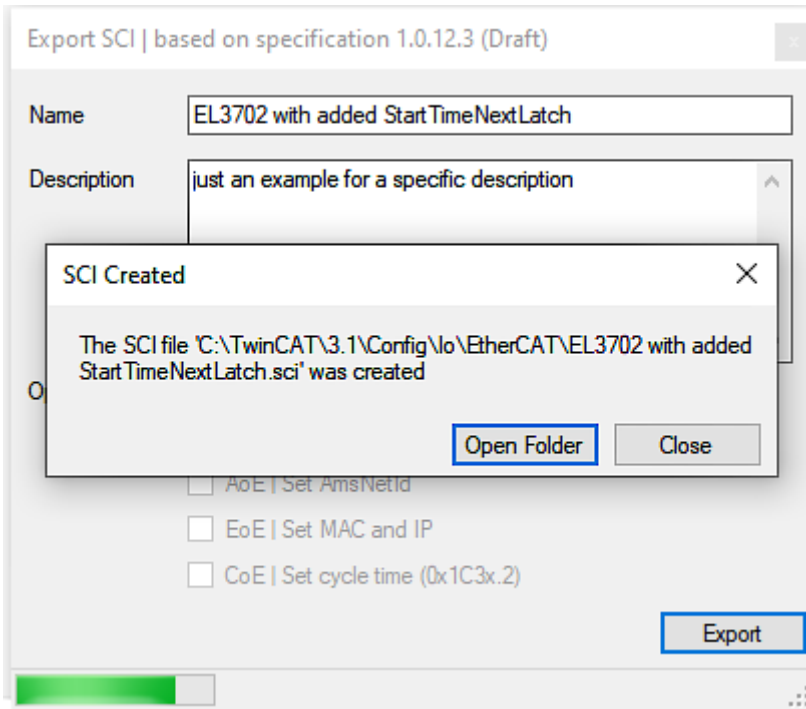


- Selection of the slaves to be exported:
  - All:
    - All slaves are selected for export.

- None:  
All slaves are deselected.
- The sci file can be saved locally:

Dateiname:   
 Dateityp:

- The export takes place:

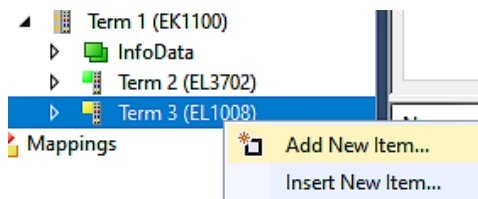


**Import**

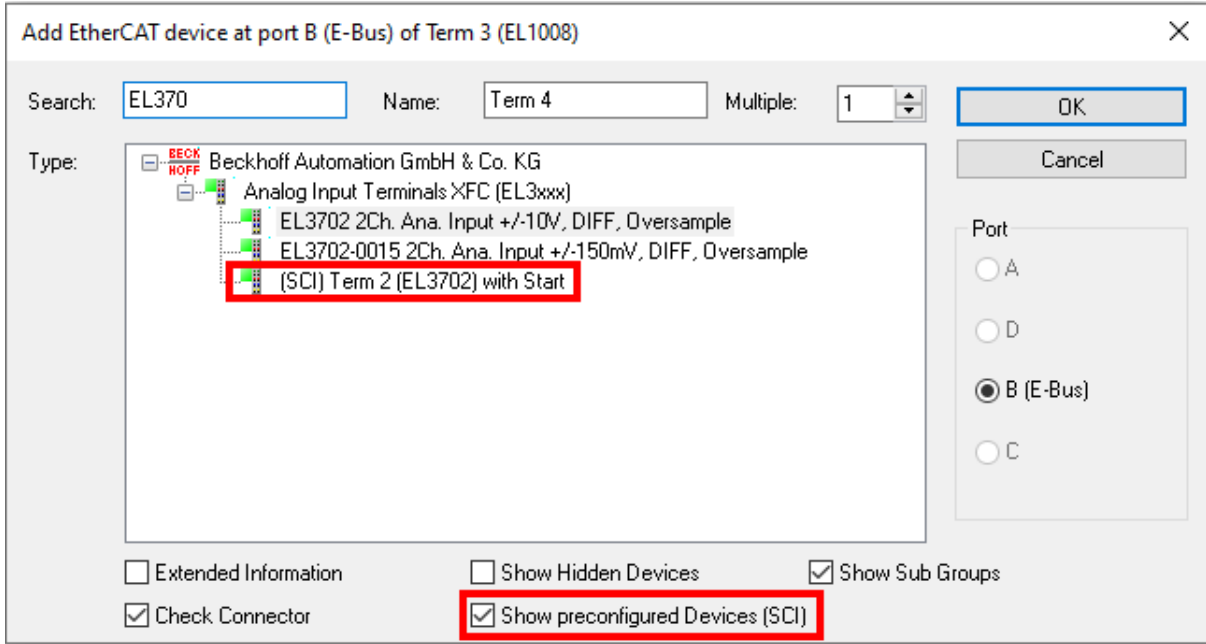
- An sci description can be inserted manually into the TwinCAT configuration like any normal Beckhoff device description.
- The sci file must be located in the TwinCAT ESI path, usually under:  
C:\TwinCAT\3.1\Config\Io\EtherCAT

	EL3702 with added StartTimeNextLatch.sci	11.01.2021 13:29	SCI-Datei	6 KB
--	--	------------------	-----------	------

- Open the selection dialog:

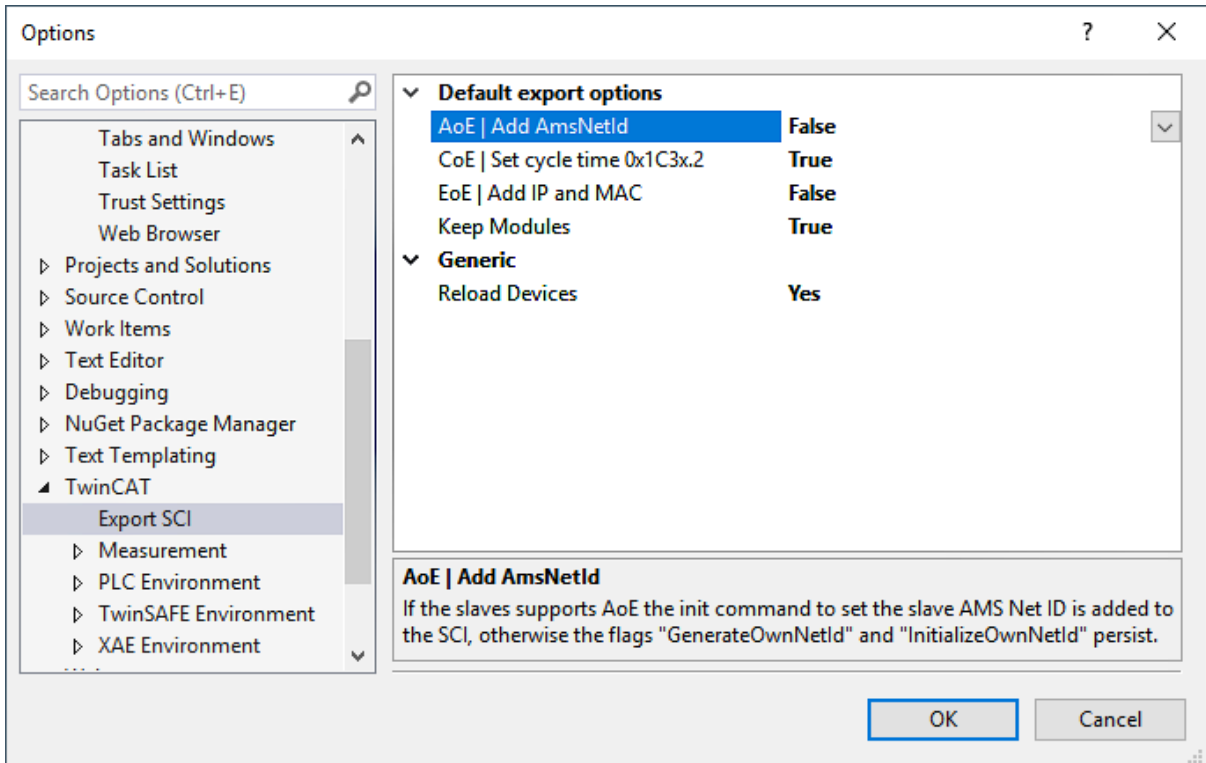


- Display SCI devices and select and insert the desired device:



**Additional Notes**

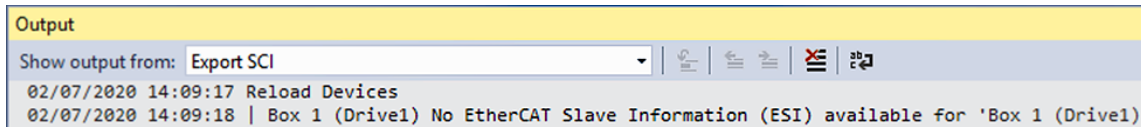
- Settings for the SCI function can be made via the general Options dialog (Tools → Options → TwinCAT → Export SCI):



Explanation of the settings:

Default export options	AoE   Set AmsNetId	Default setting whether the configured AmsNetId is exported.
	CoE   Set cycle time(0x1C3x.2)	Default setting whether the configured cycle time is exported.
	EoE   Set MAC and IP	Default setting whether the configured MAC and IP addresses are exported.
	Keep modules	Default setting whether the modules persist.
Generic	Reload Devices	Setting whether the Reload Devices command is executed before the SCI export. This is strongly recommended to ensure a consistent slave configuration.

SCI error messages are displayed in the TwinCAT logger output window if required:



## 6.3 General Notes - EtherCAT Slave Application

This summary briefly deals with a number of aspects of EtherCAT Slave operation under TwinCAT. More detailed information on this may be found in the corresponding sections of, for instance, the [EtherCAT System Documentation](#).

### Diagnosis in real time: WorkingCounter, EtherCAT State and Status

Generally speaking an EtherCAT Slave provides a variety of diagnostic information that can be used by the controlling task.

This diagnostic information relates to differing levels of communication. It therefore has a variety of sources, and is also updated at various times.

Any application that relies on I/O data from a fieldbus being correct and up to date must make diagnostic access to the corresponding underlying layers. EtherCAT and the TwinCAT System Manager offer comprehensive diagnostic elements of this kind. Those diagnostic elements that are helpful to the controlling task for diagnosis that is accurate for the current cycle when in operation (not during commissioning) are discussed below.

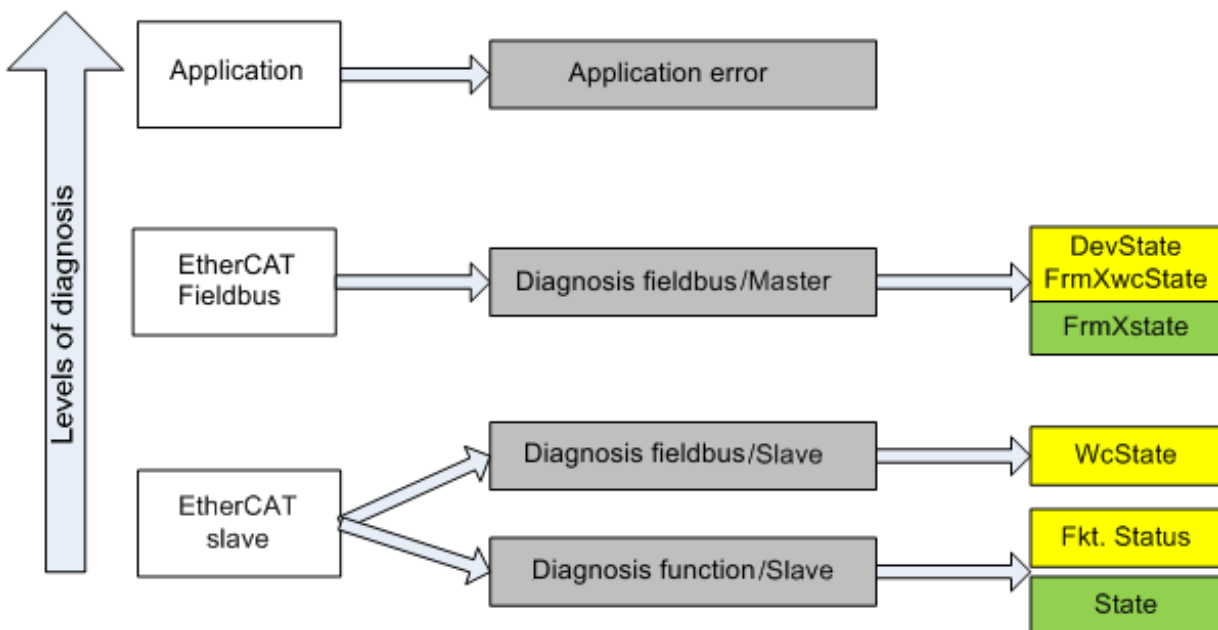


Fig. 111: Selection of the diagnostic information of an EtherCAT Slave

In general, an EtherCAT Slave offers

- communication diagnosis typical for a slave (diagnosis of successful participation in the exchange of process data, and correct operating mode)  
This diagnosis is the same for all slaves.

as well as

- function diagnosis typical for a channel (device-dependent)  
See the corresponding device documentation

The colors in Fig. *Selection of the diagnostic information of an EtherCAT Slave* also correspond to the variable colors in the System Manager, see Fig. *Basic EtherCAT Slave Diagnosis in the PLC*.

Colour	Meaning
yellow	Input variables from the Slave to the EtherCAT Master, updated in every cycle
red	Output variables from the Slave to the EtherCAT Master, updated in every cycle
green	Information variables for the EtherCAT Master that are updated acyclically. This means that it is possible that in any particular cycle they do not represent the latest possible status. It is therefore useful to read such variables through ADS.

Fig. Basic EtherCAT Slave Diagnosis in the PLC shows an example of an implementation of basic EtherCAT Slave Diagnosis. A Beckhoff EL3102 (2-channel analogue input terminal) is used here, as it offers both the communication diagnosis typical of a slave and the functional diagnosis that is specific to a channel. Structures are created as input variables in the PLC, each corresponding to the process image.

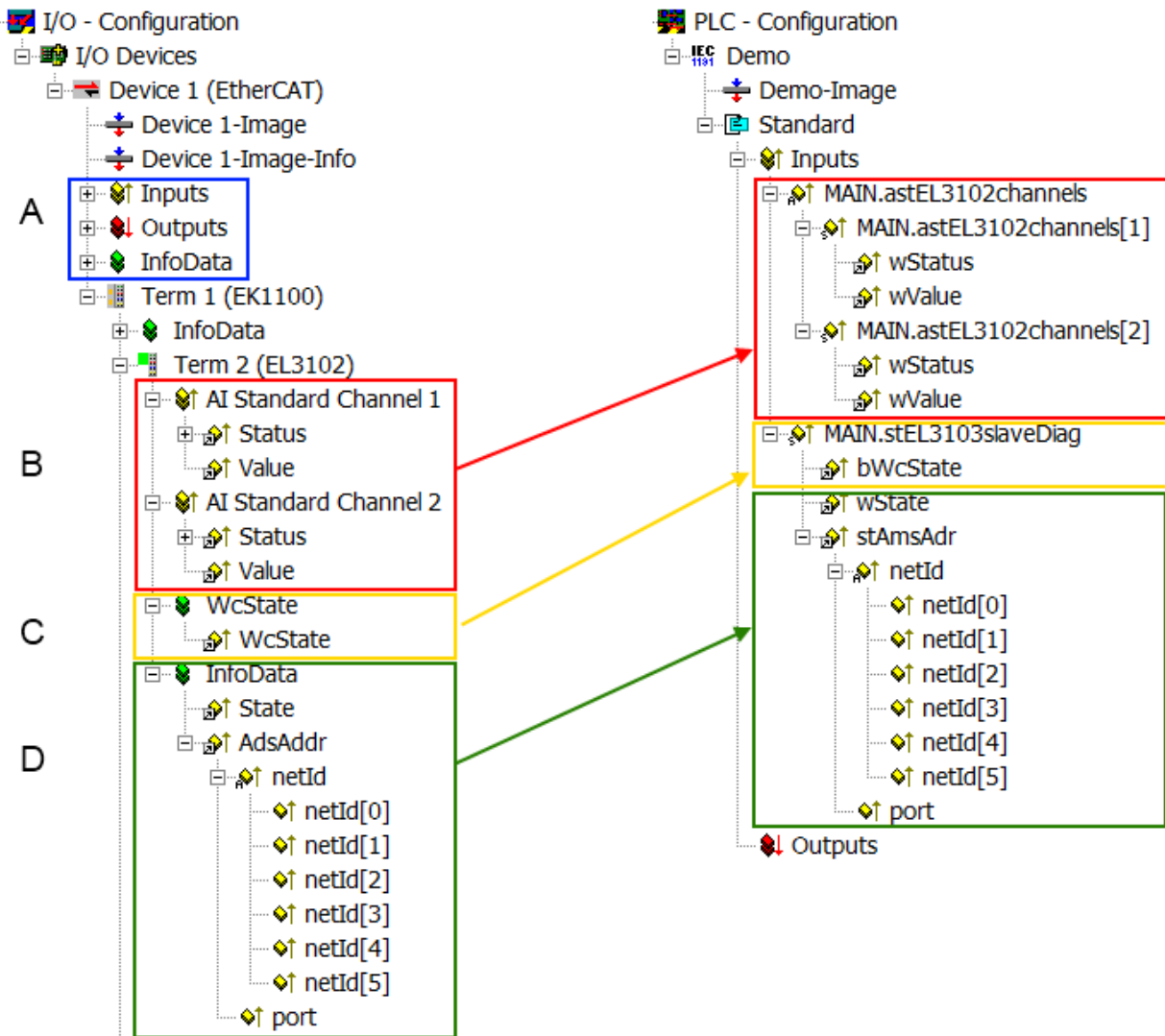


Fig. 112: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:



Code	Function	Implementation	Application/evaluation
A	The EtherCAT Master's diagnostic information updated acyclically (yellow) or provided acyclically (green).		At least the DevState is to be evaluated for the most recent cycle in the PLC. The EtherCAT Master's diagnostic information offers many more possibilities than are treated in the EtherCAT System Documentation. A few keywords: <ul style="list-style-type: none"> <li>• CoE in the Master for communication with/through the Slaves</li> <li>• Functions from <i>TcEtherCAT.lib</i></li> <li>• Perform an OnlineScan</li> </ul>
B	In the example chosen (EL3102) the EL3102 comprises two analogue input channels that transmit a single function status for the most recent cycle.	Status <ul style="list-style-type: none"> <li>• the bit significations may be found in the device documentation</li> <li>• other devices may supply more information, or none that is typical of a slave</li> </ul>	In order for the higher-level PLC task (or corresponding control applications) to be able to rely on correct data, the function status must be evaluated there. Such information is therefore provided with the process data for the most recent cycle.
C	For every EtherCAT Slave that has cyclic process data, the Master displays, using what is known as a WorkingCounter, whether the slave is participating successfully and without error in the cyclic exchange of process data. This important, elementary information is therefore provided for the most recent cycle in the System Manager <ol style="list-style-type: none"> <li>1. at the EtherCAT Slave, and, with identical contents</li> <li>2. as a collective variable at the EtherCAT Master (see Point A) for linking.</li> </ol>	WcState (Working Counter) 0: valid real-time communication in the last cycle 1: invalid real-time communication This may possibly have effects on the process data of other Slaves that are located in the same SyncUnit	In order for the higher-level PLC task (or corresponding control applications) to be able to rely on correct data, the communication status of the EtherCAT Slave must be evaluated there. Such information is therefore provided with the process data for the most recent cycle.
D	Diagnostic information of the EtherCAT Master which, while it is represented at the slave for linking, is actually determined by the Master for the Slave concerned and represented there. This information cannot be characterized as real-time, because it <ul style="list-style-type: none"> <li>• is only rarely/never changed, except when the system starts up</li> <li>• is itself determined acyclically (e.g. EtherCAT Status)</li> </ul>	State current Status (INIT..OP) of the Slave. The Slave must be in OP (=8) when operating normally. <i>AdsAddr</i> The ADS address is useful for communicating from the PLC/task via ADS with the EtherCAT Slave, e.g. for reading/writing to the CoE. The AMS-NetID of a slave corresponds to the AMS-NetID of the EtherCAT Master; communication with the individual Slave is possible via the <i>port</i> (= EtherCAT address).	Information variables for the EtherCAT Master that are updated acyclically. This means that it is possible that in any particular cycle they do not represent the latest possible status. It is therefore possible to read such variables through ADS.

## NOTICE

### Diagnostic information

It is strongly recommended that the diagnostic information made available is evaluated so that the application can react accordingly.

### CoE Parameter Directory

The CoE parameter directory (CanOpen-over-EtherCAT) is used to manage the set values for the slave concerned. Changes may, in some circumstances, have to be made here when commissioning a relatively complex EtherCAT Slave. It can be accessed through the TwinCAT System Manager, see Fig. *EL3102, CoE directory*:



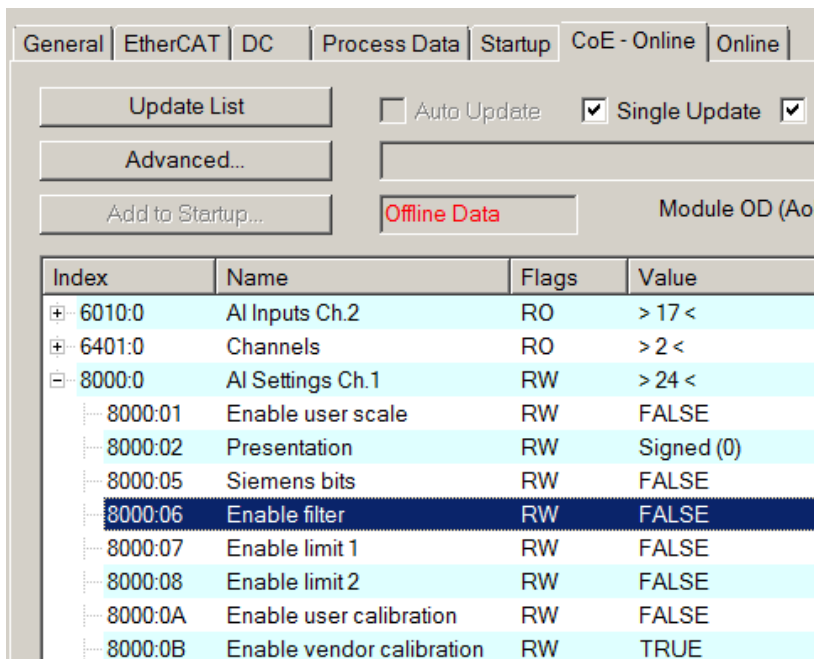


Fig. 113: EL3102, CoE directory

**i EtherCAT System Documentation**

The comprehensive description in the [EtherCAT System Documentation](#) (EtherCAT Basics --> CoE Interface) must be observed!

A few brief extracts:

- Whether changes in the online directory are saved locally in the slave depends on the device. EL terminals (except the EL66xx) are able to save in this way.
- The user must manage the changes to the StartUp list.

**Commissioning aid in the TwinCAT System Manager**

Commissioning interfaces are being introduced as part of an ongoing process for EL/EP EtherCAT devices. These are available in TwinCAT System Managers from TwinCAT 2.11R2 and above. They are integrated into the System Manager through appropriately extended ESI configuration files.

