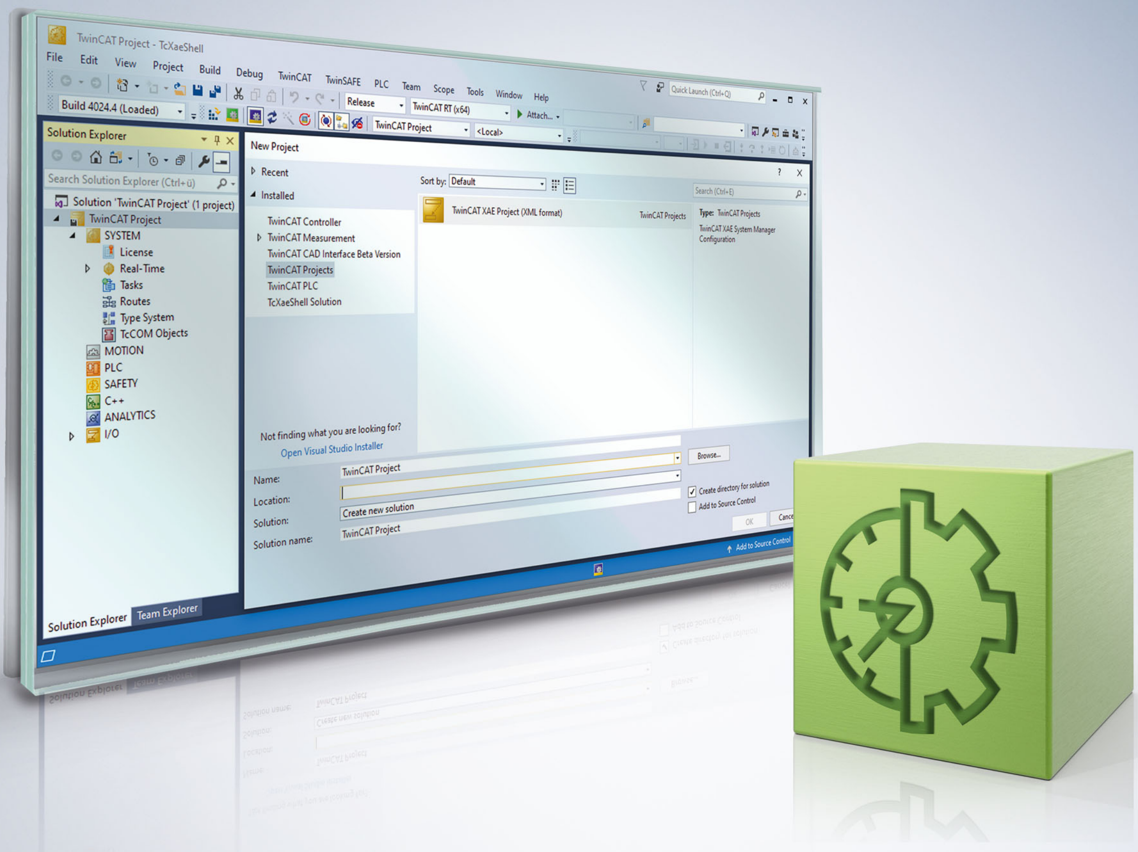
BECKHOFF New Automation Technology

Manual | EN

TwinCAT 3

The TwinCAT Project



2025-05-07 | Version: 1.1.5

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
The documentation and the following notes and explanations must be complied with when installing and commissioning the components.
The trained specialists must always use the current valid documentation.

The trained specialists must ensure that the application and use of the products described is in line with all safety requirements, including all relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been compiled with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without notice.
Claims to modify products that have already been supplied may not be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Copyright

Third-party trademarks

Trademarks of third parties may be used in this documentation. You can find the trademark notices here: https://www.beckhoff.com/trademarks.

1.2 For your safety

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Signal words

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

Personal injury warnings

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

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

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

Warning of damage to property or environment

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

Information on handling the product

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

1.3 Notes on information security

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

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

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

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

2 Overview

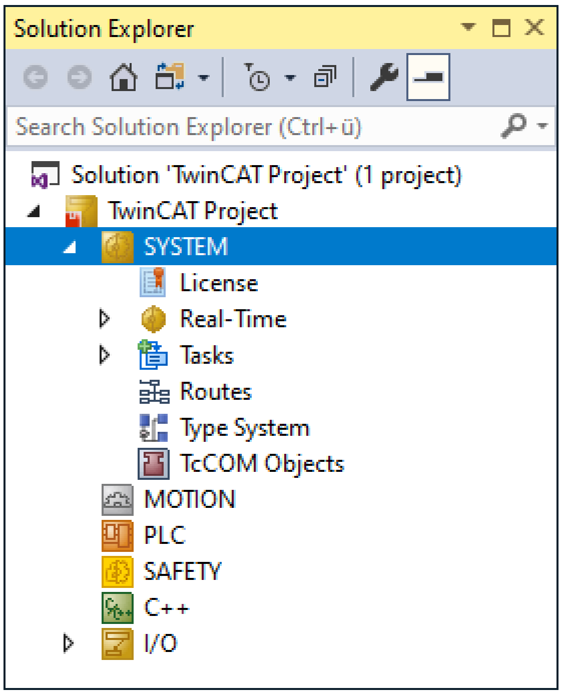
The TE1000 XAE (eXtended Automation Engineering) is the TwinCAT 3 development environment. Beckhoff controllers can be programmed, diagnosed and debugged with this software. A controller is thereby mapped by a TwinCAT project. Several TwinCAT projects can be combined in one Solution.

A TwinCAT project includes all settings relevant for commissioning and operating a controller, the sub-projects for creating and instantiating application programs (PLC and C++), and the description and parameterization of the I/Os in order to interact with the process to be controlled. The following documentation explains the individual parts of a TwinCAT project in more detail.

3 SYSTEM

Function: This node contains the system configuration. This contains, among other things, the real-time settings, the configuration of the tasks, the ADS routes as well as the type definitions and required licenses of the project.

Call: Click on the arrow in front of **SYSTEM**.

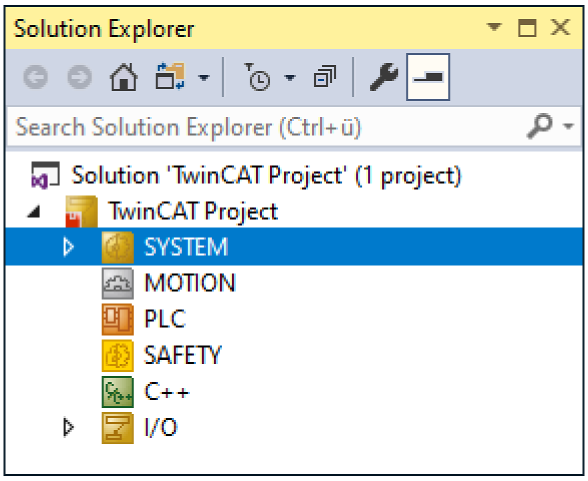


The individual sub-nodes are explained in more detail below.

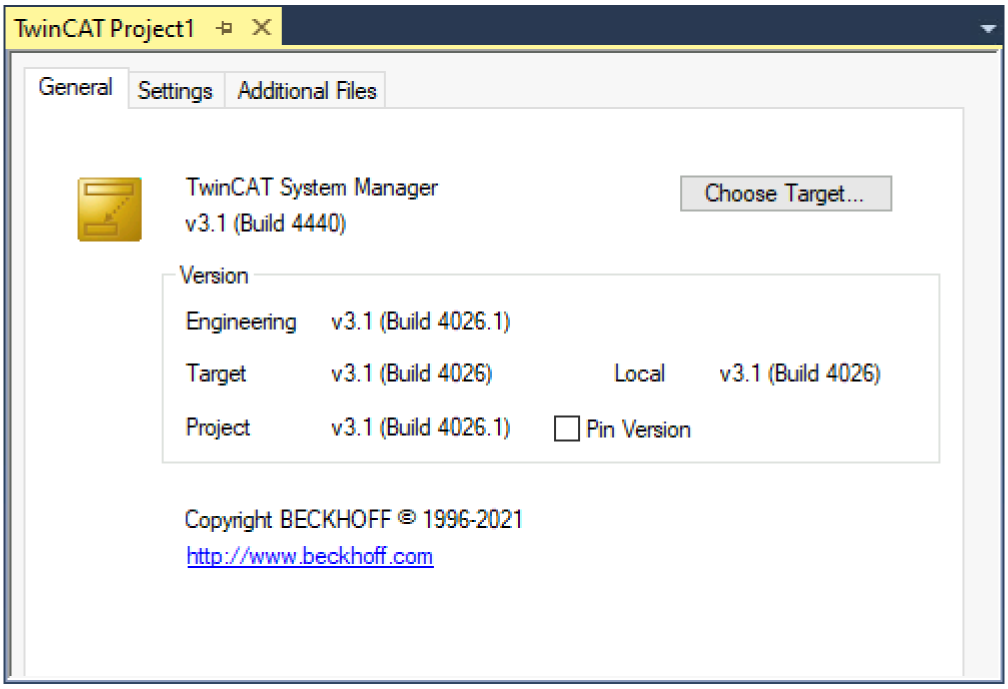
3.1 System

Function: This node contains general information and settings for the present TwinCAT project. These are presented in three tabs.

Call: Double-click on the node **SYSTEM**.


Solution Explorer
Search Solution Explorer (Ctrl+ü)
Solution 'TwinCAT Project' (1 project)
TwinCAT Project
SYSTEM
MOTION
PLC
SAFETY
C++
I/O

3.1.1 General tab


TwinCAT Project1

General | Settings | Additional Files

TwinCAT System Manager
v3.1 (Build 4440)

Choose Target...

Version

Engineering	v3.1 (Build 4026.1)		
Target	v3.1 (Build 4026)	Local	v3.1 (Build 4026)
Project	v3.1 (Build 4026.1)	☐ Pin Version	

Copyright BECKHOFF © 1996-2021
http://www.beckhoff.com

Choose Target..	Opens the **Choose Target System** dialog.

Version

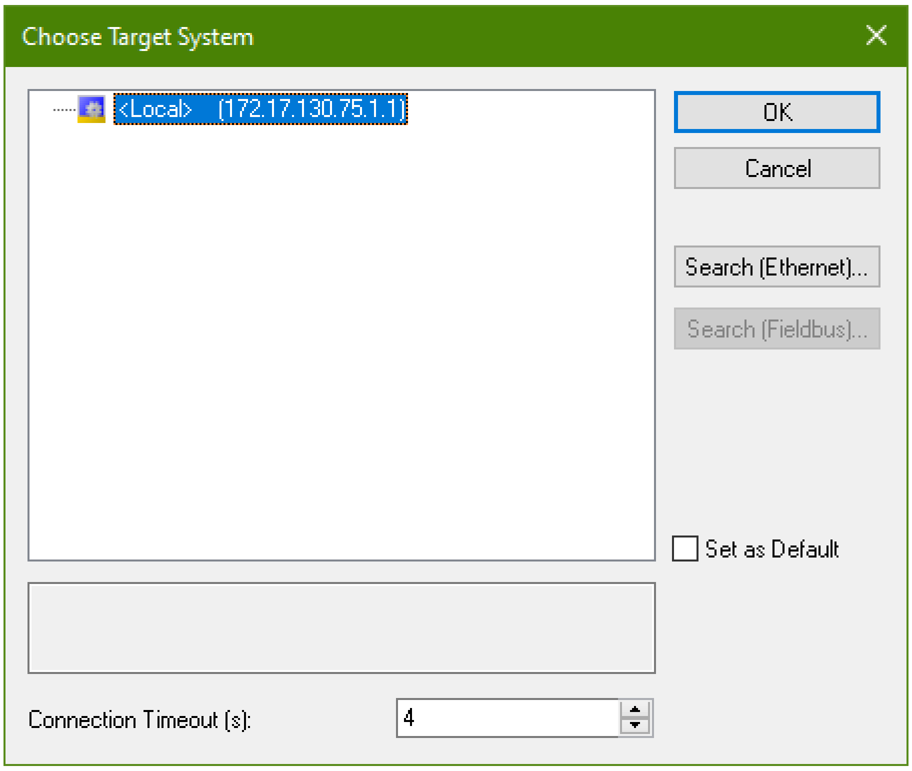
Engineering	TwinCAT Engineering version
Target	TwinCAT Runtime version of the target system
Project	TwinCAT version with which the project was created.
Local	Local TwinCAT Runtime version
Pin Version	Pins the currently used version of TwinCAT XAE Engineering to the project. If the project is opened by double-clicking from the File Explorer or with the TwinCAT version not yet selected (in the Remote Manager toolbar in the XAE no version is yet marked as "Loaded"), the pinned version is automatically loaded if it is installed. For more information, please visit in the PLC documentation the chapter Project delivery.

Choose Target System:

The TwinCAT 3 runtime system that is to be parameterized and programmed with the current project is referred to as the target system. To select the target system, proceed as follows:

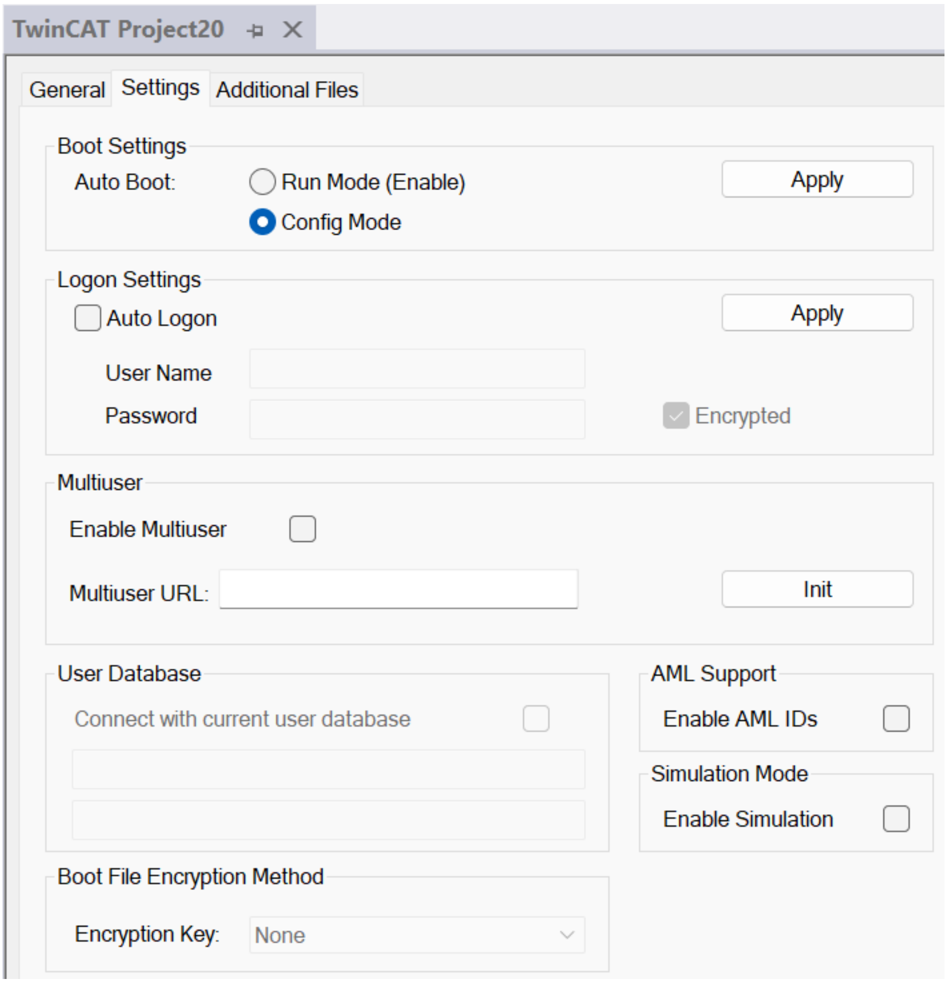
1. Press the **Choose Target...** button.

⇨ The **Choose Target System** dialog opens.



2. Select the target system you want to program from the list of target systems connected to your computer.
3. If the desired target system does not yet exist, add it to the list. See also Target systems.

3.1.2 Settings tab



Auto Boot: ( ) Run Mode (Enable) [● Config Mode]

Auto Logon: ☐

Encrypted: ✓

Enable Multiuser: ☐

Connect with current user database: ☐

Enable AML IDs: ☐

Enable Simulation: ☐

Encryption Key: None

Boot Settings

Auto Boot	• Run Mode (Enable): TwinCAT starts in Run Mode as soon as a user logs on to the Windows system. • Config Mode (default setting)
Auto Logon	If, in addition to the "Run Mode (Enable)" option, the "Auto Logon" option is also enabled, TwinCAT starts in Run Mode and the loaded and enabled boot projects are processed independently.
User Name Password	User name and password If you are working with an IPC, you can find more information about logging into a system in the Windows 10 IoT Enterprise (LTSB/LTSC) documentation in the General information chapter.
Apply	The settings will be transferred.

Multiuser

Enable Multiuser	Turns on the multiuser function for the System Manager.
Multiuser URL	Displays the multiuser URL for the Remote Repository or offers the possibility to change it.
Init	Initializes the multiuser function.

User Database

Connect with current user database	Connects the project with a user database.

AML Support

Enable AML IDs	Turns on the support of AML IDs. The AML IDs are stored in the project.

ADS Symbolic

UTF-8 Encoding	The ADS Symbolic uses a UTF-8 encoding (this option is fixed from version 3.1.4026, so this selection option is omitted from this version).

Simulation Mode

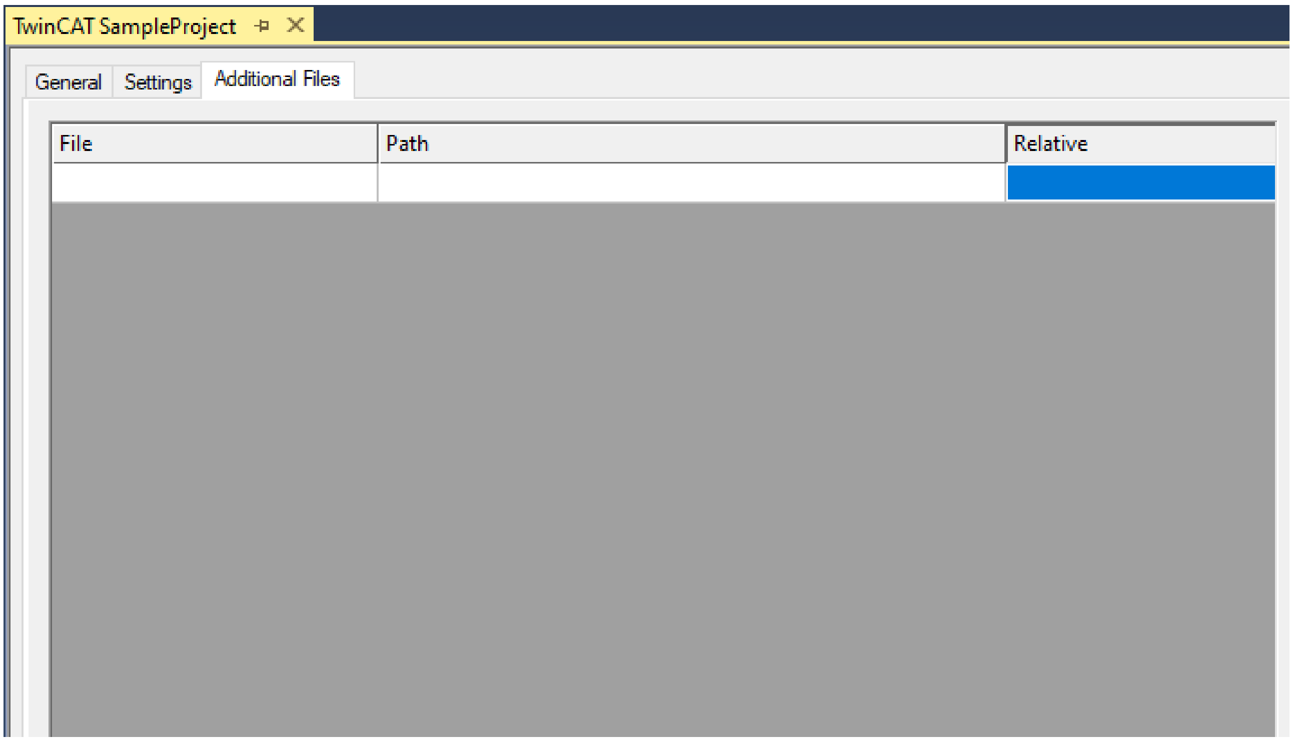
Enable Simulation	Activates the simulation mode.

Boot File Encryption Method

Encryption Key	Encryption of the boot file • "None" • "User DB Key"

3.1.3 Additional Files tab

This tab allows adding further files to the archive files of the TwinCAT project.



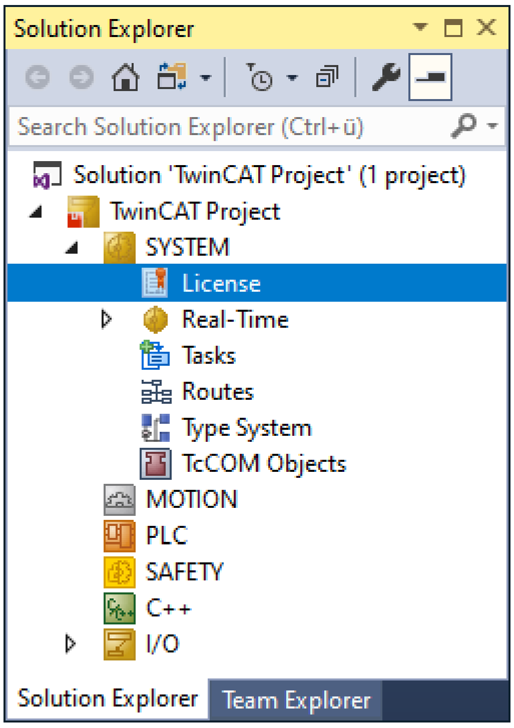
File	Double-click in the input area opens a dialog that allows you to select files you want to add to the archive.
Path	Display the path to the selected file.
Relative	Selection whether the file should be included with the absolute path or relative to the System Manager project.

3.2 License

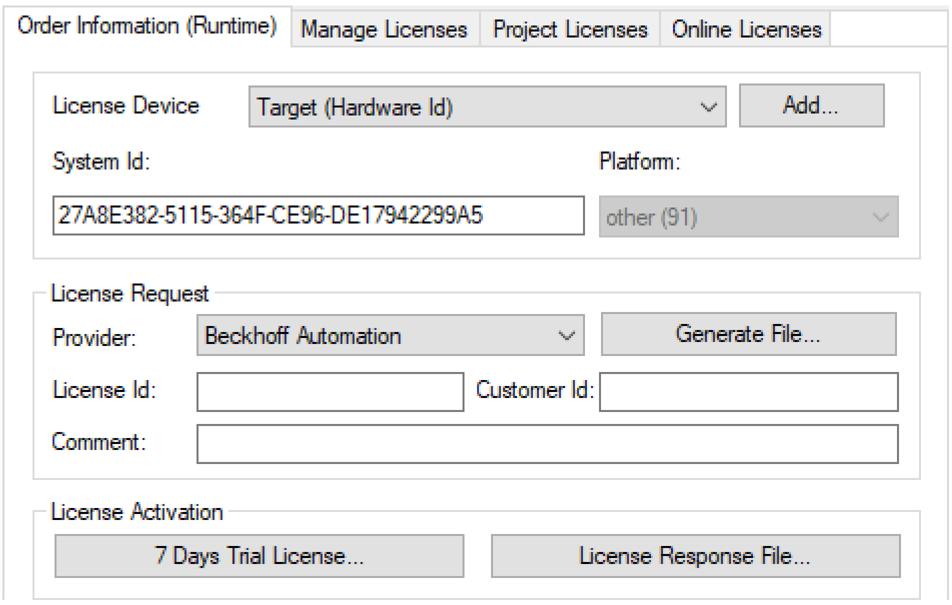
Detailed information on licensing can be found in the corresponding documentation in the Beckhoff Information System TwinCAT 3 Licensing

Function: This node contains information and setting options for your TwinCAT 3 license.

Call: Double-click on the node **License**.



A window opens containing four different tabs whose functions are briefly described in the following chapter.



License Device: Target (Hardware Id)

System Id: 27A8E382-5115-364F-CE96-DE17942299A5

Platform: other (91)

License Request

Provider: Beckhoff Automation

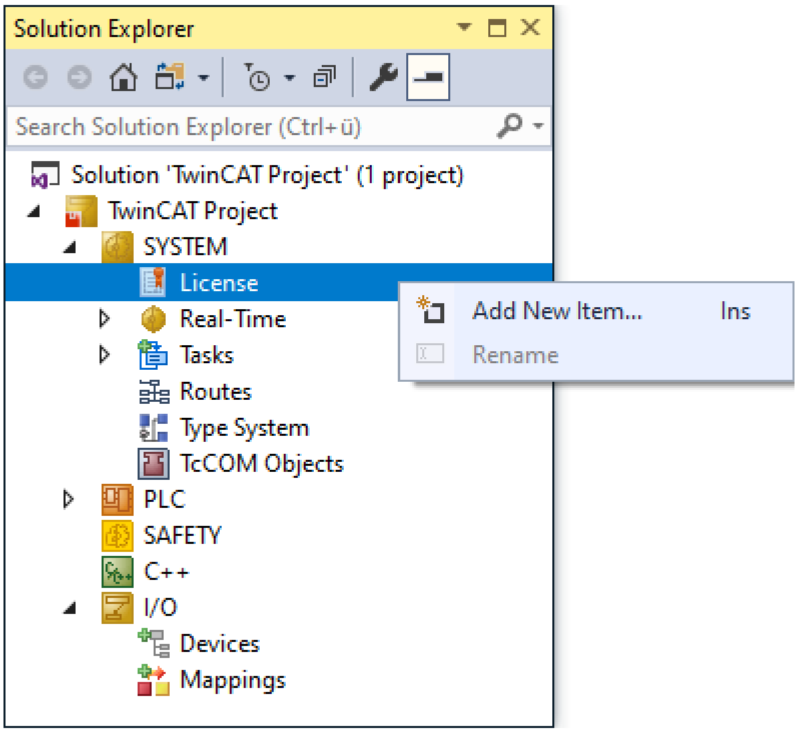
License Id:

Customer Id:

Comment:

License Activation

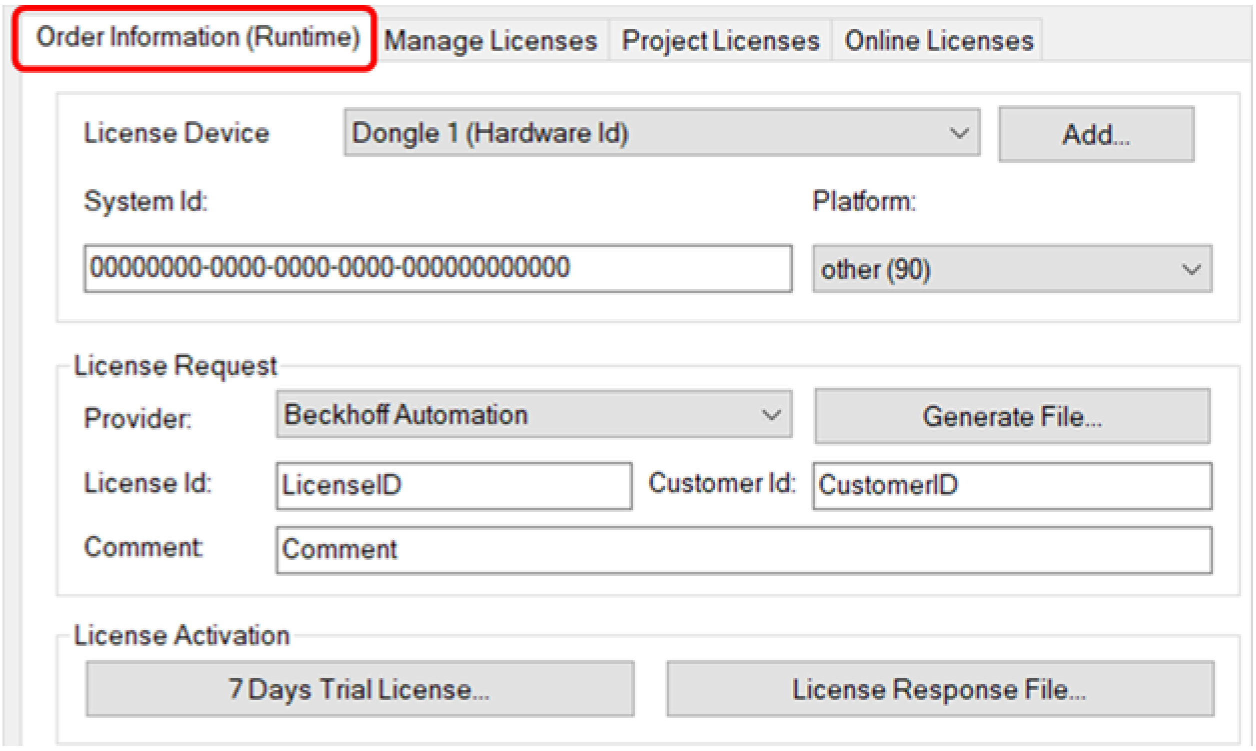
Context menu



Add New Item…	Adds a new license dongle.
Rename	Rename element.

3.2.1 Order Information (Runtime) tab

The **Order Information** tab is used to activate licenses, i.e. create a "License Request File", import a "License Response File" or create trial licenses.



Order Information (Runtime) | Manage Licenses | Project Licenses | Online Licenses

License Device: Dongle 1 (Hardware Id) [Add...]

System Id: 00000000-0000-0000-0000-000000000000

Platform: other (90)

License Request

Provider: Beckhoff Automation [Generate File...]

License Id: LicenseID

Customer Id: CustomerID

Comment: Comment

License Activation

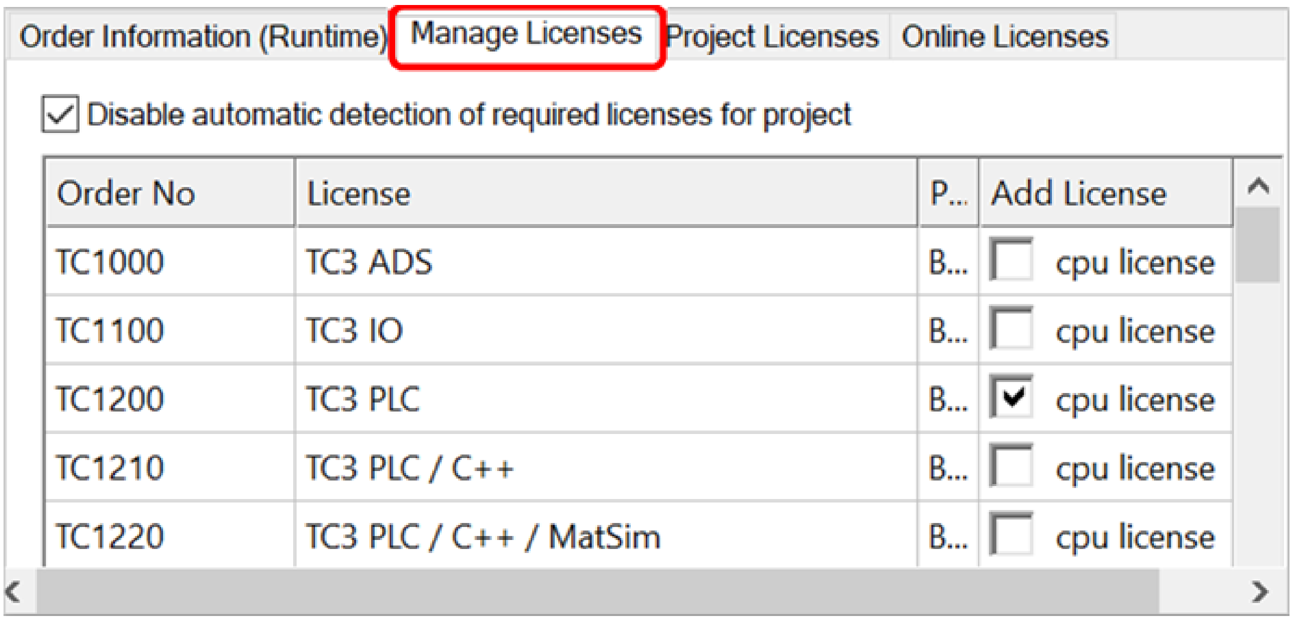
[7 Days Trial License...] [License Response File...]