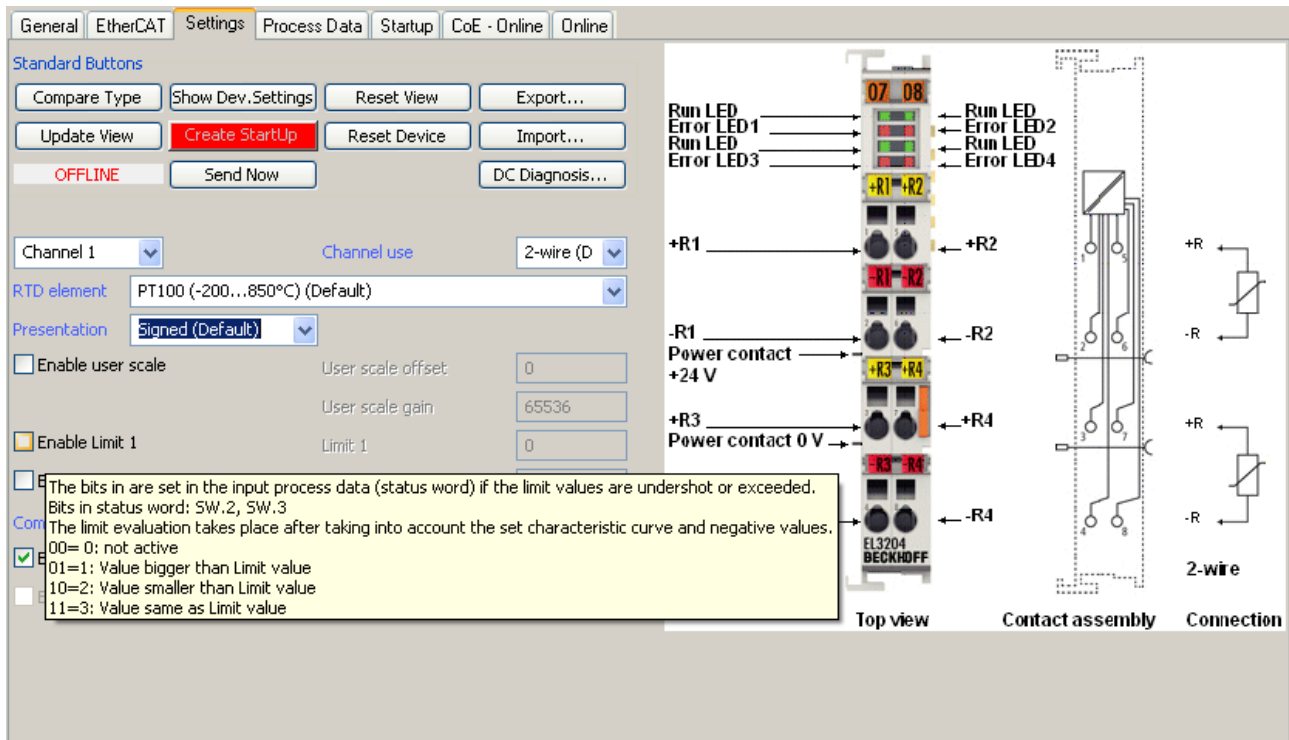


Fig. 114: Example of commissioning aid for a EL3204

This commissioning process simultaneously manages

- CoE Parameter Directory
- DC/FreeRun mode
- the available process data records (PDO)

Although the “Process Data”, “DC”, “Startup” and “CoE-Online” that used to be necessary for this are still displayed, it is recommended that, if the commissioning aid is used, the automatically generated settings are not changed by it.

The commissioning tool does not cover every possible application of an EL/EP device. If the available setting options are not adequate, the user can make the DC, PDO and CoE settings manually, as in the past.

**EtherCAT State: automatic default behaviour of the TwinCAT System Manager and manual operation**

After the operating power is switched on, an EtherCAT Slave must go through the following statuses

- INIT
- PREOP
- SAFEOP
- OP

to ensure sound operation. The EtherCAT Master directs these statuses in accordance with the initialization routines that are defined for commissioning the device by the ES/XML and user settings (Distributed Clocks (DC), PDO, CoE). See also the section on "Principles of [Communication, EtherCAT State Machine \[► 49\]](#)" in this connection. Depending how much configuration has to be done, and on the overall communication, booting can take up to a few seconds.

The EtherCAT Master itself must go through these routines when starting, until it has reached at least the OP target state.

The target state wanted by the user, and which is brought about automatically at start-up by TwinCAT, can be set in the System Manager. As soon as TwinCAT reaches the status RUN, the TwinCAT EtherCAT Master will approach the target states.

**Standard setting**

The advanced settings of the EtherCAT Master are set as standard:

- EtherCAT Master: OP
- Slaves: OP  
This setting applies equally to all Slaves.

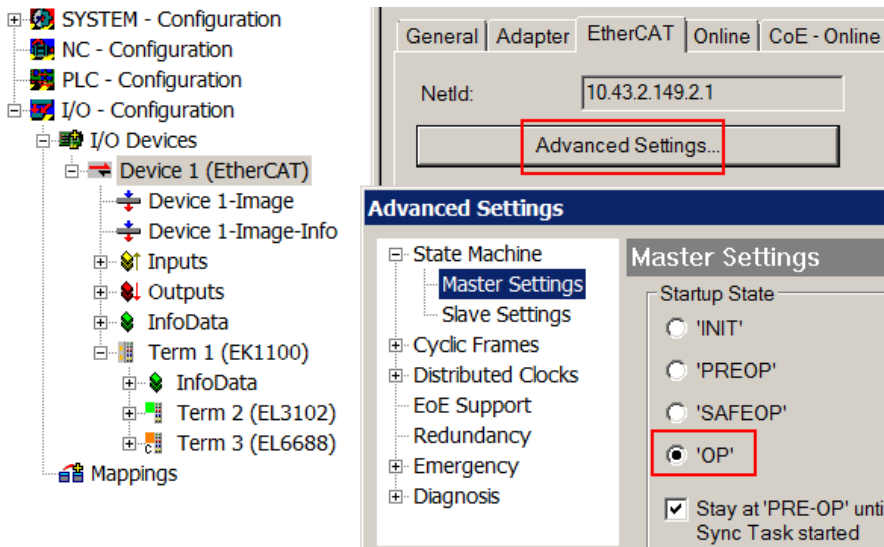


Fig. 115: Default behaviour of the System Manager

In addition, the target state of any particular Slave can be set in the “Advanced Settings” dialogue; the standard setting is again OP.

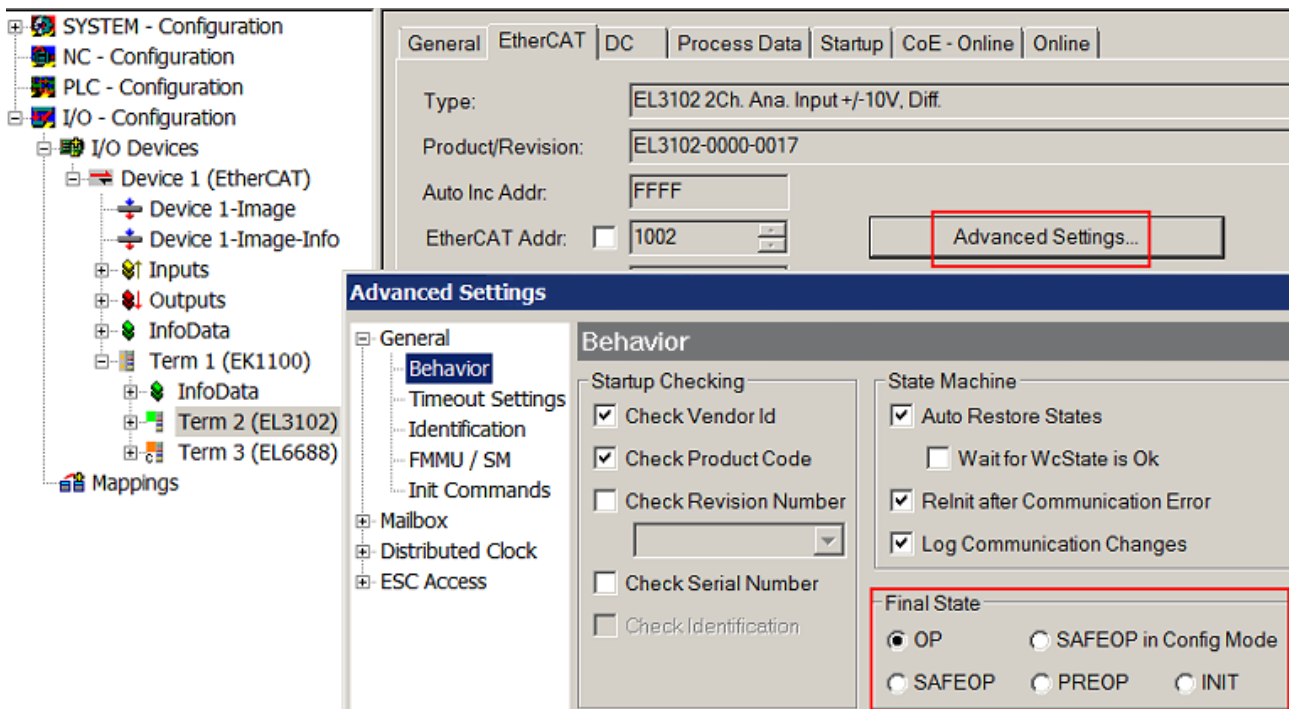


Fig. 116: Default target state in the Slave

**Manual Control**

There are particular reasons why it may be appropriate to control the states from the application/task/PLC. For instance:

- for diagnostic reasons
- to induce a controlled restart of axes

- because a change in the times involved in starting is desirable

In that case it is appropriate in the PLC application to use the PLC function blocks from the *TcEtherCAT.lib*, which is available as standard, and to work through the states in a controlled manner using, for instance, *FB\_EcSetMasterState*.

It is then useful to put the settings in the EtherCAT Master to INIT for master and slave.

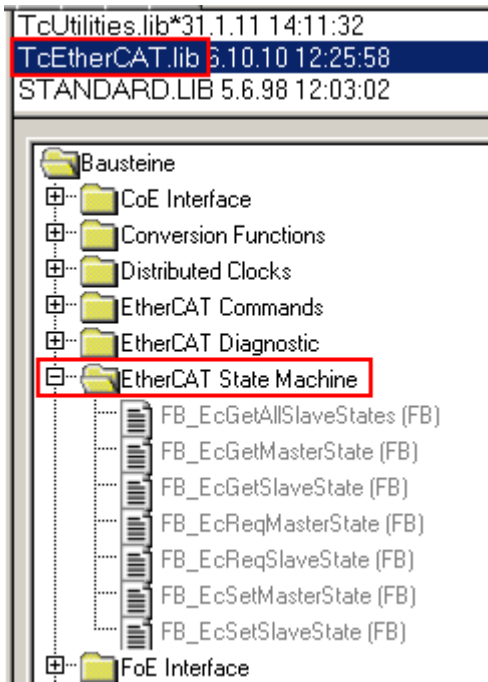


Fig. 117: PLC function blocks

## 6.4 First steps Power supplies with EtherCAT

1. Install the power supply as described in the chapter "[Technical data, mounting, wiring](#) [[▶ 17](#)]" and connect it to the controller.
2. Install the TwinCAT automation software and familiarize yourself with the program.
3. Integrate the power supply into your TwinCAT project as described in the chapter "[TwinCAT Quickstart](#)". Further information on the software and communication via EtherCAT can be found in the chapters:
  - ⇒ "[TwinCAT development environment](#)" [[▶ 70](#)] and
  - ⇒ "[General commissioning instructions for the EtherCAT slave](#)" [[▶ 108](#)].
4. Configure the power supply in TwinCAT.
  - ⇒ Select the process data as described in the chapter "[Switching the process data sets](#) [[▶ 139](#)]".
  - ⇒ Create a PLC project as described in the chapter "[Programming the PLC](#)" [[▶ 62](#)].
  - ⇒ Link the process data with PLC variables (see chapter "[Linking the process data](#)") [[▶ 116](#)].
5. Select additional info data (see chapter "[Selecting info data](#)") [[▶ 118](#)]
6. Optional: application-specific settings via the CoE directory
  - ⇒ Set warning thresholds for the warning bit (see chapter "[Setting the warning thresholds for the warning bit](#)") [[▶ 121](#)]
  - ⇒ Reset counters and maximum/minimum values via the CoE directory (see chapter "[Resetting counters and maximum/minimum values](#)") [[▶ 122](#)].
  - ⇒ For parallel use, observe the notes on [parallel use](#) [[▶ 135](#)].
7. Monitor the output voltage using the [DC-OK function](#) [[▶ 131](#)].
8. Errors and warning messages
  - ⇒ Read error and warning messages via the Diag History (see chapter "[Displaying errors and warning messages](#)") [[▶ 124](#)].
  - ⇒ Use the WCState bit to check whether errors have occurred during EtherCAT transmission (see chapter "[Detecting invalid data](#)") [[▶ 134](#)].

### NOTICE



#### Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE - Online tab (with a double click on the respective object) or via the Process Data tab (assignment of PDOs). A detailed description can be found in the EtherCAT System-Documentation in chapter "[EtherCAT subscriber configuration](#)"

Please note the general CoE notes in the EtherCAT System Documentation in chapter "[CoE-interface](#)" when using/manipulating the CoE parameters:

- Keep a startup list if components have to be replaced
- Differentiation between online/offline dictionary,
- existence of current XML description
- use "CoE reload" for resetting changes

## **i** Process data of the PS2001-xxxx-1001 power supply

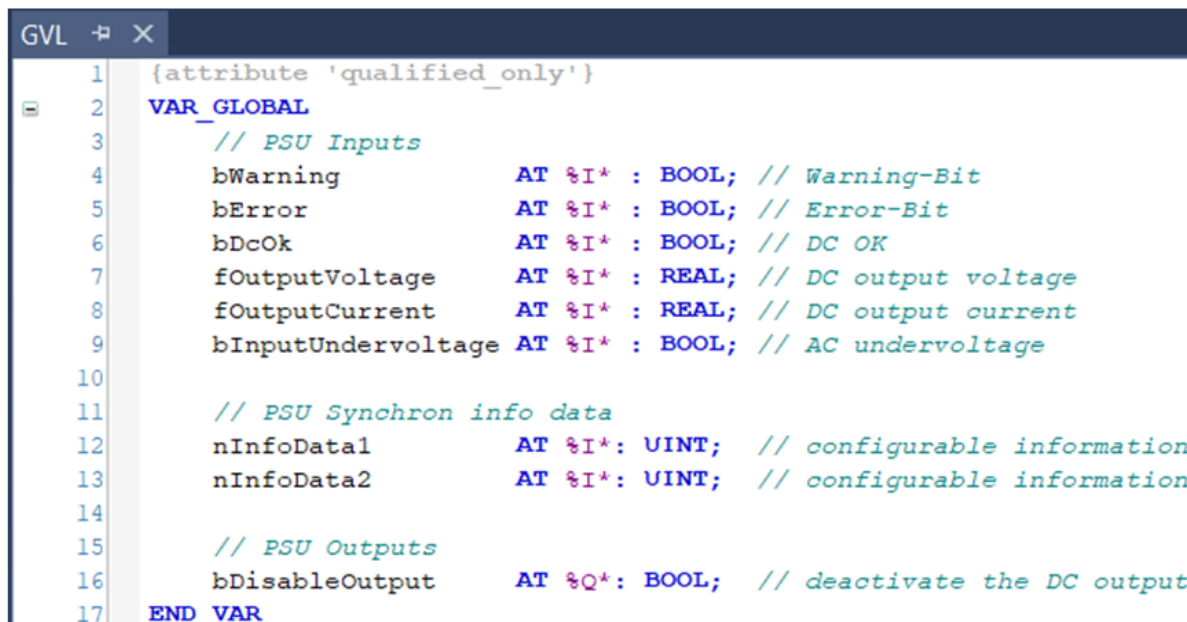
The process data objects (PDO) are cyclically transmitted process values. They are enabled via the "Process data" tab and then displayed in the TwinCAT tree. This allows them to be linked directly to variables from the PLC program.

- By default, the following process data is recorded for the PS2001-xxx-1001 power supplies and updated and transmitted by the device every 2 ms:
  - ⇒ Input undervoltage
  - ⇒ Output voltage
  - ⇒ Output current
  - ⇒ "DC-OK" bit
  - ⇒ "Warning" and "Error" bit
- After activating PDO 0x1600 "PSU Outputs", the power supply output can be disabled via the "Disable output" bit (see chapter "Switching output via EtherCAT [▶ 132]").
- Additional voltage, current and temperature information can also be transmitted as process data via the freely configurable information data "Info data 1" and "Info data 2". Further information can be found in chapter [Selecting the info data \[▶ 118\]](#).

### 6.4.1 Linking of process data

Proceed as follows to link the process data with PLC variables:

- ✓ The process data to be linked has been enabled in the "Process data [▶ 139]" tab.
1. Create corresponding PLC variables for the process data in your PLC project, e.g. in a Global Variable List (GVL), and define the data types as described in the following figure.

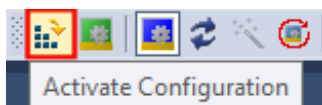


```

GVL  + X
1  {attribute 'qualified_only'}
2  VAR_GLOBAL
3  // PSU Inputs
4  bWarning          AT %I* : BOOL; // Warning-Bit
5  bError            AT %I* : BOOL; // Error-Bit
6  bDcOk             AT %I* : BOOL; // DC OK
7  fOutputVoltage    AT %I* : REAL; // DC output voltage
8  fOutputCurrent    AT %I* : REAL; // DC output current
9  bInputUndervoltage AT %I* : BOOL; // AC undervoltage
10
11 // PSU Synchron info data
12 nInfoData1        AT %I* : UINT; // configurable information
13 nInfoData2        AT %I* : UINT; // configurable information
14
15 // PSU Outputs
16 bDisableOutput    AT %Q* : BOOL; // deactivate the DC output
17 END_VAR
  
```

Fig. 118: Definition of PLC variables in a Global Variable List (GLV)

2. Activate the configuration using the "Activate Configuration" button



3. Now link the process data of the power supply to the variables.
4. Right-click on the object to be linked.

5. Select "Change Link" from the selection list. The "Attach Variable" dialog box opens.

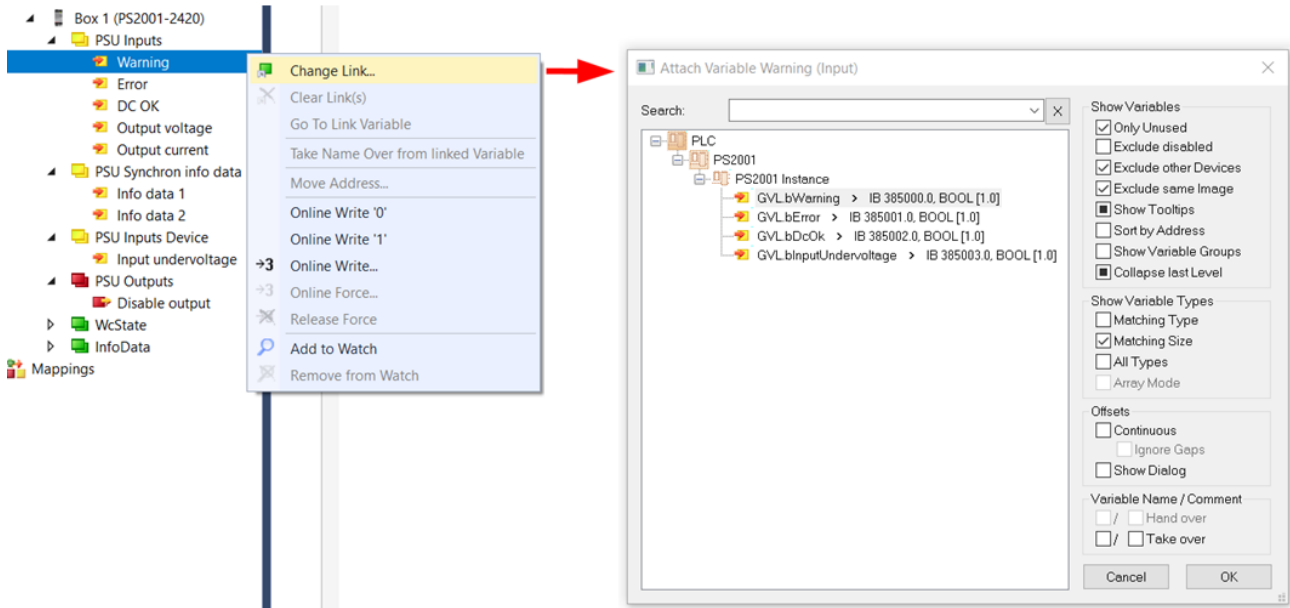



Fig. 119: Linking the warning bit from the process data with the PLC variable bWarning

6. Select the appropriate variable and confirm with "OK".

⇒ The successful linking of the variable is indicated in the TwinCAT tree by an arrow symbol  .



## 6.4.2 Selecting the info data

### ● Updating the info data objects

**i** In addition to the process data transmitted by default, some variables from the "PSU Diag Data" objects can be displayed via the two info data objects "Info data 1" "0x6000:15" and "Info data 2" and then linked to PLC variables.

Note that although the info data is transmitted on every process data cycle, it typically contains new values only every 52 ms.

The info data is selected in two steps:

1. In the "Process data" tab of the power supply, select the Predefined PDO Assignment "Standard + Synchron info data".

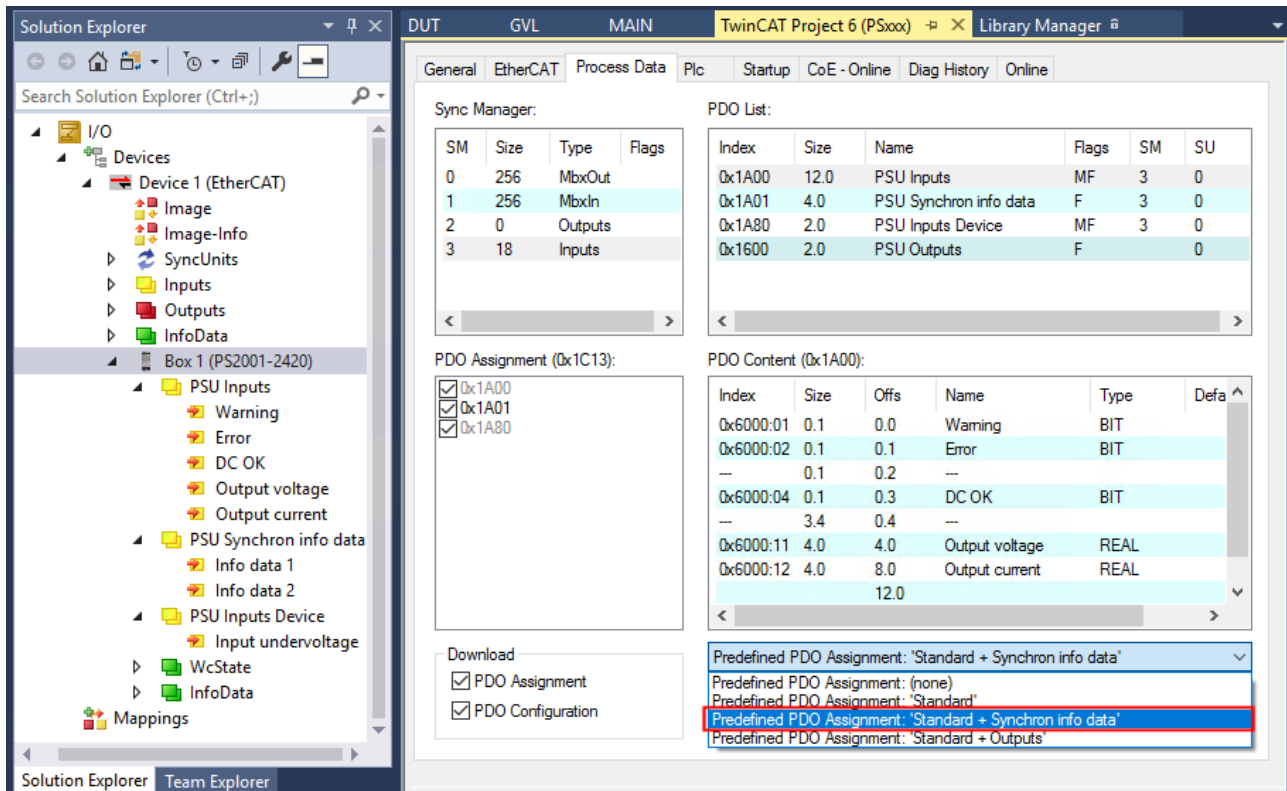


Fig. 120: Selection of process data set "Standard + Synchron info data" in the Process Data tab using the PS2001-2420-1001 as an example

2. Select the info data via the CoE-Online tab under object 0x8001 [▶ 143] "PSU Features".
  - ⇒ Double-click on the object 0x8001:19 "Select Info data 1" or 0x8001:21 "Select Info data 2"
  - ⇒ The "Set Value Dialog" dialog box opens
  - ⇒ Select the desired entry from the drop-down menu under "Enum" (see following tables).  
The following sizes from "PSU Diag Data" can be selected
  - ⇒ Close the dialog box with "OK"



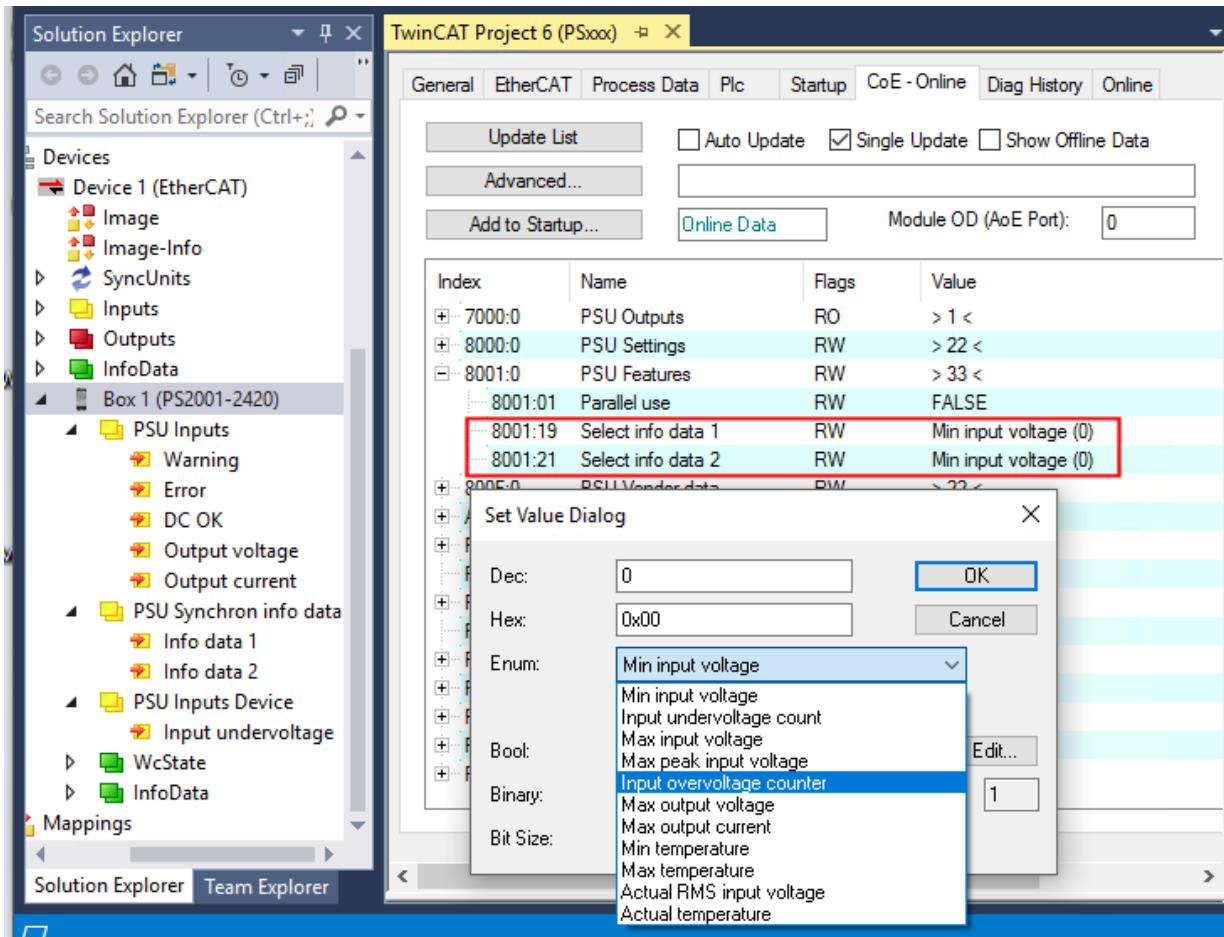


Fig. 121: Select info data under "PSU Features" using PS2001-2420-1001 as an example

The following entries of the "PSU Diag Data" can be selected in the objects 0x8001:19 and 0x8001:21:

Information on the input side	
Enum (Dec) in index 0x8001:19 or 0x8001:21	Index "PSU Diag data"
Min input voltage (0)	0xFA14:11
Input undervoltage count (1)	0xFA14:12
Max input voltage (2)	0xFA14:13
Max peak input voltage (4)	0xFA14:14
Input overvoltage counter (5)	0xFA14:16
Actual RMS input voltage (10)	0xFA14:19

Information on the output side	
Enum (Dec) in index 0x8001:19 or 0x8001:21	Index "PSU Diag data"
Max output voltage (6)	0xA000:11
Max output current (7)	0xA000:12

Temperature information	
Enum (Dec) in index 0x8001:19 or 0x8001:21	Index "PSU Diag data"
Min temperature (8)	0xFA14:17
Max temperature (9)	0xFA14:18
Actual temperature (14)	0xFA14:1B

**i Differentiation between counters, current values and maximum/minimum values**

The entries of the "PSU Diag Data" are divided into values, counters and maximum/minimum values.

Objects that contain counters or maximum/minimum values can be reset if required (see chapter [Resetting counters and maximum/minimum values \[▶ 122\]](#)).

**NOTICE****Parameterization via the CoE list (CAN over EtherCAT)**

The EtherCAT device is parameterized via the CoE - Online tab (with a double click on the respective object) or via the Process Data tab (assignment of PDOs). A detailed description can be found in the EtherCAT System-Documentation in chapter "[EtherCAT subscriber configuration](#)"

Please note the general CoE notes in the EtherCAT System Documentation in chapter "[CoE-interface](#)" when using/manipulating the CoE parameters:

- Keep a startup list if components have to be replaced
- Differentiation between online/offline dictionary,
- existence of current XML description
- use "CoE reload" for resetting changes

**Linking with the PLC variables**

Note that you must create UINT variables for the info data and link the info data with the corresponding PLC variables (see chapter "[Linking of process data \[▶ 116\]](#)").

**Using info data for network and system monitoring**

With the variables recorded as process data by default and the additional info data, both the operating state of the system and network monitoring can be realized on the input side. Details on the data recorded on the input side can be found in the chapter "[Information on input-side grid monitoring \[▶ 136\]](#)".

### 6.4.3 Setting the warning thresholds for the warning bit

#### Adjustable warnings

The adjustable warning thresholds can be used to set application-specific limits above which an overvoltage, overcurrent or too low voltage is to be detected at the output and a warning of too high a temperature in the device is to be issued. The warnings can thus be individually adapted to the system.

While the warning bit (0x6000:01 [▶ 142]) only indicates the warning state, the detailed warning messages can be read in the TwinCAT EventLogger and the Diag History.

Please note that if the absolute, device-specific overvoltage, overcurrent and overtemperature limits are exceeded, the power supply switches to error mode (see 0x6000:02 [▶ 142] "Error").

The warning thresholds are set in the CoE Online tab under 0x8000 [▶ 143] and 0xF814 [▶ 144] "PSU Settings":

Index	Name
0x8000:14	Overvoltage warn threshold
0x8000:15	Overcurrent warn threshold
0x8000:16	DC OK threshold
0xF814:11	Overtemperature warn threshold

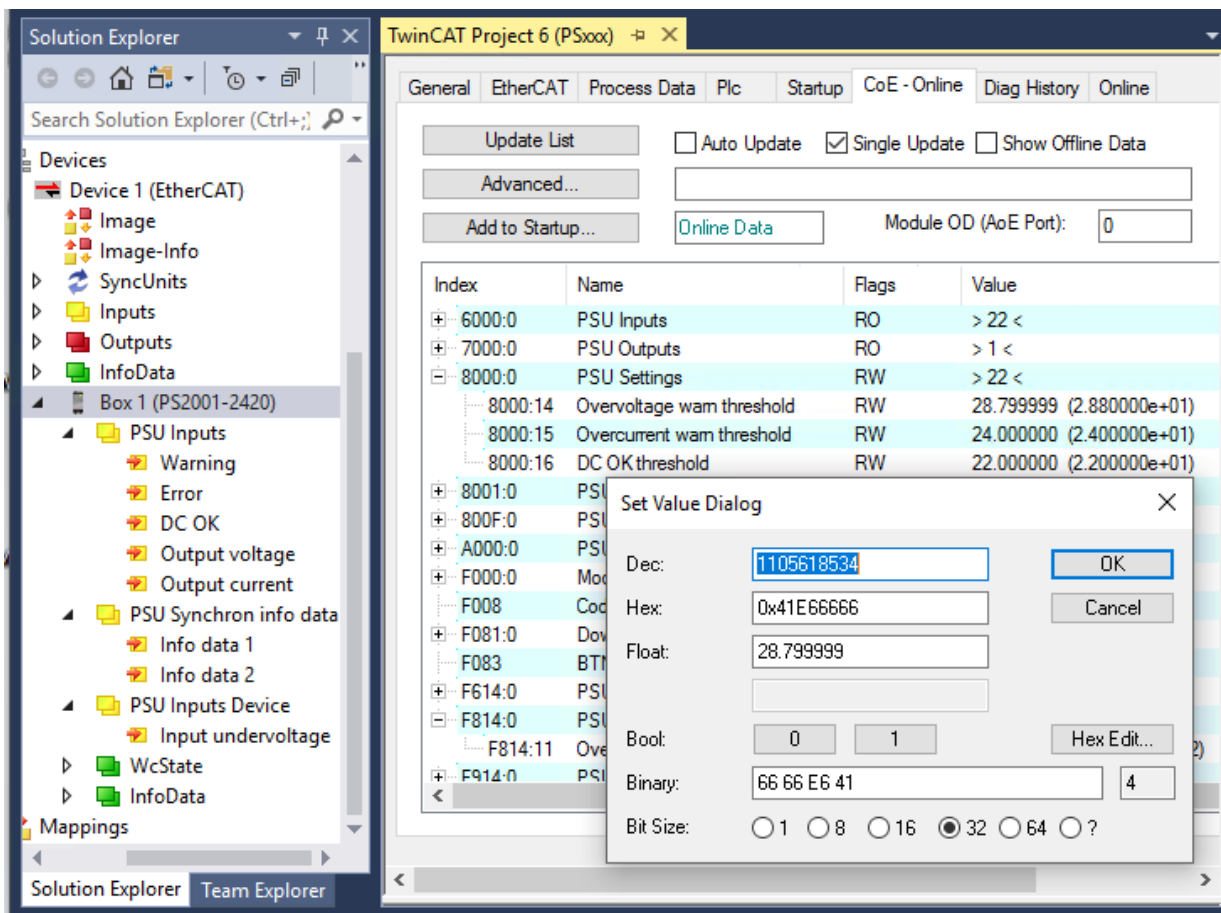


Fig. 122: Set values for warning thresholds e.g. 0x8000:14 "Overvoltage warn threshold" using the PS2001-2420-1001 as an example

1. Double-click on the corresponding warning threshold. The "Set Value Dialog" dialog box opens.
2. Enter the corresponding threshold value under "Float" and close the dialog box with "OK"

## 6.4.4 Resetting counters and maximum/minimum values

### Resetting individual counters and maximum/minimum values

Resetting individual counters and maximum/minimum values is possible in the "CoE-Online" tab under the two "PSU Diag Data" objects (0xA000, 0xFA14)

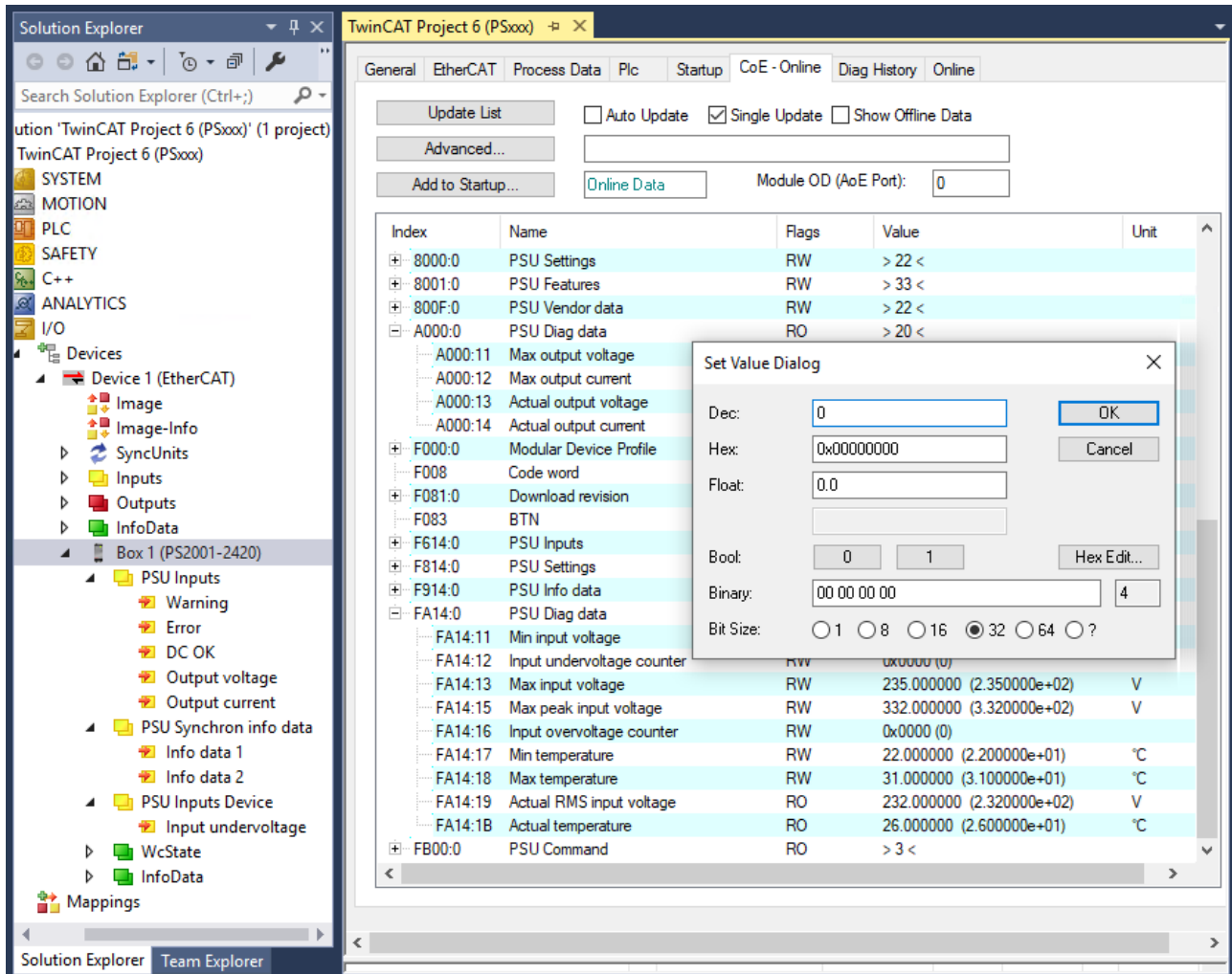


Fig. 123: Reset e.g. 0xA000:11 "Max output voltage" using the PS2001-2420-1001 as an example

1. Double-click on the counter or maximum/minimum value that you want to reset. The "Set Value Dialog" dialog box opens
2. With a write access (e.g. with the value 0), which you confirm with "OK",
  - ⇒ the corresponding counter is reset to 0,
  - ⇒ all other values have the current value after resetting.

### **i** No writing of 0xA000, 0xFA14 with certain values

Only resetting the values is possible via write access.

It is not possible to write specific values to the counters or maximum/minimum values.

- Regardless of the value entered, due to the write access
  - ⇒ the value of counters is always reset to zero,
  - ⇒ all other values have the current value after resetting.

### Resetting all counters and maximum/minimum values

Resetting all counters to 0 and minimum/maximum values to the current values is possible in the "CoE-Online" tab via the "PSU Command" object (0xFB00):

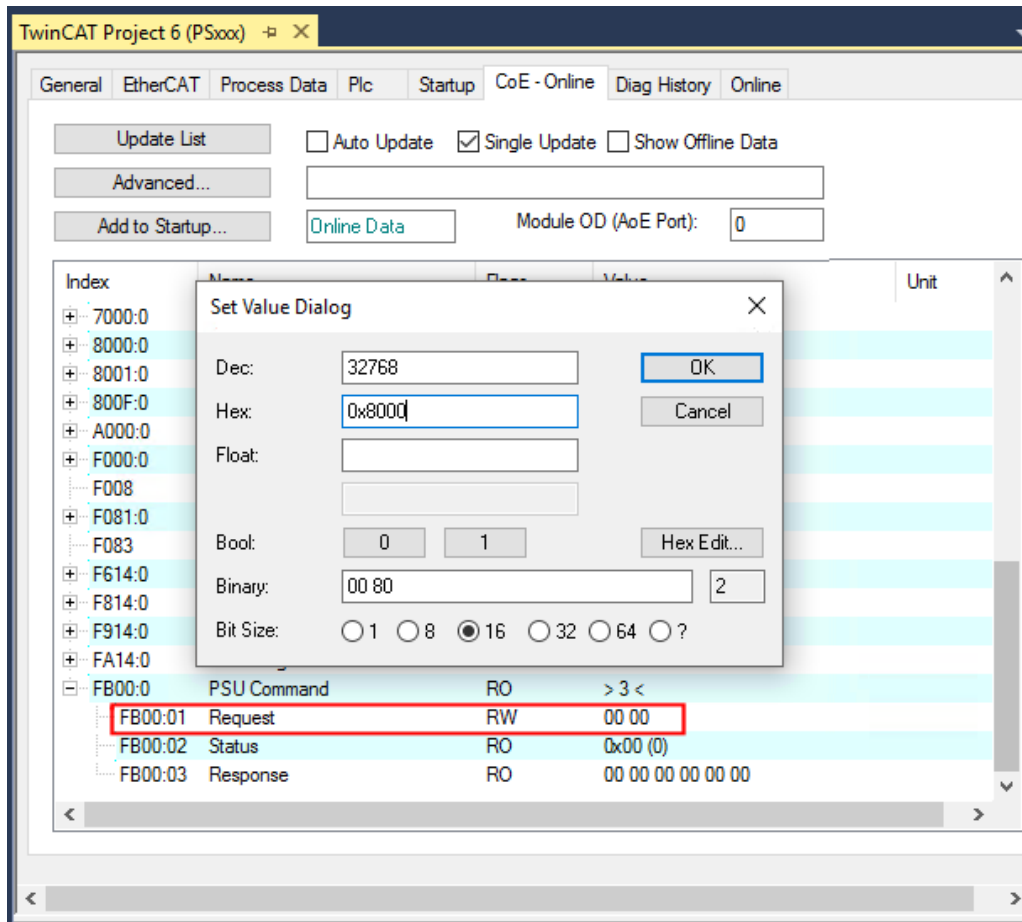


Fig. 124: Reset all counters and maximum/minimum values via 0xFB00:01

1. Double-click on the object 0xFB00:01. The "Set Value Dialog" dialog box opens
2. Enter the desired value under "Hex". The following values can be specified:
  - ⇒ 0x7001: Reset all min/max values in the objects 0xA000 and 0xFA14
  - ⇒ 0x7002: Delete exceeded threshold values from the JSON log
3. Confirm with "OK".
  - ⇒ In the "PSU Diag Data", all counters are set to zero, maximum/minimum values have the current values or
  - ⇒ Threshold value violations are deleted from the JSON log.

### **i** Reading and writing CoE values via function blocks

As an alternative to manually reading the CoE data and writing via the "Set Value Dialog" window, the CoE data of the power supply can also be read and written via corresponding function blocks from the PLC program.

This facilitates, for example, the automatic reading of certain information from the "PSU Diag Data" and the associated resetting of counters and minimum/maximum values (see chapter "Reading and writing CoE objects using function blocks") [▶ 125].

## 6.4.5 Display of errors and warning messages

### Diag History

If a warning or error state is present and the warning or error bit has been set, the detailed warning or error messages can be read in the "Diag History" tab of the power supply (see also chapter "[Complete Access to PSU Diag data](#)") [▶ 129]:

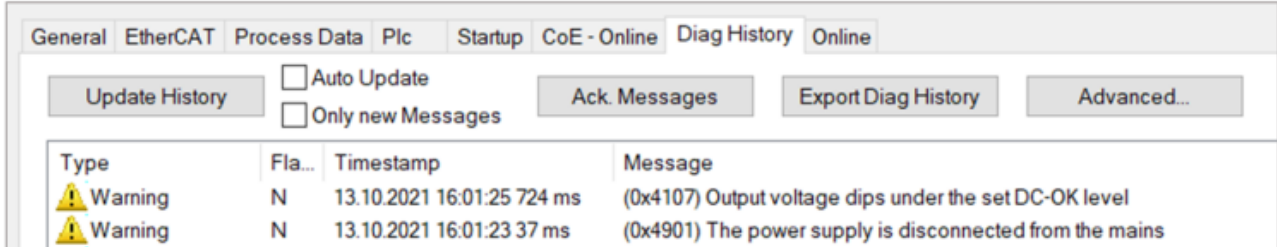


Fig. 125: Diag History tab with warning messages

### TwinCAT EventLogger

Alternatively, warnings and error messages are also displayed in the TwinCAT EventLogger.

Open the TwinCAT EventLogger via the "View" menu and select "Other Windows" -> TwinCAT Logged Events.

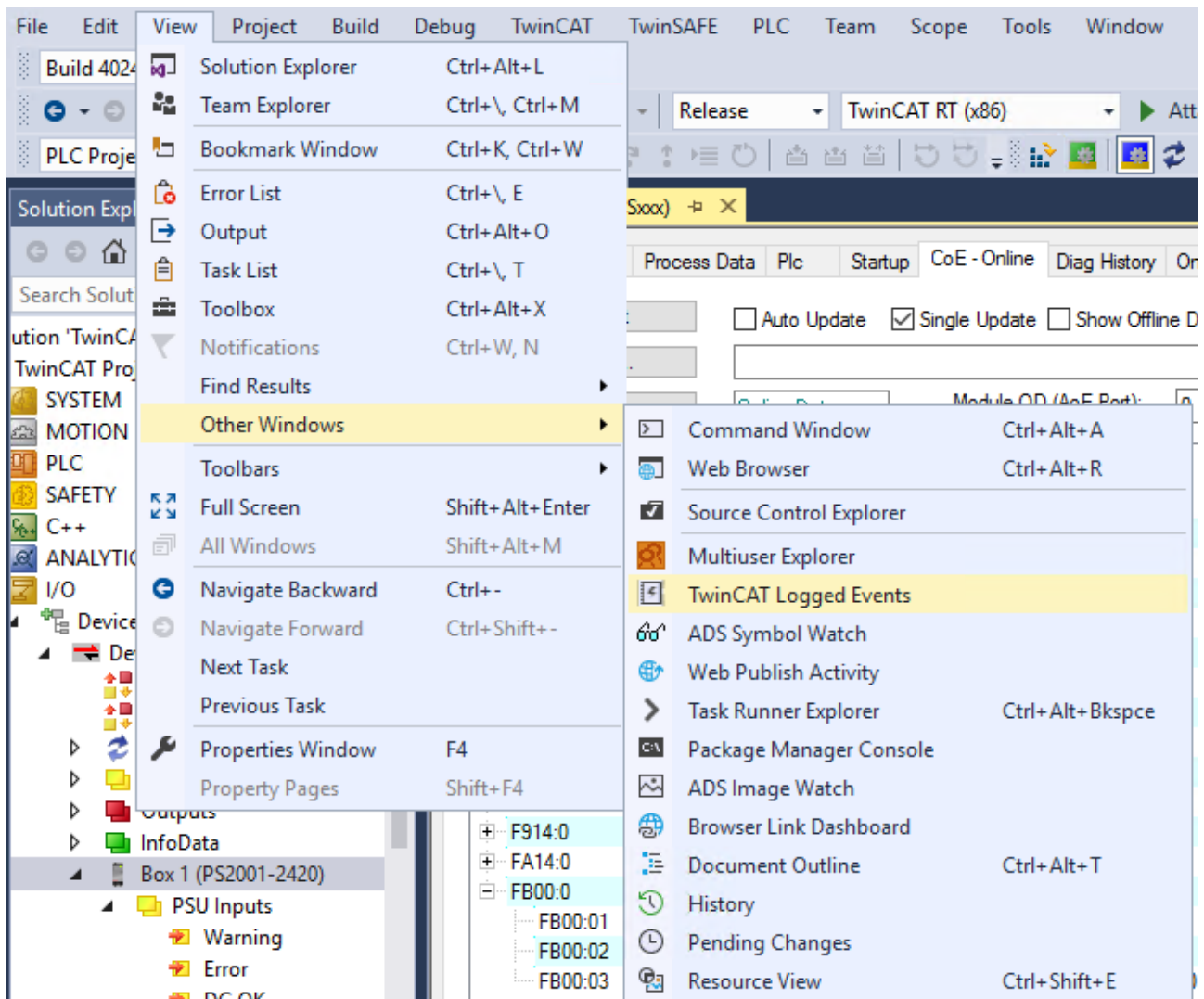
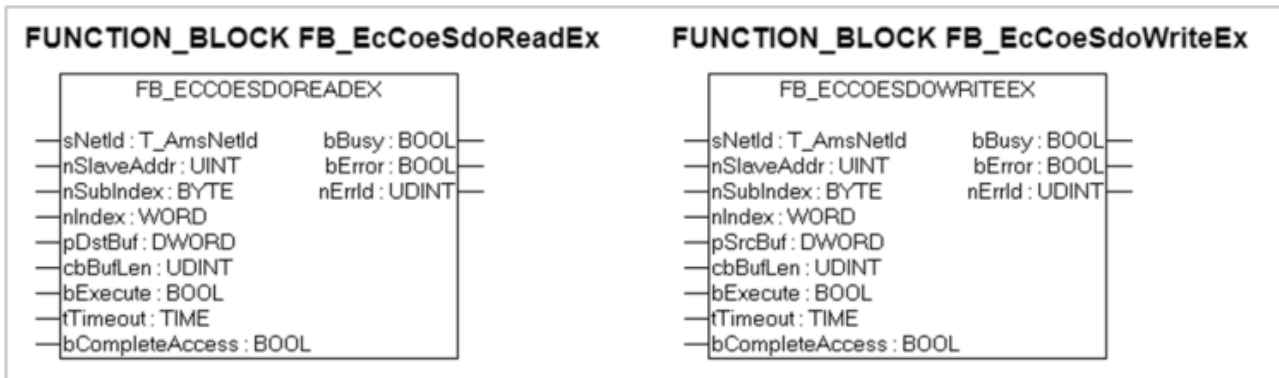


Fig. 126: Open TwinCAT EventLogger



### 6.4.6 Writing and reading CoE objects using function blocks

The function blocks "FB\_EcCoeSdoReadEx" and "FB\_EcCoeSdoWriteEx" from the Tc2\_EtherCAT library can be used to read and write CoE values in the PLC project.