The **current license status of the system** can be found in the "Online Licenses" [▶ 17] tab.

A detailed description of the functions of this tab can be found here: Activating standard licenses manually.

3.2.2 Manage Licenses tab

Licenses can be added or deselected for the creation of a "License Request File" in the **Manage Licenses** tab. You can add licenses manually only if you have checked the box at **Disable automatic detection of required licenses for project**.



☑ Disable automatic detection of required licenses for project

Order No	License	P...	Add License
TC1000	TC3 ADS	B...	☐ cpu license
TC1100	TC3 IO	B...	☐ cpu license
TC1200	TC3 PLC	B...	☑ cpu license
TC1210	TC3 PLC / C++	B...	☐ cpu license
TC1220	TC3 PLC / C++ / MatSim	B...	☐ cpu license

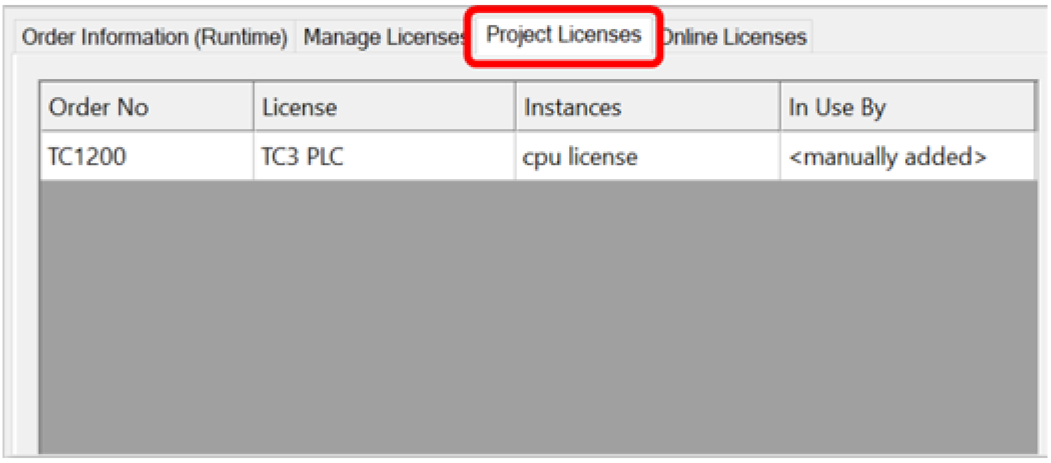
Disable automatic detection of required licenses for project	☑ Disables the automatic search for required licenses. By selecting the option **Disable automatic detection of required licenses for project** the licenses required in the project are no longer automatically detected and removed from the selection. This can be useful if you want to add licenses in a separate license file. When selecting this option, the user is responsible for ensuring that all required licenses are appropriately included in the license files. Non-existing licenses can lead to TwinCAT not starting or unlicensed functions not being executed during runtime.
Add License	Adds licenses to the project manually.

Licenses marked manually in this menu will be taken over as part of the required "Project Licenses [▸ 16]". Licenses that are automatically detected by the system are automatically selected and grayed out so that they cannot be deselected.

Related links: Quick start

3.2.3 Project Licenses tab

In the **Project Licenses** tab the licenses required for the project are displayed.



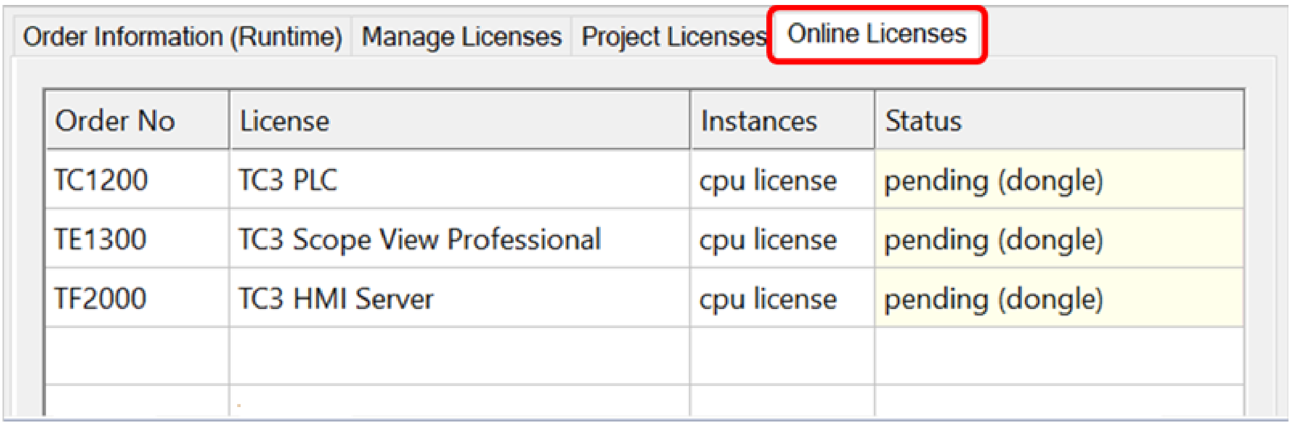
Order No	License	Instances	In Use By
TC1200	TC3 PLC	cpu license	<manually added>

Since not all required licenses can be determined at any time (e.g. in Config Mode), this list may be incomplete.

Licenses added manually in the Manage Licenses [▶ 16] tab are also displayed here as "required".

3.2.4 Online Licenses tab

The **Online Licenses** tab displays the currently recognized licenses in the system.



Order No	License	Instances	Status
TC1200	TC3 PLC	cpu license	pending (dongle)
TE1300	TC3 Scope View Professional	cpu license	pending (dongle)
TF2000	TC3 HMI Server	cpu license	pending (dongle)

The license status is not displayed in real-time
The license information is determined when the runtime is started and then approximately every two minutes. If you add a license file (or a dongle with licenses), the license status is displayed with a delay or only when TwinCAT Runtime is started.

3.2.5 Determining the license status

You can display the TwinCAT 3 license status both in TwinCAT 3 Engineering and in the TwinCAT 3 Runtime.

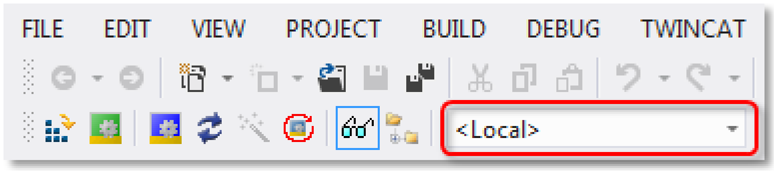
3.2.5.1 License overview in TwinCAT 3 Engineering (XAE)

Correct target system
Note that the license manager always applies to the target system for the TwinCAT 3 project!

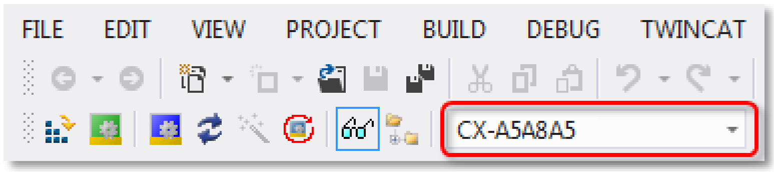
In the TwinCAT 3 development environment, you can determine the license status in the TwinCAT 3 license manager on the **Online Licenses** tab.

✓ The TwinCAT 3 development environment has been started and a project is loaded.

1. Set the desired target system. To do this, select the target system from the **Choose Target System** dropdown list in the **TwinCAT XAE Base Toolbar Options**:
   If the target system is the local computer, select <Local>.

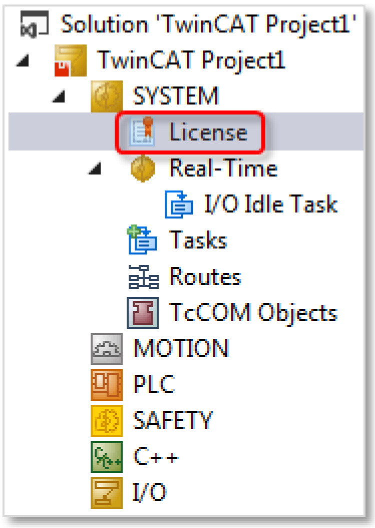


   If the target system is a remote computer, select it from the list or select "Choose Target System" and configure the target system. (If necessary, a new ADS route will be created.)



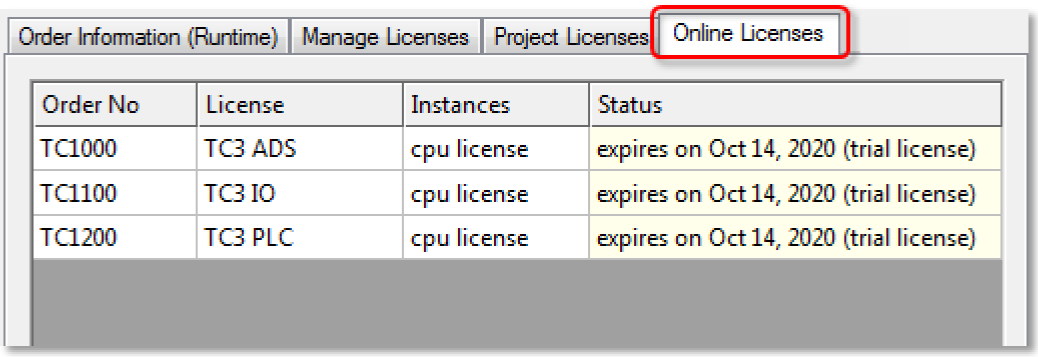
   ⇨ The licensing settings in the license manager refer precisely to the target system set here. When the project is activated on the target system, the corresponding TwinCAT 3 licenses are automatically copied to this system.

2. Open the TwinCAT 3 license manager by double-clicking **License** in the **System** subtree of the TwinCAT project tree.



3. Open the **Online Licenses** tab.

⇨ The overview shows which licenses are activated for this project (through one or more License Response Files).



Order No	License	Instances	Status
TC1000	TC3 ADS	cpu license	expires on Oct 14, 2020 (trial license)
TC1100	TC3 IO	cpu license	expires on Oct 14, 2020 (trial license)
TC1200	TC3 PLC	cpu license	expires on Oct 14, 2020 (trial license)

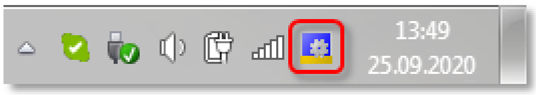
If these are License Response Files for TwinCAT 3 license dongles, the status "Pending" may be displayed there (before the TwinCAT 3 Runtime is started). This means that the License Response File has been detected as valid in principle, but the content cannot yet be released because there is no connection to the associated TwinCAT 3 license dongle. The connection to the TwinCAT 3 license dongle is only established when the TwinCAT 3 Runtime is started and the EtherCAT bus has "OP" status, for example.

3.2.5.2 License overview in the TwinCAT 3 Runtime (XAR)

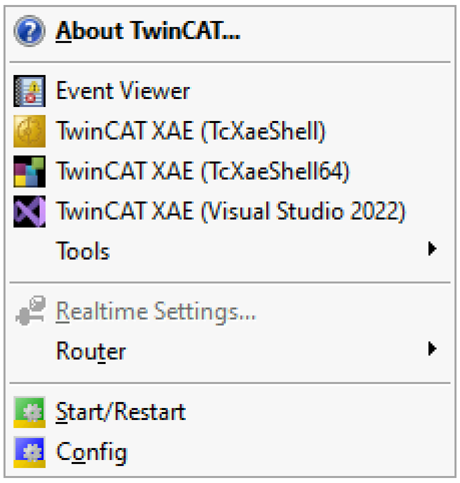
If you do not have access to the TwinCAT 3 development environment and the current project, you can determine the license status of your control computer via the TwinCAT 3 Runtime.

✓ You have access to the Windows Desktop (via remote desktop or a connected monitor and mouse) on the control computer.

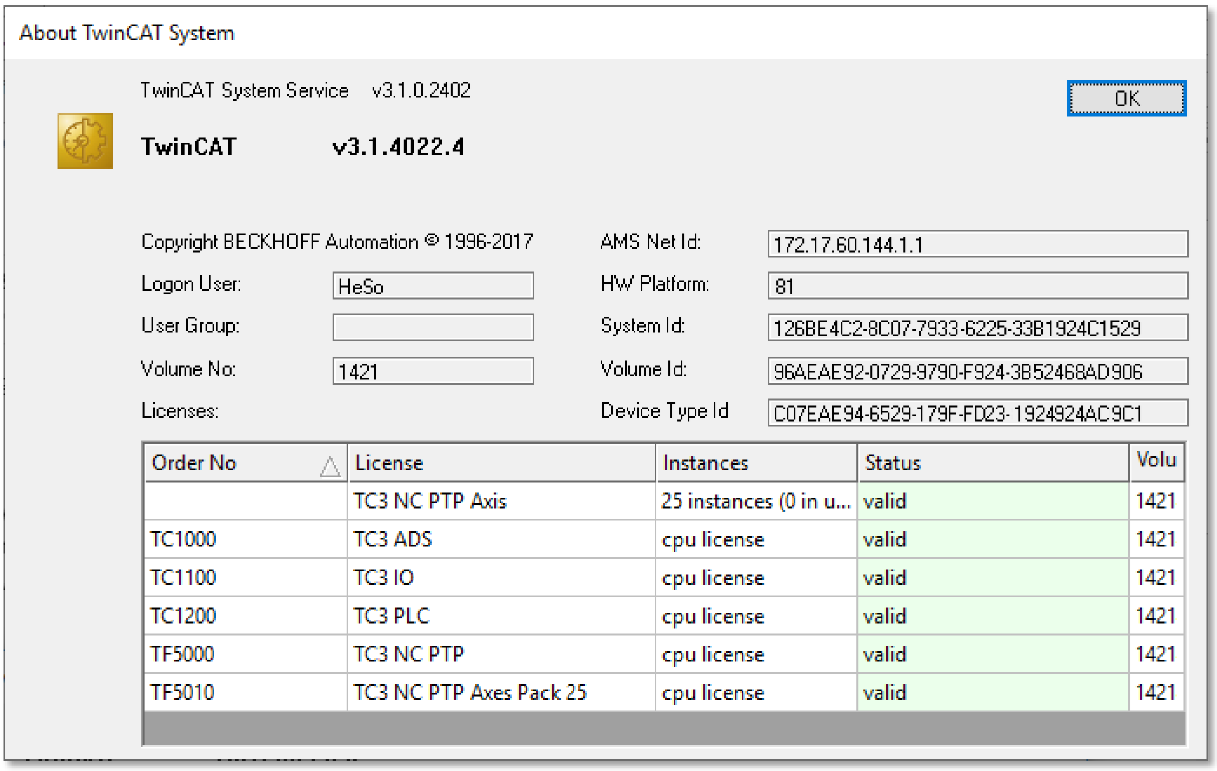
1. Right-click on the TwinCAT Runtime icon in the Windows taskbar.



2. Click on the entry **About TwinCAT** in the menu that opens.



⇨ The **About TwinCAT System** window opens, showing a list of the licenses contained in this TwinCAT 3 Runtime, the Hardware Platform Level, the System ID and Device Type ID of the computer. If the computer is equipped with a volume license, the Volume System ID will also be shown here.


About TwinCAT System

TwinCAT System Service v3.1.0.2402

TwinCAT v3.1.4022.4

Copyright BECKHOFF Automation © 1996-2017

Logon User:	HeSo	AMS Net Id:	172.17.60.144.1.1
User Group:		HW Platform:	81
Volume No:	1421	System Id:	126BE4C2-8C07-7933-6225-33B1924C1529
		Volume Id:	96AEAE92-0729-9790-F924-3B52468AD906
Licenses:		Device Type Id	C07EAE94-6529-179F-FD23-1924924AC9C1

Order No	License	Instances	Status	Volu
	TC3 NC PTP Axis	25 instances (0 in u...	valid	1421
TC1000	TC3 ADS	cpu license	valid	1421
TC1100	TC3 IO	cpu license	valid	1421
TC1200	TC3 PLC	cpu license	valid	1421
TF5000	TC3 NC PTP	cpu license	valid	1421
TF5010	TC3 NC PTP Axes Pack 25	cpu license	valid	1421

With a TwinCAT 3 license dongle the license status is "Pending" if there is no connection to the associated TwinCAT 3 license dongle.

The Device Type ID is displayed from TwinCAT Version 3.1 Build 4022.4. This is an item of internal information that identifies the device type.

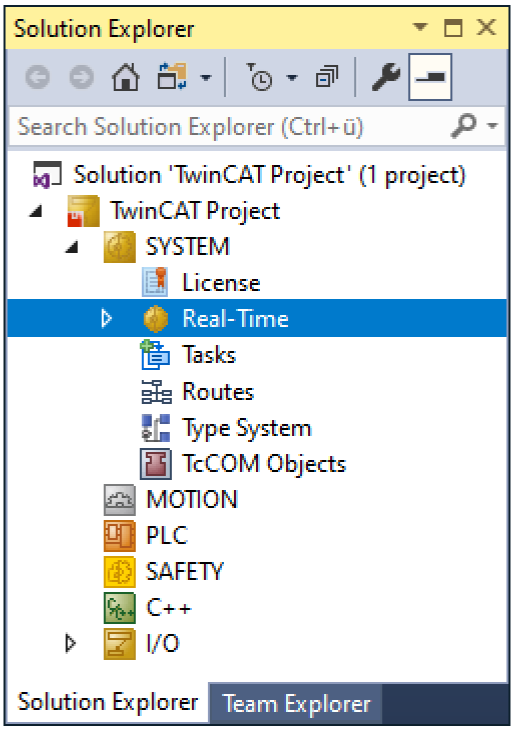
Only validated TwinCAT 3 licenses switch the EtherCAT bus to status "OP". If the EtherCAT bus in TwinCAT 3 Run mode is not in "OP" state, the required TwinCAT 3 licenses could probably not be validated. This may be the case if a TwinCAT 3 license dongle is used, for example, and the License Response File does not match the TwinCAT 3 license dongle used.

See also: Tc3 license dongle

3.3 Real-Time

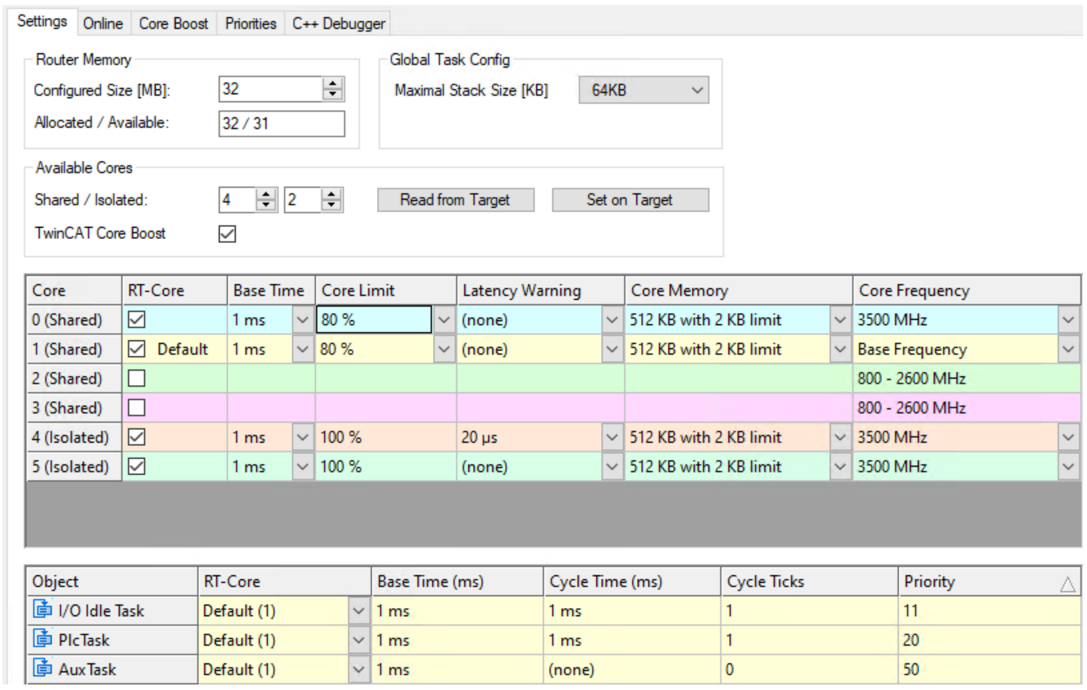
Function: Real-time setting of the (project-related) TwinCAT 3 Runtime

Call: Double-click on the node **Real-Time**.



Further information about TwinCAT 3.1 Real-Time can be found in the documentation Basics in chapter Real-Time.

3.3.1 Settings tab



Router Memory

Configured Size [MB]: 32

Allocated / Available: 32 / 31

Global Task Config

Maximal Stack Size [KB] 64KB

Available Cores

Shared / Isolated: 4 2

TwinCAT Core Boost ☑

Core	RT-Core	Base Time	Core Limit	Latency Warning	Core Memory	Core Frequency
0 (Shared)	☑	1 ms	80 %	(none)	512 KB with 2 KB limit	3500 MHz
1 (Shared)	☑ Default	1 ms	80 %	(none)	512 KB with 2 KB limit	Base Frequency
2 (Shared)	☐					800 - 2600 MHz
3 (Shared)	☐					800 - 2600 MHz
4 (Isolated)	☑	1 ms	100 %	20 µs	512 KB with 2 KB limit	3500 MHz
5 (Isolated)	☑	1 ms	100 %	(none)	512 KB with 2 KB limit	3500 MHz

Object	RT-Core	Base Time (ms)	Cycle Time (ms)	Cycle Ticks	Priority
I/O Idle Task	Default (1)	1 ms	1 ms	1	11
PlcTask	Default (1)	1 ms	1 ms	1	20
AuxTask	Default (1)	1 ms	(none)	0	50

Router Memory (MB)	Size of the router memory: Used to set the size of the router memory (see also note below the table).
Global Task Config/ Maximal Stack Size [KB]	Sets the stack size per task in kilobyte. This setting applies to all tasks in the system.
Available Cores (Shared/ Isolated)	Number of available cores (Windows cores and isolated cores)
Read from Target	Reads the number of available cores from the target system or from the local system.
Set on Target	Sets the distribution between shared and isolated cores on the target system or on the local system.
TwinCAT Core Boost	Available from TwinCAT 3.1 Build 4026 Activates the TwinCAT feature Core Boost. If this function is active and is supported by the current target system, the clock frequency can be fixed for each real-time kernel used by TwinCAT. The choice is made in the **Core Frequency** column.
Core	Contains the number of a core that can be selected for runtime operation. Brackets after the number indicate whether the core is shared or isolated.
RT-Core	Cores can be selected and deselected row by row in the **RT-Core** column of the upper table. The selected cores can then also be selected in the drop-down lists in the **RT-Core** column of the lower table. In this way, a task in a row can be assigned to one core at a time.
Base Time	Base time of the core: All tasks that are processed on this core can only use cycle times that are equal or integer multiples of the base time. Drop-down menu options: no base time, 1 ms, 500 µs, 333 µs, 250 µs, 200 µs, 125 µs, 100 µs, 83 µs, 3 µs, 76.9 µs, 71.4 µs, 66.6 µs, 62.5 µs and 50 µs. If the value of the base time is set to "(none)", then it is possible to use TwinCAT without real-time. Versions that do not require real-time, such as AMS Router, TwinCAT Scope, ADS OCX, continue to operate without restrictions. Execution of real-time tasks is not possible with this setting.
Core Limit	Maximum percentage of core occupied by TwinCAT real-time. Drop-down list options: 10 %, 20 %, 30 %, 40 %, 50 %, 60 %, 70 %, 80 %, 90 % and 100 %. The rest of the CPU is guaranteed to be reserved for the operating system and thus also for user interface programs on shared cores. The value "Core Limit" can be set very high since it is automatically reset to Windows when the real-time task has completed its cycle.
Latency Warning	The TwinCAT real-time cyclically measures the actual length of the base time. Due to the PC architecture, this may show slight fluctuations compared to the set base time. A limit value can be set in the drop-down menu; if this value is exceeded, a warning is generated by the TwinCAT system. Drop-down menu options: "(none)", 20 µs, 50 µs, 100 µs, 200 µs, 250 µs and 500 µs.
Core Memory	Size of real-time memory reserved for this core All memory requests from this core are first processed by this memory and only if it is not sufficient, memory is requested from the router.
Core Frequency	If the TwinCAT feature Core Boost is active and is supported by the target system, a drop-down menu with the possible clock frequencies that can be set for this core is displayed in this column for each real-time kernel. If this function is deactivated, **Base Frequency** is displayed in this column.
Object	Name of the TwinCAT object that is called cyclically.
Cycle Time (ms)	Cycle times of the tasks
Cycle Ticks	Number of ticks in a cycle

Priority	Priority of the respective task in the TwinCAT system: The smaller the number specified here, the higher the priority of the task within the TwinCAT system.

Router memory:

TwinCAT 3.1 versions up to and including Build 4024:
There is a global real-time memory that is used for ADS communication and for memory requests from real-time. The size of the memory can be defined via the above mentioned option in the System Manager and becomes active after a reboot of the computer.

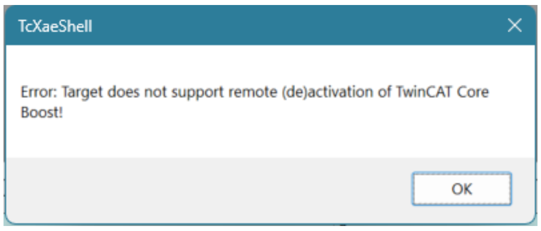
TwinCAT 3.1 from Build 4026:
The router uses separate memory areas for ADS communication between the devices and for memory requests from the real-time. The memory area for requests from real-time can be set via the above mentioned option in the System Manager and becomes active when the Solution is activated. The ADS communication memory is created when the system boots and is approximately 25 % of the size of the memory previously set up for real-time requests. The size is limited between 4 MB and 32 MB. This memory remains permanently until the next restart of the system.

TwinCAT Core Boost

Prerequisite: Both the engineering environment and the runtime environment must use at least TwinCAT version 3.1.4026.6.

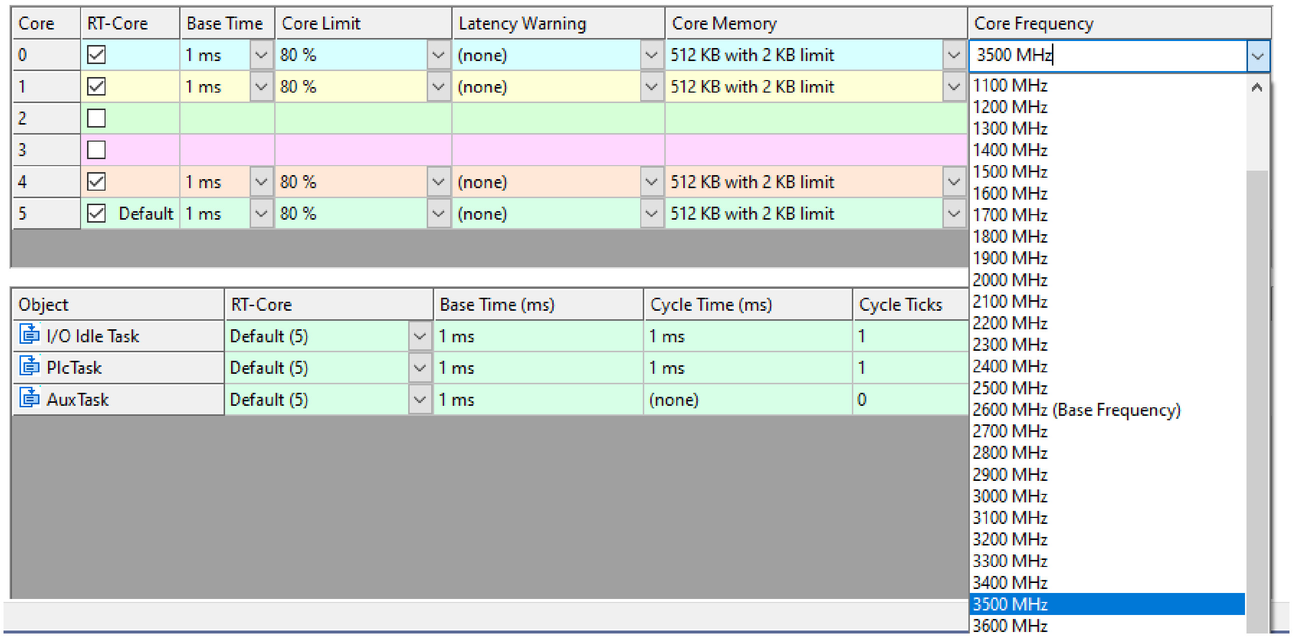
If the TwinCAT feature Core Boost is supported for a given target system and is active, this is displayed in the **TwinCAT Core Boost** selection box after pressing the **Read from Target** button. To activate or deactivate the function, these settings must be activated on the target system using the **Set on Target** button. You must restart the computer after changing this setting.

If the TwinCAT Core Boost function is not supported by a given target system, the following message appears after pressing the button **Set on Target**:


TcXaeShell

Error: Target does not support remote (de)activation of TwinCAT Core Boost!

If the TwinCAT Core Boost setting is activated, a clock frequency can be defined for each real-time core. If no clock frequency is defined for a real-time core, the **Base Frequency** is automatically selected.



Core	RT-Core	Base Time	Core Limit	Latency Warning	Core Memory	Core Frequency
0	☑	1 ms	80 %	(none)	512 KB with 2 KB limit	3500 MHz
1	☑	1 ms	80 %	(none)	512 KB with 2 KB limit	
2	☐					
3	☐					
4	☑	1 ms	80 %	(none)	512 KB with 2 KB limit	
5	☑ Default	1 ms	80 %	(none)	512 KB with 2 KB limit	

Object	RT-Core	Base Time (ms)	Cycle Time (ms)	Cycle Ticks
I/O Idle Task	Default (5)	1 ms	1 ms	1
PlcTask	Default (5)	1 ms	1 ms	1
AuxTask	Default (5)	1 ms	(none)	0

Correct selection of the core frequency

TwinCAT automatically monitors the clock frequencies of the individual cores based on the limits stored in the system for the temperature of the individual cores or the power consumption of the individual packages. If these limits are exceeded, TwinCAT reduces the clock frequencies of the individual cores accordingly (see also chapter Core Boost [▶ 29] tab). If TwinCAT is forced to reduce the clock frequencies of individual real-time cores, this may have an influence on the real-time behavior set in TwinCAT. The tasks executed on this real-time core then have longer execution times, which may lead to cycle timeouts. You therefore share the responsibility for selecting the clock frequencies of the real-time cores in such a way that TwinCAT is not permanently operated in throttle mode. If the temperature limit is permanently exceeded, the system may shut down.