To use the function blocks, proceed as follows:

1. Add the "Tc2\_EtherCAT" library:

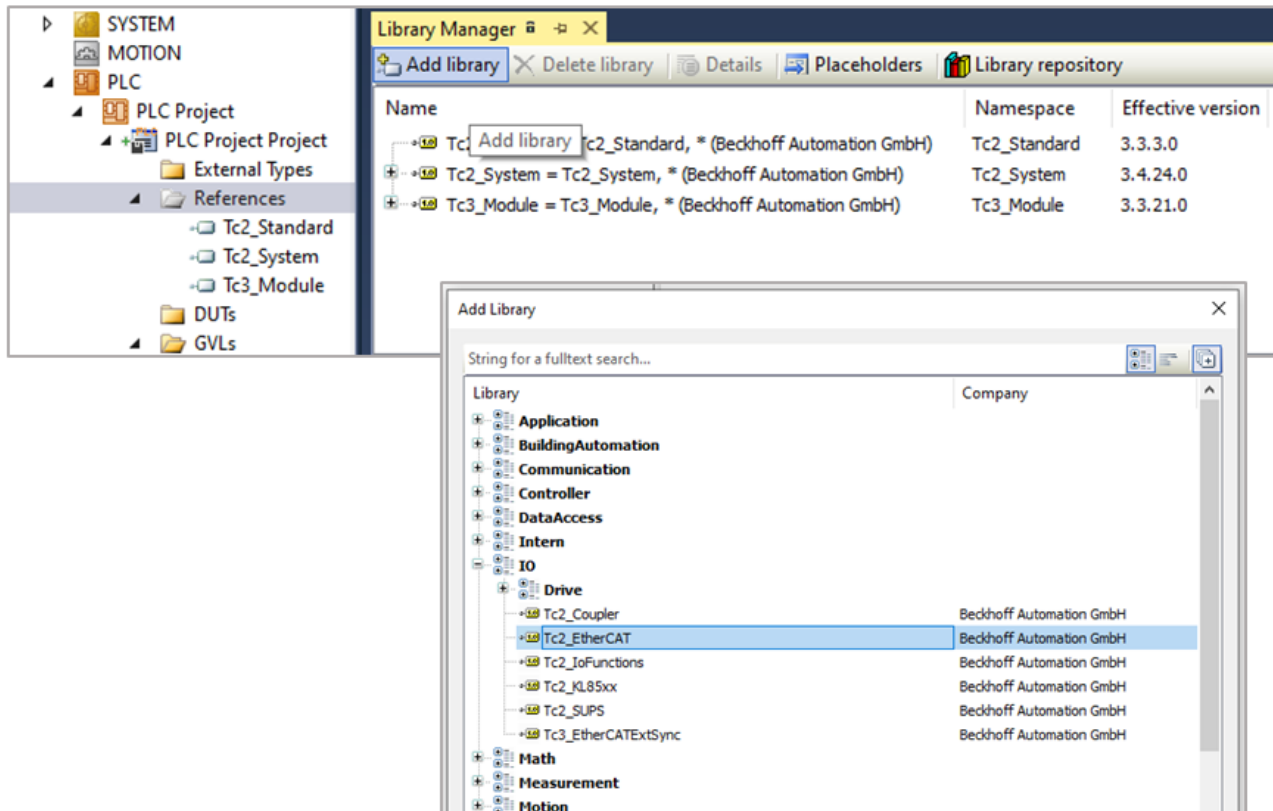


Fig. 127: Adding the "Tc2\_EtherCAT" library

2. Right-click on "References" in the PLC project and select "Add library". The "Add Library" dialog box opens.
3. Select "IO" -> "Tc2\_EtherCAT" and close the dialog box with "OK".
4. The Tc2\_EtherCAT library is now displayed under References in the TwinCAT tree.
5. Press the "F2" button to open the input wizard.

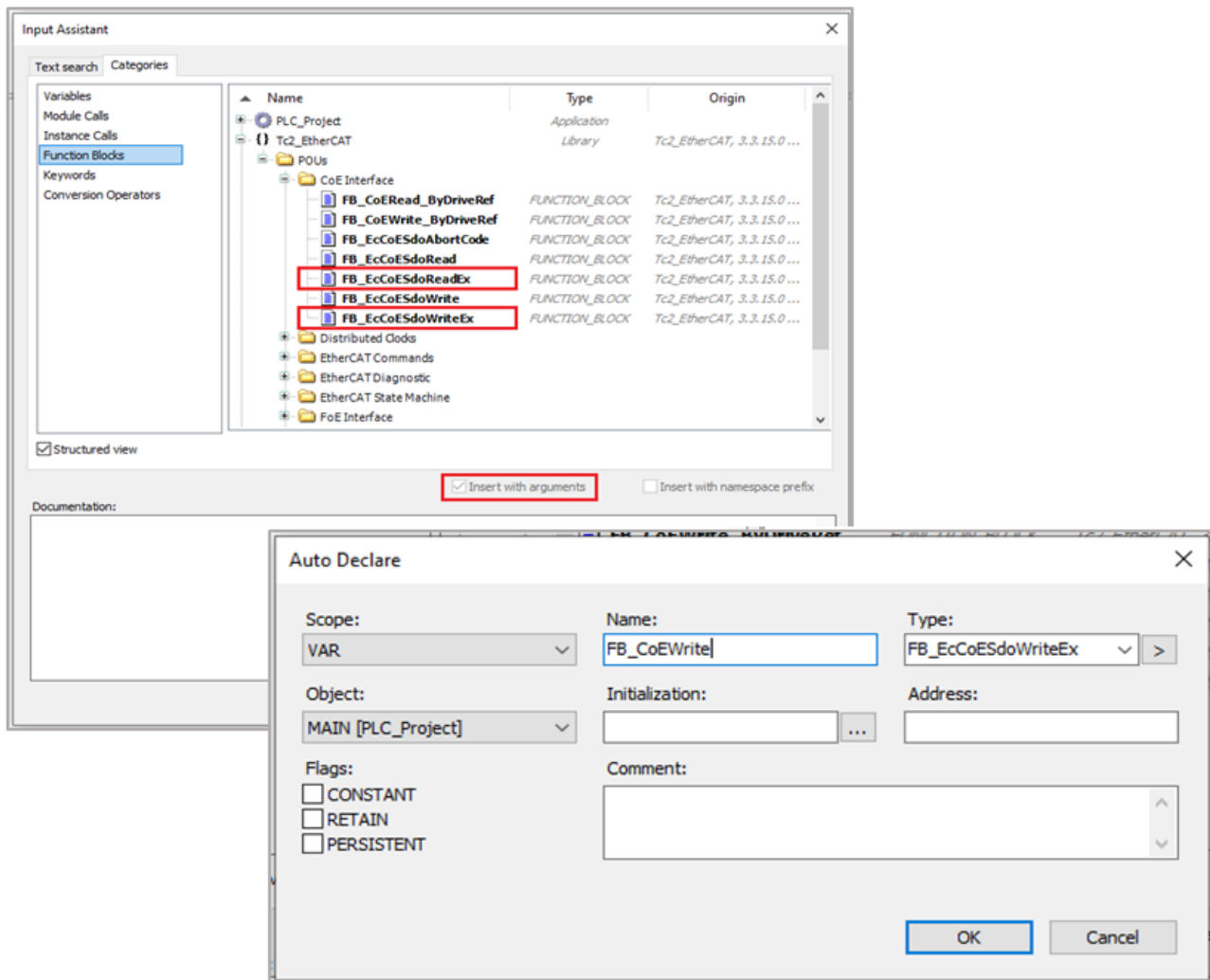


Fig. 128: Selection and instantiation of function blocks

6. Activate the checkbox "Insert with arguments" and select the function block
  - ⇒ "FB\_EcCoeSdoReadEx" for reading CoE values or
  - ⇒ "FB\_EcCoeSdoWriteEx" for writing CoE values
7. Confirm with "OK". The "Auto Declare" dialog box opens.
8. Instantiate the function block in the "Auto Declare" dialog box by entering the name and confirming with "OK".
9. Initialize the function blocks in your PLC program. Use the following sample code for reading and writing the "DC OK threshold" (`0x8000:16 [I]_143`) as a guide.



```

MAIN  ▾ ×
1  PROGRAM MAIN
2  VAR
3      // Read and Write of the DC OK threshold in the CoE object dictionary
4      FB_CoERead      : FB_EcCoESdoReadEx; // Instantiation of the read function block
5      FB_CoEWrite     : FB_EcCoESdoWriteEx; // Instantiation of the write function block
6
7      fDCOKThreshold  : REAL;             // Variable for DC OK threshold
8
9      bStartRead      : BOOL := FALSE;    // Rising edge will start the read command
10     bStartWrite     : BOOL := FALSE;    // Rising edge will start the write command
11
12     AmsAdr AT %I*   : ST_AmsAddr;       // Link to power supply
13 END_VAR

1 // Read the value from CoE 0x8000:16 (DC OK threshold)
2 FB_CoERead(
3     sNetId:= F_CreateAmsNetId(AmsAdr.netId),
4     nSlaveAddr:= AmsAdr.port,
5     nSubIndex:= 16#16,
6     nIndex:= 16#8000,
7     pDstBuf:= ADR(fDCOKThreshold),
8     cbBufLen:= sizeof(fDCOKThreshold),
9     bExecute:= bStartRead,
10    tTimeout:= T#1S,
11    bCompleteAccess:= FALSE,
12    bBusy=>,
13    bError=>,
14    nErrId=>,
15    cbRead=> );
16
17 // Write value to CoE 0x8000:16 (DC OK threshold)
18 FB_CoEWrite(
19     sNetId:= F_CreateAmsNetId(AmsAdr.netId),
20     nSlaveAddr:= AmsAdr.port,
21     nSubIndex:= 16#16,
22     nIndex:= 16#8000,
23     pSrcBuf:= ADR(fDCOKThreshold),
24     cbBufLen:= sizeof(fDCOKThreshold),
25     bExecute:= bStartWrite,
26     tTimeout:= T#1S,
27     bCompleteAccess:= FALSE,
28     bBusy=>,
29     bError=>,
30     nErrId=> );
    
```

Code sample: Reading and writing the CoE value for the DC OK threshold

10. After starting the controller and logging in, you will see the following screen:

**Read command:**

A positive edge of "startRead" starts the read command. The value of the variable "fDCOKthreshold" is written.

Expression	Type	Value	Prepared...	Address	Comment
FB_CoERead	FB_EcCoESdoR...				Instantiation of the read function bl...
FB_CoEWrite	FB_EcCoESdo...				Instantiation of t...rite function bl...
fDCOKThreshold	REAL	24			Variable for DC OK threshold
bStartRead	BOOL	TRUE			Rising edge will s... the read comm...
bStartWrite	BOOL	FALSE			Rising edge will start the write com...
AmsAdr	ST_AmsAddr			%I*	Link to power supply

Online data for reading and writing the DC OK threshold

**Write command:**

A positive edge of "startWrite" starts the write command. The CoE object 0x8000:16 "DC OK threshold" is read and the value in the variable "fDCOKthreshold" is written to the CoE object "DC OK threshold" 0x8000:16.

### 6.4.7 Complete access to "PSU Diag Data"

Using the function block "FB\_EcCoeSdoReadEx", it is also possible to access all sub-values belonging to an index with a read command. This allows all elements from the "PSU Diag Data" to be read at once, for example. Proceed as follows:

To display the "PSU Diag Data" in detail, you must first create a data structure corresponding to the CoE structure in your PLC project under DUTs:

<pre> 1  {attribute 'pack_mode' := '1'} 2  TYPE sPsuDiagData : 3      STRUCT 4          Align1          : UINT; // FA14:01...10 --&gt; Align 2 byte 5          MinInputVoltage : REAL; // FA14:11 6          InputUndervoltageCounter : UINT; // FA14:12 7          MaxInputVoltage : REAL; // FA14:13 8          Align2          : UINT; // FA14:14 --&gt; Align 2 byte 9          MaxPeakInputVoltage : REAL; // FA14:15 10         InputOvervoltageCounter : UINT; // FA14:16 11         MinTemperature : REAL; // FA14:17 12         MaxTemperature : REAL; // FA14:18 13         ActualRMSInputVoltage : REAL; // FA14:19 14         Align3          : UDINT; // FA14:1A --&gt; Align 4 byte 15         ActualTemperature : REAL; // FA14:1B 16     END_STRUCT 17 END_TYPE         </pre>	<table border="1"> <tr> <td>FA14:0</td> <td>PSU Diag data</td> <td>RO</td> <td>&gt;27&lt;</td> <td></td> </tr> <tr> <td>FA14:11</td> <td>Min input voltage</td> <td>RW</td> <td>233.000000 (2.330000e+02)</td> <td>V</td> </tr> <tr> <td>FA14:12</td> <td>Input undervoltage counter</td> <td>RW</td> <td>0x0000 (0)</td> <td></td> </tr> <tr> <td>FA14:13</td> <td>Max input voltage</td> <td>RW</td> <td>240.000000 (2.400000e+02)</td> <td>V</td> </tr> <tr> <td>FA14:15</td> <td>Max peak input voltage</td> <td>RW</td> <td>340.000000 (3.400000e+02)</td> <td>V</td> </tr> <tr> <td>FA14:16</td> <td>Input overvoltage counter</td> <td>RW</td> <td>0x0000 (0)</td> <td></td> </tr> <tr> <td>FA14:17</td> <td>Min temperature</td> <td>RW</td> <td>21.187500 (2.118750e+01)</td> <td>°C</td> </tr> <tr> <td>FA14:18</td> <td>Max temperature</td> <td>RW</td> <td>29.625000 (2.962500e+01)</td> <td>°C</td> </tr> <tr> <td>FA14:19</td> <td>Actual RMS input voltage</td> <td>RO</td> <td>234.000000 (2.340000e+02)</td> <td>V</td> </tr> <tr> <td>FA14:1B</td> <td>Actual temperature</td> <td>RO</td> <td>29.437500 (2.943750e+01)</td> <td>°C</td> </tr> </table>	FA14:0	PSU Diag data	RO	>27<		FA14:11	Min input voltage	RW	233.000000 (2.330000e+02)	V	FA14:12	Input undervoltage counter	RW	0x0000 (0)		FA14:13	Max input voltage	RW	240.000000 (2.400000e+02)	V	FA14:15	Max peak input voltage	RW	340.000000 (3.400000e+02)	V	FA14:16	Input overvoltage counter	RW	0x0000 (0)		FA14:17	Min temperature	RW	21.187500 (2.118750e+01)	°C	FA14:18	Max temperature	RW	29.625000 (2.962500e+01)	°C	FA14:19	Actual RMS input voltage	RO	234.000000 (2.340000e+02)	V	FA14:1B	Actual temperature	RO	29.437500 (2.943750e+01)	°C
FA14:0	PSU Diag data	RO	>27<																																																
FA14:11	Min input voltage	RW	233.000000 (2.330000e+02)	V																																															
FA14:12	Input undervoltage counter	RW	0x0000 (0)																																																
FA14:13	Max input voltage	RW	240.000000 (2.400000e+02)	V																																															
FA14:15	Max peak input voltage	RW	340.000000 (3.400000e+02)	V																																															
FA14:16	Input overvoltage counter	RW	0x0000 (0)																																																
FA14:17	Min temperature	RW	21.187500 (2.118750e+01)	°C																																															
FA14:18	Max temperature	RW	29.625000 (2.962500e+01)	°C																																															
FA14:19	Actual RMS input voltage	RO	234.000000 (2.340000e+02)	V																																															
FA14:1B	Actual temperature	RO	29.437500 (2.943750e+01)	°C																																															

Fig. 129: Creating a data structure (DUT) for the PSU Diag Data

Then insert the function block "FB\_EcCoeSdoReadEx" and instantiate it as described in the chapter "Writing and reading CoE objects using function blocks" [▶ 125]. Then follow the sample code below to initialize the function block.

```

MAIN_CA*  X  sPsuDiagData
1  PROGRAM MAIN_CA
2  VAR
3      FB_CoERead : FB_EcCoeSdoReadEx; // Instantiation of the read function block
4      sPSUDiagData : sPsuDiagData; // Struct which holds the device diag data (CoE 0xFA14)
5      startRead : BOOL := FALSE; // Rising edge will start the read command
6      AmsAdr AT %I+ : ST_AmsAddr; // Link to power supply
7  END_VAR
8
9
10 // Read all indices from object 0xFA14 (PSU Diag data)
11
12 FB_CoERead(
13     sNetId:= F_CreateAmsNetId(AmsAdr.netId),
14     nSlaveAddr:= AmsAdr.port,
15     nSubIndex:= 16#01,
16     nIndex:= 16#FA14,
17     pDstBuf:= ADR(sPSUDiagData),
18     cbBufLen:= SIZEOF(sPSUDiagData),
19     bExecute:= startRead,
20     tTimeout:= T#1S,
21     bCompleteAccess:= TRUE,
22     bBusy=>,
23     bError=>,
24     nErrId=>,
25     cbRead=> );
        
```

Code sample: Reading all objects of the PSU Diag Data with a read command

After starting the controller and logging in, you will see the following screen:

Expression	Type	Value	Prepared...	Address	Comment
FB_CoERead	FB_EcCoESdoR...				Instantiation of the read function bl...
sPSUdiagData	sPsuDiagData				Struct which holds the device diag ...
Align1	UINT	0			FA14:01...10 --> Align 2 byte
MinInputVoltage	REAL	233			FA14:11
InputUndervoltageCounter	UINT	0			FA14:12
MaxInputVoltage	REAL	240			FA14:13
Align2	UINT	0			FA14:14 --> Align 2 byte
MaxPeakInputVoltage	REAL	340			FA14:15
InputOvervoltageCounter	UINT	0			FA14:16
MinTemperature	REAL	21.1875			FA14:17
MaxTemperature	REAL	29.625			FA14:18
ActualRMSInputVoltage	REAL	234			FA14:19
Align3	UDINT	0			FA14:1A --> Align 4 byte
ActualTemperature	REAL	29.4375			FA14:1B
startRead	BOOL	TRUE			Rising edge will s... the read comm...
AmsAdr	ST_AmsAddr			%I*	Link to power supply
netId	AMSNETID	5.35.2.16.4.1		%IB*	
port	WORD	1001		%IB*	

Fig. 130: Online data for reading all PSU Diag Data objects

**Read command:**

A positive edge of "startRead" starts the read command. The CoE object 0xA000 PSU Diag Data is read and the values of the sub-objects are written to the corresponding places in the data structure "sPSUdiagData".

## 6.4.8 DC-OK function (parameterization)

The DC-OK function monitors the output voltage at the output terminals of an operating power supply via the process data.

1. Define the value for the DC OK threshold via the index 0x8000:16 "DC OK threshold"
  - ⇒ Adjustment range 3.0 V to 30 V for nominal voltage  $U_N = 24$  V
  - ⇒ Default:  $22 \text{ V} = 0.917 \cdot U_N$
  - ⇒ Hysteresis: typ. 0.5 V
- ⇒ If the output voltage (index 0x6000:11) falls below the DC OK threshold (index 0x8000:16), the DC OK bit (index 0x6000:04) is set to FALSE. In addition, the warning bit (index 0x6000:01) is also set to TRUE (see chapter "[Setting the warning thresholds for the warning bit \[► 121\]](#)").

### Also see about this

- 📖 DC-OK function (monitoring) [► 24]

## 6.4.9 Switching output via EtherCAT

### **i** Switching the outputs when the system is at a standstill or in the event of an error via "Disable output"

The output voltage of the device can be disabled via "Disable output". In the event of a system standstill or fault, the power supply and corresponding downstream system components can be switched off and on again via TwinCAT in a targeted and location-independent manner.

1. Enable the PDO 0x1600 "PSU Outputs" via the "Process data" tab
  - ⇒ "Disable output" is displayed in the tree structure. The power supply output can be disabled via the "Disable output" bit.
2. Restarting the EtherCAT system or reloading the configuration in Config mode (F4) restarts the EtherCAT communication and the process data is transferred.
3. If the "Disable output" bit is set to TRUE, the DC output voltage of the device is switched off.
  - ⇒ The output remains disabled as long as "Disable output" = TRUE.
  - ⇒ The voltage at the output returns when the "Disable output" bit is disabled again.

The screenshot shows the TwinCAT configuration interface for 'TwinCAT Project 6 (PSxxx)'. The 'Process Data' tab is selected, and the 'PDO Assignment (0x1C12)' section shows '0x1600' selected. The 'PDO Content (0x1600)' section shows a 'Disable output' bit at index 0x7000:01. The 'Solution Explorer' on the left shows the project structure, with 'PSU Outputs' and 'Disable output' highlighted under 'Box 1 (PS2001-2420-1001)'.

SM	Size	Type	Flags
0	256	MbxOut	
1	256	MbxIn	
2	2	Outputs	
3	16	Inputs	

Index	Size	Name	Flags	SM
0x1A00	12.0	PSU Inputs	MF	3
0x1A01	4.0	PSU Synchron info data	F	3
0x1A80	2.0	PSU Inputs Device	MF	3
0x1600	2.0	PSU Outputs	F	2

Index	Size	Offs	Name	Type
0x7000:01	0.1	0.0	Disable output	BIT
---	1.7	0.1	---	---
		2.0		

Enabling "Disable Output" via the Process Data tab using the PS2001-2420-1001 as an example

### 6.4.9.1 Start-up behavior of the output voltage

**i** **Change the behavior of the output voltage at startup via index 0x8001:41 "Default Start Up Output State" (from FW03)**

- This function only takes effect if the device is completely de-energized on both the primary and secondary sides and the internal capacitors are discharged.
- If the input voltage is only interrupted for a short time, this function does not take effect. Due to the internal capacitance, the internal auxiliary voltage can remain on for several seconds, depending on the load situation. The EtherCAT interface can also continue to communicate actively during this time. If the input voltage returns during this time without a loss of communication, the last known state of the output is retained.

When delivered, the device is parameterized so that the output voltage is applied directly when the input voltage is switched on.

1. Behavior at start (see fig. below *AC Power on*)

The behavior of the output voltage at startup can be changed via the CoE object 0x8001:41 ("Default Start Up Output State").

- 0x8001:41 = 0 "Enable Output (Default)"
- 0x8001:41 = 1 "Disable Output"  
If the bit is set to "Disable Output", the output voltage is not switched on directly after switching on.

2. Behavior after reaching the OP state (see figure below *State change to OP*)

As soon as the EtherCAT devices state changes to OP, this setting becomes ineffective and is taken over by the process data bit "Disable output" (0x1600:01).

- This is not mapped in the standard configuration, which means that the device switches on the output voltage after reaching the OP state.
- If the device is not to switch on the voltage even after reaching the OP state,
  - the "Disable output" PDO (0x1600:01) must be mapped and
  - must be set to "TRUE" before reaching the OP state.

This function is available from FW03.

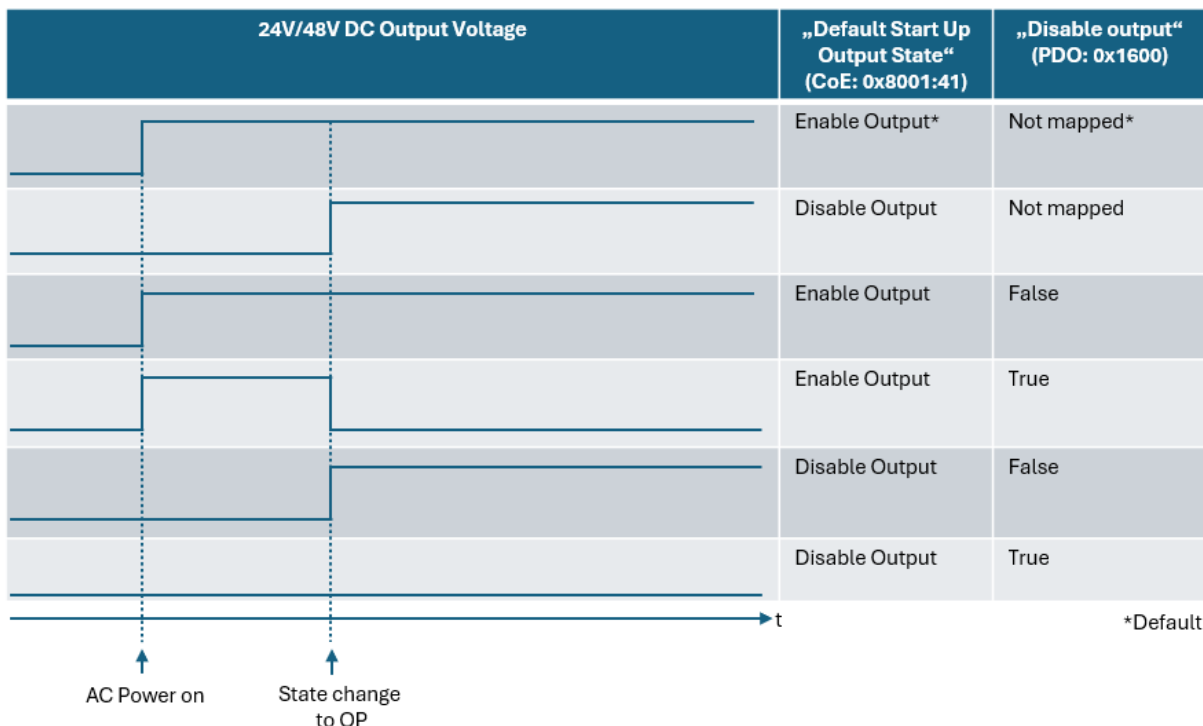



Fig. 131: Start-up behavior dependent on CoE 0x8001:41 and 0x1600

## 6.4.10 Recognizing invalid data

Each EtherCAT slave has the so-called WCState bit, which signals error-free EtherCAT communication if the value = 0. However, if the WCState is set to 1, an error has occurred during EtherCAT transmission and the process data is invalid.