Configurable clock frequencies:

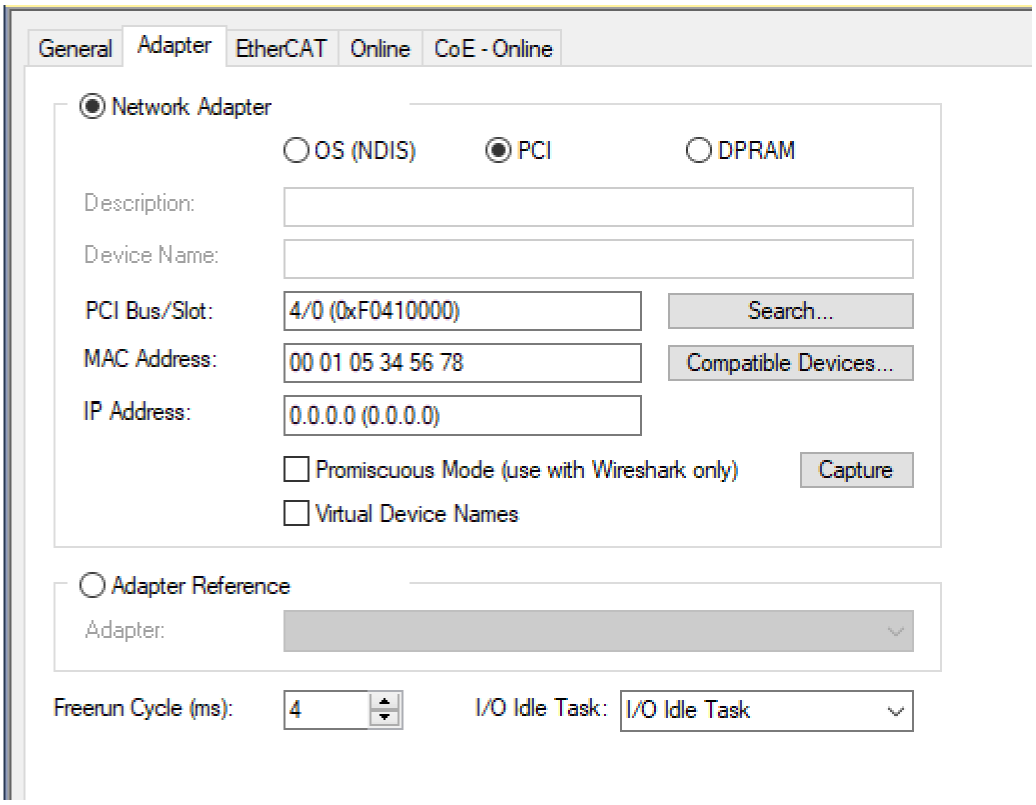
Processor	Processor generation	Base clock	Configurable Core Boost clock
Core i3-1115G4E	Intel® Celeron®, Intel® Core™ i3/i5/i7 11th generation processors, U series	2.20 GHz	3.70 GHz
Core i5-1145G7E	Intel® Celeron®, Intel® Core™ i3/i5/i7 11th generation processors, U series	1.50 GHz	3.90 GHz
Core i7-1185G7E	Intel® Celeron®, Intel® Core™ i3/i5/i7 11th generation processors, U series	1.80 GHz	4.20 GHz
Core i3-11100HE	Intel® Celeron®, Intel® Core™ i3/i5/i7 11th generation processors	2.40 GHz	4.00 GHz
Core i5-11500HE	Intel® Celeron®, Intel® Core™ i3/i5/i7 11th generation processors	2.60 GHz	4.10 GHz
Core i7-11850HE	Intel® Celeron®, Intel® Core™ i3/i5/i7 11th generation processors	2.60 GHz	4.20 GHz
Core i3-13100E	Intel® Celeron®, Intel® Pentium®, Intel® Core™ i3/i5/i7/i9 12th/13th generation processors	3.30 GHz	4.20 GHz
Core i5-13400E	Intel® Celeron®, Intel® Pentium®, Intel® Core™ i3/i5/i7/i9 12th/13th generation processors	2.40 GHz	4.10 GHz
Core i7-13700E	Intel® Celeron®, Intel® Pentium®, Intel® Core™ i3/i5/i7/i9 12th/13th generation processors	1.90 GHz	4.00 GHz
Core i9-13900E	Intel® Celeron®, Intel® Pentium®, Intel® Core™ i3/i5/i7/i9 12th/13th generation processors	1.80 GHz	3.90 GHz

Special tasks:

I/O Idle Task:

The I/O Idle Task executes the asynchronous mappings and is also responsible for the acyclic communication with the EtherCAT devices (e.g. writing of parameters).

From TwinCAT 3.1 Version 3.1.4026 there can be a separate I/O Idle Task for each EtherCAT master. This can be selected in the adapter settings of an EtherCAT master in the *I/O Idle Task* selection box.



● Network Adapter

OS (NDIS) [✓ PCI] DPRAM

Description:	
Device Name:	
PCI Bus/Slot:	4/0 (0xF0410000)
MAC Address:	00 01 05 34 56 78
IP Address:	0.0.0.0 (0.0.0.0)

☐ Promiscuous Mode (use with Wireshark only)

☐ Virtual Device Names

○ Adapter Reference

Adapter:

Freerun Cycle (ms): 4

I/O Idle Task: I/O Idle Task

PlcAuxTask:

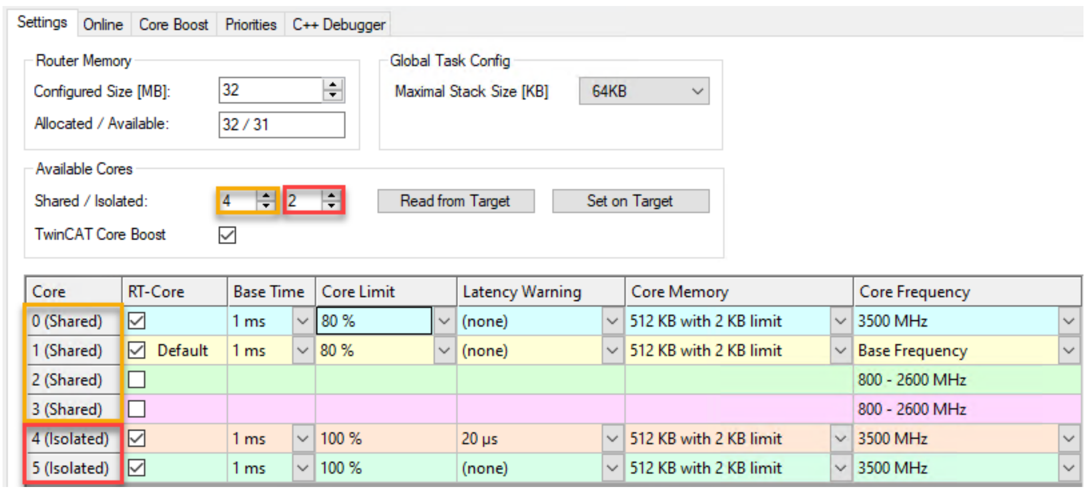
The PlcAuxTask is used for communication between the PLC editors and the PLC runtime modules. This includes the download and Online Change of PLC control code as well as debugging (monitoring of values, setting of breakpoints, etc.). In addition, the PlcAuxtask also processes the ADS messages that are sent to the runtime system independently of the development environment (TcXaeShell) (e.g. from an HMI).

3.3.1.1 Core Management

TwinCAT 3 offers extensive support for multi-core systems.

1. Select the number of cores you want to use for the TwinCAT Runtime. (Marked in red in the graphic.)
2. Select the number of cores that Windows can access and use. (Marked in orange in the graphic.)
3. Define which CPUs of TwinCAT are to be used as real-time kernels (RT core).

⇨ The cores are set appropriately.



Router Memory

Configured Size [MB]:	32
Allocated / Available:	32 / 31

Global Task Config

Maximal Stack Size [KB]	64KB

Available Cores

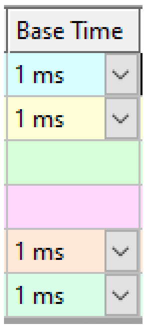
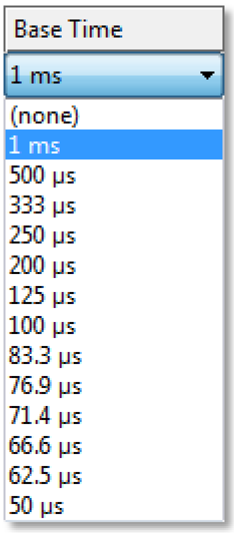
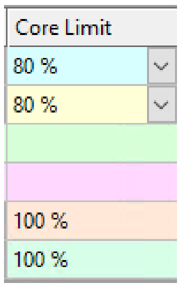
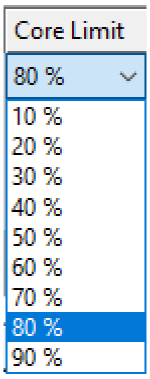
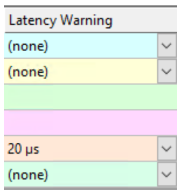
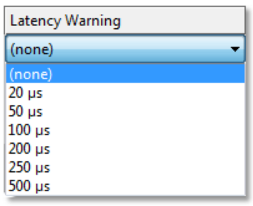
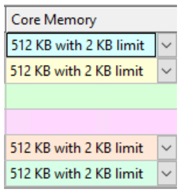
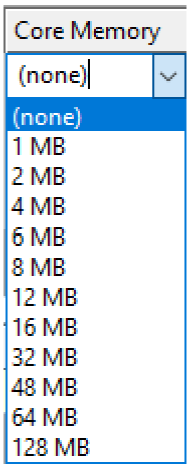
Shared / Isolated:	4 / 2
TwinCAT Core Boost	☑

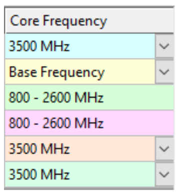
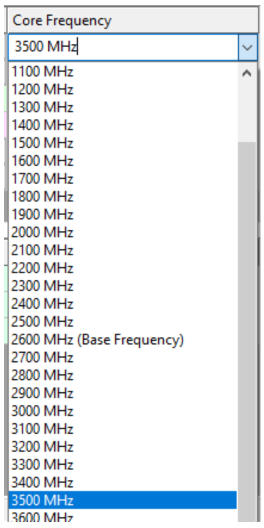
Core	RT-Core	Base Time	Core Limit	Latency Warning	Core Memory	Core Frequency
0 (Shared)	☑	1 ms	80 %	(none)	512 KB with 2 KB limit	3500 MHz
1 (Shared)	☑ Default	1 ms	80 %	(none)	512 KB with 2 KB limit	Base Frequency
2 (Shared)	☐					800 - 2600 MHz
3 (Shared)	☐					800 - 2600 MHz
4 (Isolated)	☑	1 ms	100 %	20 µs	512 KB with 2 KB limit	3500 MHz
5 (Isolated)	☑	1 ms	100 %	(none)	512 KB with 2 KB limit	3500 MHz

RT cores are required by real-time tasks like PLC tasks, software tasks and I/O tasks (EtherCAT, Profibus, ...). If no RT core(s) are selected, only tasks that do not require real-time can be executed.

- Select the RT core(s) to use the real-time.

Further information about TwinCAT real-time can be found in the documentation Basics in chapter Real-Time.

 Base Time 1 ms 1 ms 1 ms 1 ms	 Base Time 1 ms (none) 1 ms 500 µs 333 µs 250 µs 200 µs 125 µs 100 µs 83.3 µs 76.9 µs 71.4 µs 66.6 µs 62.5 µs 50 µs	By a synchronous basic tick on all real-time kernels, the scheduling for each real-time kernel is calculated independently in TwinCAT 3 Real-Time. The Base Time is the time after which scheduling is triggered on a core. Note: • Cycle times smaller than the Base Time set on a core are not possible. • The cycle times of the tasks are integer multiples of the Base Time set on the respective core. • If the application does not require it, the Base Time on Shared Cores should not be set below 1 ms. This could slow down the performance of the Windows system without noticeable performance benefits for the real-time kernel. • Base times far below 1 ms require PC systems with sufficient performance. • Setting **Base Time** to "None" means that you can only execute tasks that do not require real-time (AMS Router, TwinCAT Scope, ADS OCX, ...).
 Core Limit 80 % 80 % 100 % 100 %	 Core Limit 80 % 10 % 20 % 30 % 40 % 50 % 60 % 70 % 80 % 90 %	On shared cores, this value defines the maximum percentage of base time that the scheduler can use for real-time tasks. The remaining percentage is available for Windows. After completion of a real-time task cycle, the scheduler switches back to Windows. So the remaining time of the cycle of the real-time task is not blocked, but available for Windows.
 Latency Warning (none) (none) 20 µs (none)	 Latency Warning (none) (none) 20 µs 50 µs 100 µs 200 µs 250 µs 500 µs	TwinCAT real-time itself operates with very low real-time fluctuations (jitter). The total jitter depends on different components of a PC (hardware, BIOS, drivers, ...). TwinCAT measures and monitors the jitter and issues a warning if a preset limit value is exceeded. **We recommend to run TwinCAT on Beckhoff IPCs, because they are optimized in detail for low jitter.**
 Core Memory 512 KB with 2 KB limit 512 KB with 2 KB limit 512 KB with 2 KB limit 512 KB with 2 KB limit	 Core Memory (none) (none) 1 MB 2 MB 4 MB 6 MB 8 MB 12 MB 16 MB 32 MB 48 MB 64 MB 128 MB	With TwinCAT Version 3.1.4026 it is possible to provide a separate memory area for each real-time kernel. This memory is automatically used first by runtime modules that run on these cores and allocate memory at runtime. When this memory is used up, the memory to be allocated is used from the shared memory area held in the router. With Version 3.1.4026, the memory for communication and the memory that runtime modules can use to allocate memory are separated. A maximum of 25% of the defined router memory is used for communication.

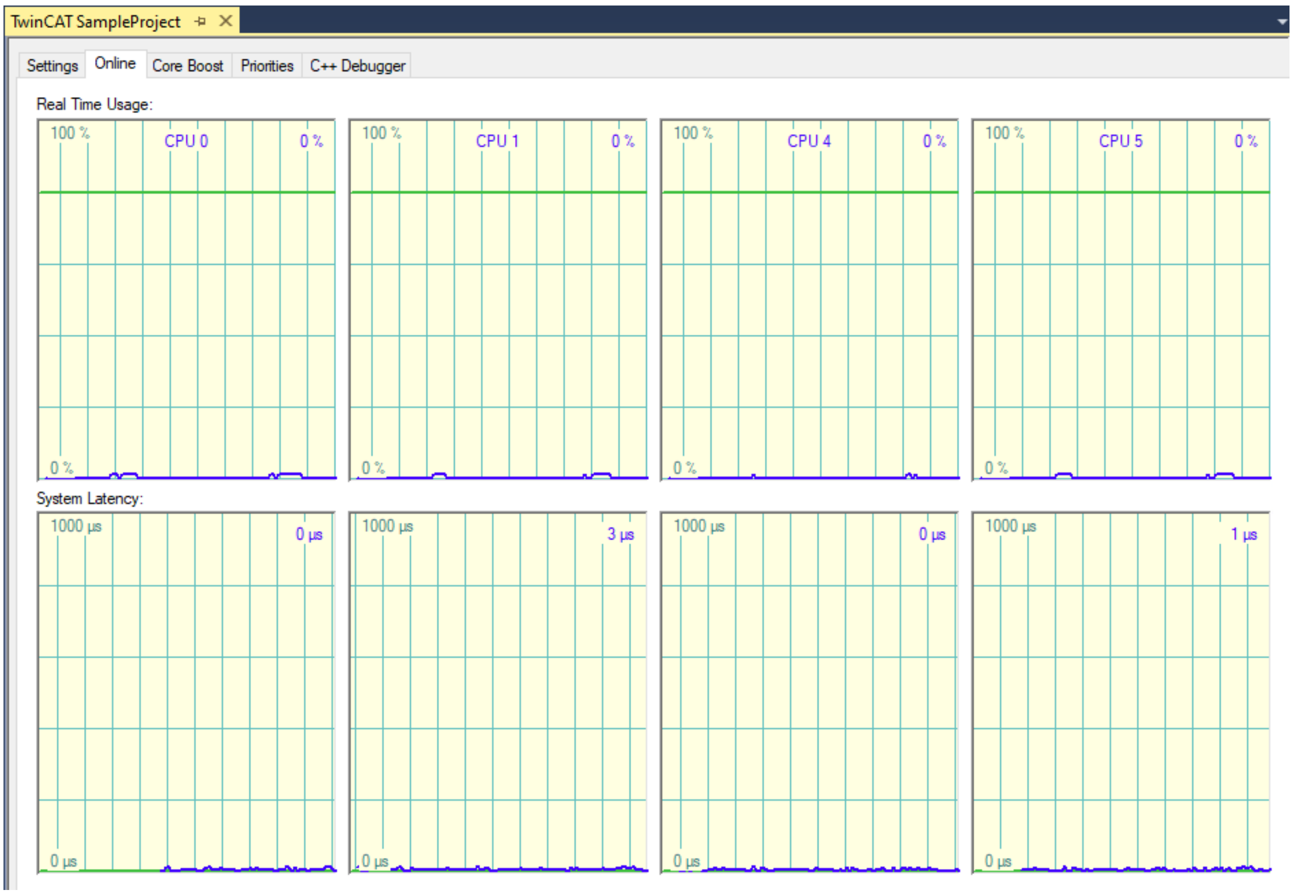
 Core Frequency 3500 MHz Base Frequency 800 - 2600 MHz 800 - 2600 MHz 3500 MHz 3500 MHz	 Core Frequency 3500 MHz 1100 MHz 1200 MHz 1300 MHz 1400 MHz 1500 MHz 1600 MHz 1700 MHz 1800 MHz 1900 MHz 2000 MHz 2100 MHz 2200 MHz 2300 MHz 2400 MHz 2500 MHz 2600 MHz (Base Frequency) 2700 MHz 2800 MHz 2900 MHz 3000 MHz 3100 MHz 3200 MHz 3300 MHz 3400 MHz 3500 MHz 3600 MHz	From TwinCAT version 3.1.4026.6 it is possible to use the TwinCAT Core Boost feature (see chapter Core Boost tab [▶ 29]). If TwinCAT Core Boost is supported on a target system and is activated, it is possible to set the clock frequency for the individual real-time kernels.

3.3.2 Online tab

The **Online** tab displays the real-time load and system latency.

The online display provides information about the current and recent CPU load. This is the load caused by the real-time tasks. The bright green line indicates the pre-set "CPU Limit" value.

The current and recent system latency is shown in the lower displays.

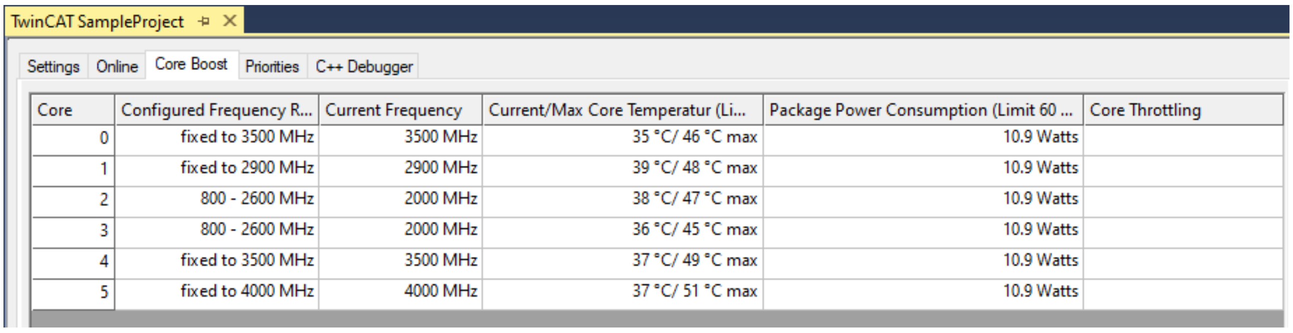

TwinCAT SampleProject

Settings | Online | Core Boost | Priorities | C++ Debugger

Real Time Usage:

System Latency:

3.3.3 Core Boost tab



Core	Configured Frequency R...	Current Frequency	Current/Max Core Temperatur (Li...	Package Power Consumption (Limit 60 ...	Core Throttling
0	fixed to 3500 MHz	3500 MHz	35 °C/ 46 °C max	10.9 Watts	
1	fixed to 2900 MHz	2900 MHz	39 °C/ 48 °C max	10.9 Watts	
2	800 - 2600 MHz	2000 MHz	38 °C/ 47 °C max	10.9 Watts	
3	800 - 2600 MHz	2000 MHz	36 °C/ 45 °C max	10.9 Watts	
4	fixed to 3500 MHz	3500 MHz	37 °C/ 49 °C max	10.9 Watts	
5	fixed to 4000 MHz	4000 MHz	37 °C/ 51 °C max	10.9 Watts	

Core	Contains the ID of the CPU core.
Configured Frequency Range	Configured clock frequency of the CPU core. The possible clock frequency range of a CPU core is displayed for non-real-time cores.
Current Frequency	Current clock frequency of the core
Current/Max Core Temperature (Limit *X*°C)	Current and maximum measured temperature of the CPU core. The maximum permissible temperature is shown in brackets in the column description.
Package Power Consumption (Limit *X* Watts)	Power consumption per package: The maximum consumption is shown in brackets in the column description. The individual lines of the CPU cores each contain the total consumption of the package.
Core Throttling	Indicates when the clock frequency is throttled due to exceeding the temperature or power consumption limit. If the value falls below the respective limit, the clock frequency is increased again.

Exceeding the limits for temperature and power consumption

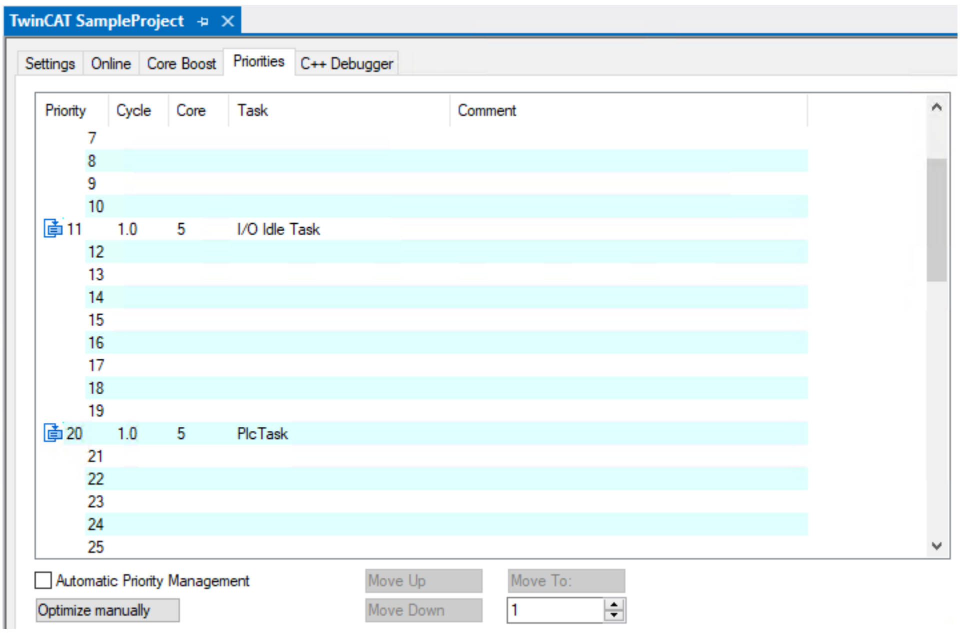
The limits for the permissible temperature or the permissible power consumption are shown in brackets in the column description. These limits are platform-dependent and can therefore differ between different target systems. If these limits are exceeded, throttling will occur. The behavior differs depending on which limit is exceeded. If the power consumption of the package exceeds the limit, the processing power of the non-real-time cores is throttled first to bring the power consumption back below the limit. If this is not sufficient, the processing power of the real-time cores is throttled. However, if the temperature limit of a real-time core is exceeded, the clock frequency of that specific core is reduced. If the current value falls below the limit, the configured clock frequencies of the cores are restored.

Supported hardware:

Motherboard	Processor generation	Minimum BIOS version	TwinCAT Build
CB6472	Compact motherboard for Intel® Celeron®, Intel® Core™ i3/i5/i7 11th generation processors	0.64	3.1.4026.6
CB3072	3½-inch motherboard for Intel® Celeron®, Intel® Core™ i3/i5/i7 11th generation processors	0.67	3.1.4026.6
CB7476	Compact motherboard for Intel® Celeron®, Intel® Pentium®, Intel® Core™ i3/i5/i7/i9 12th/13th generation processors	0.13	3.1.4026.14
CB1076	ATX motherboard for Intel® Celeron®, Intel® Pentium®, Intel® Core™ i3/i5/i7/i9 12th/13th generation processors	0.13	3.1.4026.14

3.3.4 Priorities tab

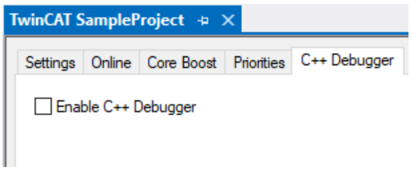
Manual priority management should only be performed by experienced TwinCAT 3 users.



Priority	Cycle	Core	Task	Comment
7				
8				
9				
10				
11	1.0	5	I/O Idle Task	
12				
13				
14				
15				
16				
17				
18				
19				
20	1.0	5	PlcTask	
21				
22				
23				
24				
25				

Priority	Priority of the respective task in the TwinCAT system The type of tasks is differentiated by different icons. The smaller the number specified here, the higher the priority of the task within the TwinCAT system. A task that serves ADS communication should have the lowest priority for data consistency reasons.
Cycle	Cycle time of the respective task in milliseconds (ms).
Core	The core on which the task will be executed, i.e. the CPU selected in the drop-down list of the RT-Core column of the table.
Task	Name of the task
Comment	Comment
Automatic Priority Management	✓ The priorities of the existing tasks are prioritized automatically.
Optimize manually	Automatically optimizes the priorities of existing tasks.
Move Up	Moves a selected task up to the next free row. This will give the selected task a higher priority.
Move Down	Moves a selected task down to the next free row. This will give the selected task a lower priority.
Move To	Moves a selected task to a selected location. The numbers 1 to 61 can be selected for the 61 rows in the priority table. If a task to be moved is marked and a free row is selected, the marked task will be moved to the free selected row when the button is pressed. If a task to be moved is selected and a row is selected that already contains a task, the button is grayed out and disabled.

3.3.5 C++ Debugger tab

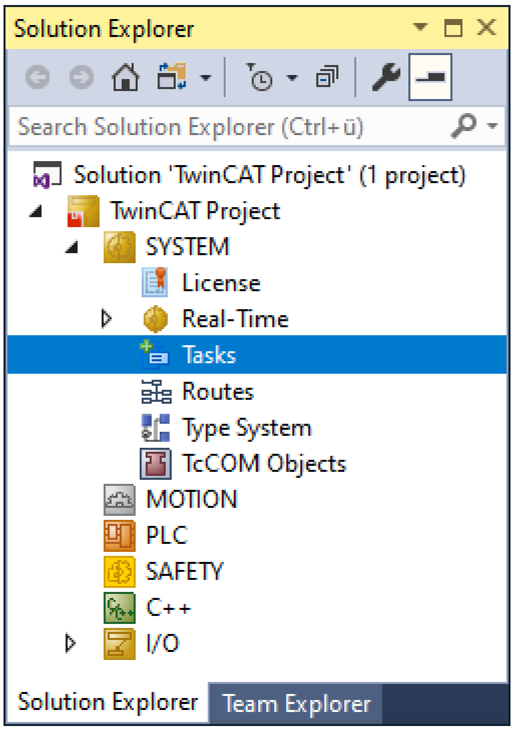


Enable C++ Debugger	✓ The C++ Debugger is supported on the TwinCAT Runtime. When the C++ Debugger is enabled, the overhead associated with the debugger can affect system performance. The checkbox should not be checked in the release version of your project.

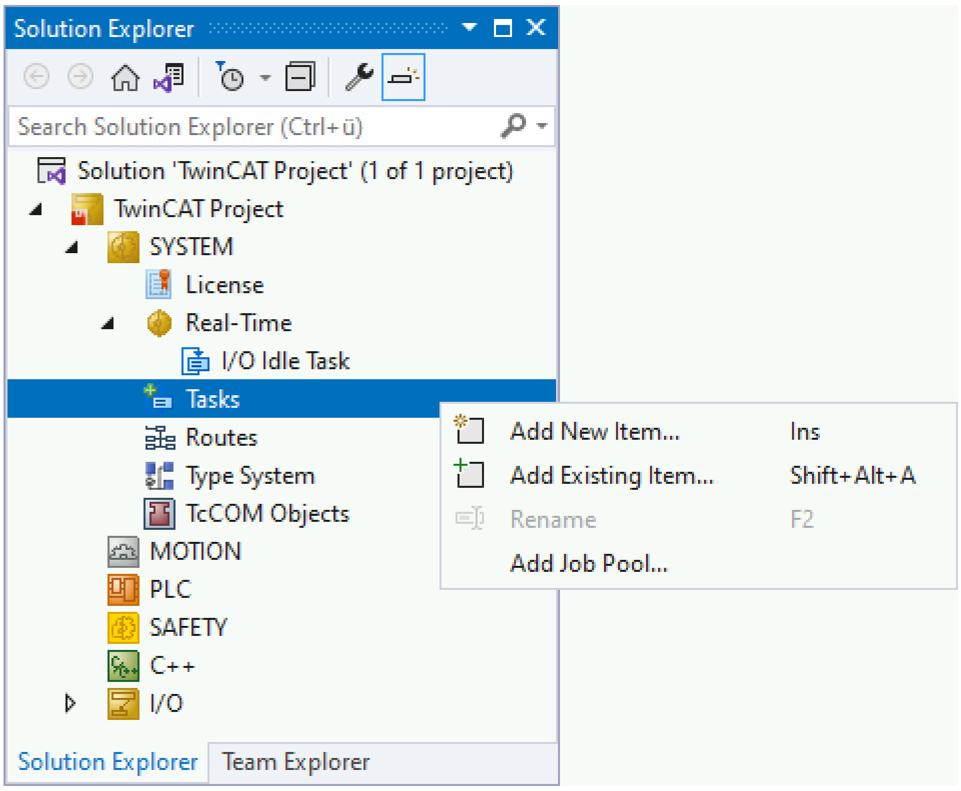
3.4 Tasks

Function: Task settings

Call: Double-click on the node **Tasks**.



Context menu

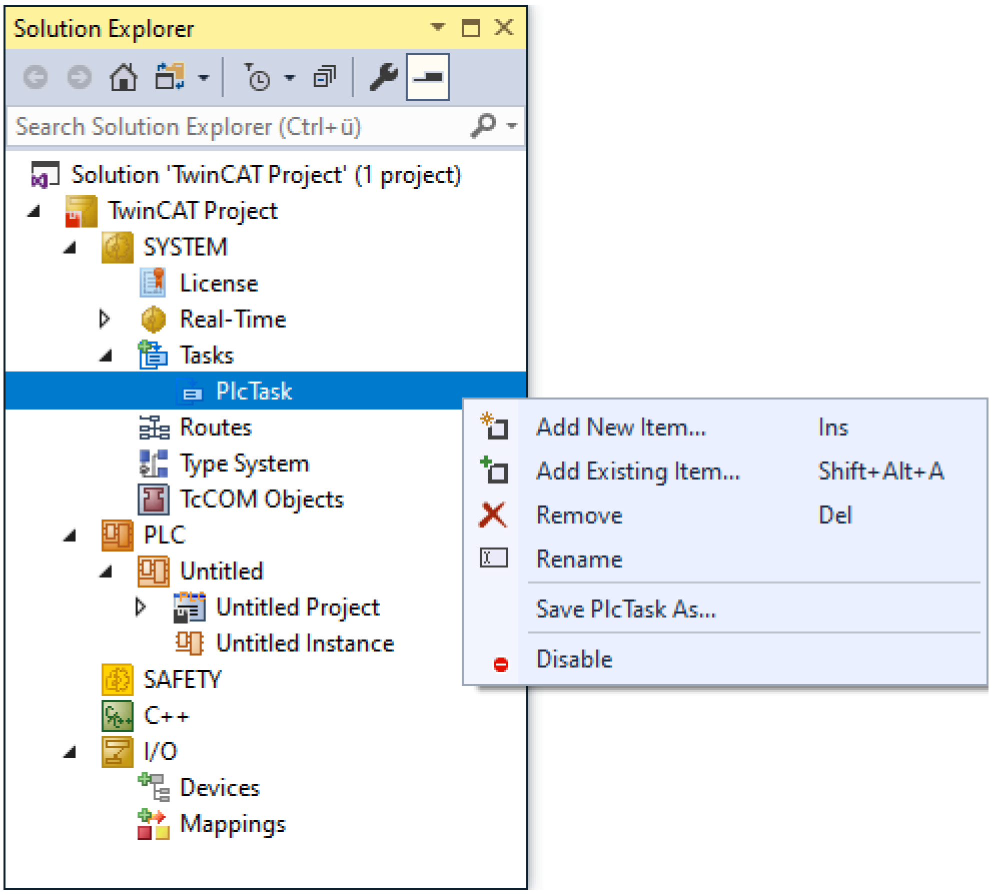


Add New Item…	The **Insert Task** dialog opens. (See chapter Task Management [▶ 40].)
Add Existing Item …	Inserts a task based on the description of an XTI file.
Rename	Rename task.
Add Job Pool…	Adds a Job Pool.

3.4.1 TwinCAT Task

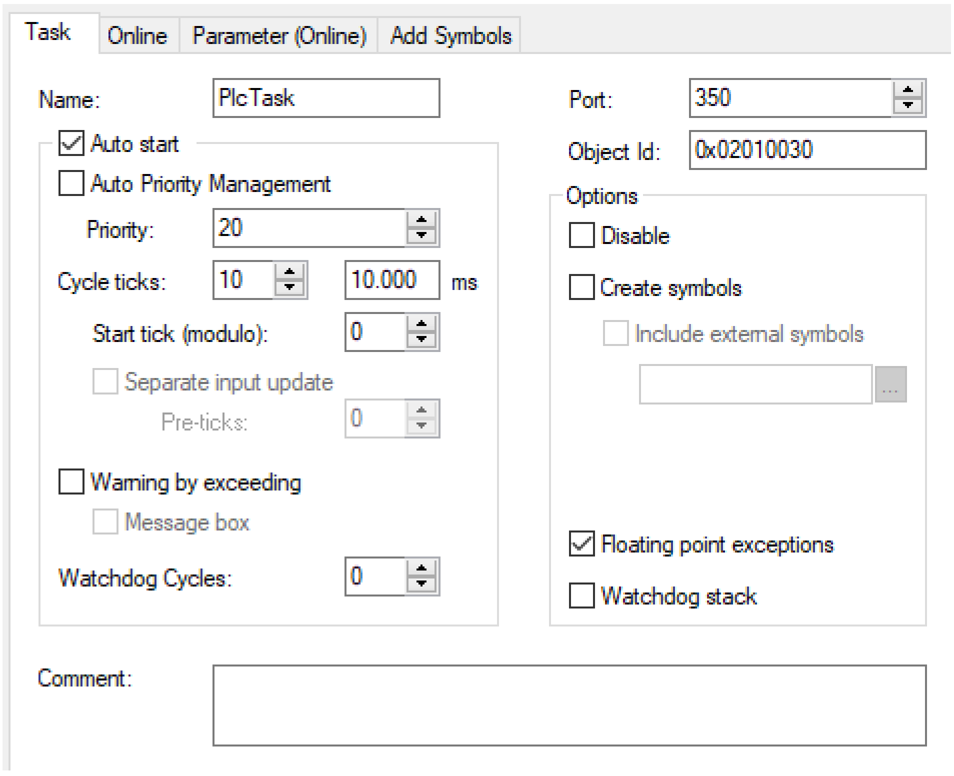
The following chapter describes the settings and menus of a TwinCAT Task.

Context menu of a TwinCAT Task



Add New Item…	Opens the **Insert TcCom Object** dialog for adding TcCOM objects directly under the task.
Add Existing Item	Importing a TcCOM object from an XTI file.
Remove	Removing the task.
Rename	Renaming the task.
Save <Task-Name> As…	Saves the task settings in an XTI file.
Disable	Disabling the task. (The task is not created when the configuration is enabled)

3.4.1.1 Task tab



Name: PlcTask

[✓] Auto start

[ ] Auto Priority Management

Priority: 20

Cycle ticks: 10 10.000 ms

Start tick (modulo): 0

[ ] Separate input update

Pre-ticks: 0

[ ] Warning by exceeding

[ ] Message box

Watchdog Cycles: 0

Port: 350

Object Id: 0x02010030

Options

[ ] Disable

[ ] Create symbols

[ ] Include external symbols

[✓] Floating point exceptions

[ ] Watchdog stack

Comment:

Name	Internal name of a task
Port	Defines the AMS port number of the task. This value must be specified. For some tasks, such as PLC tasks, this value is preset.
Object Id	A task represents an object in TwinCAT. The **Object Id** text box contains the TwinCAT object identification number.
Auto start	✓ TwinCAT generates the start command for the task so that the task is automatically started with the specified data when TwinCAT is restarted.
Auto Priority Management	✓ The priorities are assigned automatically. Sorting is carried out based on the cycle time of the tasks. As a rule, the faster a task is, the higher its priority.
Priority	The priority valid in TwinCAT for the task Priorities may not be assigned twice.
Cycle ticks	Refresh time of the task and any variables associated with it The cycle time of the task is defined in ticks and thus depends on the set TwinCAT base time. The text box to the right shows the cycle time. It results from the product of cycle ticks and base time.
Start tick (modulo)	Shifting the start time of a task compared to the actual sheduling in ticks of the base time.
Separate input update	Before the task execution starts, the inputs can be refreshed. This function is to ensure that all inputs are up to date before the cyclic processing of the task is started.
Pre-ticks	Enabled if the checkbox **Separate input update** is checked. Specifies how many ticks before the task starts execution the inputs are to be refreshed.
Warning by exceeding	✓ The TwinCAT system issues a message if the set task cycle time is exceeded.
Message box	✓ The warning that the set task cycle time has been exceeded is additionally output as a message box.
Watchdog Cycles	Specifies how often the cycle time may be exceeded before a warning is triggered.

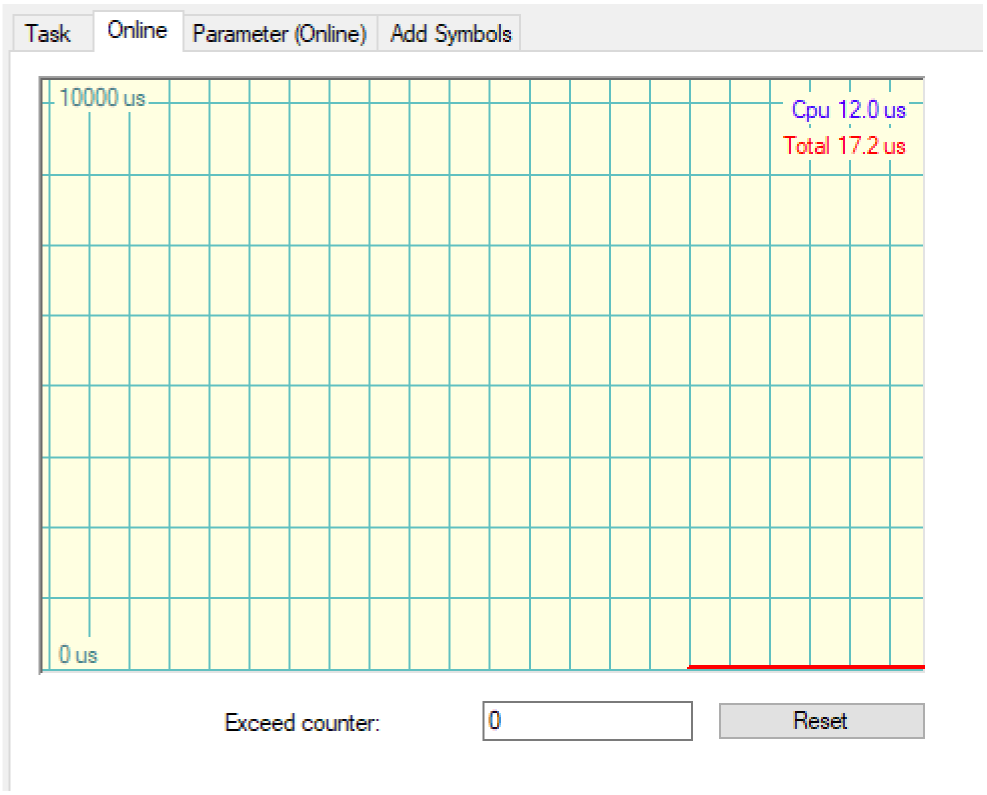
Options

Disable	Disables or enables the task. A disabled task is not taken into account when generating the I/O information. This can be useful during commissioning, for example. The created link information of the Task is preserved when disabling it.
Create symbols	✓ Variables of the task can be accessed via ADS. **Create symbols** means here that these variables are created as symbolic names.
Include external symbols	✓ This task is synchronized with a configured device with "Master Sync Interrupt". (e.g. a SERCOS card).
Floating point exception	Enables **Floating Point Exception**. Thus an exception is generated for division by zero and "Invalid Operations" (e.g. root of -1).
Watchdog stack	The task creates an object for managing nested watchdogs. This is required internally by TwinCAT Vision. This option has no influence on the "normal" watchdog function.
Comment	Comment on the task

3.4.1.2 Online tab

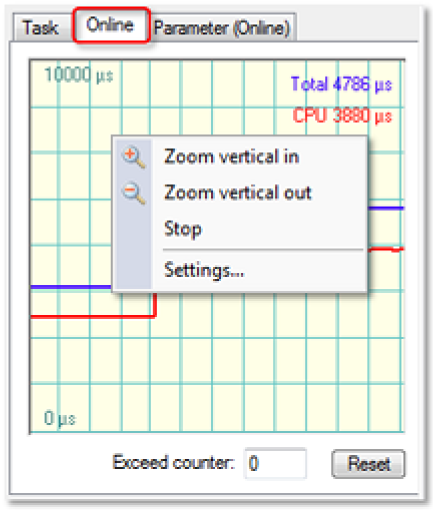
Online View

The task must be downloaded and activated on the target system to enable online display of task values.



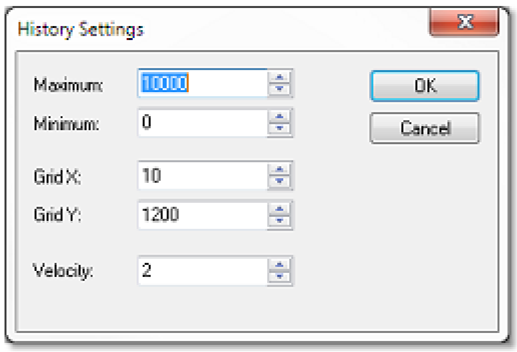
CPU display	The curve records the pure CPU time that was needed for the execution of the task.
Total display	The curve records the time that has elapsed between the start and end of task execution. The difference between pure CPU execution time and absolute (total) execution time can result, for example, from an interruption of the task by tasks that have a higher priority or by Windows.
Exceed counter	The number of times the task has exceeded the cycle time is counted in the Exceed counter.
Reset	Resets the Exceed counter to zero.

Online



Zoom vertical in	Enlarges the view.
Zoom vertical out	Reduces the view.
Stop	As long as this option is active, the recording of the curve is stopped.
Settings	Opens the **History Settings** dialog.

History Settings



History Settings

Maximum:	10000
Minimum:	0
Grid X:	10
Grid Y:	1200
Velocity:	2

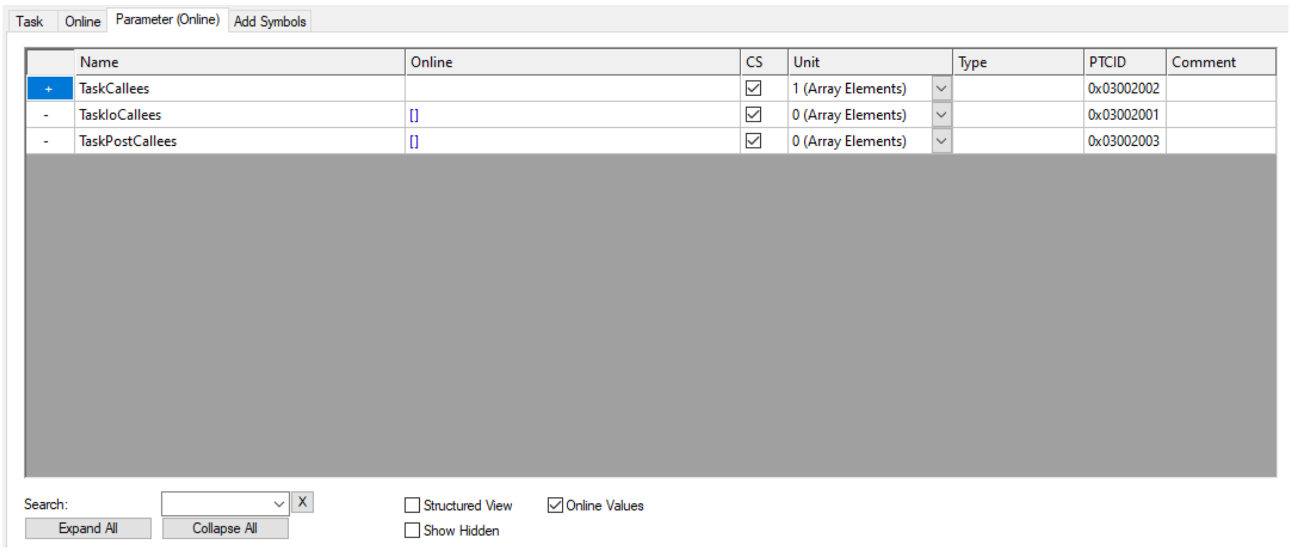
OK
Cancel

Maximum	Defines the maximum value for the History Trace View (upper row).
Minimum	Defines the minimum value (bottom row).
Grid X	Defines the grid scaling of the time axis (horizontal). Default value is "10".
Grid Y	Defines the vertical grid scaling of the Trace View. Maximum / Grid Y = Number of vertical fields Example: Maximum value = 100000, Grid Y value = 20000 => 5 vertical fields are displayed.
Velocity	Sets the feed rate of the trace. The default velocity is "2".
OK	Saves the changes and closes the dialog.
Cancel	Closes the dialog without saving the changes.

3.4.1.3 Parameter (Online) tab

Online View

The task must be downloaded and activated on the target system to enable online display of task values.



Task | Online | Parameter (Online) | Add Symbols

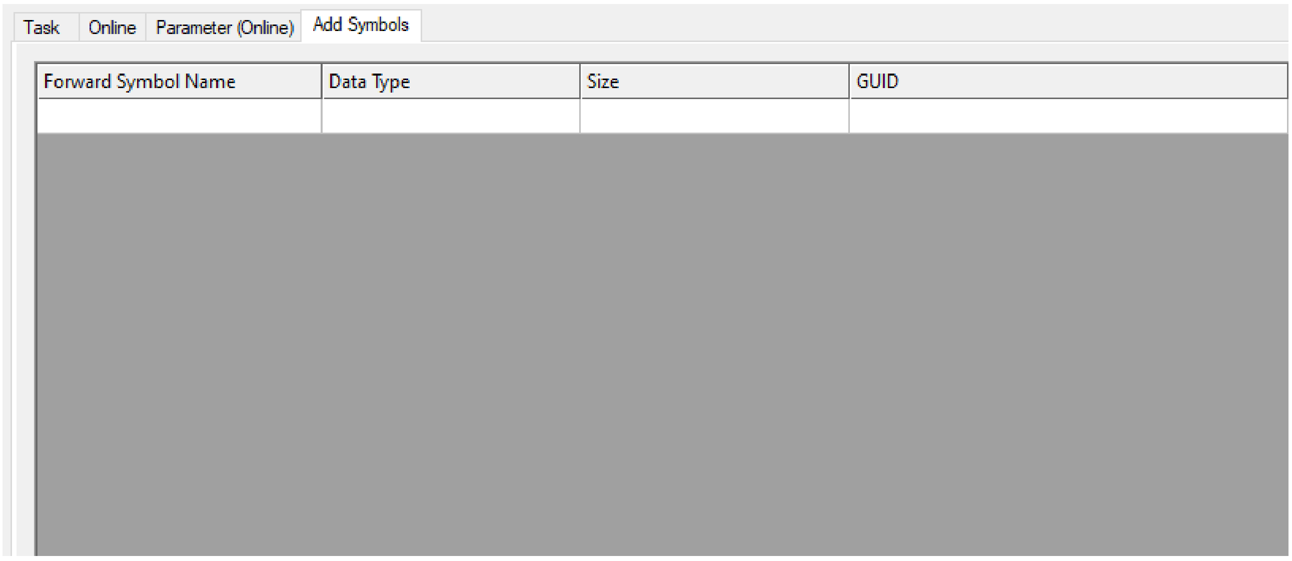
	Name	Online	CS	Unit	Type	PTCID	Comment
+	TaskCallees		☑	1 (Array Elements)		0x03002002	
-	TaskIoCallees	[]	☑	0 (Array Elements)		0x03002001	
-	TaskPostCallees	[]	☑	0 (Array Elements)		0x03002003	

Search:
Expand All | Collapse All
☐ Structured View
☐ Show Hidden
☑ Online Values

Name	Name of the parameter
Online	Online value of the parameter
CS	Option **Create Symbol** If this is active for the parameter, the parameter can be accessed symbolically (by name) via ADS.
Unit	Unit of the parameter
Type	Type of the parameter
PTCID	ID of the parameter
Comment	Comment on the corresponding parameter (if available)

3.4.1.4 Add Symbols tab

This tab is used to create Forward(-ADS) symbols to which ADS symbols can be forwarded. This function is currently only suitable for internal Beckhoff use.



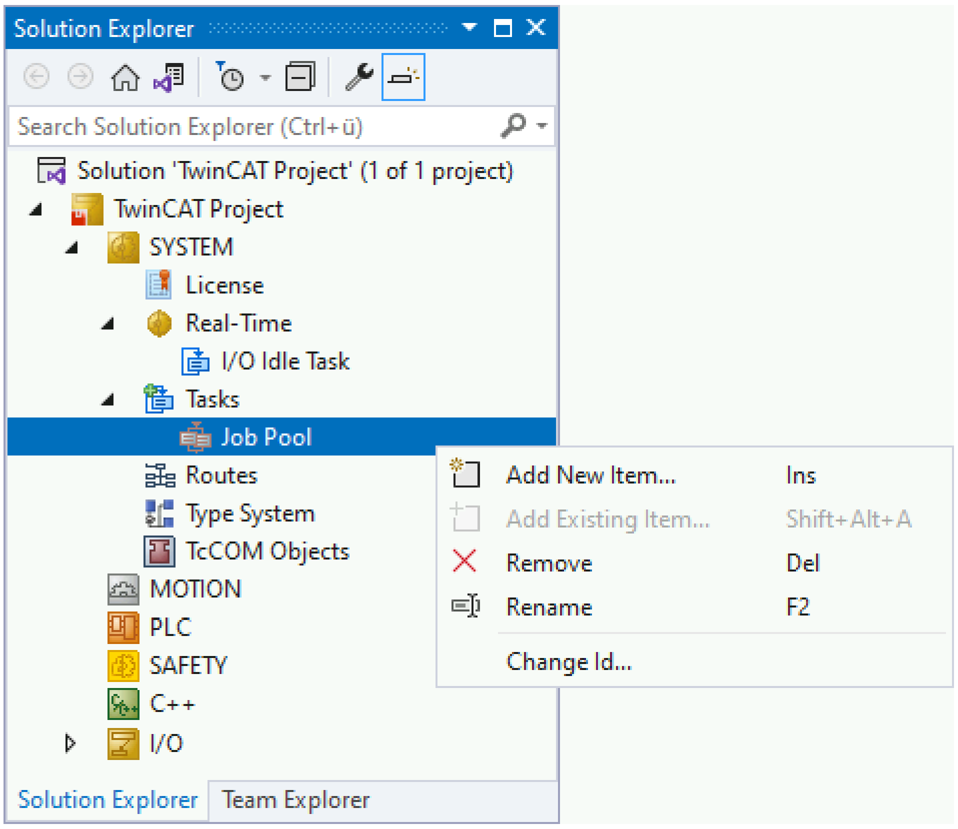
Forward Symbol Name	Name of the Forward Symbol
Data Type	Data type
Size	Data type size
GUID	GUID of the data type

3.4.2 TwinCAT Job Pool

The following chapter describes the settings and menus of the TwinCAT Job Pool.

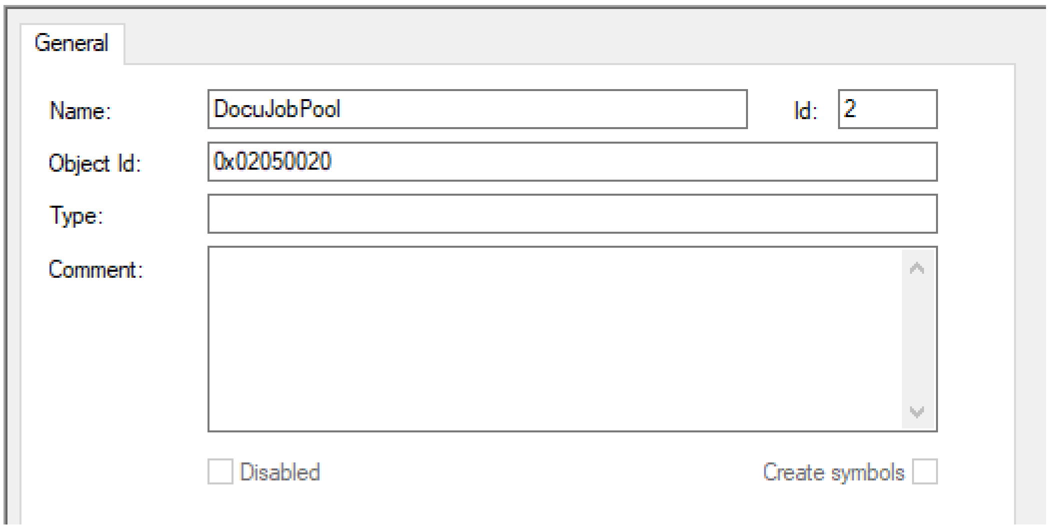
The TwinCAT Job Pool is a collection of Job Tasks that are managed by the Job Pool. If a task is assigned to the Job Pool from an application, the Job Pool automatically distributes the task to the next available Job Task from its pool.

Context menu



Add New Item…	Creates a new Job Task.
Add Existing Item…	Loads an existing Job Pool configuration from an XTI file.
Remove	Removes the Job Pool from the configuration.
Rename	Rename the Job Pool.
Change Id…	Change the ID of the Job Pool.

3.4.2.1 General tab



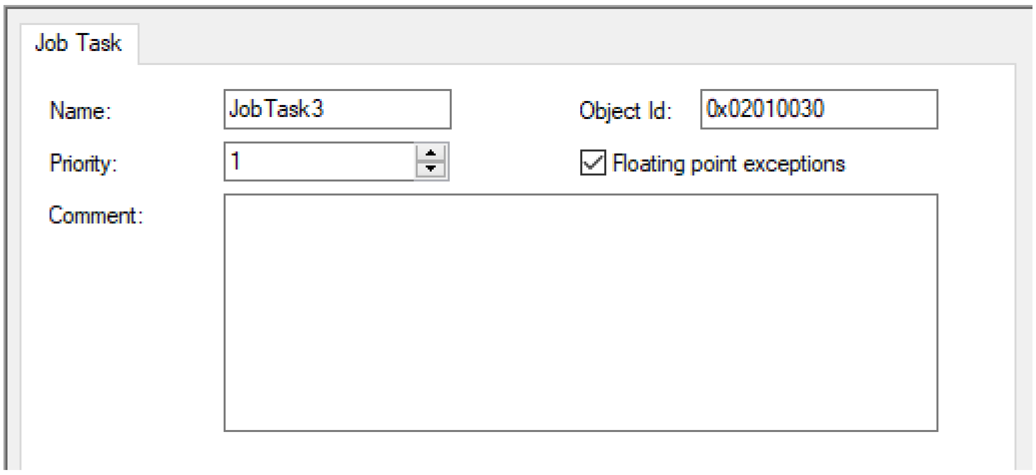
Name: DocuJobPool
Id: 2
Object Id: 0x02050020
Type:
Comment:

Name	Job Pool name
Id	Id of the Job Pool
Object Id	TwinCAT Object Id of the Job Pool
Type	-
Comment	Optional comment on the Job Pool

3.4.2.2 TwinCAT Job Task (Worker Task)

A Job Task is a task that is executed on demand. It is called from an application and is NOT executed cyclically. A Job Task can be created directly under Tasks or in a Job Pool. If you create the Job Task directly under Tasks, tasks (jobs) can be passed to it directly from a client application.

However, if you create the Job Task under a Job Pool, the tasks are transferred to the Job Pool from the application. This then assigns the task to the next Job Task in its pool that is next available.


Job Task

Name: JobTask3

Object Id: 0x02010030

Priority: 1

☑ Floating point exceptions

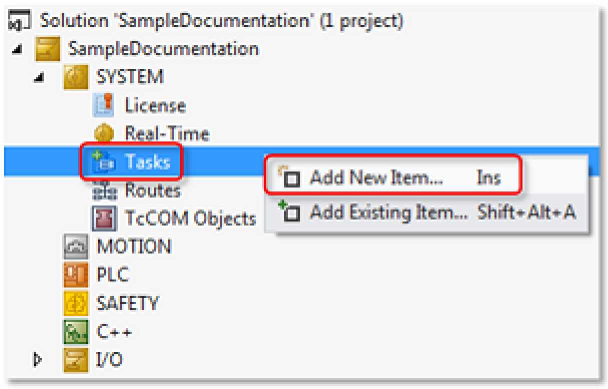
Comment:

Name	Name of the Job Task
Object Id	Object ID of the Job Task
Priority	Priority of the Job Task
Floating point exception	Specifies whether or not TwinCAT checks for floating point exceptions.
Comment	Optional comment on the Job Task

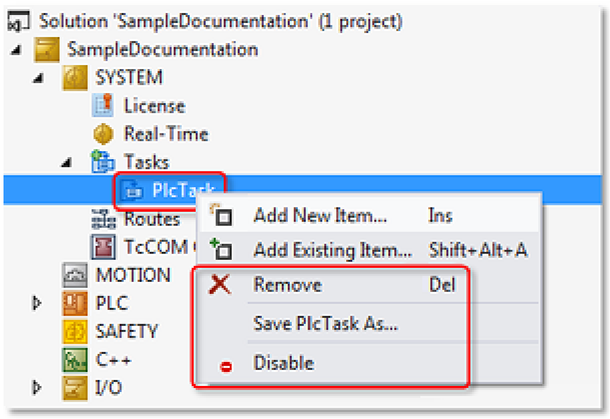
Select the setting **Floating point exception** for the calling task and for the Job Task.

3.4.3 Task Management

Create new task

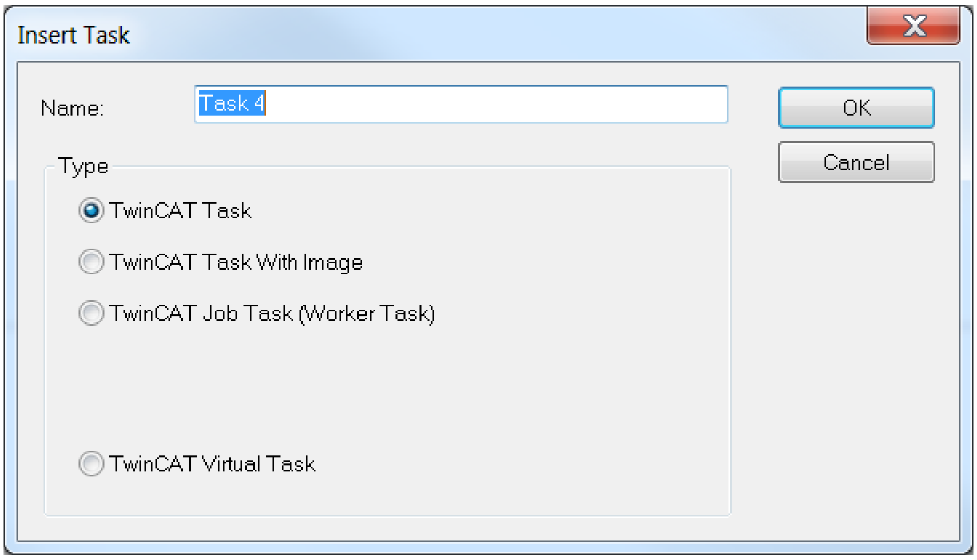


Add New Item	Opens the dialog **Insert Task**, where a new task can be configured.
Add Existing Item	Opens the standard dialog for selecting a file, via which an already existing task can be added to the project.



Add New Item	Opens the dialog **Insert Task**, where a new task can be configured.
Add Existing Item	Opens the standard dialog for selecting a file, via which an already existing task can be added to the project.
Remove	Removes the selected task from the project.
Save Plc Task as	Saves the task with the possibility to change the name.
Disable	Sets the selected task to the inactive state (not considered for the project creation), but the configurations and shortcuts are preserved and can be reactivated.

Dialog Insert Task

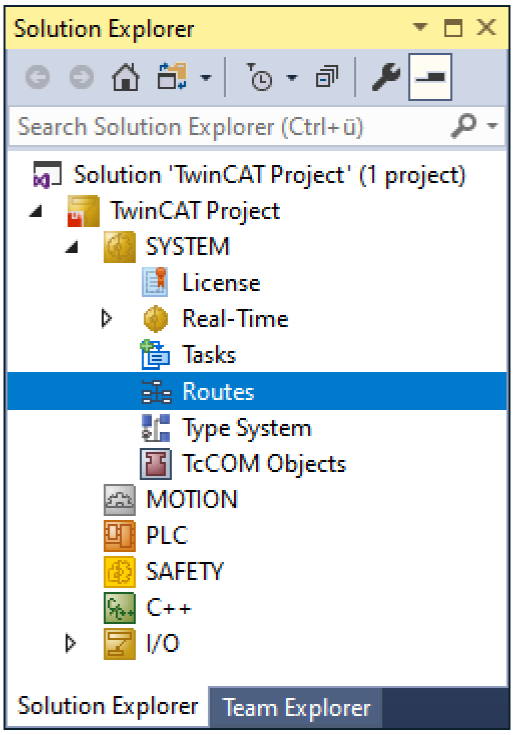


Name	Name of the task
OK	The task is created.
Cancel	The process is canceled and the dialog is closed.
TwinCAT Task	Selection of a "normal" TwinCAT Task.
TwinCAT Task With Image	Selection of a task with its own process image.
TwinCAT Job Task (Worker Task)	Selection of a Job Task.