If, for example, the 230 V at the input is switched off, the output voltage slowly drops. As soon as the output voltage falls below the voltage required for EtherCAT communication, no more data is transmitted. The output voltage therefore remains at the last transmitted value, although in reality it continues to fall. However, as EtherCAT communication has been interrupted, WCState = 1 signals that the process data and in particular the data for the output voltage are no longer valid.

Name	Online	Type
Warning	X 0	BIT
Error	0	BIT
DC OK	1	BIT
Output voltage	24.035156	REAL
Output current	0.01171875	REAL
Info data 1	2330	UINT
Info data 2	2330	UINT
Input undervolts	0	BIT
WcState	0	BIT
Input toggle	0	BIT
State	8	UINT
AdsAddr	X 5.35.2.16.4.1:1001	AMSADDR
netId	5 35 2 16 4 1	AMSNETID



Name	Online	Type
Warning	X 0	BIT
Error	0	BIT
DC OK	1	BIT
Output voltage	24.019531	REAL
Output current	0.01171875	REAL
Info data 1	2330	UINT
Info data 2	2330	UINT
Input undervolts	0	BIT
WcState	1	BIT
Input toggle	0	BIT
State	257	UINT
AdsAddr	X 5.35.2.16.4.1:1001	AMSADDR
netId	5 35 2 16 4 1	AMSNETID

Invalid process data when switching off the input voltage.

Left: Normal operation, WCState bit = 0.

Right: Input voltage has been switched off, the output voltage remains at the last transmitted value, WCState = 1 indicates the invalidity of the process data.



### 6.4.11 Parallel use

If the device is operated in parallel with another device,

- ✓ the conditions for parallel use must be met (see chapter "Parallel use for power increase [▶ 41]" and "Parallel use for redundancy [▶ 42]").
1. In addition, parallel use must be enabled via the "PSU Features" object in index 0x8001:01 "Parallel use".

#### Enabling parallel use

2. Select "PSU Features" 0x8001 in the CoE online tab.
3. Double-click on index 0x8001:01 "Parallel use". The "Set Value dialog" is opened.
4. In the "Set value dialog", select the value "1" in the field "BOOL" and confirm with "OK".

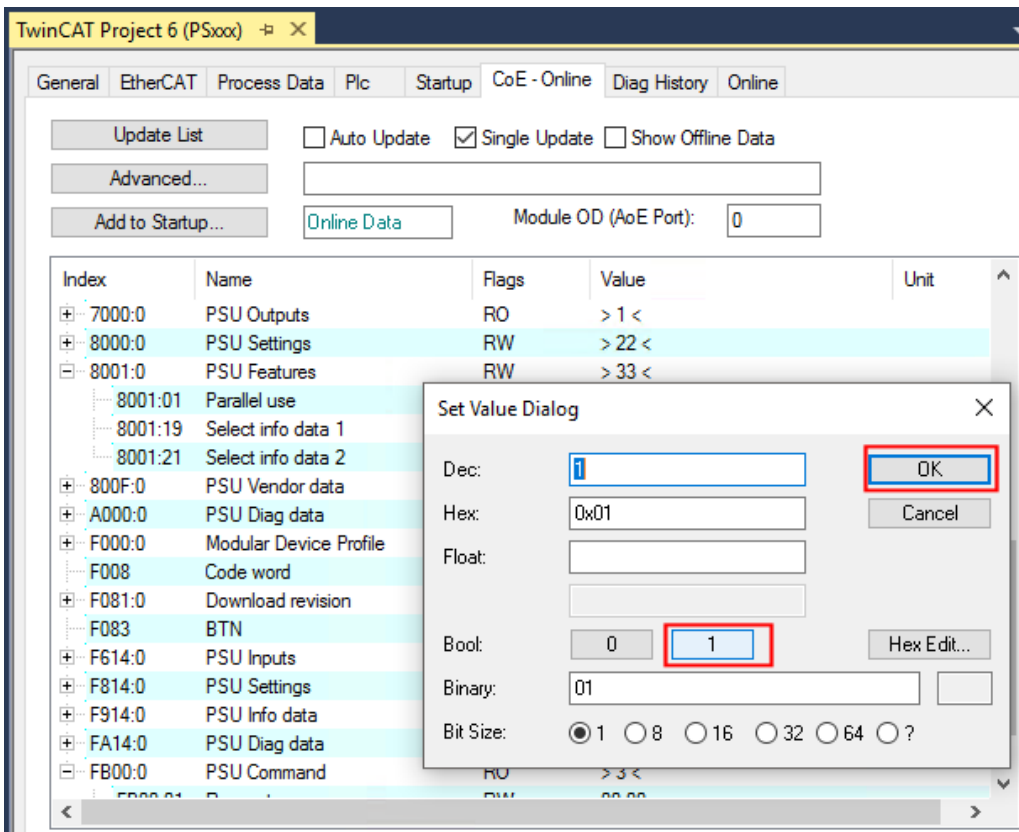


Fig. 132: Enabling parallel use via "Parallel Use" 0x8001:01

## 6.5 Information on grid monitoring on the input side

Using the measurement data recorded on the input side in object 0xFA14 "PSU Diag Data" [▶ 129], it is possible to monitor the mains voltage over a longer period of time. Creeping changes and quality problems can thus be detected at an early stage and prevented with suitable measures before they become a problem for the power supply or other connected devices.

### RMS input voltage

The AC input voltage is determined with an effective power measurement. In the nominal voltage range of 85 - 264 V AC, an average value is calculated over 10 mains periods.

### Undervoltages and overvoltages

A supply voltage that is too low or too high at the input can be detected using the PS2001-xxxx-1001 power supplies.

The devices themselves are largely protected against voltage fluctuations within the nominal voltage range, as the wide-range input 100 - 240 V AC (-15%, +10%) in the device ensures that the 24 V DC (PS2001-24xx-1001) / 48 V DC (PS2001-48xx-1001) output remains stable even in the event of major deviations in the supply voltage.

The undervoltage range begins when the voltage falls below 75 V AC and is exited again when it exceeds 85 V AC. A hysteresis of 10 V AC is stored here. There are two measuring principles within the undervoltage range:

- from 0 - 65 V AC: effective power measurement over ½ mains period
- from 65 - 75 V AC: effective power measurement over 1 mains period

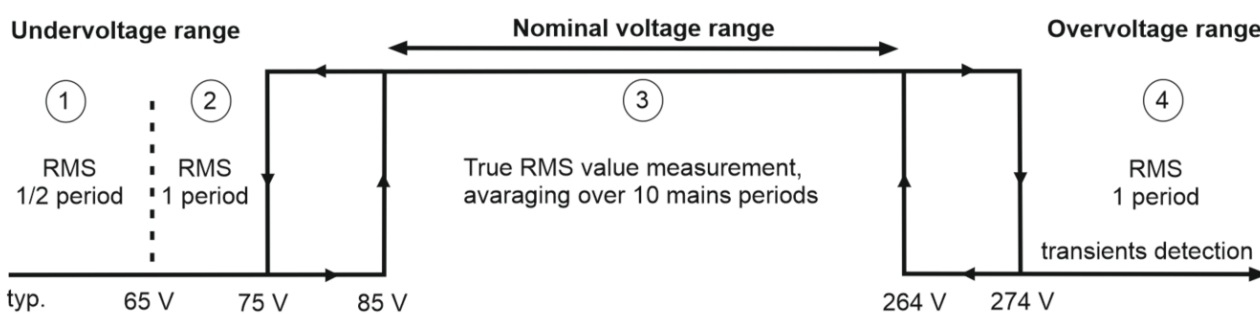
The overvoltage range begins when 274 V AC is exceeded and is exited again when the value falls below 264 V AC. A hysteresis of 10 V AC is stored here.

Peak voltages are measured up to a level of 700 V.

### Rapid overvoltages and transients

Rapid overvoltages and transients are also detected by the PS2001-xxxx-1001 power supplies. For example, the VDE-0160 pulse and surge pulses from 750 V are detected.

### Measuring times of the effective power measurement as a function of the input voltage



Input voltage range		Measuring principle:	
Undervoltage	1	< 65 V AC	Effective power measurement over ½ mains period
	2	65 - 75 V AC	Effective power measurement over 1 mains period
Nominal voltage	3	85 - 264 V AC	Effective power measurement, averaging over 10 mains periods
Overvoltage	4	> 274 V AC	Effective power measurement over 1 mains period, transient detection

Device-specific transmission times of typically 3 ms and the set cycle time are added to the measuring times.

## 6.6 Process data overview

### 6.6.1 Parameterization

In the TwinCAT System Manager, a PS2001-xxxx-1001 power supply is parameterized via two dialog boxes/ tabs.

- The Process Data tab (A) for the communication-specific settings only displays the process data.
- The CoE-Online tab (B) for settings in the slave contains all data exchanged via the EtherCAT interface.

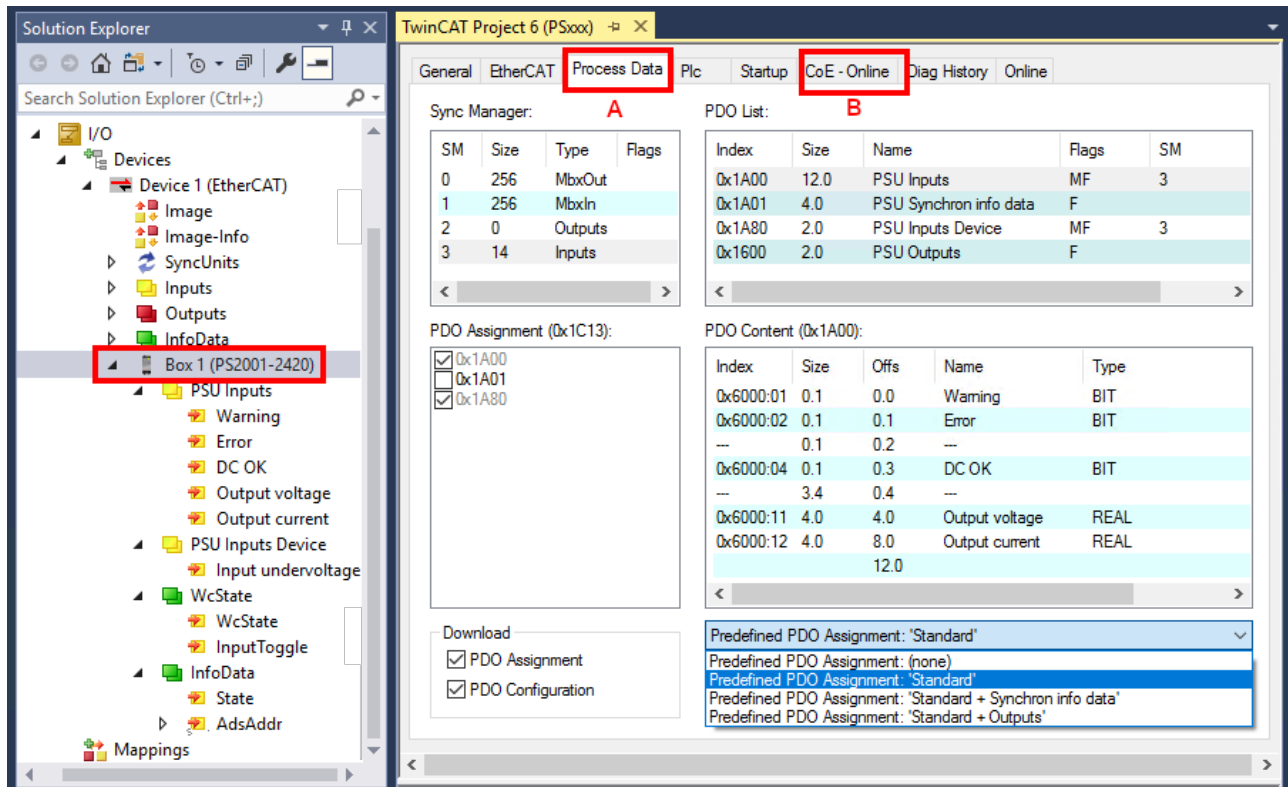


Fig. 133: Parameterization using PS2001-2420-1001 as an example

- Changes to the process data-specific settings are generally only effective after a restart of the EtherCAT master:
  - restart TwinCAT in RUN or CONFIG mode; RELOAD in CONFIG mode
- Changes in CoE - Online
  - are in general immediately effective.
  - are in general stored non-volatile *only* in the terminal/in the slave and should therefore be entered in the CoE StartUp list. This list is processed at each EtherCAT start and the settings are loaded into the slave.
  - Observe the Notes for parameterization via the CoE directory [► 115].

## 6.6.2 Process data

In the PS2001-xxxx-1001 power supplies, the process data is divided into the categories:

- "PSU Inputs"  
The process data *Warning*, *Error*, *DC-OK*, *Output voltage* (32-bit) and *Output current* (32-bit) are summarized under "PSU Inputs".
- "PSU Outputs"  
The "PSU Outputs" contain the values *Disable output*.
- "PSU Synchron info data"  
In addition, certain additional diagnostic information can be transmitted in the "PSU Synchron info data" category via the freely configurable information data *info data 1* and *info data 2*.

There are three process data settings "Predefined PDO Assignments" to select from (see chapter "[Switching the process data \[► 139\]](#)"):

- "Standard":  
Only the "PSU Inputs" and "PSU Inputs Device" are transmitted.
- "Standard + Synchron info data":  
"PSU Inputs", "PSU Inputs Device" and "PSU Synchron info data" are transmitted.
- Standard + Outputs  
"PSU Inputs", "PSU Inputs Device" and "PSU Outputs" are transmitted.

### Interpretation of Value & Status variable

Name	Online	Type	Size	>Addr...	In/Out	User ID	Linked to
Warning		BIT	0.1	39.0	Input	0	
Error		BIT	0.1	39.1	Input	0	
DC OK		BIT	0.1	39.3	Input	0	
Output voltage		REAL	4.0	43.0	Input	0	
Output current		REAL	4.0	47.0	Input	0	
Input undervoltage		BIT	0.1	51.2	Input	0	
WcState		BIT	0.1	1522.1	Input	0	
InputToggle		BIT	0.1	1524.1	Input	0	
State		UINT	2.0	1548.0	Input	0	
AdsAddr		AMSADDR	8.0	1550.0	Input	0	

Fig. 134: Default process data using the PS2001-2420-1001 as an example

The plain text display of the bit meanings of the status word is particularly helpful for commissioning, but also for linking with the PLC program.

By right-clicking on the status variable in the configuration tree (A), the structure can be opened for linking (B).

### 6.6.2.1 Switching process data sets

The process data to be transmitted (PDO, Process Data Objects) can be selected by the user

- via the selection dialog "Predefined PDO Assignment" [▶ 139] (all TwinCAT versions)
- selectively for individual PDOs [▶ 140], taking into account the excluded elements.

#### Selection dialog "Predefined PDO Assignment"

Defined PDO sets can be preselected if they exist in the XML description.

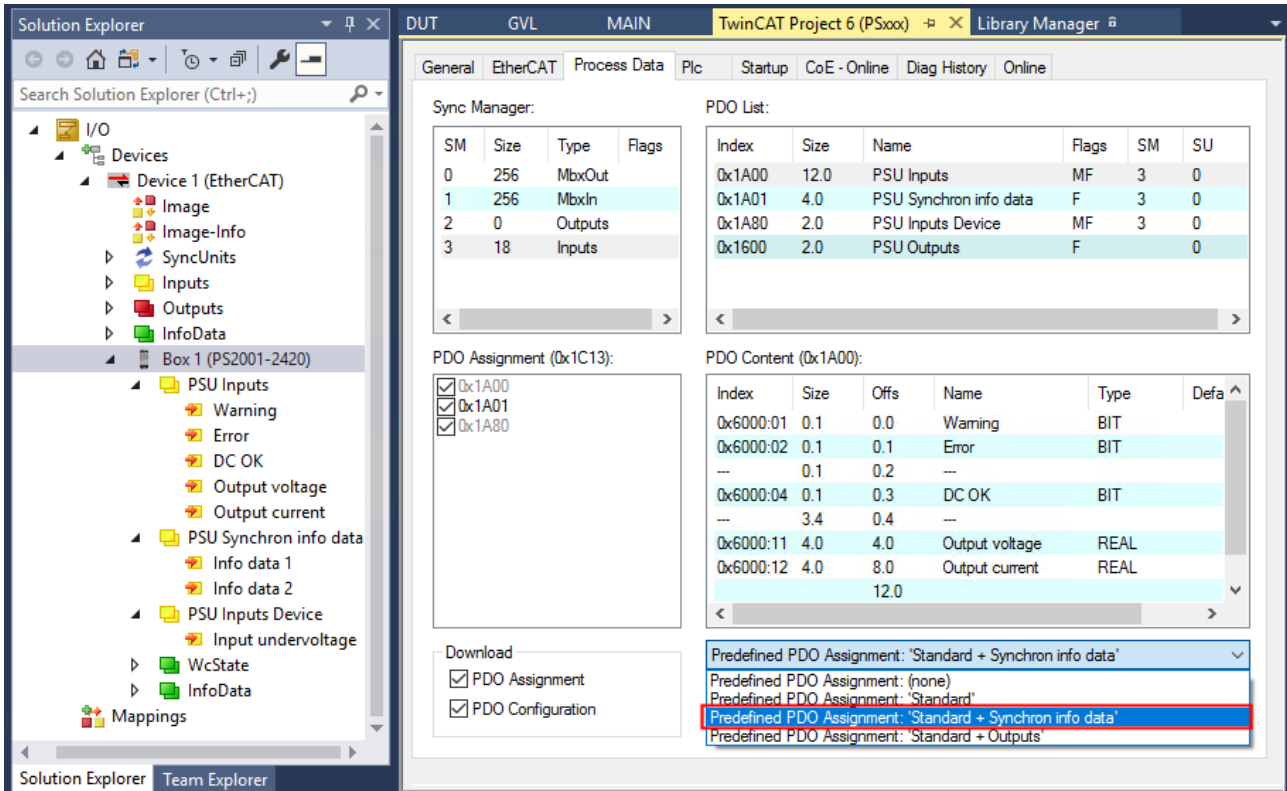


Fig. 135: Predefined PDO assignments using the PS2001-2420-1001 as an example

This sets the PDOs of the PS2001-xxxx-1001 to a process image:

- "Standard" or
- "Standard + Synchron info data" or
- "Standard + Outputs".

## Selective PDO selection

All TwinCAT versions support the selective selection of individual PDOs, as defined in the XML description. Exclusion criteria prevent invalid combinations. A detailed description can be found in the [EtherCAT system documentation](#).

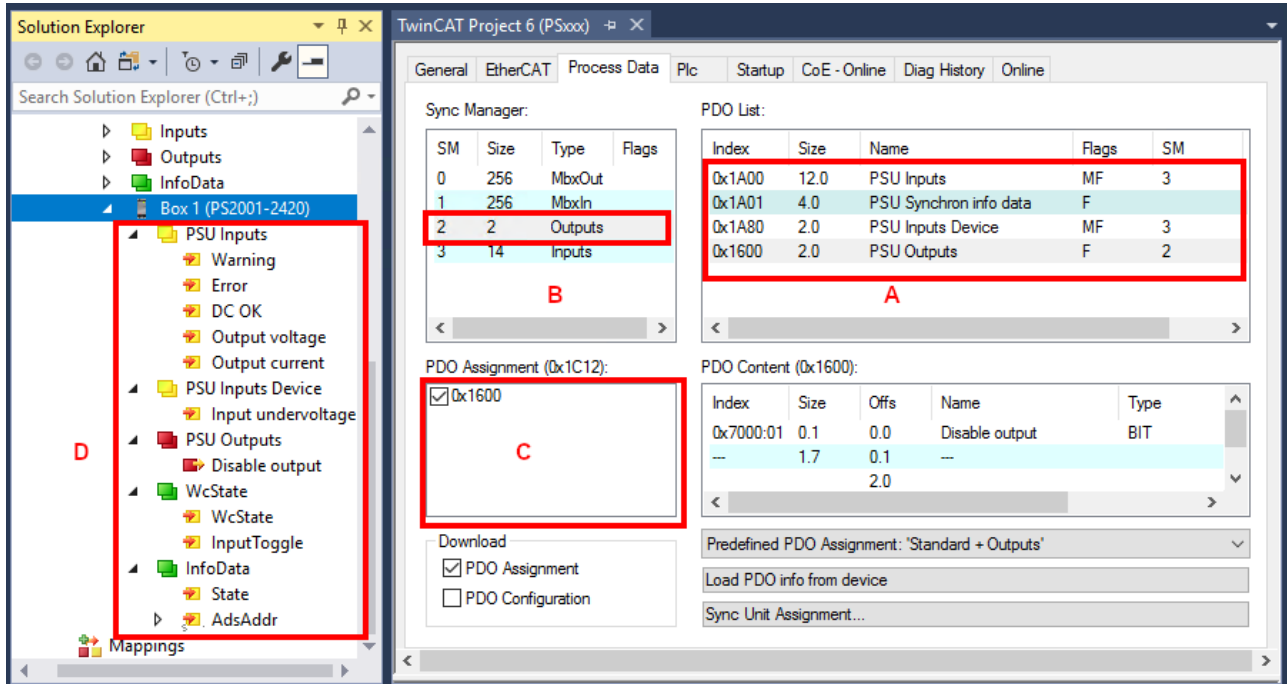


Fig. 136: Selective PDO selection using the PS2001-2420-1001 as an example

Explanations for Fig. *Selective PDO selection*: In the "Process Data" tab under (A) it can be seen that the PS2001-2420-1001 power supply offers several different PDOs for transmission:

- "PSU Inputs" from the CoE index 0x1A00 (12 bytes):
  - Warning
  - Error
  - DC-OK
  - Output voltage
  - Output current
- "PSU Synchron info data" under CoE index 0x1A01 (4 bytes):
  - Info data 1
  - Info data 2
- "PSU Inputs Device" from the CoE index 0x1A80 (2 bytes):
  - Input undervoltage
- "PSU Output" from the CoE index 0x1600 (2 bytes)
  - Disable output

By selecting the necessary Input SyncManager in (B), the PDO Assignment under (C) can be carried out manually. The process data can then be linked in the TwinCAT tree (D).

## 6.7 Object description and parameterization

### **i** EtherCAT XML Device Description

The display matches that of the CoE objects from the EtherCAT XML Device Description. We recommend downloading the latest XML file from the download area of the Beckhoff website and installing it according to installation instructions.

### **i** Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE-Online tab [▶ 97] (double-click on the respective object) or via the Process Data tab [▶ 94](allocation of PDOs). Please note the following general CoE notes [▶ 50] when using/manipulating the CoE parameters:

- Keep a startup list if components have to be replaced
- Differentiation between online/offline dictionary, existence of current XML description
- use “CoE reload [▶ 166]” for resetting changes

### Introduction

The CoE overview contains objects for different intended applications:

- Objects required for parameterization during commissioning:
  - Restore Object Index 0x1011 [▶ 141]
  - Configuration data index 0x80n0 [▶ 143]
- Objects intended for regular operation, e.g. through ADS access.
- Profile-specific objects:
  - Configuration data (vendor-specific) index 0x80nF [▶ 144]
  - Input data Index 0x60n0 [▶ 142]
  - Information and diagnostic data [▶ 146] Index 0xF000, 0xF008, 0xA000, 0xF914, 0xFA14
- Standard objects [▶ 147]

The following section first describes the objects required for normal operation, followed by a complete overview of missing objects.