3.5 Routes

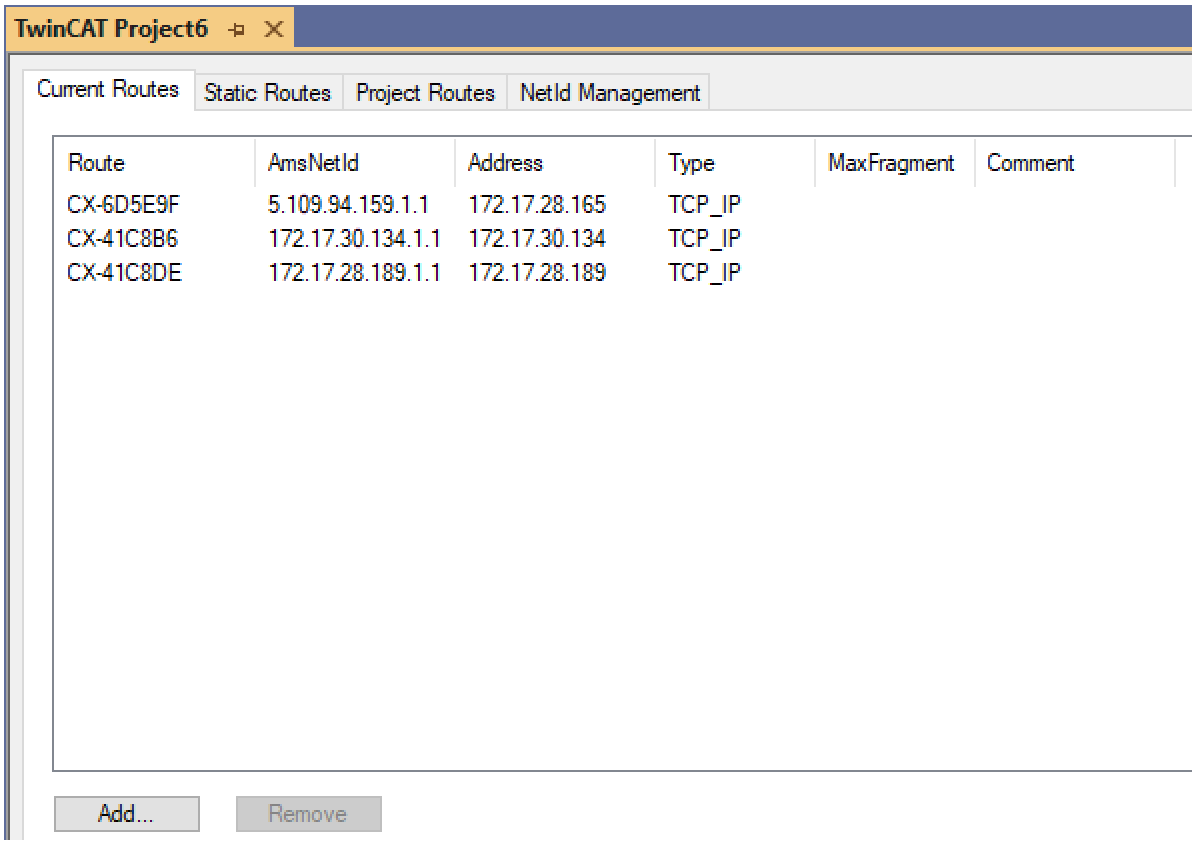
Function: Routing settings (See also documentation Basics, chapter Routing.)

Call: Double-click on the node **Routes**.



3.5.1 Current Routes tab

Current routes are routes that are currently active in the router.

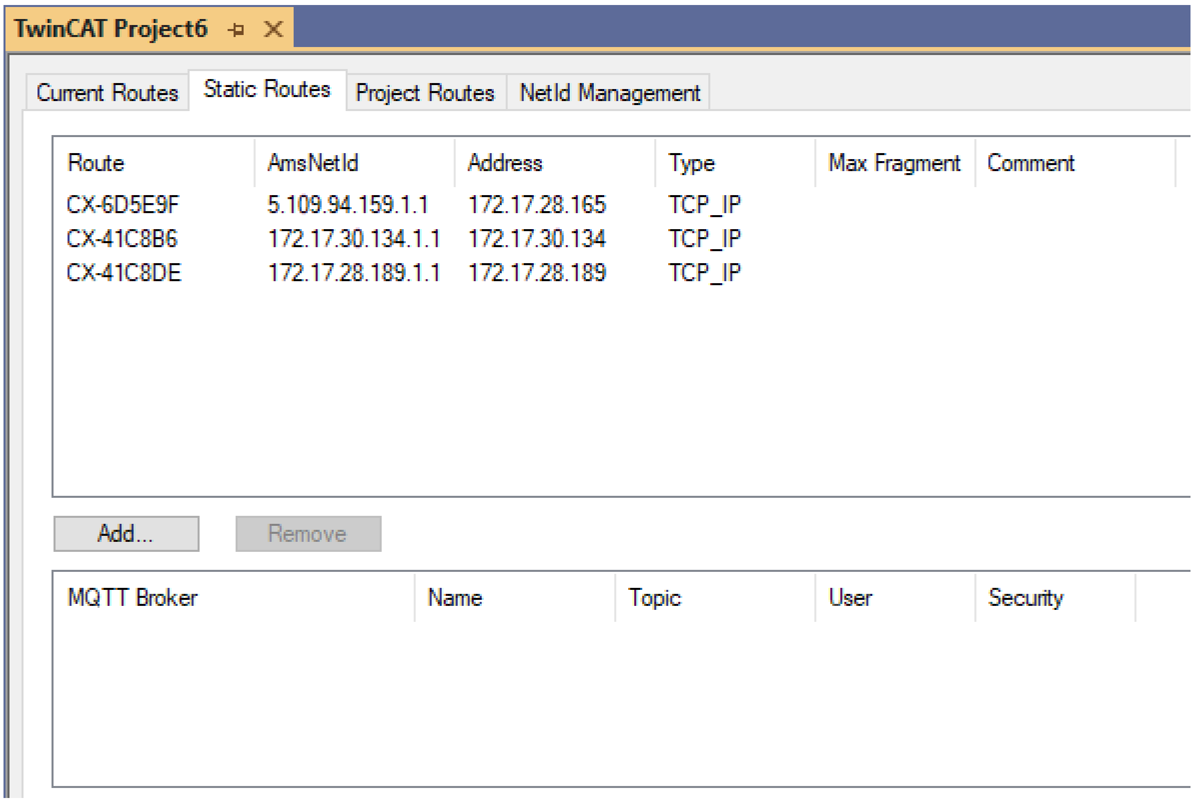


Route	AmsNetId	Address	Type	MaxFragment	Comment
CX-6D5E9F	5.109.94.159.1.1	172.17.28.165	TCP_IP		
CX-41C8B6	172.17.30.134.1.1	172.17.30.134	TCP_IP		
CX-41C8DE	172.17.28.189.1.1	172.17.28.189	TCP_IP		

Route	Name of the possible target system logged on to the current TwinCAT router. The name is freely selectable.
AmsNetId	AmsNetId of the listed target system
Address	Address of the listed TwinCAT target system The address depends on the transport protocol being used. In addition to TCP/IP addresses, addresses of Profibus devices are possible, which in turn must support the ADS protocol in order to be addressed as "target system" or "remote system".
Type	Protocol used for communication with the target system.
MaxFragment	Max. fragment size of the Ams commands. If this size is exceeded, the Ams command is automatically split into several fragments by the router.
Comment	Comment for the respective system, if configured.
Add	Calls the **Add Route dialog**, which can be used to add further TwinCAT target systems to the routing table.
Remove	Activated when an element has been selected in the list. If the button is pressed, a dialog opens asking whether the element should be deleted. The button **Cancel** cancels the deletion. The button **OK** confirms the deletion and the entry marked in the list is deleted from the routing table of the local TwinCAT router.

3.5.2 Static Routes tab

Static routes are routes that are already configured in the currently selected router.



Route	AmsNetId	Address	Type	Max Fragment	Comment
CX-6D5E9F	5.109.94.159.1.1	172.17.28.165	TCP_IP		
CX-41C8B6	172.17.30.134.1.1	172.17.30.134	TCP_IP		
CX-41C8DE	172.17.28.189.1.1	172.17.28.189	TCP_IP		

MQTT Broker	Name	Topic	User	Security

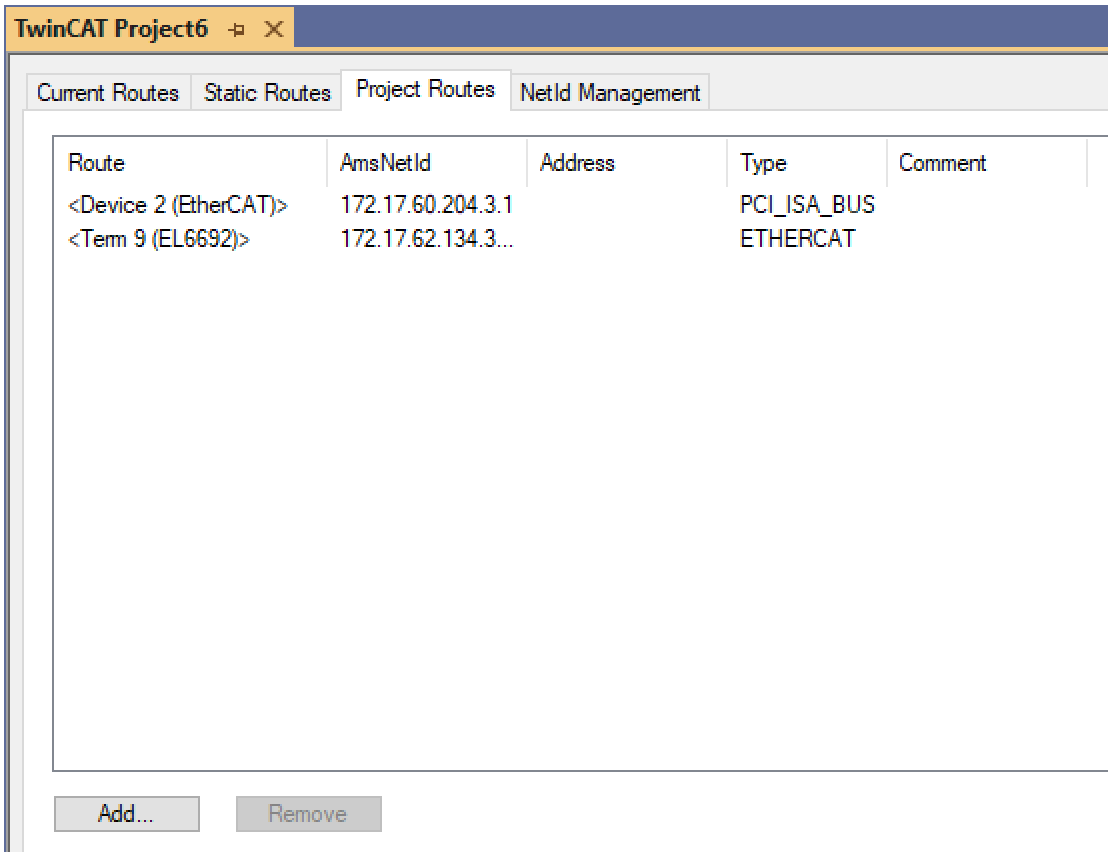
Route	Name of the possible target system logged on to the current TwinCAT router The name is freely selectable.
AmsNetId	AmsNetId of the listed target system
Address	Address of the listed TwinCAT target system The address depends on the transport protocol being used. In addition to TCP/IP addresses, addresses of Profibus devices are possible, which in turn must support the ADS protocol in order to be addressed as "target system" or "remote system".
Type	Protocol used for communication with the target system.
MaxFragment	Max. fragment size of the Ams commands. If this size is exceeded, the Ams command is automatically split into several fragments by the router.
Comment	Comment for the respective system, if configured.
Add	Calls the **Add Route dialog**, which can be used to add further TwinCAT target systems to the routing table.
Remove	Activated when an element has been selected in the list. If the button is pressed, a dialog opens asking whether the element should be deleted. The button **Cancel** cancels the deletion. The button **OK** confirms the deletion and the entry marked in the list is deleted from the routing table of the local TwinCAT router.

The configuration of ADS-over-MQTT is currently done via XML files. For more information, see chapter ADS-over-MQTT.

MQTT Broker	Address of the MQTT Broker
Name	Broker name
Topic	Topic to be connected to.
User	User name
Security	Securing communication

3.5.3 Project Routes tab

Project Routes are only stored in the project. When a new project is created, new routes are created.



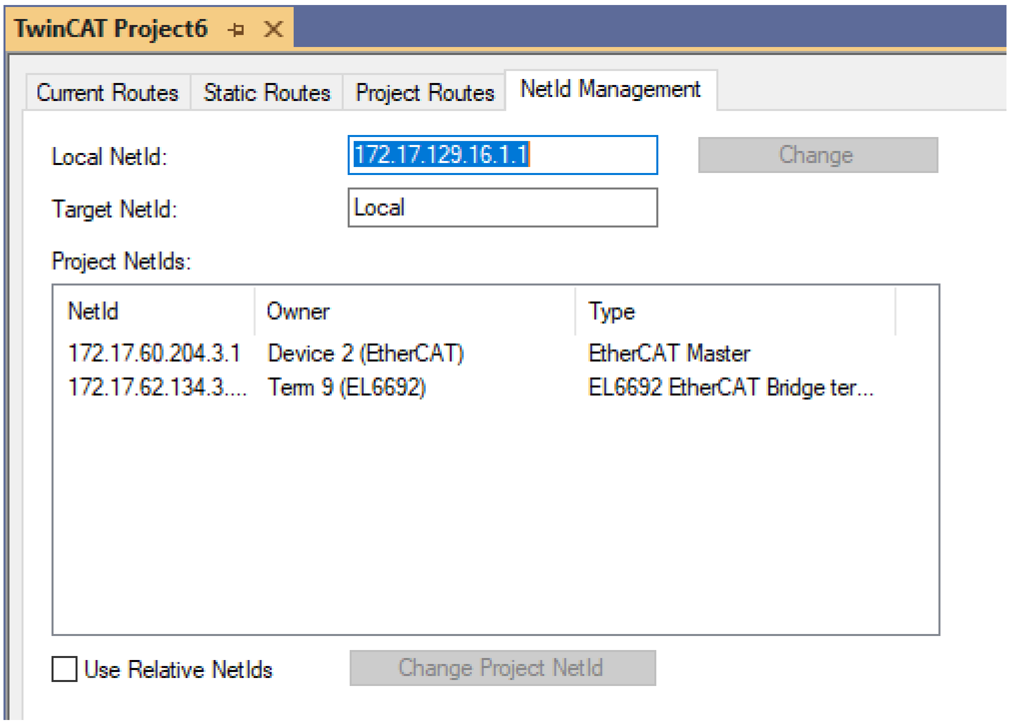
Route	AmsNetId	Address	Type	Comment
<Device 2 (EtherCAT)>	172.17.60.204.3.1		PCI_ISA_BUS	
<Term 9 (EL6692)>	172.17.62.134.3...		ETHERCAT	

Route	Name of the possible target system logged on to the current TwinCAT router. The name is freely selectable.
AmsNetId	AmsNetId of the listed target system
Address	Address of the listed TwinCAT target system The address depends on the transport protocol being used. In addition to TCP/IP addresses, addresses of Profibus devices are possible, which in turn must support the ADS protocol in order to be addressed as "target system" or "remote system".
Type	Protocol used for communication with the target system.
MaxFragment	Max. fragment size of the Ams commands. If this size is exceeded, the Ams command is automatically split into several fragments by the router.
Comment	Comment for the respective system, if configured.
Add	Calls the **Add Route dialog**, which can be used to add further TwinCAT target systems to the routing table.
Remove	Activated when an element is selected in the list. If the button is pressed, a dialog opens asking whether the element should be deleted. The button **Cancel** cancels the deletion. The button **OK** confirms the deletion and the entry marked in the list is deleted from the routing table of the local TwinCAT router.

3.5.4 NetId Management tab

Address conflict

All changes to AmsNetIds must not result in two or more identical AmsNetIds within a system. Two AmsNetIds representing the same address are identical even if one is defined absolutely and the other relatively.



Local NetId: 172.17.129.16.1.1

Target NetId: Local

Project NetIds:

NetId	Owner	Type
172.17.60.204.3.1	Device 2 (EtherCAT)	EtherCAT Master
172.17.62.134.3....	Term 9 (EL6692)	EL6692 EtherCAT Bridge ter...

☐ Use Relative NetIds

Local NetId	AmsNetId of the local system
Change	Change the local AmsNetId.
Target NetId	AmsNetId of the target system If the local system has been selected as the target system, then the "Local" entry indicates that the "Target NetId" matches the AmsNetId of the local system.
Project NetIds	Lists the AmsNetIds of the projected I/O devices. If an entry is selected in the "Project NetIds" table, then the **Change Project NetId** button becomes active.
NetId	AmsNetId of the device listed in the respective table row
Owner	Name of the device to which the AmsNetId in the respective table row belongs.
Type	Type of the device to which the AmsNetId in the respective table row belongs.
Use Relative NetIds	Converts AmsNetIds to relative AmsNetIds In order for an AmsNetId to be converted to a relative AmsNetId, the first four numbers of an AmsNetId must match the first four numbers of the "Target NetId". For the AmsNetIds listed in the table column "NetId" for which this match is fulfilled, if the checkbox **Use Relative NetIds** is checked, the respective first four numbers are combined into a group. This related group is identified by being enclosed in square brackets. Also in the **Project Routes** tab, AmsNetIds are recognizable as relative NetIds by the fact that square brackets enclose the first four numbers. The square bracket surrounding the first four numbers of a relative NetId means that these numbers are set equal to the first four numbers of the Target NetID. In this way, addressing is based on the fifth and sixth numbers relative to the "Target NetId". For a configured I/O device, the first four numbers of a relative NetId are internally represented as zero. For example, the relative NetId [172.17.12.1].2.1 is stored internally as 0.0.0.0.2.1. You can change an absolute AmsNetId into a relative AmsNetId for a projected I/O device in the dialog **Change NetId** if you set its first four numbers to the value zero. As a consequence, the AmsNetId of the corresponding projected device is displayed in the table column "NetId" and also in the **Project Routes** tab as relative NetId. On a local system, a NetId is registered with a local router and can be used locally. In contrast, a remote router generally does not know a NetId. It tries to find a remote computer where the first four digits of the remote computer's address match the first four digits of the projected I/O device's NetId. Therefore a changed value compared to the "Target NetId" within the first four numbers of the AmsNetId generally results in the fact that the projected I/O device cannot be addressed remotely unless all four first numbers carry the value zero internally. If one of the first four numbers has been changed from the "Target NetId" and a relative NetId has not been defined, then this AmsNetId cannot be converted to a relative "Ams NetId". The use of relative NetIds ensures that configured I/O devices are addressed with valid AmsNetIds, provided that the fifth and sixth numbers of the assigned AmsNetIds exclude ambiguities. The fifth and the sixth number of a respective relative AmsNetId must make a respective relative AmsNetId distinguishable from all other assigned relative AmsNetIds.
Change Project NetId	Opens the **Change NetId** dialog to change the AmsNetId of the entry marked in the "Project NetIds" table.

Open "Change NetId" dialog

1. In the **Project NetIds** table, select the projected device whose NetId you want to change.
   ⇨ The button **Change Project NetId** is activated.
2. Click on the button **Change Project NetId**.

⇨ The **Change NetId** dialog opens.

Alternative:

1. Right-click on the device in the I/O tree whose AmsNetId you want to change.
   ⇨ A context menu opens.

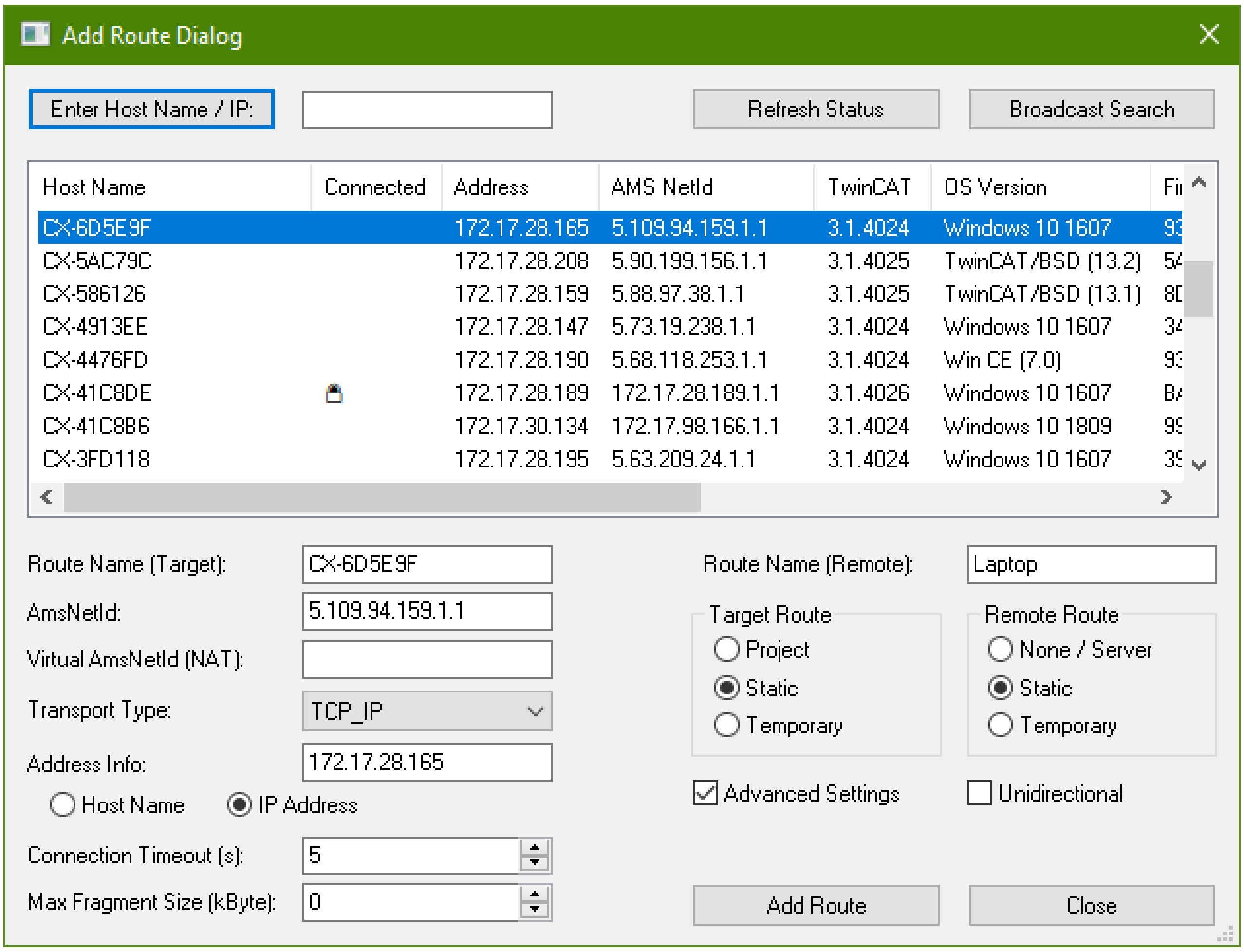
2. From the context menu, select the command **Change NetId...**.

⇨ The **Change NetId** dialog opens.

⇨ The button **OK** of the dialog **Change NetId** changes the AmsNetId and closes the dialog. The **Cancel** button below closes the dialog without changing the AmsNetId.

3.5.5 Adding routes

If you want to add a new route, this is done via the **Add Route** dialog.



Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Fi
CX-6D5E9F		172.17.28.165	5.109.94.159.1.1	3.1.4024	Windows 10 1607	93
CX-5AC79C		172.17.28.208	5.90.199.156.1.1	3.1.4025	TwinCAT/BSD (13.2)	54
CX-586126		172.17.28.159	5.88.97.38.1.1	3.1.4025	TwinCAT/BSD (13.1)	8D
CX-4913EE		172.17.28.147	5.73.19.238.1.1	3.1.4024	Windows 10 1607	34
CX-4476FD		172.17.28.190	5.68.118.253.1.1	3.1.4024	Win CE (7.0)	93
CX-41C8DE	🔒	172.17.28.189	172.17.28.189.1.1	3.1.4026	Windows 10 1607	BA
CX-41C8B6		172.17.30.134	172.17.98.166.1.1	3.1.4024	Windows 10 1809	99
CX-3FD118		172.17.28.195	5.63.209.24.1.1	3.1.4024	Windows 10 1607	39

Route Name (Target):	CX-6D5E9F
AmsNetId:	5.109.94.159.1.1
Virtual AmsNetId (NAT):	
Transport Type:	TCP_IP
Address Info:	172.17.28.165
	Host Name [✓ IP Address]
Connection Timeout (s):	5
Max Fragment Size (kByte):	0
Route Name (Remote):	Laptop
Target Route	Project [✓ Static] Temporary
Remote Route	None / Server [✓ Static] Temporary
[✓ Advanced Settings]	Unidirectional

Search for target systems:

Enter Host Name / IP	Enter a host name or IP address in the input field and press the button to search for a TwinCAT target system or a remote system in the network (subnet).
Refresh Status	Updates the list of potential target systems.
Broadcast Search	Search for all TwinCAT systems connected to the current subnet. The targets must be in TwinCAT configuration mode or run mode.

List of target systems

Host Name	Name of the target or remote system found
Connected	Shows the status of the connection to the target system.
Address	Address of the device The type of address depends on the transport protocol used. In the case of "TCP/IP" it is the TCP/IP address.
AmsNetId	The TwinCAT identification address of the device for the TwinCAT Router
TwinCAT	Version and build number of the TwinCAT target system
OS Version	Operating system on the target

Data of the selected target system

When a target is selected from the list, these fields display the associated connection data. The fields can be edited.

Route Name (Target)	Name of the selected target system
AmsNetId	AmsNetId of the selected target system
Virtual AmsNetId (NAT)	
Transport Type	Transport type of the selected target system
Address Info	Either the name or the IP address of the corresponding target system in the network. If the name cannot be recognized, switch to the IP address.
Host Name	Selection that the route should be saved by target system name.
IP Adress	Selection that the route should be saved by IP address.
Connection Timeout (s)	Setting the connection timeout in seconds
Max Fragment Size (kByte)	Max. fragment size of the Ams commands. If this size is exceeded, the Ams command is automatically split into several fragments by the router.

Route related settings

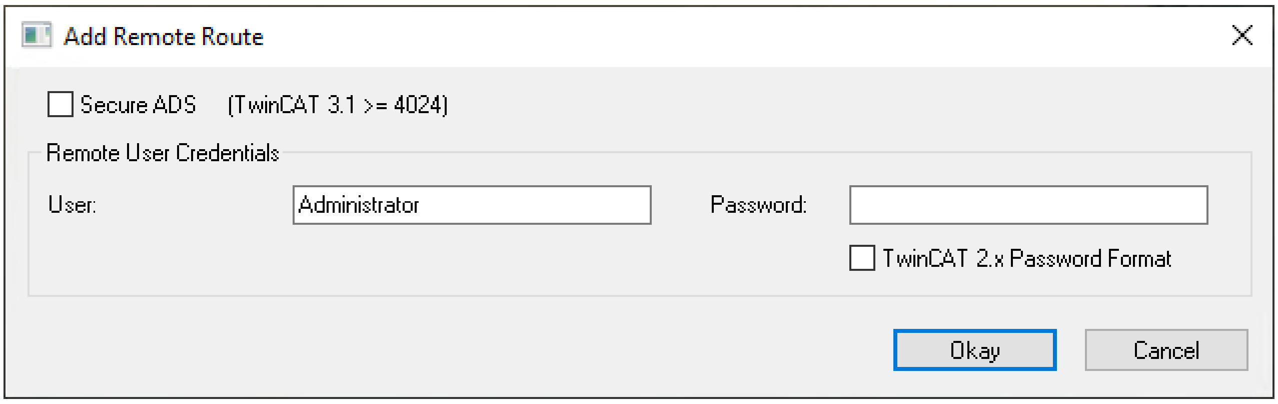
Route Name (Remote)	Name with which the route is entered on the remote system.
Target Route	Route from the local system to the target system • Project • Static • Temporary
Remote Route	Route from remote system to local system • None • Static • Temporary
Project	Saves the route in the project.
Static	Saves the route on the system (project-independent).
None	Only Secure ADS routes can be added. These are stored on the target system as a server with the corresponding certificate or preshared key (PSK).
Temporary	Route is saved only until the next restart of the target system.
Advanced Settings	Displays or hides the route settings described in this chapter.
Unidirectional	Create a directed ADS communication (ADS command calls received from the opposite system).

Adding route

Add Route	Adds the selected target system with the settings made to the "Routes" (list of configured target systems).
Close	Closes the dialog box.

Login Information

To access and configure a target system, you must be logged on to it. After clicking the button **Add Route** you can enter user name and password. This data is stored in the route, so you only have to enter it once. If you use a TwinCAT version > 3.1.4024 or if you do not want to use Secure ADS, the dialog for entering the login information is as follows:



Add Remote Route

☐ Secure ADS (TwinCAT 3.1 >= 4024)

Remote User Credentials

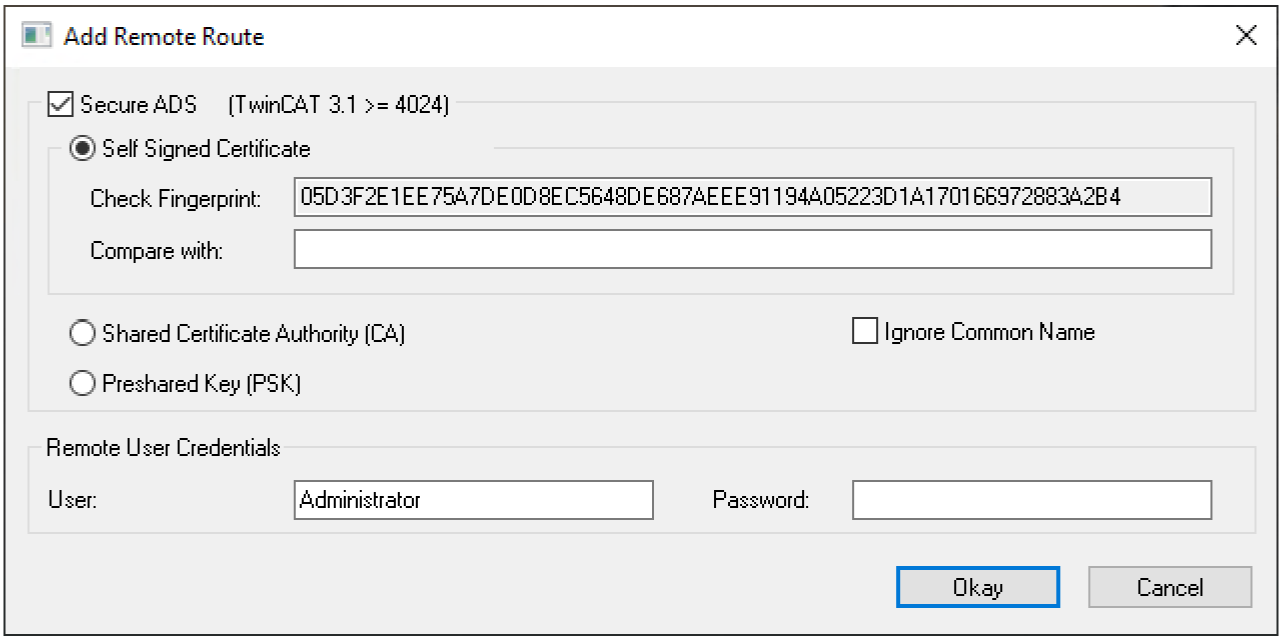
User: Administrator

Password:

☐ TwinCAT 2.x Password Format

User	User name with administrator rights, which should add the route.
Password	User password

Starting from a TwinCAT 3.1.4024 version, if the option **Secure ADS** is switched on, the dialog appears as follows:



Add Remote Route

☑ Secure ADS (TwinCAT 3.1 >= 4024)

● Self Signed Certificate

Check Fingerprint: 05D3F2E1EE75A7DE0D8EC5648DE687AEEE91194A05223D1A170166972883A2B4

Compare with:

○ Shared Certificate Authority (CA)

☐ Ignore Common Name

○ Preshared Key (PSK)

Remote User Credentials

User: Administrator

Password:

Further information can be found in the documentation Secure ADS.