### 6.7.1 Restore object

#### Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	<u>Restore default parameters</u> [▶ 166]	Restore default parameters	UINT8	RO	0x01 (1 <sub>dec</sub> )
1011:01	SubIndex 001	If this object is set to “ <b>0x64616F6C</b> ” in the set value dialog, all backup objects are reset to their delivery state.	UINT32	RW	0x00000000 (0 <sub>dec</sub> )

## 6.7.2 Input data

### Index 6000 PSU Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	PSU Inputs		UINT8	RO	0x16 (22 <sub>dec</sub> )
6000:01	Warning	<p>Is set if one of the following conditions applies:</p> <ul style="list-style-type: none"> <li>The output voltage (0x6000:11) falls below the DC OK threshold (0x8000:16)</li> <li>The output voltage (0x6000:11) exceeds the overvoltage threshold (0x8000:14)</li> <li>The output current (0x6000:12) exceeds the overcurrent threshold (0x8000:15)</li> <li>The internal device temperature (0xFA14:1B) exceeds the internal temperature threshold (0xF814:11)</li> <li>The input voltage "Actual RMS input voltage" (0xFA14:19) falls below the minimum value of the nominal input voltage</li> <li>The device has been disconnected from the power supply</li> <li>If the warning bit = 1, the cause of the warning can be viewed in the <a href="#">Diag History</a> or via the <a href="#">TwinCAT EventLogger</a> [► 124].</li> </ul>	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6000:02	Error	<p>Becomes TRUE and results in the output voltage being switched off if one of the following conditions applies:</p> <ul style="list-style-type: none"> <li>Device-specific overvoltage protection active The output voltage has exceeded the maximum permissible value. See chapter <a href="#">Protective functions</a> [► 33].</li> <li>Device-specific overcurrent protection active. The device was switched off due to the <a href="#">hiccup mode</a> [► 21]. The error bit is an image of the on/off behavior of the hiccup mode.</li> <li>Device-specific overtemperature protection active. The internal device temperature has exceeded the maximum permissible value (130°C).</li> <li>If the error bit = 1, the cause of the error can be viewed in the <a href="#">Diag History</a> or via the <a href="#">TwinCAT EventLogger</a> [► 124].</li> </ul>	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6000:04	DC OK	The output voltage (0x6000:11) falls below the DC OK threshold (0x8000:16)	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6000:11	Output voltage	Current output voltage in V Accuracy: typ. ±0.2% FSV (full scale value, typ. 0 to 65 V)	REAL32	RO	0x00000000 (0 <sub>dec</sub> )
6000:12	Output current	Current output current in A Accuracy: typ. ±0.2% FSV (full scale value, typ. 0 to 65 A)	REAL32	RO	0x00000000 (0 <sub>dec</sub> )
6000:15	Info data 1	Synchronous information (selection via subindex 0x8001:19)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
6000:16	Info data 2	Synchronous information (selection via subindex 0x8001:21)	UINT16	RO	0x0000 (0 <sub>dec</sub> )

### Index F614 PSU Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
F614:0	PSU Inputs		UINT8	RO	0x03 (3 <sub>dec</sub> )
F614:03	Input undervoltage	The input voltage "Actual RMS input voltage" (0xFA14:19) falls below the minimum value of the nominal input voltage.	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )



### 6.7.3 Output data

#### Index 7000 PSU Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7000:0	PSU Outputs		UINT8	RO	0x10 (16 <sub>dec</sub> )
7000:01	Disable output	Switch off output voltage	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )

### 6.7.4 Configuration data

#### Index 8000 PSU Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8000:0	PSU Settings		UINT8	RO	0x16 (22 <sub>dec</sub> )
8000:14	Overvoltage warn threshold	Adjustable overvoltage warning threshold [► 121] from: 0.0 V to 30.0 V  Default value: 1.2 * Nominal Voltage (0x800F:15 [► 144]) Hysteresis: 0.5 V  If the output voltage (0x6000:11 [► 142]) exceeds this value, the warning bit (0x6000:01) is set in the input process data.	REAL32	RW	0x41E66666 (1105618534 <sub>dec</sub> )
8000:15	Overcurrent warn threshold	Adjustable overcurrent warning threshold [► 121] from 0.0 A to 4 * Nominal current (0x800F:16).  Default value: 1.2 * Nominal Current (0x800F:16) Hysteresis: 0.5 A  If the output current (0x6000:12) exceeds this value, the warning bit (0x6000:01) is set in the input process data.	REAL32	RW	0x41C00000 (1103101952 <sub>dec</sub> )
8000:16	DC OK threshold	Adjustable DC OK threshold [► 24] from 3.0 V to 30.0 V  Default value: 0.917 * Nominal Voltage (0x800F:15) Hysteresis: 0.5 V  If the output voltage (0x6000:11) falls below this value, the warning bit (0x6000:01) is set in the input process data	REAL32	RW	0x41B00000 (1102053376 <sub>dec</sub> )

## Index 8001 PSU Features

Index (hex)	Name	Meaning	Data type	Flags	Default
8001:0	PSU Features		UINT8	RO	0x21 (33 <sub>dec</sub> )
8001:01	Parallel use	Must be set if several power supplies are connected in parallel. See chapter: <a href="#">Parallel use for power increase [► 41]</a> and <a href="#">Parallel use for redundancy [► 42]</a>	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
8001:19	Select info data 1	This object can be used to select which value is to be displayed in the Info Data 1 object (0x6000:15). See chapter <a href="#">Selecting the info data [► 118]</a> . permitted values: <ul style="list-style-type: none"> <li>0: Min input voltage in 0.1 V (0xFA14:11)</li> <li>1: Input undervoltage counter (0xFA14:12)</li> <li>2: Max input voltage in 0.1 V (0xFA14:13)</li> <li>4: Max peak input voltage in 0.1 V (0xFA14:15)</li> <li>5: Input overvoltage counter (0xFA14:16)</li> <li>6: Max output voltage in 0.1 V (0xA000:11)</li> <li>7: Max output current in 0.01 A (0xA000:12)</li> <li>8: Min temperature in 0.1°C (0xFA14:17)</li> <li>9: Max temperature in 0.1°C (0xFA14:18)</li> <li>10: Actual RMS input voltage in 0.1 V (0xFA14:19)</li> <li>14: Actual temperature in 0.1°C (0xFA14:18)</li> </ul>	UINT8	RW	0x00 (0 <sub>dec</sub> )
8001:21	Select info data 2	This object can be used to select which value is to be displayed in the Info Data 2 object (0x6000:16). See chapter <a href="#">Selecting the info data [► 118]</a> . permitted values: see subindex 0x8001:19	UINT8	RW	0x00 (0 <sub>dec</sub> )
8001:41	Default Start Up Output State	This object can be used to change the start-up behavior (see chapter <a href="#">Start-up behavior of the output voltage [► 133]</a> ) 0: Enable Output (Default) 1: Disable Output	UINT8	RW	0x00 (0 <sub>dec</sub> )

## Index F814 PSU Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
F814:0	PSU Settings		UINT8	RO	0x11 (17 <sub>dec</sub> )
F814:11	Overtemperature warn threshold	Adjustable warning threshold for overtemperature from -40°C to 130°C Default value: 120°C Hysteresis: 2°C If the internal device temperature (0xFA14:1B) exceeds this value, the warning bit (0x6000:01) is set in the input process data.	REAL32	RW	0x42F00000 (1123024896 <sub>dec</sub> )

## 6.7.5 Configuration data (vendor-specific)

## Index 800F PSU Vendor data

Index (hex)	Name	Meaning	Data type	Flags	Default
800F:0	PSU Vendor data		UINT8	RO	0x16 (22 <sub>dec</sub> )
800F:15	Nominal voltage	Output-side nominal voltage of the device in V, cannot be changed.	REAL32	RO	0x00 (0 <sub>dec</sub> )
800F:16	Nominal current	Output-side nominal current of the device in A, cannot be changed.	REAL32	RO	0x00 (0 <sub>dec</sub> )

## 6.7.6 Command object

### Index FB00 PSU Command

Index (hex)	Name	Meaning	Data type	Flags	Default
FB00:0	PSU Command		UINT8	RO	0x03 (3 <sub>dec</sub> )
FB00:01	Request	A write access to this object executes the corresponding command. 0x7001: Reset all counters and min. and max. values in the "PSU Diag Data" 0xA000 and 0xFA14 0x7002: Reset all events from the <a href="#">JSON file</a> [►_154]	OCTET-STRING[2]	RW	{0}
FB00:02	Status	0: last command completed, no errors, no response data 1: last command completed, no errors, response data available 2: last command completed, error, no response data 3: last command completed, error, response data available 4-99: reserved for future use 100-200: indicates how much of the command was executed (in %, 100 = 0%, 200 = 100%) 201-254: reserved for future use 255: command is executed (if the percentage display is not supported)	UINT8	RO	0x00 (0 <sub>dec</sub> )
FB00:03	Response	Byte 0: see SubIndex 2 Byte 1: unused 2-n: Service response data	OCTET-STRING[6]	RO	{0}

## 6.7.7 Information - diagnostic data

### Index A000 PSU Diag data

Index (hex)	Name	Meaning	Data type	Flags	Default
A000:0	PSU Diag data		UINT8	RO	0x20 (32 <sub>dec</sub> )
A000:11	Max output voltage	<p>Maximum output voltage in V.</p> <ul style="list-style-type: none"> <li>The value for the maximum output voltage is updated again if the old value is exceeded</li> <li>Is reset to the current value by writing any value.</li> <li>Is retained beyond one PowerCycle.</li> </ul>	REAL32	RW	0x00000000 (0 <sub>dec</sub> )
A000:12	Max output current	<p>Maximum output current in A.</p> <ul style="list-style-type: none"> <li>The value for the maximum output current is updated again if the old value is exceeded.</li> <li>Is reset to the current value by writing any value.</li> <li>Is retained beyond one PowerCycle.</li> </ul>	REAL32	RW	0x00000000 (0 <sub>dec</sub> )
A000:13	Actual output voltage	<p>Value of the current output voltage in V</p> <p>Accuracy: typ. ±0.2% FSV (full scale value, typ. 0 to 65 V)</p>	REAL32	RO	0x00000000 (0 <sub>dec</sub> )
A000:14	Actual output current	<p>Value of the current output current in A</p> <p>Accuracy: typ. ±0.2% FSV (full scale value, typ. 0 to 65 A)</p>	REAL32	RO	0x00000000 (0 <sub>dec</sub> )

### Index F914 PSU Info data

Index (hex)	Name	Meaning	Data type	Flags	Default
F914:0	PSU Info data		UINT8	RO	0x14 (20 <sub>dec</sub> )
F914:12	Operating time:Operating time	Total operating time of the device in s (cannot be reset)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
F914:13	Uptime	Operating time since last AC-side switch-on process in s (cannot be reset)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
F914:14	Turn on counter	Total number of AC-side turn-on operations of the power supply during the complete operating time (cannot be reset)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )

Index FA14 PSU Diag data

Index (hex)	Name	Meaning	Data type	Flags	Default
FA14:0	PSU Diag data		UINT8	RO	0x14 (20 <sub>dec</sub> )
FA14:11	Min input voltage	Minimum RMS input voltage in V <sub>RMS</sub> . The value is updated again if it falls below the old value.  The input voltage averaged over 10 mains periods is used as the reference value.  Undervoltages due to a grid disconnection are not taken into account.	UINT32	RW	0x00000000 (0 <sub>dec</sub> )
FA14:12	Input undervoltage counter	Number of undervoltage events of the RMS input voltage.	UINT16	RW	0x0000 (0 <sub>dec</sub> )
FA14:13	Max input voltage	Maximum RMS input voltage in V <sub>RMS</sub> . The value is updated again if the old value is exceeded.  The input voltage averaged over 10 mains periods is used as the reference value.	UINT32	RW	0x00000000 (0 <sub>dec</sub> )
FA14:15	Max peak input voltage	Maximum peak input voltage from 0 V to 700 V. The value is updated again if the old value is exceeded.	REAL32	RW	0x00000000 (0 <sub>dec</sub> )
FA14:16	Input overvoltage counter	Number of overvoltage events of the RMS input voltage.  The counter for the input overvoltages is also increased if fast input voltage transients such as the VDE 0160 pulse or a surge pulse (EN 61000-4-5) occur.	UINT16	RW	0x0000 (0 <sub>dec</sub> )
FA14:17	Min temperature	Minimum internal device temperature in °C. The value for the minimum device temperature is updated again if it falls below the old value.	REAL32	RW	0x00000000 (0 <sub>dec</sub> )
FA14:18	Max temperature	Maximum internal device temperature in °C. The value for the maximum device temperature is updated again if the old value is exceeded.	REAL32	RW	0x00000000 (0 <sub>dec</sub> )
FA14:19	Actual RMS input voltage	RMS value of the current input voltage in V <sub>RMS</sub> . Accuracy: typ. ±0.7% FSV (typ. 0 to 700 V)	REAL32	RO	0x00000000 (0 <sub>dec</sub> )
FA14:1B	Actual temperature	Value of the current internal device temperature in °C Accuracy: typ ±1 °C	REAL32	RO	0x00000000 (0 <sub>dec</sub> )
<p>All values of object 0xFA14 with flag: "RW"</p> <ul style="list-style-type: none"> <li>• are retained beyond one PowerCycle,</li> <li>• can be reset with any value via write access (counter to 0, all other values to the current value) see chapter <a href="#">Resetting counters and max/min values</a>  ▶ 122 </li> </ul>					

Index F000 Modular Device profile

Index (hex)	Name	Meaning	Data type	Flags	Default
F000:0	Modular Device Profile	General information for the Modular Device Profile	UINT8	RO	0x02 (2 <sub>dec</sub> )
F000:01	Index distance	Index distance of the objects of the individual channels	UINT16	RO	0x0010 (16 <sub>dec</sub> )
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0001 (1 <sub>dec</sub> )

Index F008 Code word

Index (hex)	Name	Meaning	Data type	Flags	Default
F008:0	Code word		UINT32	RW	0x00000000 (0 <sub>dec</sub> )

Index F081 Download revision

Index (hex)	Name	Meaning	Data type	Flags	Default
F081:0	Download revision		UINT8	RO	0x01 (1 <sub>dec</sub> )
F081:01	Revision number		UINT32	RW	0x00000000 (0 <sub>dec</sub> )

## 6.7.8 Standard objects

### Standard objects (0x1000-0x1FFF)

The standard objects have the same meaning for all EtherCAT slaves.

**Index 1000 Device type**

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: the Lo-Word contains the used CoE profile (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x038E1389 (59642761 <sub>dec</sub> )

**Index 1008 Device name**

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	PS2001-2420

**Index 1009 Hardware version**

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	

**Index 100A Software version**

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	00

**Index 100B Bootloader version**

Index (hex)	Name	Meaning	Data type	Flags	Default
100B:0	Bootloader version		STRING	RO	N/A

**Index 1018 Identity**

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4 <sub>dec</sub> )
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2 <sub>dec</sub> )
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x07D14D02 (131157250 <sub>dec</sub> )
1018:03	Revision	Revision number of the EtherCAT slave; the Low Word (bit 0-15) indicates the special terminal number, the High Word (bit 16-31) refers to the device description	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1018:04	Serial number	Serial number of the EtherCAT slave; the Low Byte (bit 0-7) of the Low Word contains the year of production, the High Byte (bit 8-15) of the Low Word contains the week of production, the High Word (bit 16-31) is 0	UINT32	RO	0x00000000 (0 <sub>dec</sub> )

**Index 10E2 Manufacturer-specific Identification Code**

Index (hex)	Name	Meaning	Data type	Flags	Default
10E2:0	Manufacturer-specific Identification Code	s. chapter <a href="#">Beckhoff Identification Code (BIC)</a> [▶ 13]	UINT8	RO	0x01 (1 <sub>dec</sub> )
10E2:01	SubIndex 001		STRING	RO	

**Index 10F0 Backup parameter handling**

Index (hex)	Name	Meaning	Data type	Flags	Default
10F0:0	Backup parameter handling	Information for standardized loading and saving of backup entries	UINT8	RO	0x01 (1 <sub>dec</sub> )
10F0:01	Checksum	Checksum across all backup entries of the EtherCAT slave	UINT32	RO	0x00000000 (0 <sub>dec</sub> )

**Index 10F3 Diagnosis History**

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:0	Diagnosis History		UINT8	RO	0x1E (30 <sub>dec</sub> )
10F3:01	Maximum Messages		UINT8	RO	0x00 (0 <sub>dec</sub> )
10F3:02	Newest Message		UINT8	RO	0x00 (0 <sub>dec</sub> )
10F3:03	Newest Acknowledged Message		UINT8	RW	0x00 (0 <sub>dec</sub> )
10F3:04	New Messages Available		BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
10F3:05	Flags		UINT16	RW	0x0000 (0 <sub>dec</sub> )
10F3:06	Diagnosis Message 001		OCTET-STRING[27]	RO	{0}
10F3:07	Diagnosis Message 002		OCTET-STRING[27]	RO	{0}
10F3:08	Diagnosis Message 003		OCTET-STRING[27]	RO	{0}
10F3:09	Diagnosis Message 004		OCTET-STRING[27]	RO	{0}
10F3:0A	Diagnosis Message 005		OCTET-STRING[27]	RO	{0}
10F3:0B	Diagnosis Message 006		OCTET-STRING[27]	RO	{0}
10F3:0C	Diagnosis Message 007		OCTET-STRING[27]	RO	{0}
10F3:0D	Diagnosis Message 008		OCTET-STRING[27]	RO	{0}
10F3:0E	Diagnosis Message 009		OCTET-STRING[27]	RO	{0}
10F3:0F	Diagnosis Message 010		OCTET-STRING[27]	RO	{0}
10F3:10	Diagnosis Message 011		OCTET-STRING[27]	RO	{0}
10F3:11	Diagnosis Message 012		OCTET-STRING[27]	RO	{0}
10F3:12	Diagnosis Message 013		OCTET-STRING[27]	RO	{0}
10F3:13	Diagnosis Message 014		OCTET-STRING[27]	RO	{0}
10F3:14	Diagnosis Message 015		OCTET-STRING[27]	RO	{0}
10F3:15	Diagnosis Message 016		OCTET-STRING[27]	RO	{0}
10F3:16	Diagnosis Message 017		OCTET-STRING[27]	RO	{0}
10F3:17	Diagnosis Message 018		OCTET-STRING[27]	RO	{0}
10F3:18	Diagnosis Message 019		OCTET-STRING[27]	RO	{0}
10F3:19	Diagnosis Message 020		OCTET-STRING[27]	RO	{0}
10F3:1A	Diagnosis Message 021		OCTET-STRING[27]	RO	{0}
10F3:1B	Diagnosis Message 022		OCTET-STRING[27]	RO	{0}
10F3:1C	Diagnosis Message 023		OCTET-STRING[27]	RO	{0}
10F3:1D	Diagnosis Message 024		OCTET-STRING[27]	RO	{0}
10F3:1E	Diagnosis Message 025		OCTET-STRING[27]	RO	{0}

**Index 10F8 Timestamp Object**

Index (hex)	Name	Meaning	Data type	Flags	Default
10F8:0	Timestamp Object		UINT64	RO	

**Index 10F9 Time Distribution Object**

Index (hex)	Name	Meaning	Data type	Flags	Default
10F9:0	Time Distribution Object		UINT8	RO	0x1 (1 <sub>dec</sub> )
10F9:01	Distributed Time Value	This object can be used to inform the EtherCAT slave of the current time. The time of this object corresponds to the notation for 64-bit EtherCAT timestamps see chapter <a href="#">JSON [►_154]</a> .	UINT64	RW	

**Index 1600 PSU RxPDO-Map Outputs**

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	PSU RxPDO-Map Outputs	PDO Mapping RxPDO 1	UINT8	RO	0x02 (2 <sub>dec</sub> )
1600:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (PSU Outputs), entry 0x01 (Disable output))	UINT32	RO	0x7000:01, 1
1600:02	SubIndex 002	2. PDO Mapping entry (15 bits align)	UINT32	RO	0x0000:00, 15

**Index 1A00 PSU TxPDO-Map Inputs**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	PSU TxPDO-Map Inputs	PDO Mapping TxPDO 1	UINT8	RO	0x07 (7 <sub>dec</sub> )
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (PSU Inputs), entry 0x01 (Warning))	UINT32	RO	0x6000:01, 1
1A00:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (PSU Inputs), entry 0x02 (Error))	UINT32	RO	0x6000:02, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (1 bit align)	UINT32	RO	0x0000:00, 1
1A00:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (PSU Inputs), entry 0x04 (DC OK))	UINT32	RO	0x6000:04, 1
1A00:05	SubIndex 005	5. PDO Mapping entry (28 bits align)	UINT32	RO	0x0000:00, 28
1A00:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (PSU Inputs), entry 0x11 (Output voltage))	UINT32	RO	0x6000:11, 32
1A00:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (PSU Inputs), entry 0x12 (Output current))	UINT32	RO	0x6000:12, 32

**Index 1A01 PSU TxPDO-Map Synchron info data**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A01:0	PSU TxPDO-Map Synchron info data	PDO Mapping TxPDO 2	UINT8	RO	0x02 (2 <sub>dec</sub> )
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (PSU Inputs), entry 0x15 (Info data 1))	UINT32	RO	0x6000:15, 16
1A01:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (PSU Inputs), entry 0x16 (Info data 2))	UINT32	RO	0x6000:16, 16

**Index 1A80 PSU TxPDO-Map Inputs Device**

Index (hex)	Name	Meaning	Data type	Flags	Default
1A80:0	PSU TxPDO-Map Inputs Device	PDO Mapping TxPDO 2	UINT8	RO	0x03 (3 <sub>dec</sub> )
1A80:01	SubIndex 001	1. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A80:02	SubIndex 002	2. PDO Mapping entry (object 0xF614 (PSU Inputs), entry 0x03 (Input undervoltage))	UINT32	RO	0xF614:03, 1
1A80:03	SubIndex 003	3. PDO Mapping entry (13 bits align)	UINT32	RO	0x0000:00, 13



**Index 1C00 Sync manager type**

Index (hex)	Name	Meaning	Data type	Flags	Default
1C00:0	Sync manager type	Using the Sync Managers	UINT8	RO	0x04 (4 <sub>dec</sub> )
1C00:01	SubIndex 001	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01 (1 <sub>dec</sub> )
1C00:02	SubIndex 002	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02 (2 <sub>dec</sub> )
1C00:03	SubIndex 003	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	0x03 (3 <sub>dec</sub> )
1C00:04	SubIndex 004	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	0x04 (4 <sub>dec</sub> )

**Index 1C12 RxPDO assign**

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RO	0x01 (1 <sub>dec</sub> )
1C12:01	Subindex 001		UINT16	RO	

**Index 1C13 TxPDO assign**

For operation on masters other than TwinCAT it must be ensured that the channels are entered in the PDO assignment (“TxPDO assign”, object 0x1C13) successively.

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x01 (1 <sub>dec</sub> )
1C13:01	Subindex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656 <sub>dec</sub> )
1C13:02	Subindex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A80 (6784 <sub>dec</sub> )

## Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32 <sub>dec</sub> )
1C32:01	Sync mode	Current synchronization mode: <ul style="list-style-type: none"> <li>0: Free Run</li> <li>1: Synchron with SM 2 Event</li> <li>2: DC-Mode - Synchron with SYNC0 Event</li> <li>3: DC-Mode - Synchron with SYNC1 Event</li> </ul>	UINT16	RW	0x0000 (0 <sub>dec</sub> )
1C32:02	Cycle time	Cycle time (in ns): <ul style="list-style-type: none"> <li>Free Run: cycle time of the local timer</li> <li>Synchron with SM 2 Event: cycle time of the master</li> <li>DC-Mode: SYNC0/SYNC1 Cycle Time</li> </ul>	UINT32	RW	0x00000000 (0 <sub>dec</sub> )
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:04	Sync modes supported	Supported synchronization modes: <ul style="list-style-type: none"> <li>Bit 0 = 1: Free Run is supported</li> <li>Bit 1 = 1: Synchron with SM 2 Event is supported</li> <li>Bit 2-3 = 01: DC-Mode is supported</li> <li>Bit 4-5 = 10: Output Shift with SYNC1 Event (only DC mode)</li> <li>Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08)</li> </ul>	UINT16	RO	0x0001 (1 <sub>dec</sub> )
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:07	Minimum delay time		UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:08	Get Cycle Time	<ul style="list-style-type: none"> <li>0: Measurement of the local cycle time is stopped</li> <li>1: Measurement of the local cycle time is started</li> </ul> <p>The entries 0x1C32:03, 0x1C32:05, 0x1C32:06, 0x1C32:09, 0x1C33:03, 0x1C33:06, 0x1C33:09 [► 153] are updated with the maximum measured values. For a subsequent measurement the measured values are reset</p>	UINT16	RW	0x0000 (0 <sub>dec</sub> )
1C32:09	Maximum delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C32:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )

**Index 1C33 SM input parameter**

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32 <sub>dec</sub> )
1C33:01	Sync mode	Current synchronization mode: <ul style="list-style-type: none"> <li>• 0: Free Run</li> <li>• 1: Synchron with SM 3 Event (no outputs available)</li> <li>• 2: DC - Synchron with SYNC0 Event</li> <li>• 3: DC - Synchron with SYNC1 Event</li> <li>• 34: Synchron with SM 2 Event (outputs available)</li> </ul>	UINT16	RW	0x0000 (0 <sub>dec</sub> )
1C33:02	Cycle time	Cycle time (in ns): <ul style="list-style-type: none"> <li>• Free Run: cycle time of the local timer</li> <li>• Synchron with SM 2 Event: cycle time of the master</li> <li>• DC-Mode: SYNC0/SYNC1 Cycle Time</li> </ul>	UINT32	RW	0x000F4240 (1000000 <sub>dec</sub> )
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:04	Sync modes supported	Supported synchronization modes: <ul style="list-style-type: none"> <li>• Bit 0: Free Run is supported</li> <li>• Bit 1: Synchron with SM 2 Event is supported (outputs available)</li> <li>• Bit 1: Synchron with SM 3 Event is supported (no outputs available)</li> <li>• Bit 2-3 = 01: DC-Mode is supported</li> <li>• Bit 4-5 = 01: Input shift through local event (outputs available)</li> <li>• Bit 4-5 = 10: Input shift with SYNC1 event (no outputs available)</li> <li>• Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08 [▶ 152] or 0x1C33:08)</li> </ul>	UINT16	RO	0x0001 (1 <sub>dec</sub> )
1C33:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x000186A0 (100000 <sub>dec</sub> )
1C33:06	Calc and copy time	Time between reading of the inputs and the inputs being available for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:07	Minimum delay time		UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:08	Get Cycle Time	<ul style="list-style-type: none"> <li>• 0: Measurement of the local cycle time is stopped</li> <li>• 1: Measurement of the local cycle time is started</li> </ul> <p>The entries 0x1C32:03, 0x1C32:05, 0x1C32:06, 0x1C32:09 [▶ 152], 0x1C33:03, 0x1C33:06, 0x1C33:09 are updated with the maximum measured values. For a subsequent measurement the measured values are reset</p>	UINT16	RW	0x0000 (0 <sub>dec</sub> )
1C33:09	Maximum delay time	Time between SYNC1 event and reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C33:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C33:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C33:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )

## 6.8 JSON

The power supplies of the PS2000 series with EtherCAT interface have the option of retrieving a log file in the form of a JSON document. This contains, for example, the currently set threshold values, minimum and maximum voltages and currents that have occurred, as well as an error history. The file also contains static product properties such as the nominal current, nominal voltage, product name and a unique serial number (BIC [[▶ 13](#)]/eBIC [[▶ 15](#)]).

The error history contains up to ten undervoltage, overvoltage, overcurrent and overtemperature events. The value of the over/underrun and the time of occurrence are stored for each event.

### Exemplary representation of the JSON file.

```
{
  "Product": "PS2001-4810",
  "BIC": "1P551049SBTNasdfghjklKPS2001-4810-1001          Q1    52SP12345678          ",
  "NominalCurrent": 20.000000,
  "NominalVoltage": 24.000000,
  "MinInputVoltage": 228.000000,
  "MaxInputVoltage": 240.000000,
  "MaxPeakInputVoltage": 340.000000,
  "InputUndervoltageCounter": 0,
  "InputOvervoltageCounter": 0,
  "MaxOutputVoltage": 28.531250,
  "MaxOutputCurrent": 0.062500,
  "MinTemperature": 18.656250,
  "MaxTemperature": 30.156250,
  "DcOkThreshold": 23.000000,
  "OvervoltageWarnThreshold": 28.799999,
  "OvercurrentWarnThreshold": 24.000000,
  "OvertemperatureWarnThreshold": 120.000000,
  "UndervoltageEvents":
  [
    {
      "Value": 22.386719,
      "DcTimestamp": 695747772293000000,
      "Time": "2022-01-17T15:16:12"
    }
  ]
}
```

### Reading the JSON log file

The file can be read via a FoE upload

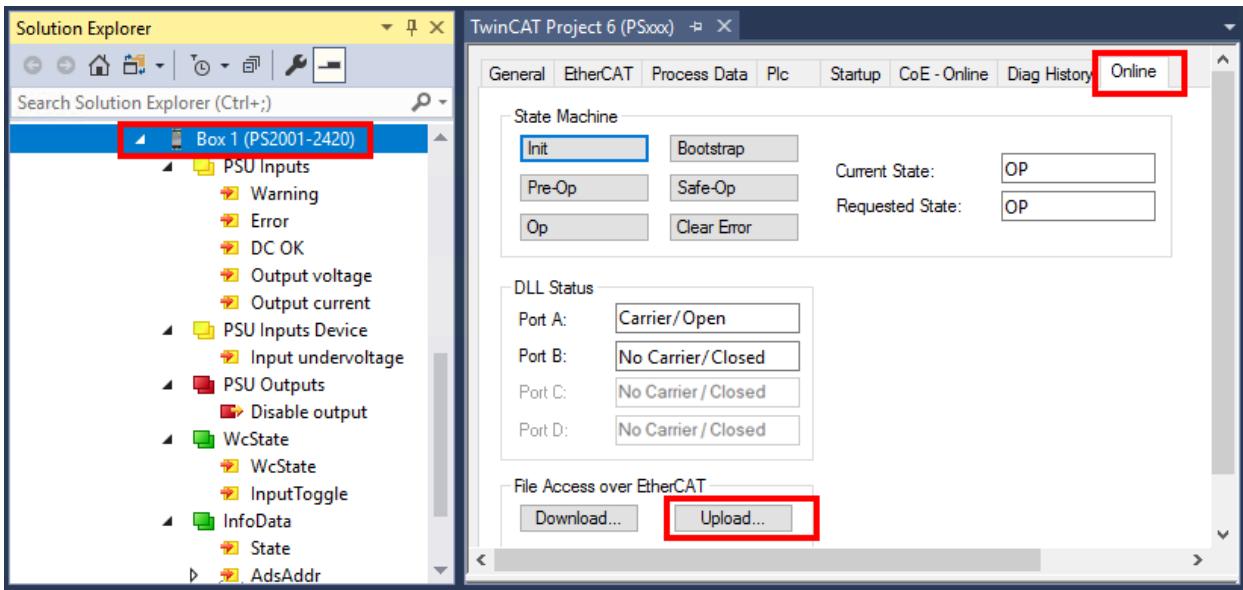


Fig. 137: Reading the JSON log file using the PS2001-2420-1001 as an example

- In the following dialog, enter "Logfile.json" as the file name and press "Save".
- Confirm the following dialog "Edit FoE Name" with "OK".

### Update the time for the JSON file

The power supply has no information on the current local time. For this reason, the time of the events is always displayed as 0 by default.

The CoE object 0x10F9:01 can be used to inform the EtherCAT slave of the current time. The time of this object corresponds to the notation for 64-bit EtherCAT timestamps.

Theoretically, it is sufficient to transfer the current time once after switching on. The internal clock of the power supply unit is set and continues to run automatically. However, to ensure that the time of the master and the power supply unit do not drift apart, a regular update is recommended. (e.g. every minute).

```

MAIN  X Source Control Explorer
1  PROGRAM MAIN
2  VAR
3      ActTime      : T_DCTIME64;
4      Delay        : TON;
5      CoeWrite     : FB_EcCoESdoWrite;
6      AmsAddr AT %I* : ST_AmsAddr;
7  END_VAR
8
9  Delay(IN:=NOT Delay.Q, PT:=T#60S);
10
11 ActTime := F_GetActualDcTime64();
12
13 IF Delay.Q THEN
14     CoeWrite(bExecute := TRUE);
15 END_IF
16
17 CoeWrite.bExecute := FALSE;;
18 CoeWrite.cbBufLen := SIZEOF(T_DCTIME);
19 CoeWrite.nIndex := 16#10F9;
20 CoeWrite.nSubIndex := 16#01;
21 CoeWrite.nSlaveAddr := AmsAddr.port;
22 CoeWrite.sNetId := F_CreateAmsNetId(AmsAddr.netId);
23 CoeWrite.pSrcBuf := ADR(ActTime);
24 CoeWrite();

```

Fig. 138: Sample code for writing the time every minute

### PLC library:Tc3\_JsonXml

With the aid of the [PLC library Tc3\\_JsonXml](#), SAX and DOM parser technologies can be used to create and navigate through JSON and XML documents.

## 7 Disposal



Products marked with a crossed-out wheeled bin shall not be discarded with the normal waste stream. The device is considered as waste electrical and electronic equipment. The national regulations for the disposal of waste electrical and electronic equipment must be observed.

## 8 Appendix

### 8.1 EtherCAT AL Status Codes

For detailed information please refer to the [EtherCAT system description](#).

### 8.2 Firmware update PSxxxx

This section describes the device update for Beckhoff EtherCAT slaves from the EL/ES, ELM, EM, EK and EP series. A firmware update should only be carried out after consultation with Beckhoff support.

#### NOTICE

##### Only use TwinCAT 3 software!

A firmware update of Beckhoff IO devices must only be performed with a TwinCAT 3 installation. It is recommended to build as up-to-date as possible, available for free download on the Beckhoff website <https://www.beckhoff.com/en-us/>.

To update the firmware, TwinCAT can be operated in the so-called FreeRun mode, a paid license is not required.

The device to be updated can usually remain in the installation location, but TwinCAT has to be operated in the FreeRun. Please make sure that EtherCAT communication is trouble-free (no LostFrames etc.).

Other EtherCAT master software, such as the EtherCAT Configurator, should not be used, as they may not support the complexities of updating firmware, EEPROM and other device components.

#### Storage locations

An EtherCAT slave stores operating data in up to three locations:

- Each EtherCAT slave has a device description, consisting of identity (name, product code), timing specifications, communication settings, etc.  
This device description (ESI; EtherCAT Slave Information) can be downloaded from the Beckhoff website in the download area as a [zip file](#) and used in EtherCAT masters for offline configuration, e.g. in TwinCAT.  
Above all, each EtherCAT slave carries its device description (ESI) electronically readable in a local memory chip, the so-called **ESI EEPROM**. When the slave is switched on, this description is loaded locally in the slave and informs it of its communication configuration; on the other hand, the EtherCAT master can identify the slave in this way and, among other things, set up the EtherCAT communication accordingly.

#### NOTICE

##### Application-specific writing of the ESI-EEPROM

The ESI is developed by the device manufacturer according to ETG standard and released for the corresponding product.

- Meaning for the ESI file: Modification on the application side (i.e. by the user) is not permitted.
- Meaning for the ESI EEPROM: Even if a writeability is technically given, the ESI parts in the EEPROM and possibly still existing free memory areas must not be changed beyond the normal update process. Especially for cyclic memory processes (operating hours counter etc.), dedicated memory products such as EL6080 or IPC's own NOVDRAM must be used.

- Depending on functionality and performance EtherCAT slaves have one or several local controllers for processing I/O data. The corresponding program is the so-called **firmware** in \*.efw format.
- In some EtherCAT slaves the EtherCAT communication may also be integrated in these controllers. In this case the controller is usually a so-called **FPGA** chip with \*.rbf firmware.

Customers can access the data via the EtherCAT fieldbus and its communication mechanisms. Acyclic mailbox communication or register access to the ESC is used for updating or reading of these data.



The TwinCAT System Manager offers mechanisms for programming all three parts with new data, if the slave is set up for this purpose. Generally the slave does not check whether the new data are suitable, i.e. it may no longer be able to operate if the data are unsuitable.

### Simplified update by bundle firmware

The update using so-called **bundle firmware** is more convenient: in this case the controller firmware and the ESI description are combined in a \*.efw file; during the update both the firmware and the ESI are changed in the terminal. For this to happen it is necessary

- for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. ELxxx-xxx\_REV0016\_SW01.efw
- for password=1 to be entered in the download dialog. If password=0 (default setting) only the firmware update is carried out, without an ESI update.
- for the device to support this function. The function usually cannot be retrofitted; it is a component of many new developments from year of manufacture 2016.

Following the update, its success should be verified

- ESI/Revision: e.g. by means of an online scan in TwinCAT ConfigMode/FreeRun – this is a convenient way to determine the revision
- Firmware: e.g. by looking in the online CoE of the device

#### NOTICE

##### Risk of damage to the device!

- ✓ Note the following when downloading new device files
  - a) Firmware downloads to an EtherCAT device must not be interrupted
  - b) Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.
  - c) The power supply must adequately dimensioned. The signal level must meet the specification.
- ⇒ In the event of malfunctions during the update process the EtherCAT device may become unusable and require re-commissioning by the manufacturer.

## 8.2.1 Device description ESI file/XML

#### NOTICE

##### Attention regarding update of the ESI description/EEPROM

Some slaves have stored calibration and configuration data from the production in the EEPROM. These are irretrievably overwritten during an update.

The ESI device description is stored locally on the slave and loaded on start-up. Each device description has a unique identifier consisting of slave name (9 characters/digits) and a revision number (4 digits). Each slave configured in the System Manager shows its identifier in the EtherCAT tab:

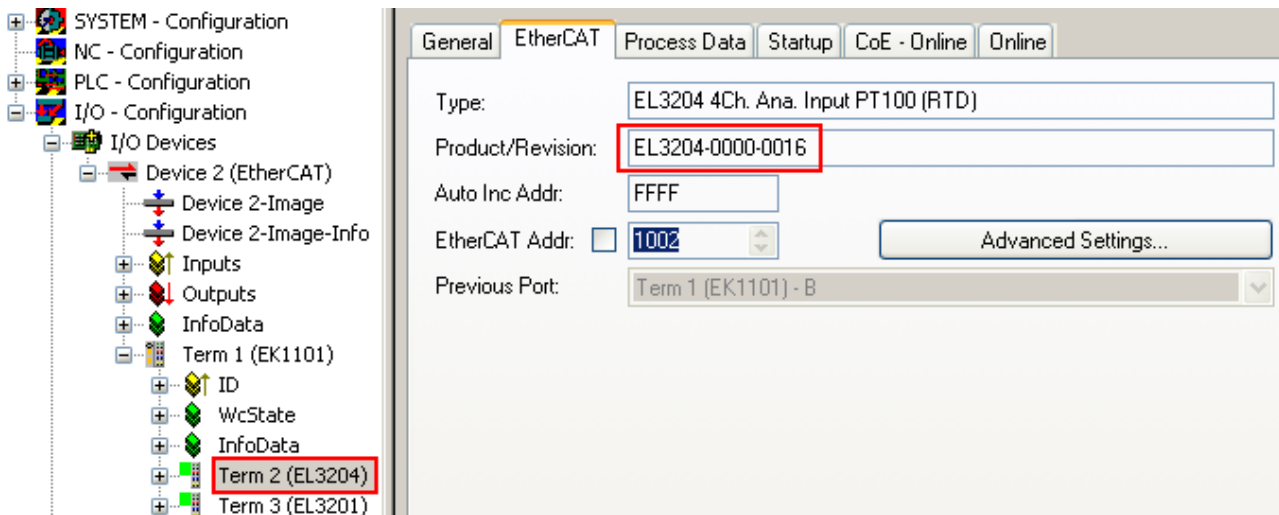


Fig. 139: Device identifier consisting of name EL3204-0000 and revision -0016

The configured identifier must be compatible with the actual device description used as hardware, i.e. the description which the slave has loaded on start-up (in this case EL3204). Normally the configured revision must be the same or lower than that actually present in the terminal network.

For further information on this, please refer to the [EtherCAT system documentation](#).

### **i** Update of XML/ESI description

The device revision is closely linked to the firmware and hardware used. Incompatible combinations lead to malfunctions or even final shutdown of the device. Corresponding updates should only be carried out in consultation with Beckhoff support.

### Display of ESI slave identifier

The simplest way to ascertain compliance of configured and actual device description is to scan the EtherCAT boxes in TwinCAT mode Config/FreeRun:

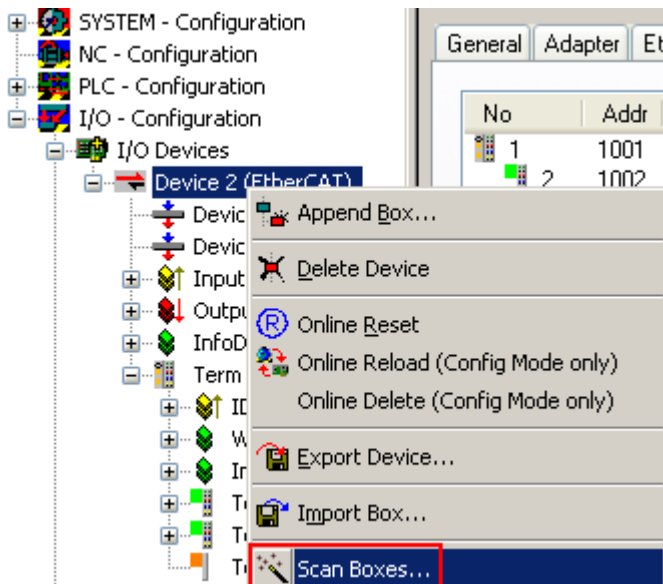


Fig. 140: Scan the subordinate field by right-clicking on the EtherCAT device

If the found field matches the configured field, the display shows



Fig. 141: Configuration is identical

otherwise a change dialog appears for entering the actual data in the configuration.

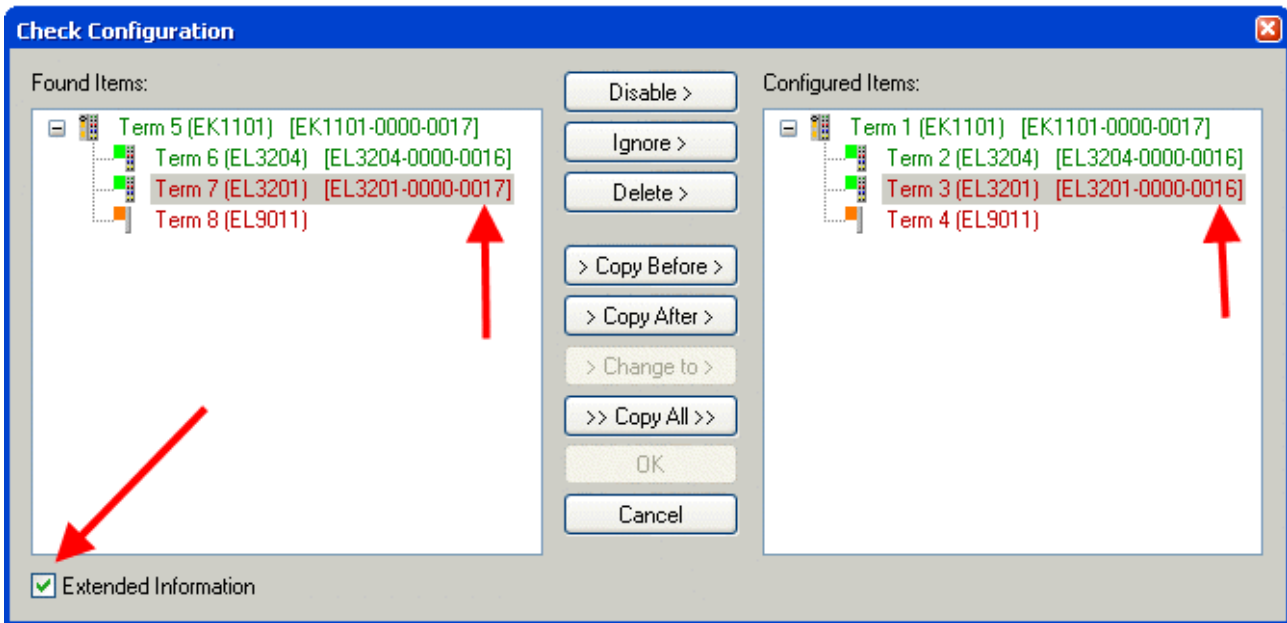


Fig. 142: Change dialog

In this example in Fig. *Change dialog*, an EL3201-0000-0017 was found, while an EL3201-0000-0016 was configured. In this case the configuration can be adapted with the *Copy Before* button. The *Extended Information* checkbox must be set in order to display the revision.

### Changing the ESI slave identifier

The ESI/EEPROM identifier can be updated as follows under TwinCAT:

- Trouble-free EtherCAT communication must be established with the slave.
- The state of the slave is irrelevant.
- Right-clicking on the slave in the online display opens the *EEPROM Update* dialog, Fig. *EEPROM Update*

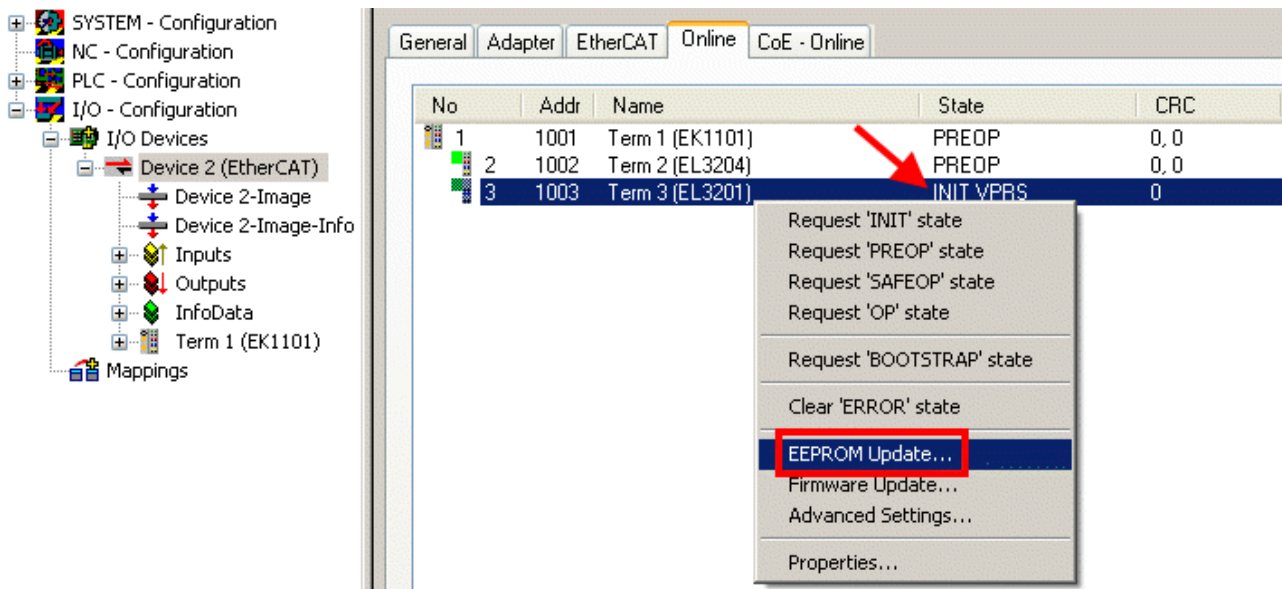


Fig. 143: EEPROM Update

The new ESI description is selected in the following dialog, see Fig. *Selecting the new ESI*. The checkbox *Show Hidden Devices* also displays older, normally hidden versions of a slave.

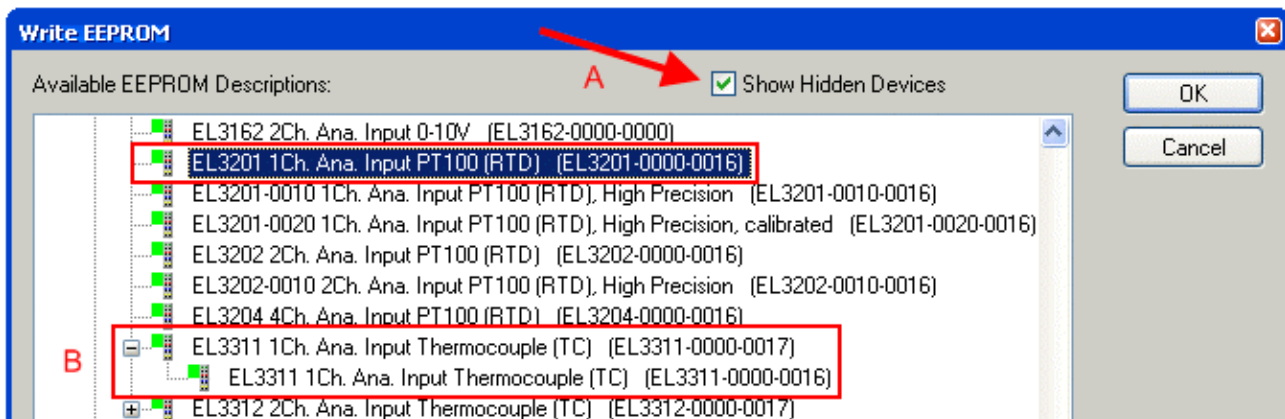


Fig. 144: Selecting the new ESI

A progress bar in the System Manager shows the progress. Data are first written, then verified.

### ● The change only takes effect after a restart.

**i** Most EtherCAT devices read a modified ESI description immediately or after startup from the INIT. Some communication settings such as distributed clocks are only read during power-on. The EtherCAT slave therefore has to be switched off briefly in order for the change to take effect.