Secure ADS offers three ways of providing the keys required for the encryption:

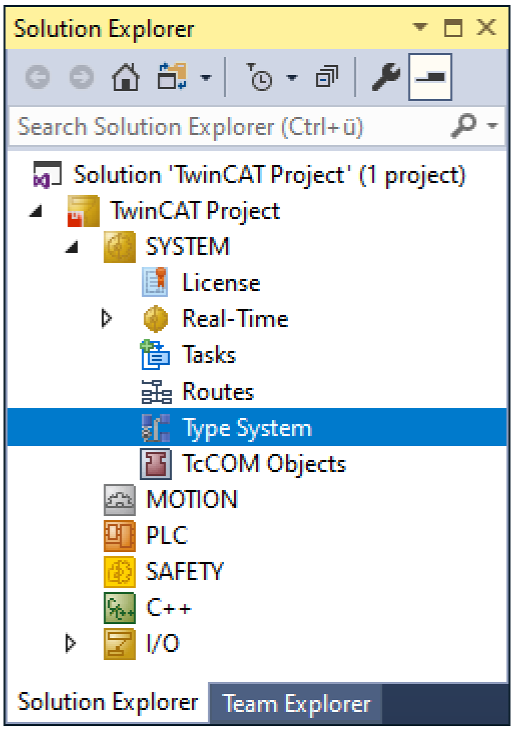
- Self Signed Certificate
- Shared Certificate Authority (CA)
- Preshared Key (PSK)

The configuration of a Secure ADS Route is described in the documentation Secure ADS.

3.6 Type System

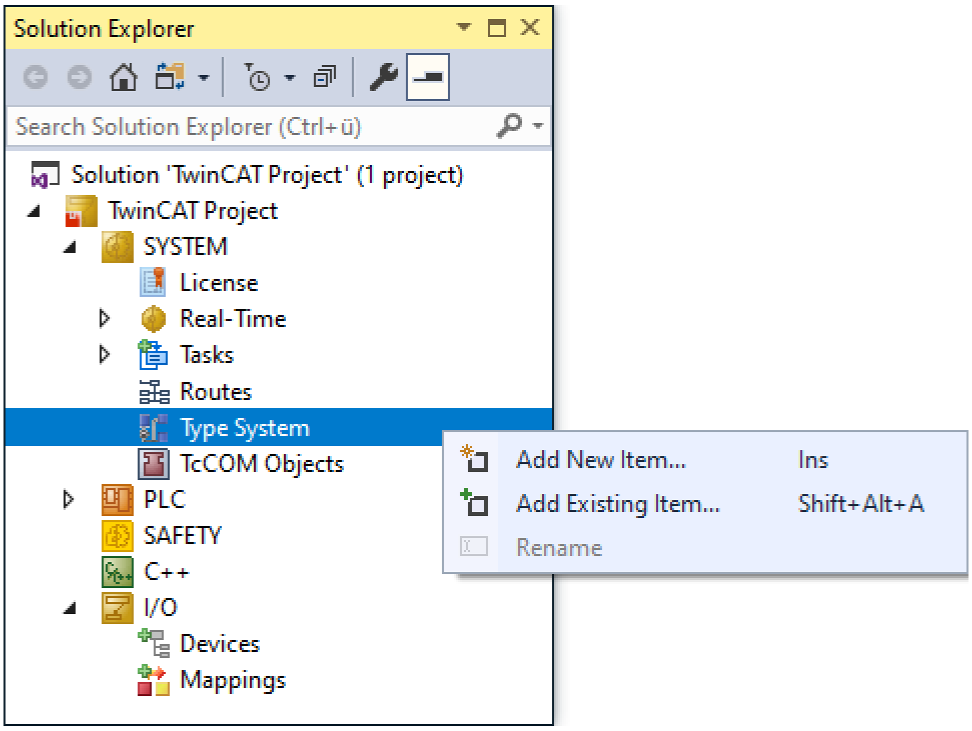
Function: Data type management

Call: Double click on the object **Type System**.



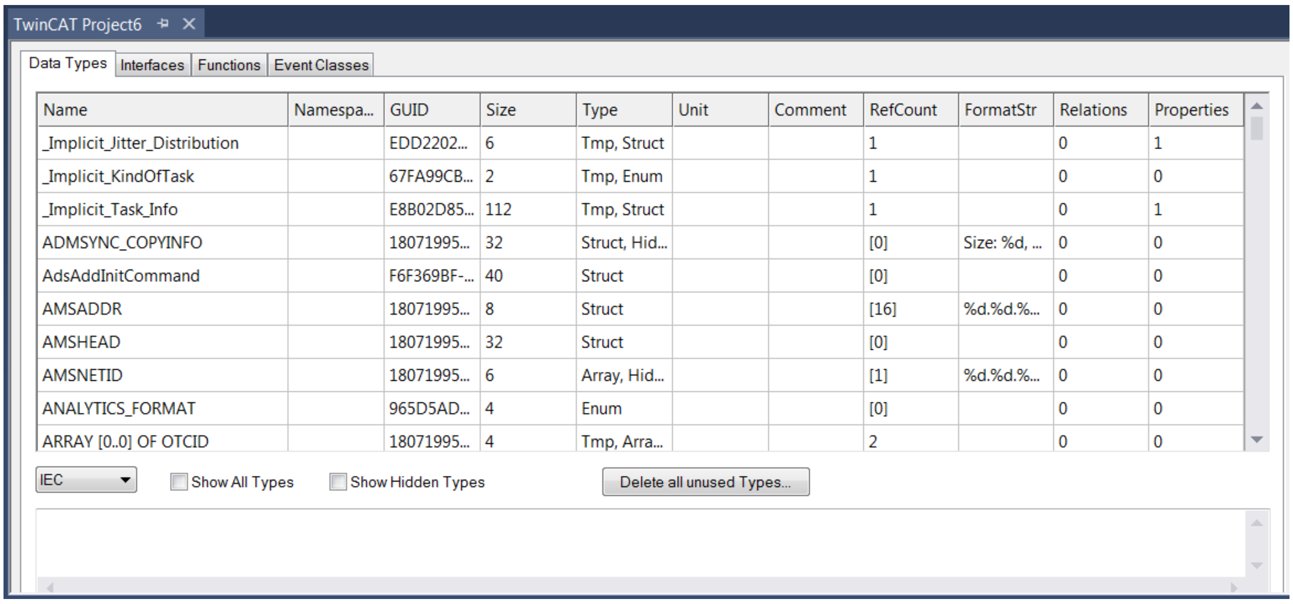
See also Type System.

Context menu



Add New Item…	Add a new configuration.
Add Existing Item…	Add an existing configuration.
Rename	Rename element.

3.6.1 Data Types tab

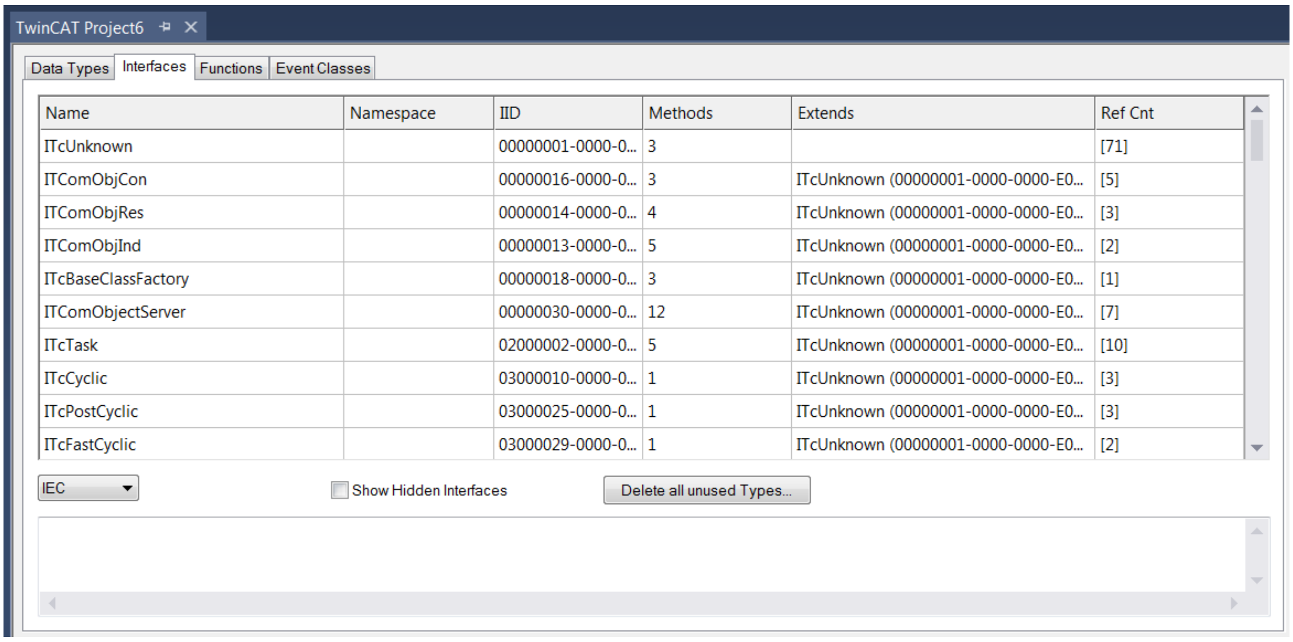


Name	Namespa...	GUID	Size	Type	Unit	Comment	RefCount	FormatStr	Relations	Properties
_Implicit_Jitter_Distribution		EDD2202...	6	Tmp, Struct			1		0	1
_Implicit_KindOfTask		67FA99CB...	2	Tmp, Enum			1		0	0
_Implicit_Task_Info		E8B02D85...	112	Tmp, Struct			1		0	1
ADMSYNC_COPYINFO		18071995...	32	Struct, Hid...			[0]	Size: %d, ...	0	0
AdsAddInitCommand		F6F369BF-...	40	Struct			[0]		0	0
AMSADDR		18071995...	8	Struct			[16]	%d.%d.%...	0	0
AMSHEAD		18071995...	32	Struct			[0]		0	0
AMSNETID		18071995...	6	Array, Hid...			[1]	%d.%d.%...	0	0
ANALYTICS_FORMAT		965D5AD...	4	Enum			[0]		0	0
ARRAY [0..0] OF OTCID		18071995...	4	Tmp, Arra...			2		0	0

IEC ▾ ☐ Show All Types ☐ Show Hidden Types [Delete all unused Types...]

Name	Name of the data type
Namespace	Namespace of the data type
GUID	GUID of the data type
Size	Data type size
Type	Type of the data type (structure, array, ...)
Unit	Unit of the data type
Comment	Comment
RefCount	Reference Counter of the data type
FormatStr	Formatting string of the data type
Relations	Number of relations to other data types
Properties	Number of attributes that specify the use of the data type.
IEC	Dropbox to toggle the view of the data type in the display field below. The representation as XML, PLC code (IEC) or C++ code can be selected. In addition, you can display the source from which the data type originates.
Show All Types	Shows all data types.
Show Hidden Types	Also shows hidden types, i.e. the data types from which others have already been derived.
Delete all unused Types	Deletes all unused data types.

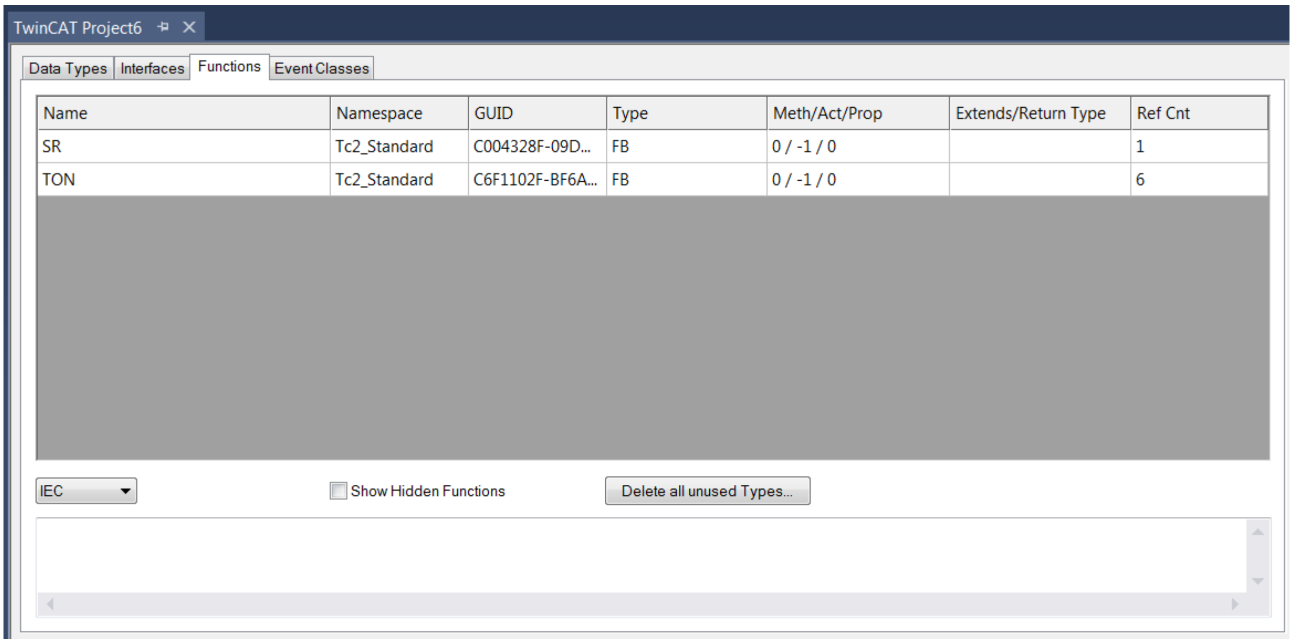
3.6.2 Interfaces tab



Name	Namespace	IID	Methods	Extends	Ref Cnt
ITcUnknown		00000001-0000-0...	3		[71]
ITComObjCon		00000016-0000-0...	3	ITcUnknown (00000001-0000-0000-E0...	[5]
ITComObjRes		00000014-0000-0...	4	ITcUnknown (00000001-0000-0000-E0...	[3]
ITComObjInd		00000013-0000-0...	5	ITcUnknown (00000001-0000-0000-E0...	[2]
ITcBaseClassFactory		00000018-0000-0...	3	ITcUnknown (00000001-0000-0000-E0...	[1]
ITComObjectServer		00000030-0000-0...	12	ITcUnknown (00000001-0000-0000-E0...	[7]
ITcTask		02000002-0000-0...	5	ITcUnknown (00000001-0000-0000-E0...	[10]
ITcCyclic		03000010-0000-0...	1	ITcUnknown (00000001-0000-0000-E0...	[3]
ITcPostCyclic		03000025-0000-0...	1	ITcUnknown (00000001-0000-0000-E0...	[3]
ITcFastCyclic		03000029-0000-0...	1	ITcUnknown (00000001-0000-0000-E0...	[2]

Name	Interface name
Namespace	Namespace of the interface
IID	GUID of the interface
Methods	Number of methods of the interface
Extends	"Parent" interfaces
Ref Cnt	Reference counter of the interface

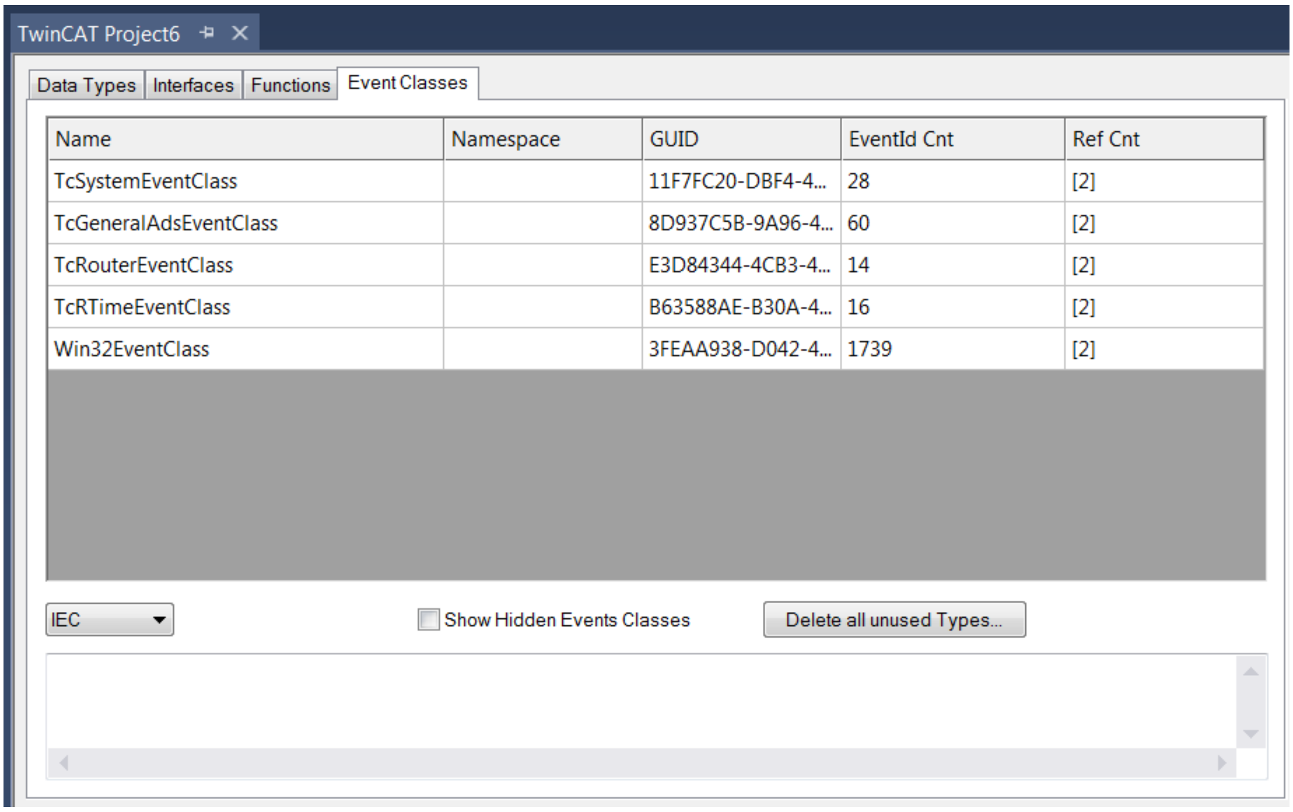
3.6.3 Functions tab



Name	Namespace	GUID	Type	Meth/Act/Prop	Extends/Return Type	Ref Cnt
SR	Tc2_Standard	C004328F-09D...	FB	0 / -1 / 0		1
TON	Tc2_Standard	C6F1102F-BF6A...	FB	0 / -1 / 0		6

Name	Name of the function
Namespace	Namespace of the function
GUID	GUID of the function
Type	Type of the function
Meth/Act/Prop	Number of methods/actions and properties
Extends/Return Type	Parent function/ return value
Ref Cnt	Reference counter
IEC	Representation of the function in the display window below as PLC code (IEC), XML or C++ code or the source of the function (TMC file in which it is declared).
Show Hidden Functions	Shows hidden functions.
Delete all unused Types…	Deletes unused types.

3.6.4 Event Classes tab



Name	Namespace	GUID	EventId Cnt	Ref Cnt
TcSystemEventClass		11F7FC20-DBF4-4...	28	[2]
TcGeneralAdsEventClass		8D937C5B-9A96-4...	60	[2]
TcRouterEventClass		E3D84344-4CB3-4...	14	[2]
TcRTimeEventClass		B63588AE-B30A-4...	16	[2]
Win32EventClass		3FEAA938-D042-4...	1739	[2]

Name	Name of the Event Class
Namespace	Namespace of the Event Class
GUID	GUID of the Event Class
EventId Cnt	Number of event IDs
Ref Cnt	Reference counter
IEC	Representation of the event class in the display window below as PLC code (IEC), XML or C++ code or the source of the function (TMC file in which it is declared).
Show Hidden Events Classes	Also shows hidden Event Classes.
Delete all unused Types…	Deletes types that are no longer used.

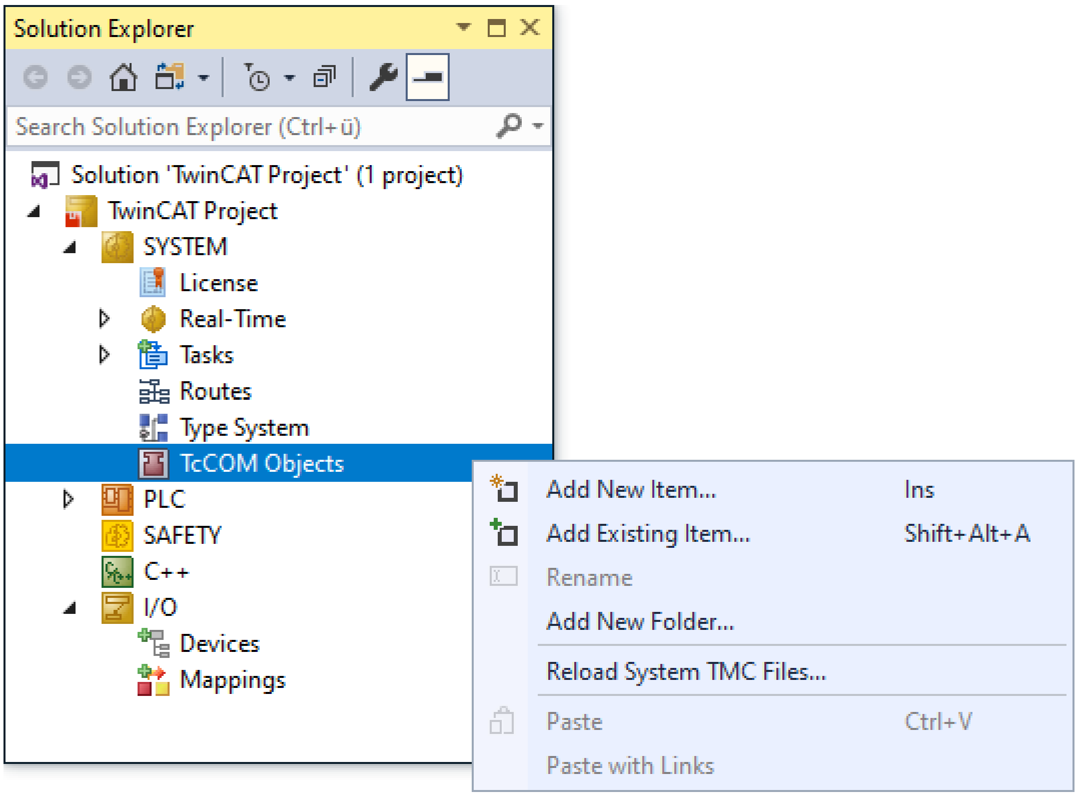
3.7 TcCOM Objects

Function: Used to configure and handle TcCOM objects in the project (add, load, Online Change, state switching).

Call: Double-click on the node **TcCOM Objects.**

See documentation TcCOM objects.

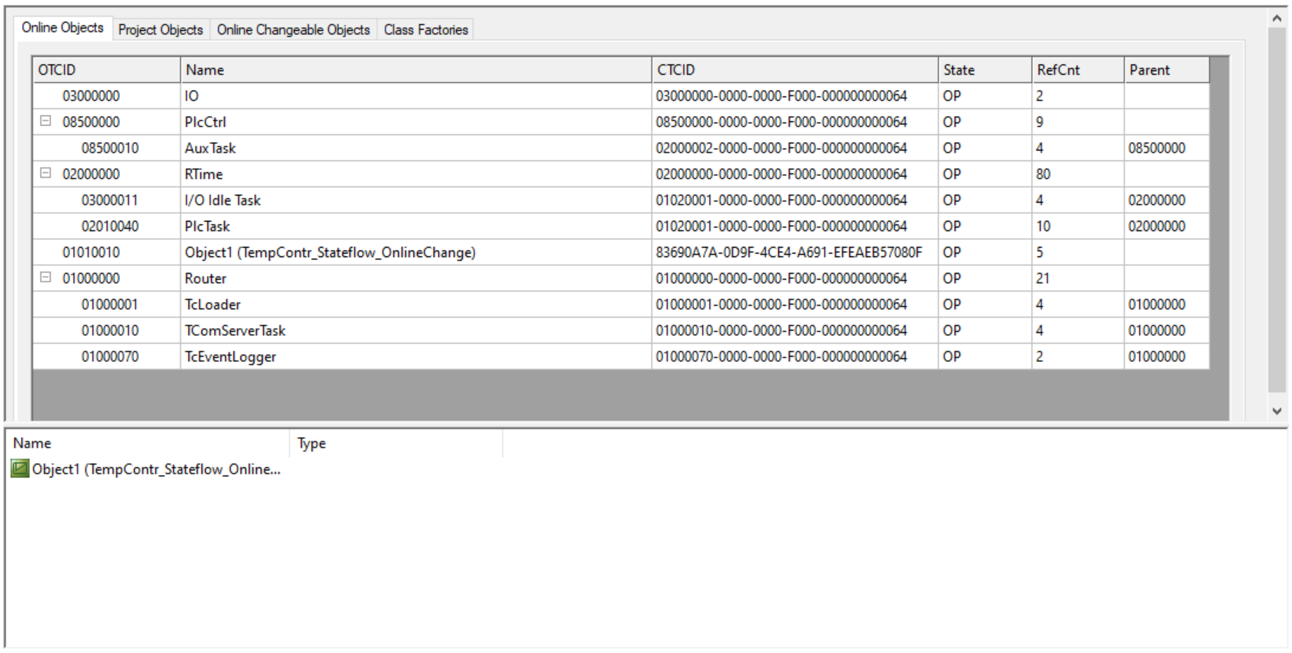
Context menu



Add New Item…	Add new TcCOM object.
Add Existing Item…	Add existing TcCOM object.
Rename	Rename TcCOM object.
Add New Folder…	Add new folder.
Reload System TMC Files…	Reloads the TMC files. This is necessary if the TMC files have been changed in the background, e.g. by a generation of new TMC files by the code generation targets that are not triggered from TwinCAT. (See e.g. TE1400)
Paste	Copy TcCOM object.
Paste with Links	Copy TcCOM object with links.

3.7.1 Online Objects tab

The **Online Objects** tab shows all TcCOM objects that are currently loaded at runtime.


Online Objects | Project Objects | Online Changeable Objects | Class Factories

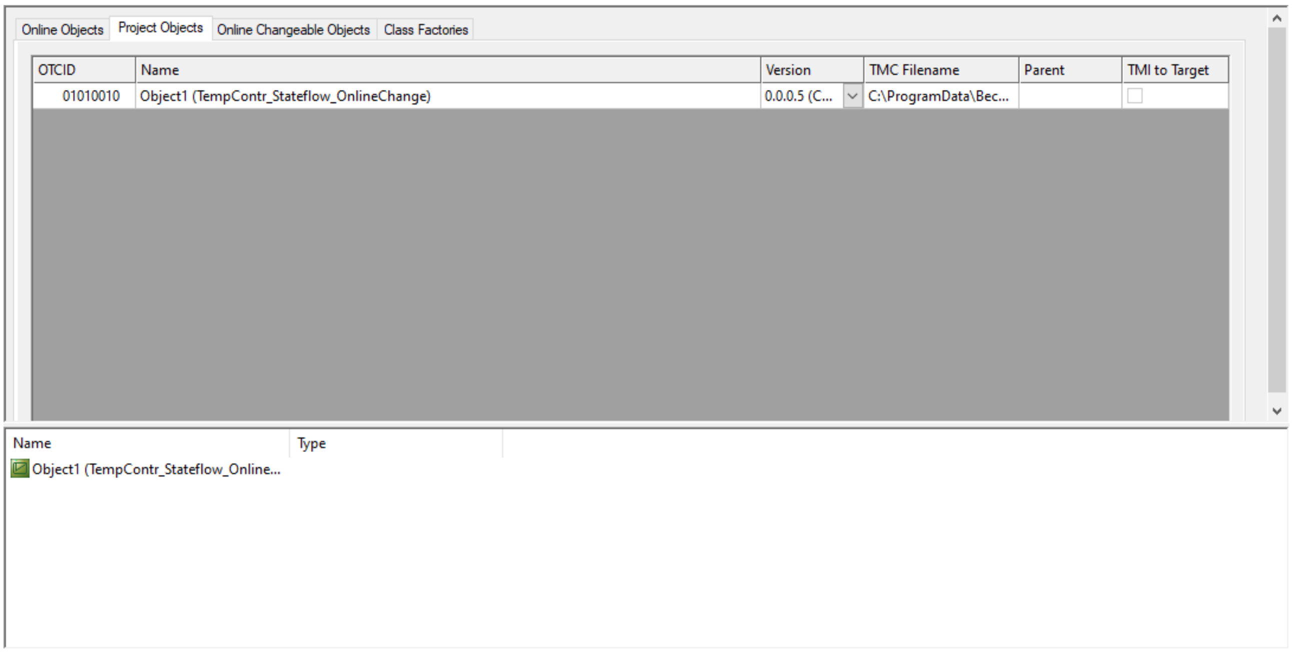
OTCID	Name	CTCID	State	RefCnt	Parent
03000000	IO	03000000-0000-0000-F000-000000000064	OP	2	
⊟ 08500000	PlcCtrl	08500000-0000-0000-F000-000000000064	OP	9	
08500010	AuxTask	02000002-0000-0000-F000-000000000064	OP	4	08500000
⊟ 02000000	RTime	02000000-0000-0000-F000-000000000064	OP	80	
03000011	I/O Idle Task	01020001-0000-0000-F000-000000000064	OP	4	02000000
02010040	PlcTask	01020001-0000-0000-F000-000000000064	OP	10	02000000
01010010	Object1 (TempContr_Stateflow_OnlineChange)	83690A7A-0D9F-4CE4-A691-EFEAEB57080F	OP	5	
⊟ 01000000	Router	01000000-0000-0000-F000-000000000064	OP	21	
01000001	TcLoader	01000001-0000-0000-F000-000000000064	OP	4	01000000
01000010	TComServerTask	01000010-0000-0000-F000-000000000064	OP	4	01000000
01000070	TcEventLogger	01000070-0000-0000-F000-000000000064	OP	2	01000000

Name	Type
Object1 (TempContr_Stateflow_Online...	

OTCID	Object Id of the TcCOM object
Name	Name of the TcCOM object
CTCID	Class-Id of the TcCOM object
State	State of the TcCOM object
RefCnt	Reference counter of the TcCOM object
Parent	Parent object

3.7.2 Project Objects tab

This tab contains a list of all TcCOM objects that have been added to the project.


Online Objects | Project Objects | Online Changeable Objects | Class Factories

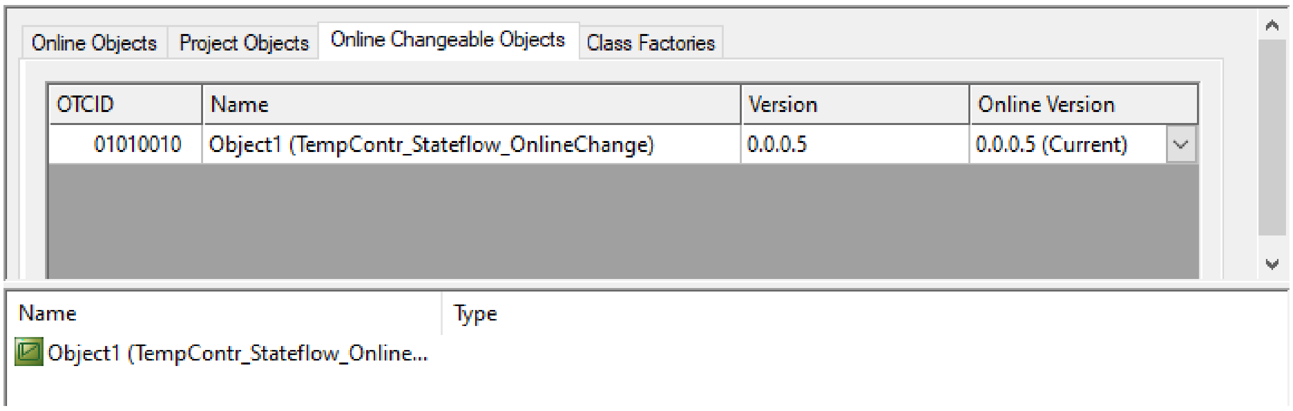
OTCID	Name	Version	TMC Filename	Parent	TMI to Target
01010010	Object1 (TempContr_Stateflow_OnlineChange)	0.0.0.5 (C...	C:\ProgramData\Bec...		☐

Name	Type
Object1 (TempContr_Stateflow_Online...	

OTCID	Object ID of the TcCOM object
Name	Name of the TcCOM object
Version	Drop-down menu for selecting the version of the TcCOM object to be loaded in the project.
TMC Filename	Name of the added TMC file
Parent	Parent TcCOM object
TMI to Target	Selection whether the TMI file (TwinCAT Module Instance) should also be written to the target system.

3.7.3 Online Changeable Objects tab

This tab lists all TcCOM objects that are capable of an Online Change. If several versions of a TcCOM object are contained in the target system repository, you can select a version to be loaded from the drop-down menu **Online Version**.



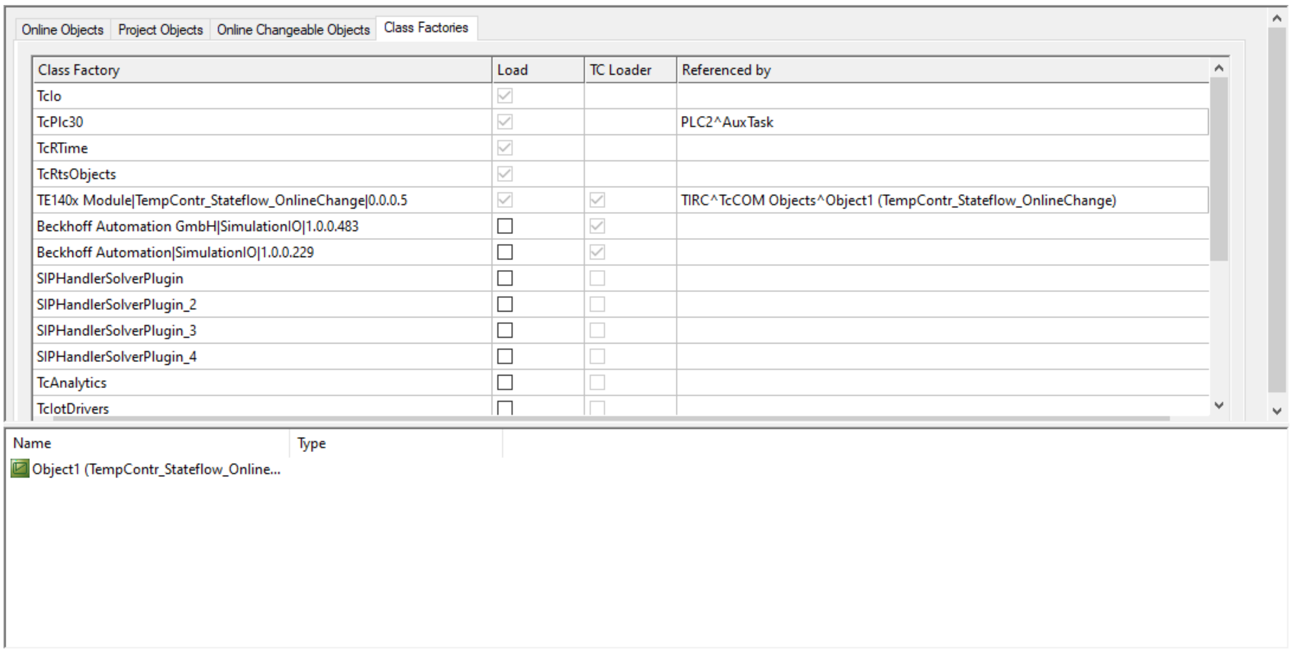
OTCID	Name	Version	Online Version
01010010	Object1 (TempContr_Stateflow_OnlineChange)	0.0.0.5	0.0.0.5 (Current)

Name	Type
Object1 (TempContr_Stateflow_Online...	

OTCID	Object ID of the TcCOM object
Name	Name of the TcCOM object
Version	Loaded version of the TcCOM object
Online Version	Selection of the version to be loaded.

3.7.4 Class Factories tab

Each TwinCAT driver contains exactly one Class Factory. When TwinCAT is loaded, these register with the TwinCAT object server. The Class Factories that are available on a system are listed in this tab. If these are to be loaded when TwinCAT is started, even if no TcCOM module exists in the project yet, this can be set accordingly on this tab.



Class Factory	Load	TC Loader	Referenced by
TcIo	✓		
TcPlc30	✓		PLC2^AuxTask
TcRTime	✓		
TcRtsObjects	✓		
TE140x Module|TempContr_Stateflow_OnlineChange|0.0.0.5	✓	✓	TIRC^TcCOM Objects^Object1 (TempContr_Stateflow_OnlineChange)
Beckhoff Automation GmbH|SimulationIO|1.0.0.483	☐	✓	
Beckhoff Automation|SimulationIO|1.0.0.229	☐	✓	
SIPHandlerSolverPlugin	☐	☐	
SIPHandlerSolverPlugin_2	☐	☐	
SIPHandlerSolverPlugin_3	☐	☐	
SIPHandlerSolverPlugin_4	☐	☐	
TcAnalytics	☐	☐	
TcIotDrivers	☐	☐	

Name	Type
Object1 (TempContr_Stateflow_Online...	

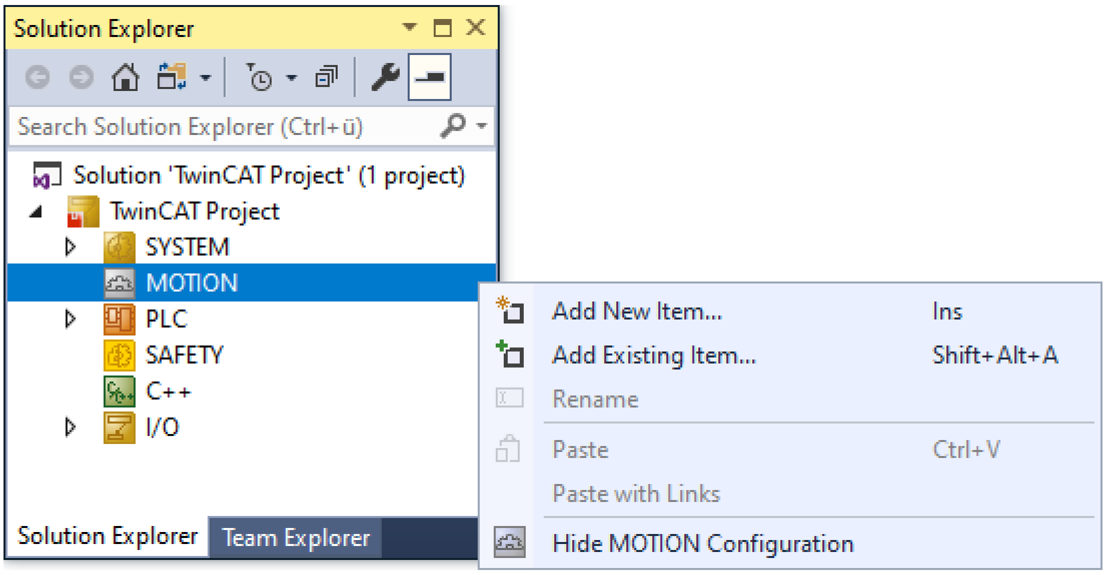
Class Factory	Name of the Class Factory
Load	Selection whether this Class Factory should be loaded.
TC Loader	Selection of whether the Class Factory is to be loaded via TwinCAT or via Windows Loader.
Referenced by	Display in which TcCOM module this Class Factory is referenced.

4 MOTION

Function: This node contains the configuration for linking with drive technologies.

Call: Select **Add New Item...** in the context menu and create a new Motion configuration.

Context menu



Add New Item…	Create new Motion configuration.
Add Existing Item…	Add existing Motion configuration.
Rename	Rename Motion configuration.
Paste	Paste Motion configuration.
Paste with Links	Paste Motion configuration with linked devices.
Hide MOTION Configuration	Hide Motion configuration.

The connection as well as the use of drive technologies in a TwinCAT project are described in detail in the TF5500 section of the Beckhoff Information System:

- TF50x0 | TwinCAT 3 NC PTP
- TF5100 | NC I
- TE1500 | Valve Diagram Editor
- TE1510 | CAM Design Tool

4.1 NC/PTP NCI Configuration

A NC/PTP NCI Configuration is divided into the following sections:

NC-Task 1 SAF [▸ 59]	• SAF task • Task for the block execution • Task in which the setpoint generation takes place. • Task that serves the fieldbus IO of the NC.
NC-Task 1 SVB [▸ 61]	• SVB task • Task for the block preparation • Linking and "Look-ahead" function of NCI segments • no effect on single-axis movements (PTP) • not responsible for the fieldbus IO of the NC
Image [▸ 62]	• NC process image
Tables [▸ 62]	• Tables, e.g. for cam plates
Objects [▸ 62]	• additional TcCOM objects
Axes	• NC axis configuration.

Additionally, channels of the following types can be added to the configuration:

NC Channel (for Interpolation)	• TF5100 | TwinCAT 3 NC I
NC Channel (for FIFO Axes)	• TF5055 | TwinCAT 3 NC FIFO Axes
NC Channel (for Kinematic Transformation)	• TF511x | TwinCAT 3 Kinematic Transformation

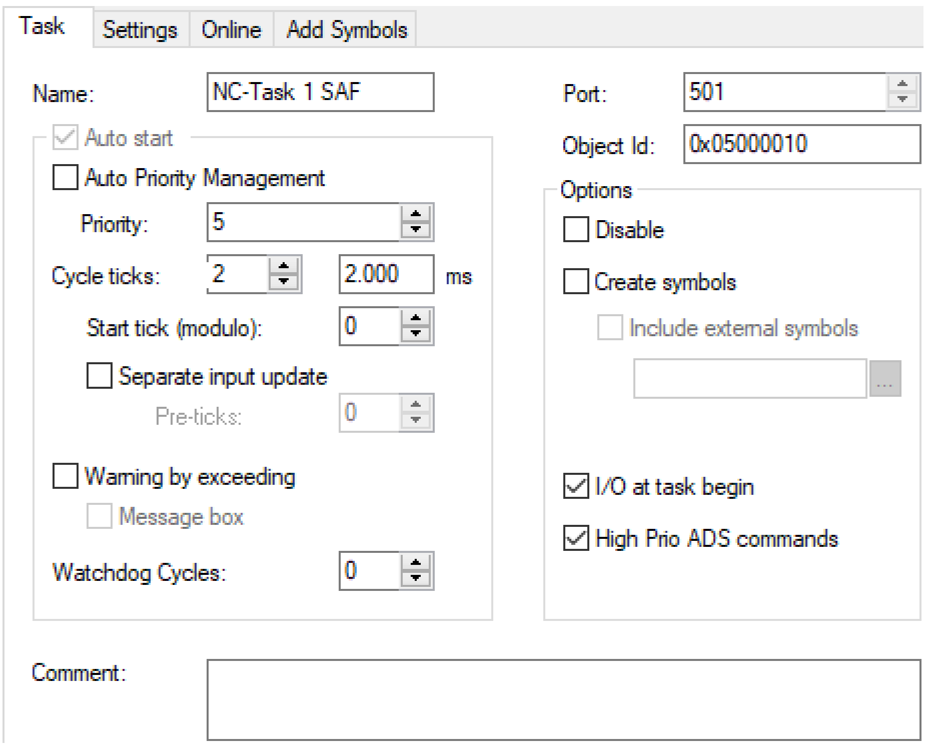
4.1.1 SAF task

The block execution task (SAF task) directly executes commands that do not require preprocessing or have been prepared by the SVB task [▸ 61] and handles cyclic communication with the drive devices.

- motion commands of the Tc2_MC2 library, like MC_MoveAbsolute, MC_MoveRelative, MC_MoveVeloctiy etc.
- coupling axes
- cyclic setpoint generation for all axes and output to drives
- cyclic acquisition of the actual position and position control, if not directly controlled by the drive
- I/O communication, e.g. for the evaluation of latch positions

Task tab

The SAF task is configured via the **NC/PTP NCI Configuration** node below the **MOTION** node, which is called **NC-Task 1 SAF** by default. Details about the task dialog, see TE1000 | The TwinCAT Project.


Task | Settings | Online | Add Symbols

Name: NC-Task 1 SAF
Port: 501
Object Id: 0x05000010
[✓ Auto start]
[ ] Auto Priority Management
Priority: 5
Cycle ticks: 2 | 2.000 ms
Start tick (modulo): 0
[ ] Separate input update
Pre-ticks: 0
[ ] Warning by exceeding
[ ] Message box
Watchdog Cycles: 0

Options
[ ] Disable
[ ] Create symbols
[ ] Include external symbols
[✓ I/O at task begin]
[✓ High Prio ADS commands]

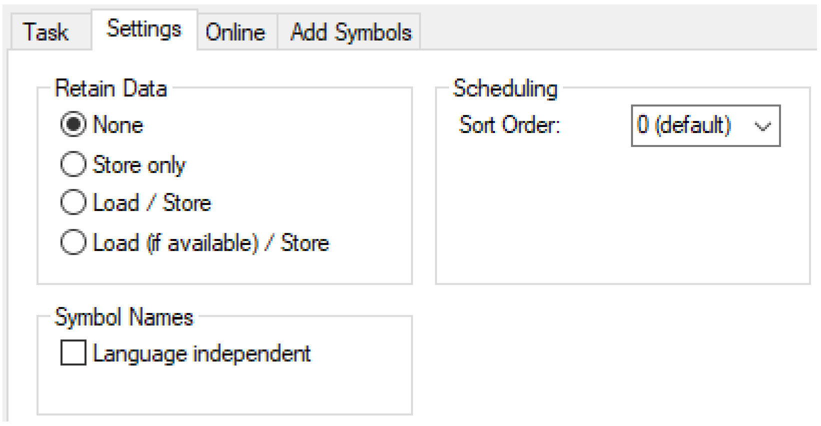
Comment:

High Prio ADS commands

As of TwinCAT 3.1 Build 4026, ADS communication takes place at the start of an SAF task cycle by default. This corresponds to a higher prioritization of ADS communication.

Up to and including TwinCAT 3.1 Build 4024, ADS communication takes place at the end of an SAF task cycle. By unchecking **High Prio ADS commands**, ADS communication also takes place on a TwinCAT 3.1 Build 4026 system at the end of an SAF task cycle.

Settings tab


Task | Settings | Online | Add Symbols

Retain Data
[✓ None]
Store only
Load / Store
Load (if available) / Store

Scheduling
Sort Order: 0 (default)

Symbol Names
[ ] Language independent

Retain data

TwinCAT-NC uses retain data to restore the position of individual axes with absolute measured value system at system startup. The Retain setting determines globally whether data required for this purpose is saved when the system is stopped and loaded when the system is started. In addition, the storage must be parameterized for each axis that requires such data (see Data Persistence [▸ 80]).

NOTICE

Prevent data loss

If retain data is used, it is recommended that the system is backed up with a UPS so that the data can be stored safely even if the supply voltage fails.

- *None:* No retain data is saved or loaded.

- *Store only:* No retain data is loaded at system startup. Retain data is saved when the system is stopped. This setting is only used for compatibility with old configurations.
- *Load/Store:* Retain data is stored at system stop and loaded at system start. If there is no data or only corrupt data at system startup, the system aborts the startup process with an error.
  The system cannot be started in this mode when the retain data is configured for the first time, so the *Mode Load (if available)/Store* should be set first and then reset to *Load/Store* after a successful system start.
- *Load (if available)/Store:* Retain data is stored at system stop and loaded at system start. If there is no data or only corrupt data at system startup, the system starts without retain data.
  Axes that rely on retain data are in the "not referenced" state. The application should check this state and take action.

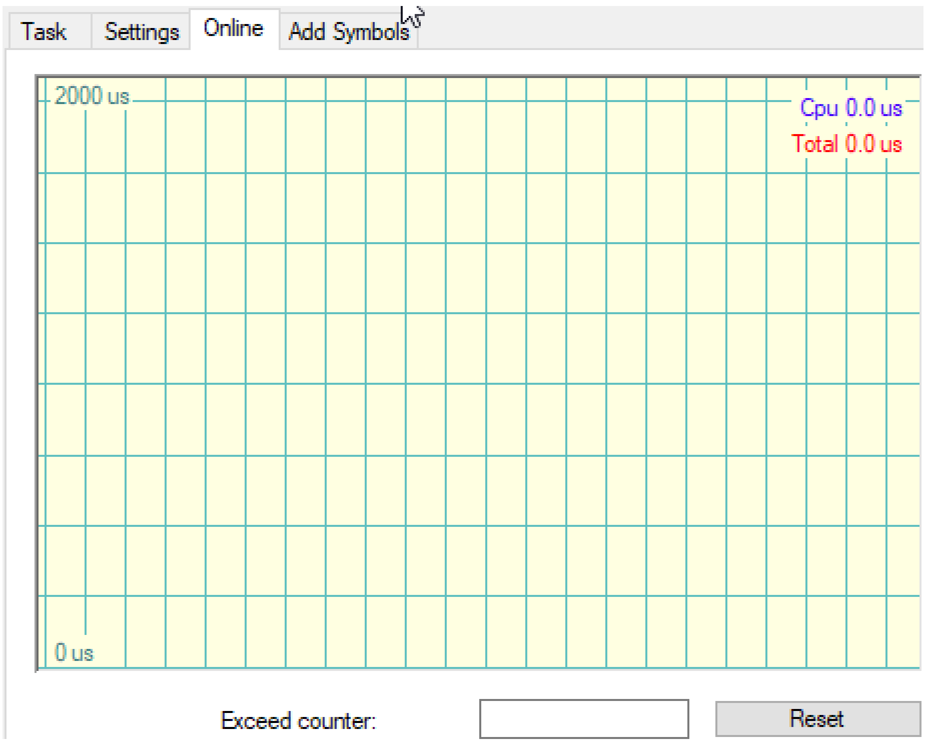
Symbol Names

The setting *Symbol Names, Language independent* specifies that the generic part of the symbol name is not changed. This is then always held in English.

For example, without this setting *Axes.Axis 1.SetPos* would change to *Achsen.Axis 1.SetPos* when switching language from English to German, while with the setting it would remain English.

Online tab

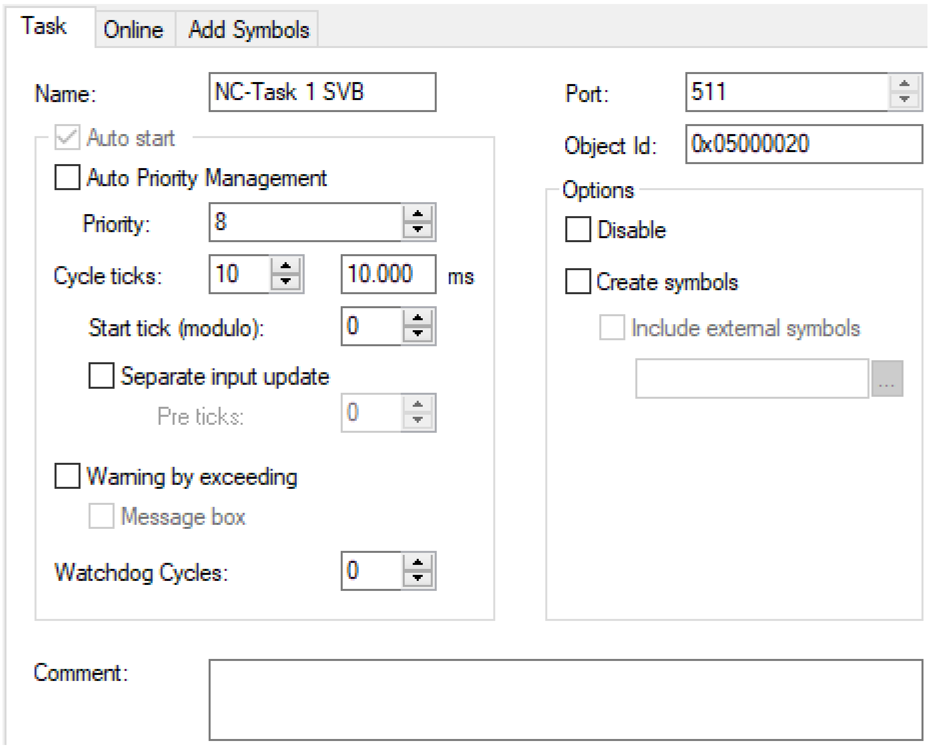
The Online tab shows the utilization of the task over time and indicates the number of cycle time overruns. Details can be found in the TE1000 | TwinCAT 3 XAE documentation.



4.1.2 SVB task

The block preparation task (SVB task) prepares selected commands so that they can subsequently be executed quickly in the SAF task [▶ 59].
These are for example:

- Operation of the axes in the development environment via the Axis online dialog [▶ 81].
- Homing sequence, which can be started e.g. with MC_Home.
- Group commands for kinematic, FIFO and NCI groups


Task | Online | Add Symbols

Name: NC-Task 1 SVB
Port: 511
Object Id: 0x05000020

[✓] Auto start
[ ] Auto Priority Management
Priority: 8
Cycle ticks: 10 10.000 ms
Start tick (modulo): 0
[ ] Separate input update
Pre ticks: 0
[ ] Warning by exceeding
[ ] Message box
Watchdog Cycles: 0

Options
[ ] Disable
[ ] Create symbols
[ ] Include external symbols

Comment:

Settings in relation to the SAF task

In relation to the SAF task, a higher cycle time and a lower priority should be selected for the SVB task, as is also the case by default.

4.1.3 Image

Under Image the process image of the SAF task is displayed.

General information on the process image of a task, see SAF task [▸ 59].

4.1.4 Tables

Electronic cam plates describe a nonlinear relationship between a master and a slave axis in a coupled axis system.

Valve characteristic curves are required for the control of hydraulic axes. They describe a non-linear relationship between the target velocity of an NC axis and the variable output to the controlled device.

Electronic cam plates and valve characteristic curves can be managed under Tables.

Further information:

- Motion Diagrams
  TE1510 | TwinCAT 3 CAM Design Tool
  TF5050 | TwinCAT 3 NC Camming
- Valve Diagrams
  TE1500 | TwinCAT 3 Valve Diagram Editor

4.1.5 Objects

TwinCAT 3 has a modular structure.

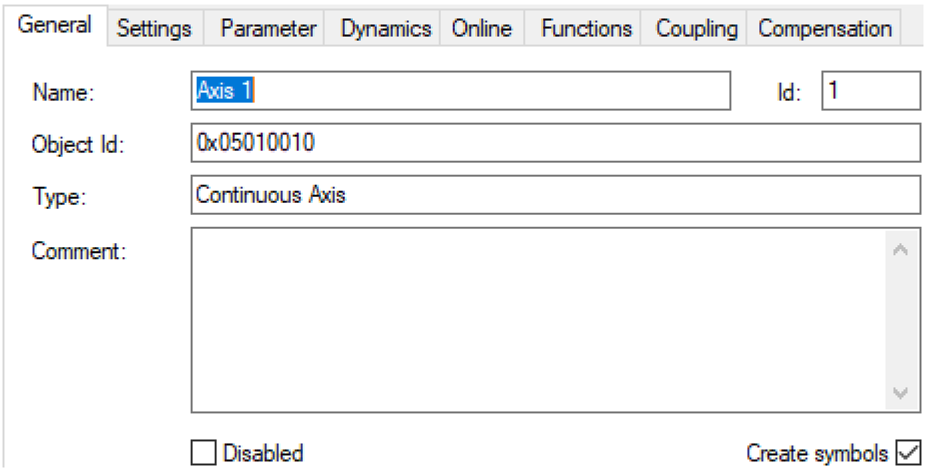
This modular structure is also used for Motion Control.

The individual motion objects can be channel-dependent, e.g. axes in the axis channel or kinematic transformations from the "NC Channel (for Kinematic Transformation)" or channel-independent.

The independent objects are managed via the Objects node. These are, for example, the Collision Avoidance and Coordinated Motion groups.

4.2 Axis dialog box

4.2.1 General

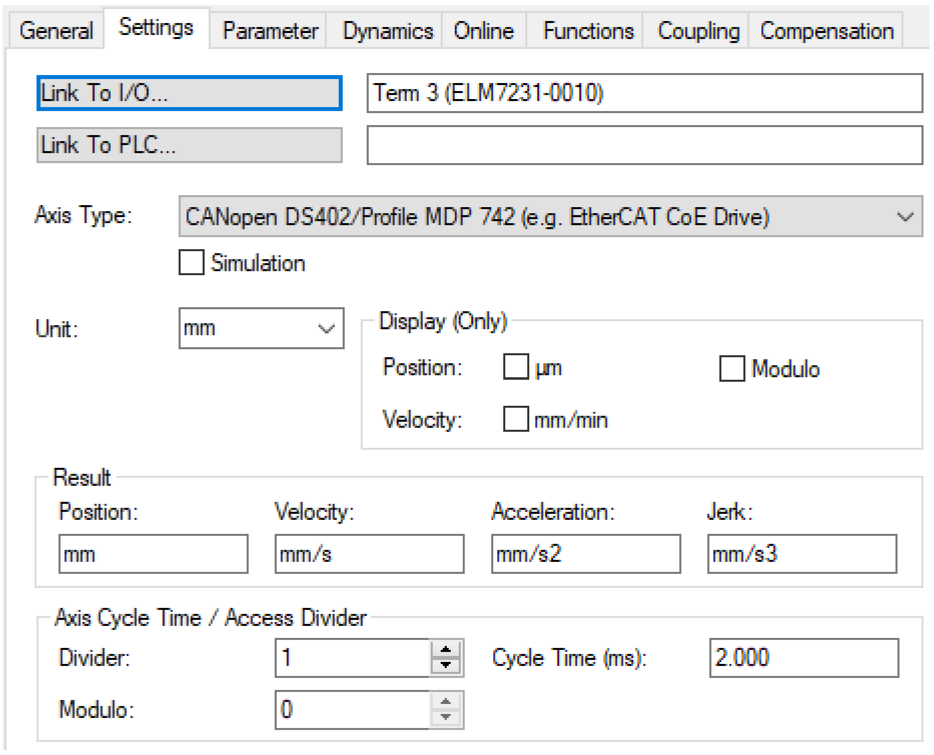


Name: Axis 1 Id: 1
Object Id: 0x05010010
Type: Continuous Axis
Comment:
☐ Disabled Create symbols ☑

Property	Description
Name	Selected name of the object. **Do not assign an object name more than once!**
Object Id	32-bit identification number that is automatically assigned and unique within a project.
Type	Axis type The axis type is defined when the axis is created and cannot be changed later.
Comment	Freely editable field for user notes.
Id	Identification number of the axis, which is assigned consecutively. When an axis is deleted, the number becomes free again and can be assigned to a new axis.
Disabled	Option to disable an axis
Create symbols	This option generates typical symbol names that are used in TwinCAT Scope, for example.

4.2.2 Settings

In the **Settings** tab, essential settings like linking the NC axis with hardware and PLC as well as type and unit can be executed.