## 8.2.2 Firmware explanation

### Determining the firmware version

#### Determining the version via the System Manager

The TwinCAT System Manager shows the version of the controller firmware if the master can access the slave online. Click on the E-Bus Terminal whose controller firmware you want to check (in the example terminal 2 (EL3204)) and select the tab *CoE Online* (CAN over EtherCAT).

**i CoE Online and Offline CoE**

Two CoE directories are available:

- **online:** This is offered in the EtherCAT slave by the controller, if the EtherCAT slave supports this. This CoE directory can only be displayed if a slave is connected and operational.
  - **offline:** The EtherCAT Slave Information ESI/XML may contain the default content of the CoE. This CoE directory can only be displayed if it is included in the ESI (e.g. “Beckhoff EL5xxx.xml”).
- The Advanced button must be used for switching between the two views.

In Fig. *Display of EL3204 firmware version* the firmware version of the selected EL3204 is shown as 03 in CoE entry 0x100A.

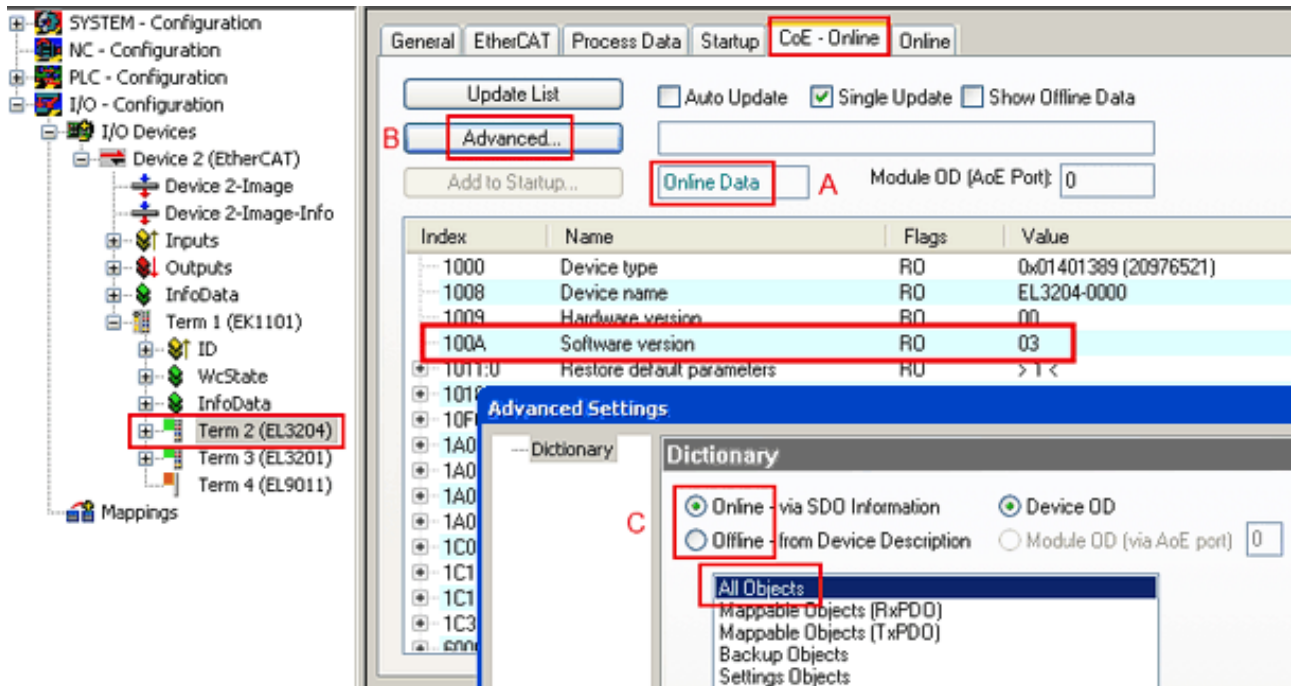


Fig. 145: Display of EL3204 firmware version

In (A) TwinCAT 2.11 shows that the Online CoE directory is currently displayed. If this is not the case, the Online directory can be loaded via the *Online* option in Advanced Settings (B) and double-clicking on *AllObjects*.

**8.2.3 Updating controller firmware \*.efw**

**i CoE directory**

The Online CoE directory is managed by the controller and stored in a dedicated EEPROM, which is generally not changed during a firmware update.

Switch to the *Online* tab to update the controller firmware of a slave, see Fig. *Firmware Update*.

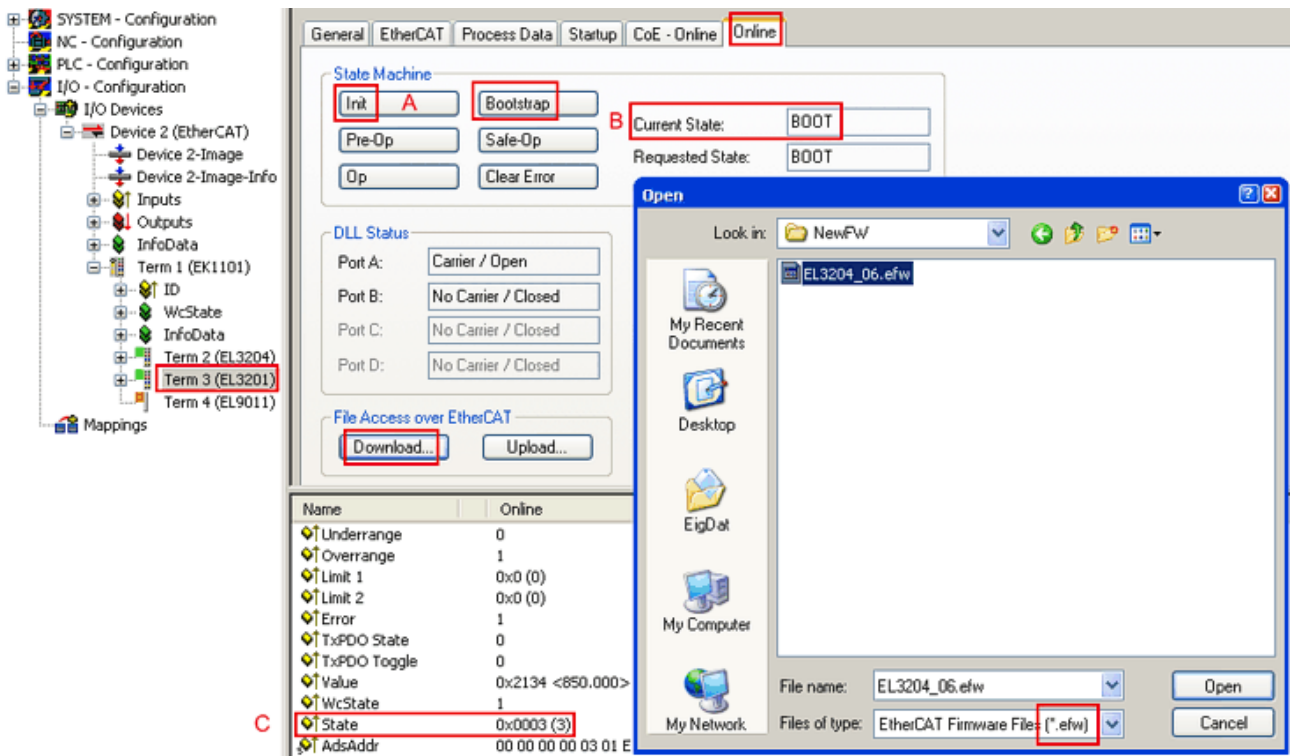
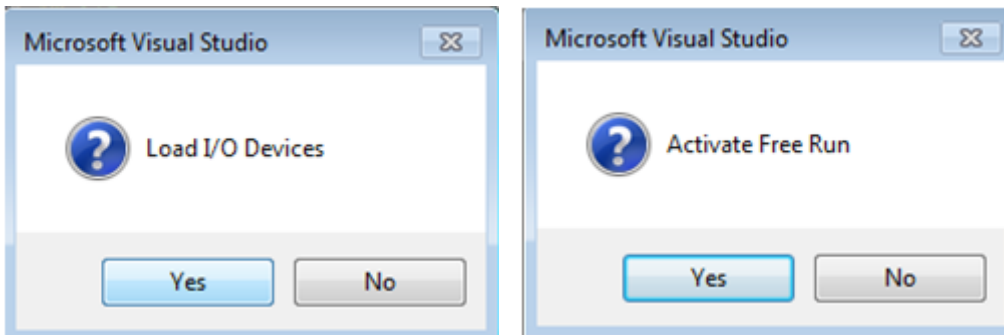


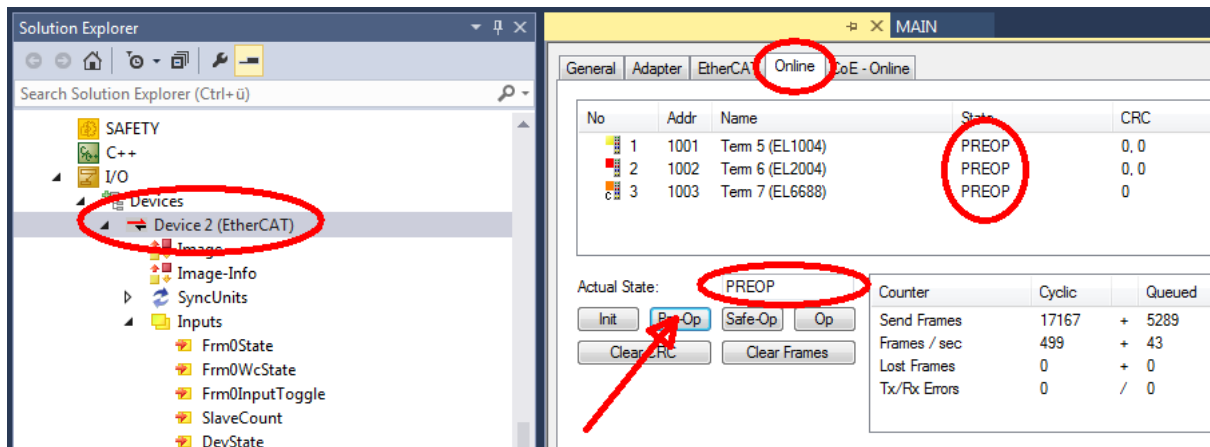
Fig. 146: Firmware Update

Proceed as follows, unless instructed otherwise by Beckhoff support. Valid for TwinCAT 2 and 3 as EtherCAT master.

- Switch TwinCAT system to ConfigMode/FreeRun with cycle time  $\geq 1$  ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.

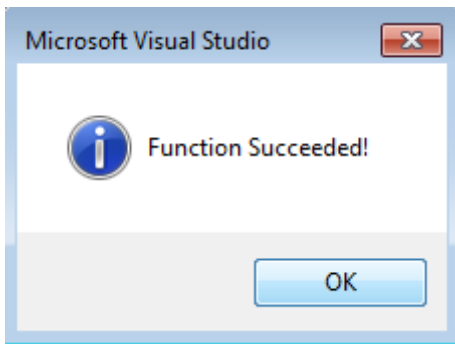


- Switch EtherCAT Master to PreOP



- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP

- Check the current status (B, C)
- Download the new \*efw file (wait until it ends). A password will not be necessary usually.



- After the download switch to INIT, then PreOP
- Switch off the slave briefly (don't pull under voltage!)
- Check within CoE 0x100A, if the FW status was correctly overtaken.

### 8.2.4 Simultaneous updating of several EtherCAT devices

The firmware and ESI descriptions of several devices can be updated simultaneously, provided the devices have the same firmware file/ESI.

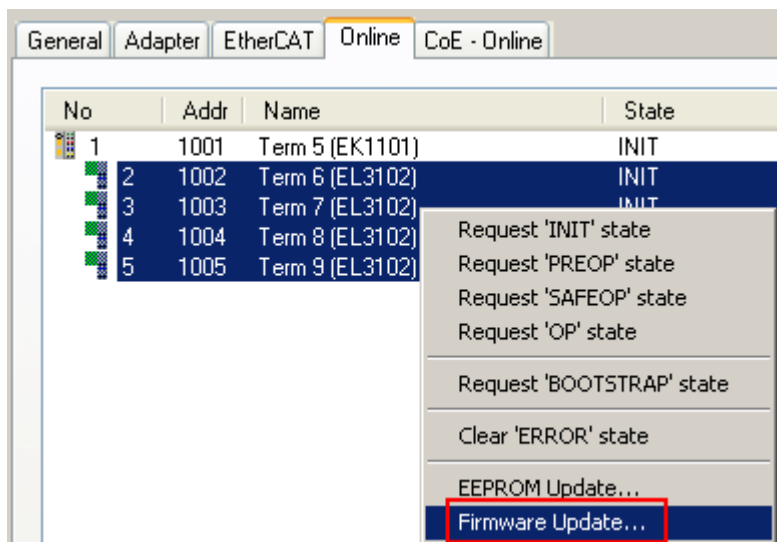


Fig. 147: Multiple selection and firmware update

Select the required slaves and carry out the firmware update in BOOTSTRAP mode as described above.



## 8.3 Restoring the delivery state

To restore the delivery state (factory settings) of CoE objects for EtherCAT devices (“slaves”), the CoE object *Restore default parameters*, SubIndex 001 can be used via EtherCAT master (e.g. TwinCAT) (see Fig. *Selecting the Restore default parameters PDO*).

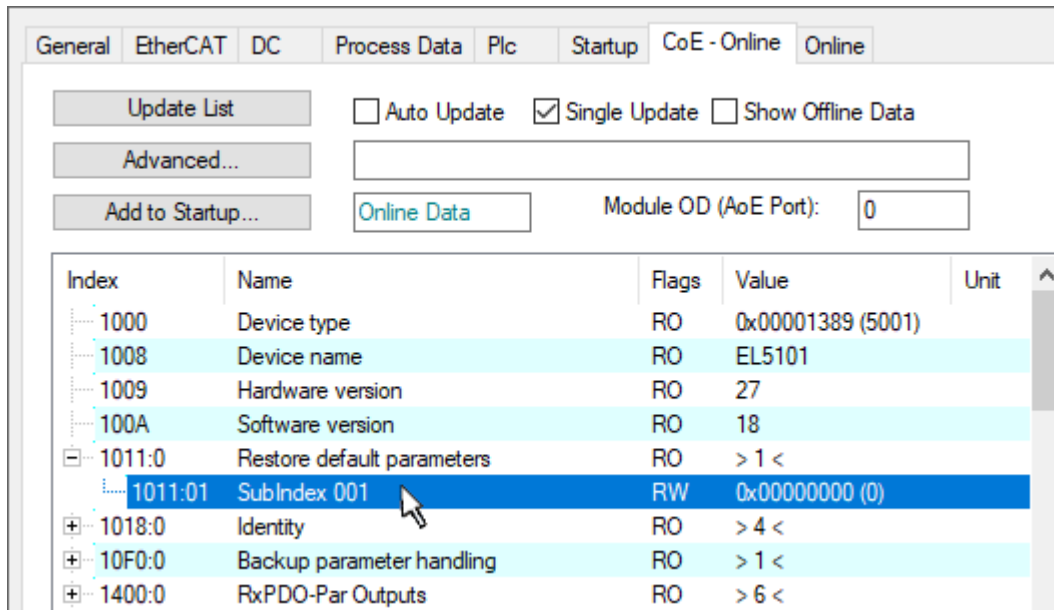


Fig. 148: Selecting the *Restore default parameters* PDO

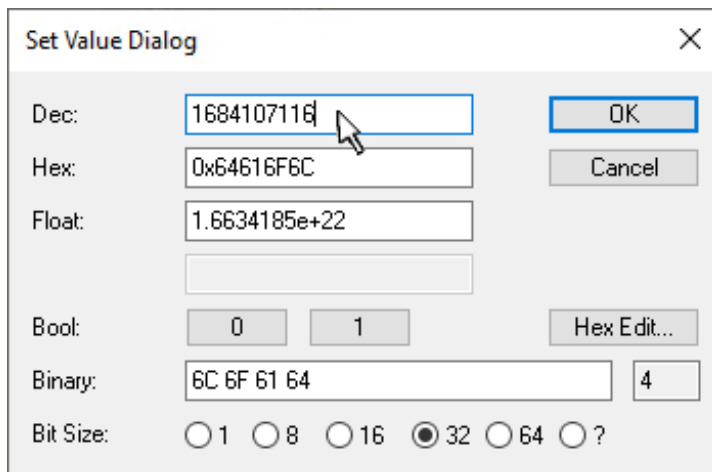


Fig. 149: Entering a restore value in the Set Value dialog

Double-click on *SubIndex 001* to enter the Set Value dialog. Enter the reset value **1684107116** in field *Dec* or the value **0x64616F6C** in field *Hex* (ASCII: “load”) and confirm with *OK* (Fig. *Entering a restore value in the Set Value dialog*).

- All changeable entries in the slave are reset to the default values.
- The values can only be successfully restored if the reset is directly applied to the online CoE, i.e. to the slave. No values can be changed in the offline CoE.
- TwinCAT must be in the RUN or CONFIG/Freerun state for this; that means EtherCAT data exchange takes place. Ensure error-free EtherCAT transmission.
- No separate confirmation takes place due to the reset. A changeable object can be manipulated beforehand for the purposes of checking.
- This reset procedure can also be adopted as the first entry in the startup list of the slave, e.g. in the state transition PREOP->SAFEOP or, as in Fig. *CoE reset as a startup entry*, in SAFEOP->OP.

All backup objects are reset to the delivery state.



**i Alternative restore value**

In some older terminals (FW creation approx. before 2007) the backup objects can be switched with an alternative restore value: Decimal value: 1819238756, Hexadecimal value: 0x6C6F6164.

An incorrect entry for the restore value has no effect.

## 8.4 Accessories

### 8.4.1 Redundancy and Buffer Modules

Power supply	Accessories		
	Redundancy module	Buffer module	UPS component
PS2001-2420-1001	PS9401-2440-0000	PS9011-2420-0001	CU8130-0xxx

#### PS9401-2440-0000 – Redundancy Module



The PS9401-2440-0000 is a redundancy module that can be used for the construction of 1+1 and N+1 redundancy systems.

It has two input channels, to which power supplies with output currents of up to 20 A can be connected, and one output that can carry nominal currents of up to 40 A.

The redundancy module uses MOSFET technology instead of diodes for decoupling the two input channels. This reduces the heat development and the voltage drop between input and output. Due to the low power loss, the device is very narrow and requires a width of just 36 mm on the DIN rail.

Large connection terminals enable secure and fast installation.

The redundancy module requires no additional auxiliary voltage.

Due to the international approvals package, the device is suitable for virtually every application.

Observe the connection instructions in the chapter "[Parallel use for redundancy \[► 42\]](#)"

#### PS9011-2420-0001 – Buffer Module



The PS9011-2420-0001 buffer module is an additional device for 24 V DC power supply units. It supplies power to bridge typical mains power failures or extends the hold-up time after switching off the AC voltage.

If the power supply unit supplies sufficient voltage, the buffer module stores energy in the integrated electrolytic capacitors. In the event of a failure of the mains voltage, the stored energy is supplied to the DC bus in a controlled process.

The buffer module can be connected at any point in parallel with the load circuit and does not require any control wiring.

A buffer module can provide 20 A of additional power and can be added in parallel to increase the output current or hold-up time.

The PS9011-2440-0000 buffer module can optionally be used to achieve longer hold-up times.

## 8.4.2 USV component

### CU8130-0xxx – UPS component



CU8130-0xxxs are battery-backed, uninterruptible power supplies (UPS). If the external 24 V DC input voltage is lost, the UPS takes over the supply of the devices connected to it thanks to its regulated and buffered 24 V DC output voltage.

The CU81xx UPS series is suitable for all Beckhoff components, particularly Industrial PCs, Embedded PCs, Panels and Panel PCs.

The main features of this UPS are:

- battery module based on NiMH cells
- mounting of the UPS on the DIN rail or on the rear panel of the control cabinet
- protocol-based communication with the Industrial PC either via OCT (**O**ne **C**able **T**echnology) or USB
- digital signals for communication with non-protocol-capable end devices
- TwinCAT PLC function blocks for querying the UPS operation

A special feature of the Beckhoff CU81xx devices is OCT (**O**ne **C**able **T**echnology) as communication technology between UPS and Industrial PC. This means that the two connecting lines (+24 V, 0 V) between Industrial PC and UPS are used not only to supply the Industrial PC, but also for bidirectional data transmission.

If both sides are OCT-capable, no further connection, e.g., via USB, is required.

Further information on UPS components can be found on the Beckhoff [homepage](#).

### 8.4.3 Accessories for mounting

#### ZS5301-0003 – Bracket for wall mounting

This bracket is used to mount the device on a level surface or panel without the use of a DIN rail. The bracket can be mounted without loosening the DIN rail brackets.

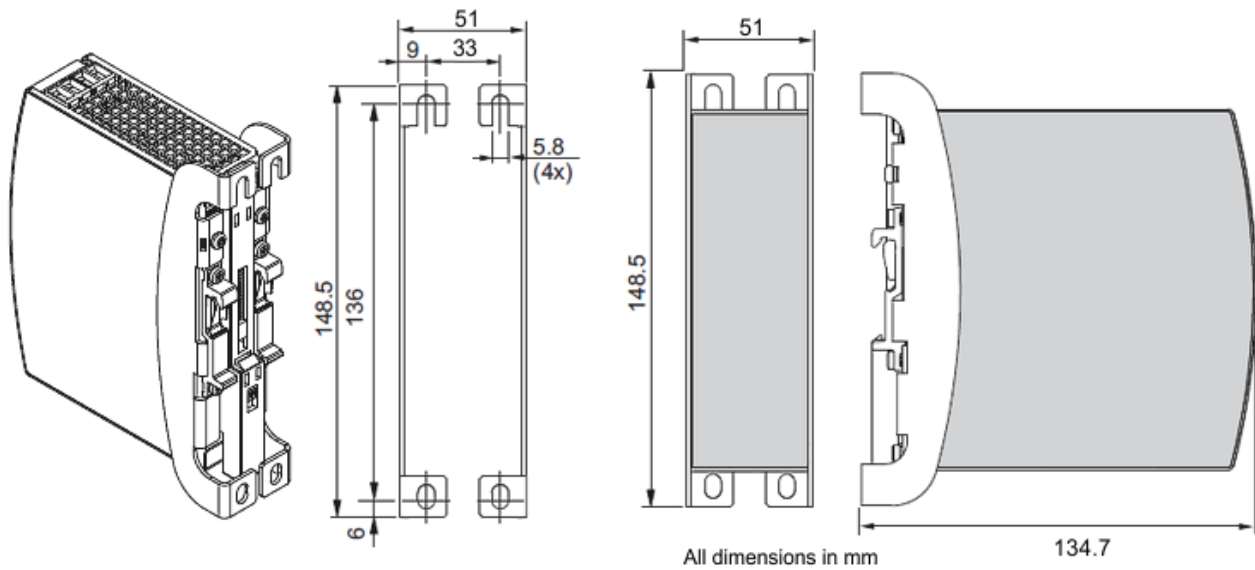


Fig. 150: Isometric view, installation dimensions bracket for wall mounting

## 8.5 Documentation issue status

Version	Comment
1.1	<ul style="list-style-type: none"><li>• Update chapter "DC-OK function (monitoring)"</li><li>• Update chapter "DC-OK function (parameterization)"</li></ul>
1.0	<ul style="list-style-type: none"><li>• First release</li></ul>
0.4	<ul style="list-style-type: none"><li>• Addenda and corrections</li><li>• Chapter "Installation position" added</li><li>• Chapter "Use in sealed housing" removed</li></ul>
0.3	<ul style="list-style-type: none"><li>• Chapter "Start-up behavior of the output voltage" added</li></ul>
0.2	<ul style="list-style-type: none"><li>• Chapter "Installation positions" removed</li></ul>
0.1	<ul style="list-style-type: none"><li>• Preliminary documentation for PS2001-2420-1001</li></ul>

## 8.6 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: [www.beckhoff.com](http://www.beckhoff.com)

You will also find further documentation for Beckhoff components there.

### Support

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

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

Hotline: +49 5246 963 157  
e-mail: [support@beckhoff.com](mailto:support@beckhoff.com)  
web: [www.beckhoff.com/support](http://www.beckhoff.com/support)

### Service

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

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

Hotline: +49 5246 963 460  
e-mail: [service@beckhoff.com](mailto:service@beckhoff.com)  
web: [www.beckhoff.com/service](http://www.beckhoff.com/service)

### Headquarters Germany

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20  
33415 Verl  
Germany

Phone: +49 5246 963 0  
e-mail: [info@beckhoff.com](mailto:info@beckhoff.com)  
web: [www.beckhoff.com](http://www.beckhoff.com)



More Information:

[www.beckhoff.com/ps2001-2420-1001](http://www.beckhoff.com/ps2001-2420-1001)

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Germany  
Phone: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