Link To I/O...: Term 3 (ELM7231-0010)

Axis Type: CANopen DS402/Profile MDP 742 (e.g. EtherCAT CoE Drive)

☐ Simulation

Unit: mm

Display (Only)
Position: ☐ µm ☐ Modulo
Velocity: ☐ mm/min

Result

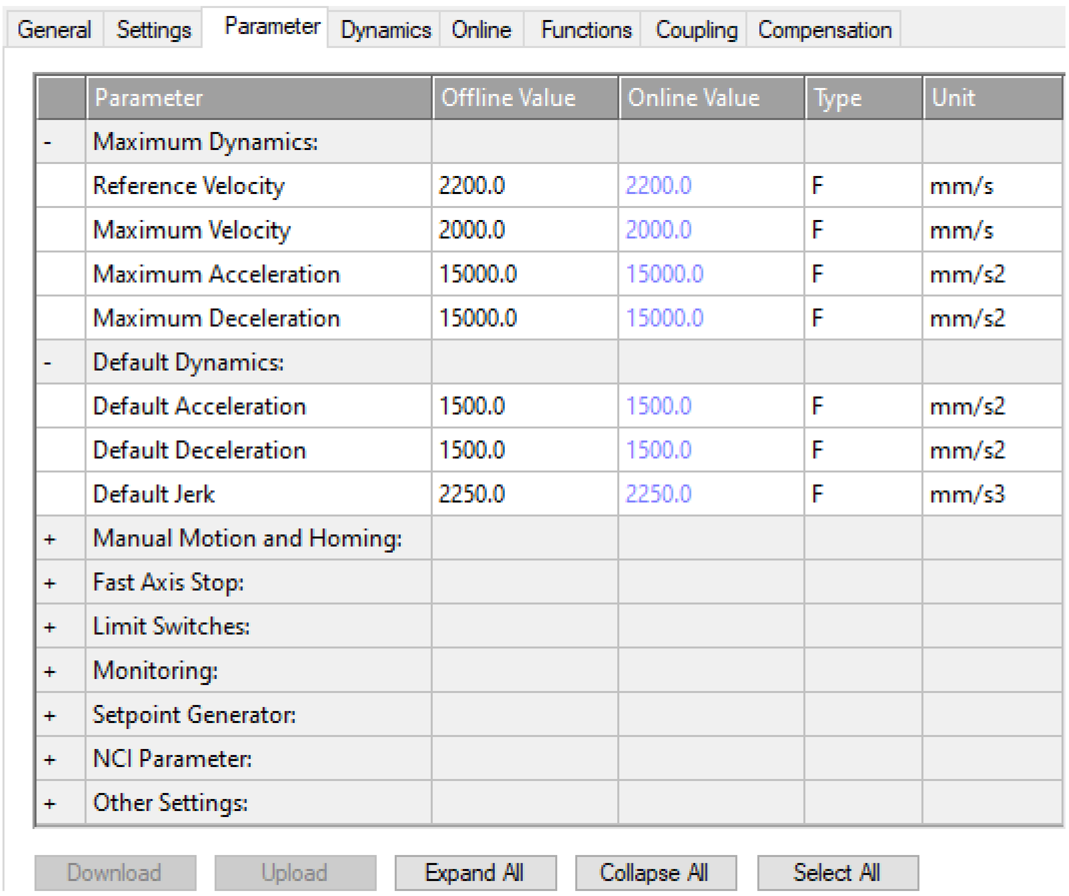
Position:	Velocity:	Acceleration:	Jerk:
mm	mm/s	mm/s2	mm/s3

Axis Cycle Time / Access Divider
Divider: 1 Cycle Time (ms): 2.000
Modulo: 0

Setting		Description
Link To I/O…		The Link button opens a dialog for linking the NC axis with the drive hardware under I/O. The link is displayed in the field on the right.
Link To PLC…		The Link button opens a dialog for linking the NC axis with the PLC instance of the axis. The link is displayed in the field on the right.
Axis Type		Type of connected drive hardware and protocol used.
Simulation		From TwinCAT 3.1 Build 4026 Enables the axis to be set to simulation mode. If simulation mode is active, the link to the I/O is ignored and a simulation drive and encoder are used instead of the configured axis type. If simulation mode is active, the axis is marked with a light blue symbol in the project tree .
Unit		Physical unit of the position of the axis. The unit can be chosen arbitrarily and edited in the Unit input field. It should be noted that the scaling factor of the axis must be set accordingly (see Encoder parameters). Standard: Millimeter (mm)
Display (only)		Adjustments to the display in the online axis dialog. These settings do not affect data in the process image.
	Position	Changes the position display by one thousand place value (mm/µm)
	Velocity	Display of velocity in mm/min instead of mm/s. The default time reference in NC and PLC remains the second regardless of the display setting.
	Modulo	Display of the modulo position instead of the absolute position.
Result		
	Position	Position
	Velocity	Velocity
	Acceleration	Acceleration
	Jerk	Jerk
Axis Cycle Time / Access Divider		
	Divider	The axis is executed in every nth cycle of the NC-SAF task with the cycle time divider. The divider can be set to a value greater than 1 to reduce the system load for low-priority axes.
	Modulo	With a cycle time divider greater than 1, the modulo value determines in which NC-SAF cycle the axis is processed. Axes with the same modulo are processed in the same cycle. For even distribution of the system load, the axes should be distributed to different modulo values. Example: Divider=4, Modulo 0..3
	Cycle Time (ms)	Axis cycle time

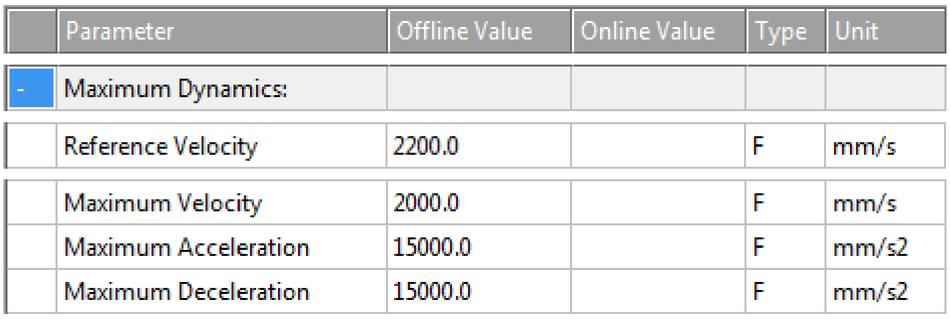
4.2.3 Parameter

Various axis settings can be made via the **Parameter** tab, which are described below.

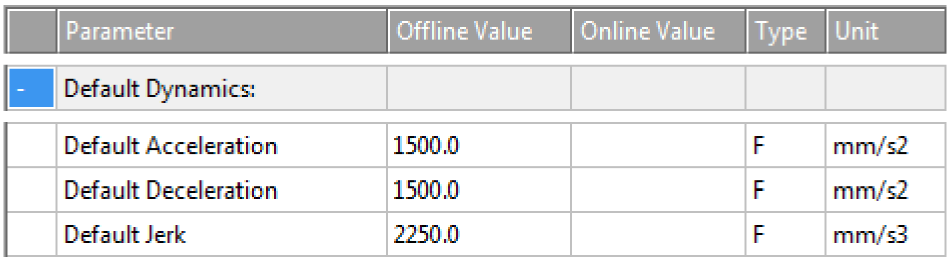


	Parameter	Offline Value	Online Value	Type	Unit
-	Maximum Dynamics:				
	Reference Velocity	2200.0	2200.0	F	mm/s
	Maximum Velocity	2000.0	2000.0	F	mm/s
	Maximum Acceleration	15000.0	15000.0	F	mm/s2
	Maximum Deceleration	15000.0	15000.0	F	mm/s2
-	Default Dynamics:				
	Default Acceleration	1500.0	1500.0	F	mm/s2
	Default Deceleration	1500.0	1500.0	F	mm/s2
	Default Jerk	2250.0	2250.0	F	mm/s3
+	Manual Motion and Homing:				
+	Fast Axis Stop:				
+	Limit Switches:				
+	Monitoring:				
+	Setpoint Generator:				
+	NCI Parameter:				
+	Other Settings:				

4.2.3.1 Maximum Dynamics, Default Dynamics



	Parameter	Offline Value	Online Value	Type	Unit
-	Maximum Dynamics:				
	Reference Velocity	2200.0		F	mm/s
	Maximum Velocity	2000.0		F	mm/s
	Maximum Acceleration	15000.0		F	mm/s2
	Maximum Deceleration	15000.0		F	mm/s2



	Parameter	Offline Value	Online Value	Type	Unit
-	Default Dynamics:				
	Default Acceleration	1500.0		F	mm/s2
	Default Deceleration	1500.0		F	mm/s2
	Default Jerk	2250.0		F	mm/s3

Dynamic-Parameters

- Velocity *Vel*,
- Acceleration *Acc*,
- Deceleration *Dec*,
- Jerk.

The *jerk* is the derivative of acceleration or deceleration with respect to time. Thus, it describes how quickly acceleration or deceleration change.

Reference Velocity

For drives that are not directly controlled by a digital velocity value, e.g. a voltage or current interface, the Reference Velocity is used to scale the drive output. The Reference Velocity is at the same time an upper velocity limit that cannot be exceeded in addition to the maximum velocity. For all drive types, the Reference Velocity must be set greater than or equal to the maximum velocity.

(Strictly speaking, the velocity upper limit is the Reference Velocity divided by the Output Ratio, if an Output Ratio smaller than 1.0 is parameterized)

For details, see the Drive Parameters > Reference Velocity.

"Maximum Dynamics" and "Default Dynamics"

The dynamic parameters are absolute, unsigned values. The default values are used if the user has not specified any values, e.g. for a motion command. The maximum values limit the axis dynamics and must be parameterized greater than or equal to the default dynamics.

The maximum values are observed by newer products like the Tc3_McCoordinatedMotion Library. However, for some products, such as the Tc2_MC2 library, the maximum acceleration and the maximum jerk are not taken into account.

Tc2_MC2 library

Default Dynamics	• If the input value "0.0" is assigned to one of the dynamic parameters "Acceleration, Deceleration, Jerk" at a motion function block or this input is left empty, then a default value is used instead.
Maximum Dynamics	• Velocity values that exceed the maximum velocity will not be accepted and will result in an error. • Values for Acceleration, Deceleration and Jerk are not checked for exceeding the maximum parameters, but are accepted.
Coupled axes	• In the case of a coupled slave axis, its dynamics depend exclusively on the master movement and maximum values are not checked. • When decoupling a slave axis, various measures are taken to prevent the maximum velocity from being exceeded or the direction of movement from being reversed. • Examples of such measures are increasing the jerk or increasing the acceleration or deceleration to the maximum value.

Tc3_McCoordinatedMotion Library, Tc3_McCollisionAvoidance Library

Tc3_Mc CoordinatedMotion Tc3_Mc CollisionAvoidance Default Values	• If for one of the dynamic-parameters “Acc, Dec, jerk” the input value “0.0” is assigned to a motion function block, this assignment leads to an error that means that this value is not allowed. • If for one of the dynamic-parameters “Acc, Dec, jerk” you would like to refer to a default value at a motion function block, this parameter has to be set to the constant value “*MC_Default*”.

Tc3_Mc CoordinatedMotion Tc3_Mc CollisionAvoidance Maximum Dynamics

Vel, Acc, Dec

- For the dynamic-parameters “Vel, Acc, Dec” the parameterized values are used.
- For the dynamic-parameters “Vel, Acc, Dec” maximum values can be parameterized at a motion function block using the constant value “*MC_Maximum*”.

Jerk

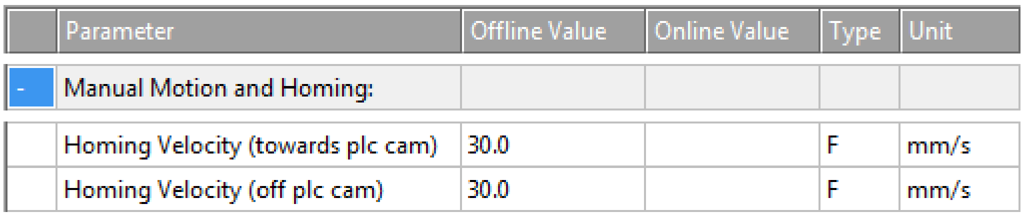
- There is no maximum value for the jerk.
- The jerk is set to the value “unlimited”. Simultaneously, a three-phase-profile or a three-phase-acceleration-setter is applied for motion.

Default Values

- It is allowed to parameterize default values that exceed their corresponding maximum values.
- If a default value is parameterized that exceeds ist corresponding maximum value, a warning will be given, but no error is thrown.
- At a Tc3_McCoordinatedMotion-function block or a Tc3_McCollisionAvoidance-function block parameterized default values using the constant value *MC_Default* will be mutually limited to the corresponding maximum values without giving an error message.

4.2.3.2 Manual Motion and Homing

Homing Velocity



	Parameter	Offline Value	Online Value	Type	Unit
-	Manual Motion and Homing:				
	Homing Velocity (towards plc cam)	30.0		F	mm/s
	Homing Velocity (off plc cam)	30.0		F	mm/s

bCalibrationCam

A boolean input of `MC_Home`. It evaluates the signal of a referencing cam. This reference signal can be coupled into the control unit via a digital input.

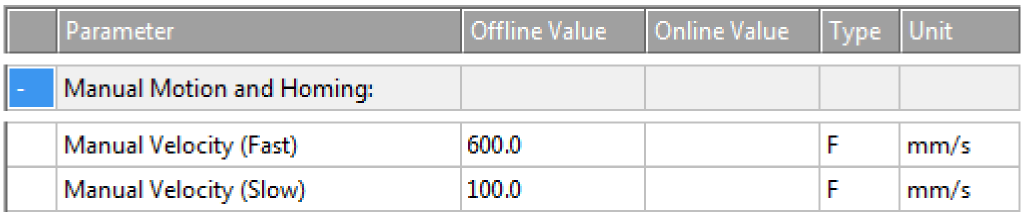
Homing Velocity (towards plc cam)

Velocity used by a `MC_Home` function block when homing to a referencing cam in the standard homing sequence when the `HomingMode MC_DefaultHoming` is selected and the `bCalibrationCam` input is evaluated.

Homing Velocity (off plc cam)

Velocity used by a function block `MC_Home` when moving away from a referencing cam in the standard homing sequence when the `HomingMode MC_DefaultHoming` is selected and the `bCalibrationCam` input is evaluated.

Manual Velocity



	Parameter	Offline Value	Online Value	Type	Unit
-	Manual Motion and Homing:				
	Manual Velocity (Fast)	600.0		F	mm/s
	Manual Velocity (Slow)	100.0		F	mm/s

Manual Velocity (Fast)

Online Dialog:

- Used velocity for MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online | -- F1.
- Used velocity for MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online | ++ F4.
- Similar for other identifiers

MC_Jog:

- Velocity used by a function block `MC_Jog` applied to the axis when its input is `JogForward` or its input is `JogBackwards` `TRUE` and selected as its `Mode` `MC_JOGMODE_STANDARD_FAST`.

Manual Velocity (Slow)

Online Dialog:

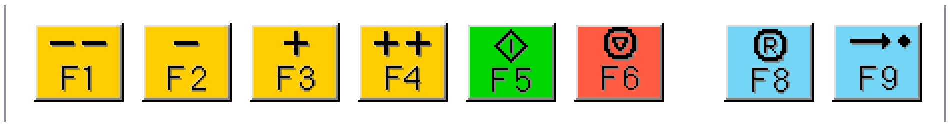
- Used velocity for MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online | - F2.
- Used velocity for MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online | + F3.
- Similar for other identifiers.

MC_Jog:

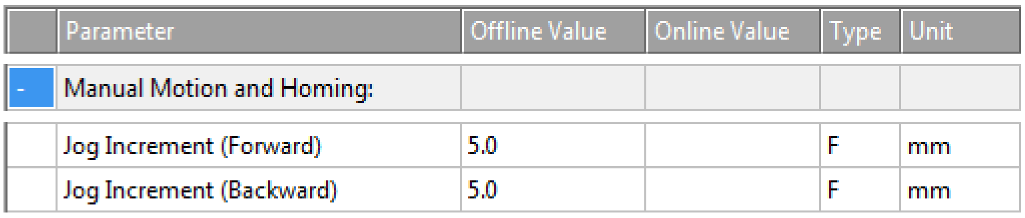
- Velocity used by a function block `MC_Jog` applied to the axis when its input is `JogForward` or its input is `JogBackwards` `TRUE` and selected as its `Mode` `MC_JOGMODE_STANDARD_SLOW`.

Buttons in the Online dialog

In the "MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online" dialog, there are the buttons -- F1, - F2, + F3 and ++ F4.



Jog Increment



	Parameter	Offline Value	Online Value	Type	Unit
-	Manual Motion and Homing:				
	Jog Increment (Forward)	5.0		F	mm
	Jog Increment (Backward)	5.0		F	mm

Jog Increment (Forward)

Not used.

This parameter is not currently used explicitly in TC3 Motion libraries. However, the parameter itself can be read or written or inserted indirectly by the user, e.g. in a function block created by the user or in an HMI.

Jog Increment (Backward)

Not used.

This parameter is not currently used explicitly in TC3 Motion libraries. However, the parameter itself can be read or written or inserted indirectly by the user, e.g. in a function block created by the user or in an HMI.

MC_JOGMODE_INCHING

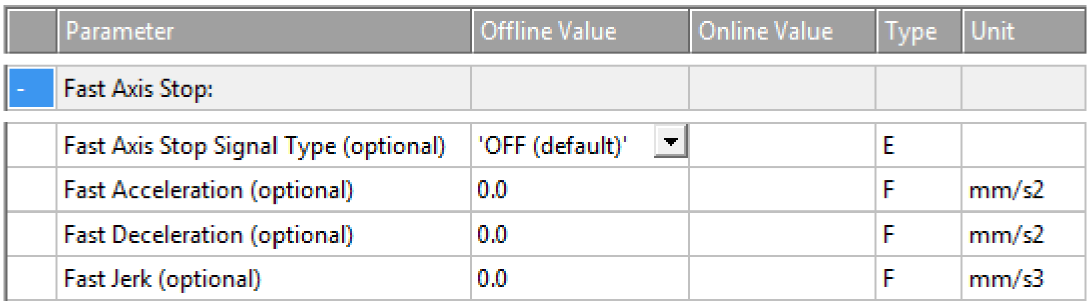
The function block `MC_Jog` enables an axis to be moved via manual keys. The key signal can be connected directly to the `JogForward` or `JogBackwards` input. The desired operation mode is specified by input `Mode`. When using the `MC_JOGMODE_INCHING` mode, a rising edge at one of the jog inputs moves the axis over a certain distance that is assigned at input `Position`.

More Information:

- MC_Jog (PLC library Tc2_MC2)

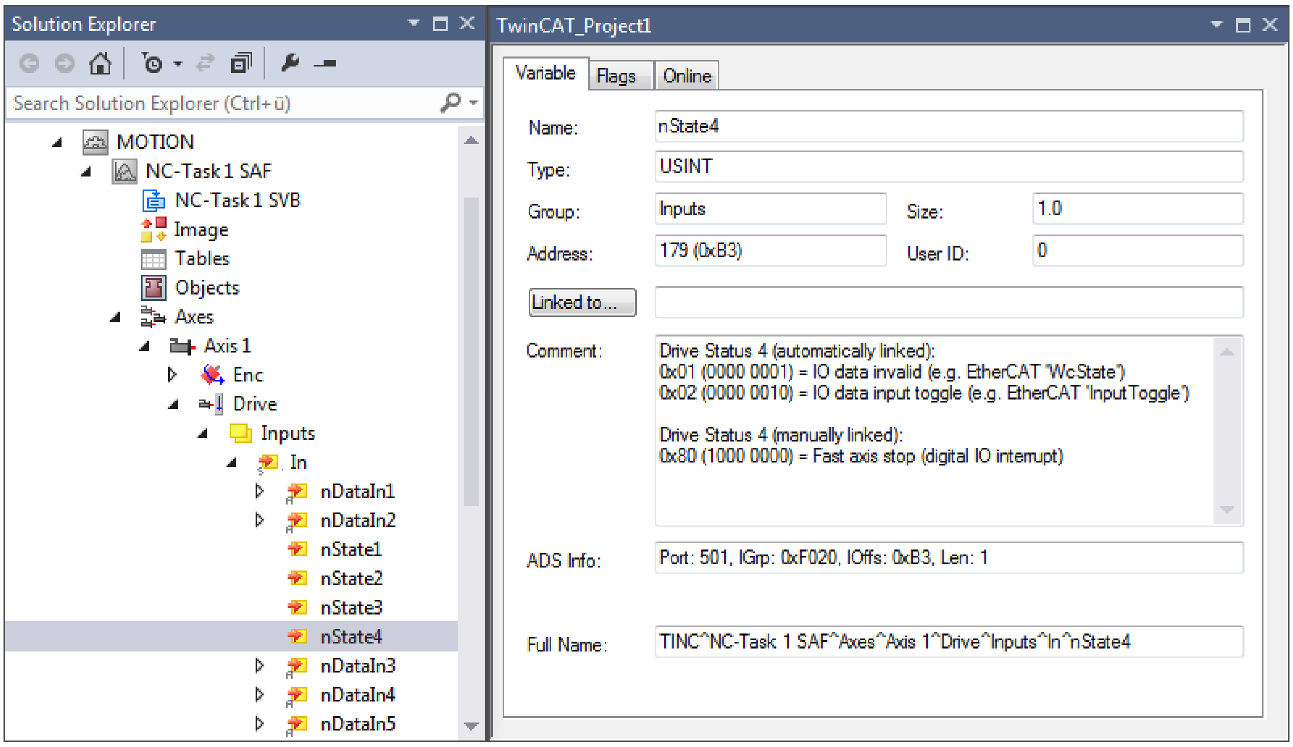
4.2.3.3 Fast Axis Stop

Fast Axis Stop



	Parameter	Offline Value	Online Value	Type	Unit
-	Fast Axis Stop:				
	Fast Axis Stop Signal Type (optional)	'OFF (default)'		E	
	Fast Acceleration (optional)	0.0		F	mm/s2
	Fast Deceleration (optional)	0.0		F	mm/s2
	Fast Jerk (optional)	0.0		F	mm/s3

A stop is usually triggered by PLC code using MC_Stop. However, there are special applications that require the time delay for the stop to be as small as possible. Here, bit 7 of Drive.Inputs.In.nState4 comes into play. This bit can trigger a stop directly without having to go through the PLC process image.



Name: nState4
Type: USINT
Group: Inputs Size: 1.0
Address: 179 (0xB3) User ID: 0
Comment: Drive Status 4 (automatically linked):
0x01 (0000 0001) = IO data invalid (e.g. EtherCAT 'WcState')
0x02 (0000 0010) = IO data input toggle (e.g. EtherCAT 'InputToggle')

Drive Status 4 (manually linked):
0x80 (1000 0000) = Fast axis stop (digital IO interrupt)
ADS Info: Port: 501, IGrp: 0xF020, IOffs: 0xB3, Len: 1
Full Name: TINC^NC-Task 1 SAF^Axes^Axis 1^Drive^Inputs^In^nState4

Drive Status 4 (manually linked):
Bit 7 = 0x80 (1000 0000) = Fast Axis Stop (digital IO interrupt)

nState4->Bit 7 variable

Bit 7 of Drive.Inputs.In.nState4 can be assigned to any event source.

Fast Axis Stop Signal Type

The "Fast Axis Stop Signal Type (optional)" list comprises six elements:

- OFF (default)
  No Fast Axis Stop is executed via the Drive.Inputs.In.nState4.7 bit.
- Rising Edge
  A Fast Axis Stop is executed on a rising edge of bit 7 of Drive.Inputs.In.nState4.
- Falling Edge
  A Fast Axis Stop is executed on a falling edge of bit 7 of Drive.Inputs.In.nState4.
- Both Edges
  A Fast Axis Stop is executed on a rising or falling edge of bit 7 of Drive.Inputs.In.nState4
- High Active
  A Fast Axis Stop is executed if bit 7 of Drive.Inputs.In.nState4 is set.

- Low Active
  A Fast Axis Stop is executed if bit 7 of Drive.Inputs.In.nState4 is not set

Fast Acceleration, Fast Deceleration, Fast Jerk

This parameterization is optional. If no values are specified, the default dynamics are applied.

Further Information:

- MC_Stop (PLC library Tc2_MC2)

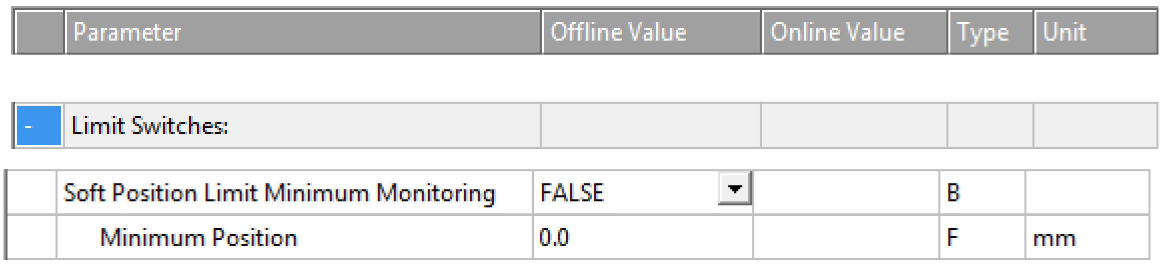
4.2.3.4 Limit Switches

The Limit Switches parameters can be set under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Parameter.

Alternatively, the Limit Switches parameters can be set under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Enc | Parameter.

Similar for other identifiers.

Soft Position Limit Minimum Monitoring



	Parameter	Offline Value	Online Value	Type	Unit
-	Limit Switches:				
	Soft Position Limit Minimum Monitoring	FALSE		B	
	Minimum Position	0.0		F	mm

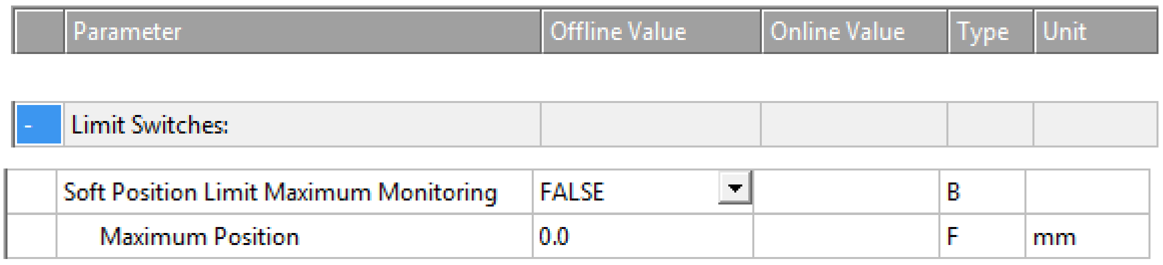
`FALSE`: Soft Position Limit Minimum Monitoring is not enabled.

`TRUE`: Soft Position Limit Minimum Monitoring is enabled.

Minimum Position

Position lower limit for the axis, which must not be violated when Soft Position Limit Minimum Monitoring is enabled. Commands that violate this lower limit will be rejected.

Soft Position Limit Maximum Monitoring



	Parameter	Offline Value	Online Value	Type	Unit
-	Limit Switches:				
	Soft Position Limit Maximum Monitoring	FALSE		B	
	Maximum Position	0.0		F	mm

`FALSE`: Soft Position Limit Maximum Monitoring is not enabled.

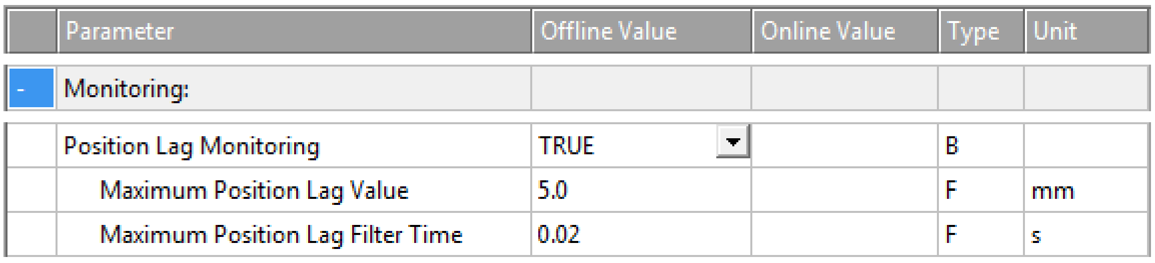
`TRUE`: Soft Position Limit Maximum Monitoring is enabled.

Maximum Position

Position upper limit for the axis, which must not be violated when Soft Position Limit Maximum Monitoring is enabled. Commands that violate this upper limit will be rejected.

4.2.3.5 Monitoring

Position Lag Monitoring



	Parameter	Offline Value	Online Value	Type	Unit
-	Monitoring:				
	Position Lag Monitoring	TRUE		B	
	Maximum Position Lag Value	5.0		F	mm
	Maximum Position Lag Filter Time	0.02		F	s

The position lag monitoring monitors the position lag value. If the parameterized limits for position and time are exceeded, a runtime error is output.

Position lag value = current set position - actual position

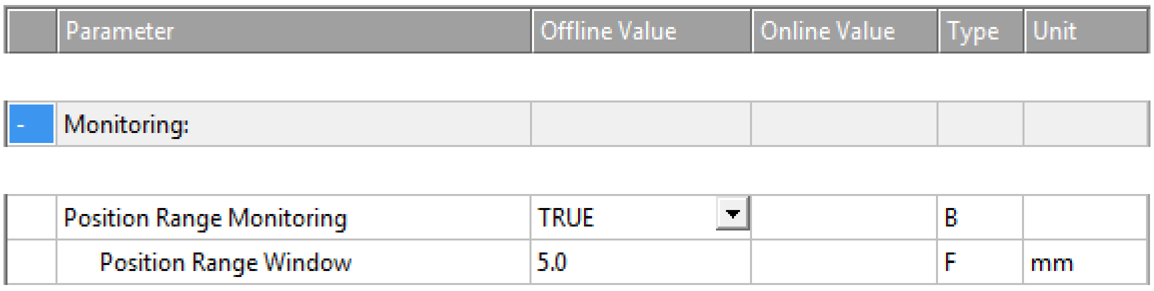
`TRUE`: Position Lag Monitoring is enabled.

`FALSE`: Position Lag Monitoring is not enabled.

Maximum Position Lag Value and Maximum Position Lag Filter Time

The Maximum Position Lag Value is the upper limit, which must not be exceeded for longer than the Maximum Position Lag Filter Time. Otherwise the NC axis is stopped immediately by direct shutdown and set to the logical state "Error", whereby the error `0x4550` is output.

Position Range Monitoring



	Parameter	Offline Value	Online Value	Type	Unit
-	Monitoring:				
	Position Range Monitoring	TRUE		B	
	Position Range Window	5.0		F	mm

Position Range Monitoring monitors whether the actual position of the NC axis reaches a window around the target position. Once the window is reached, the status flag `Axis.Status.InPositionArea` is set to `TRUE`.

`TRUE`: Position Range Monitoring is enabled.

`FALSE`: Position Range Monitoring is not enabled.

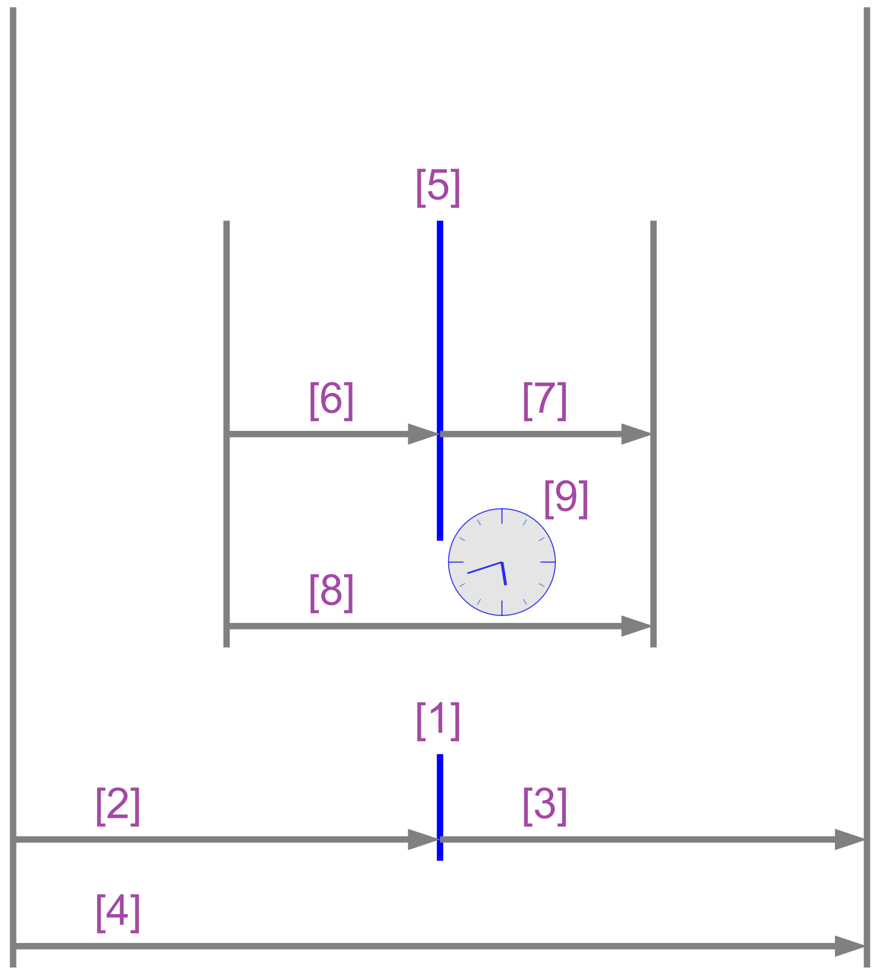
Position Range Window

Specifies the tolerance of the actual position of the NC axis with respect to the target position so that the status flag `Axis.Status.InPositionArea` is set to `TRUE`.

NC-Online: “In Pos. Range” – `Axis.Status.InPositionArea`

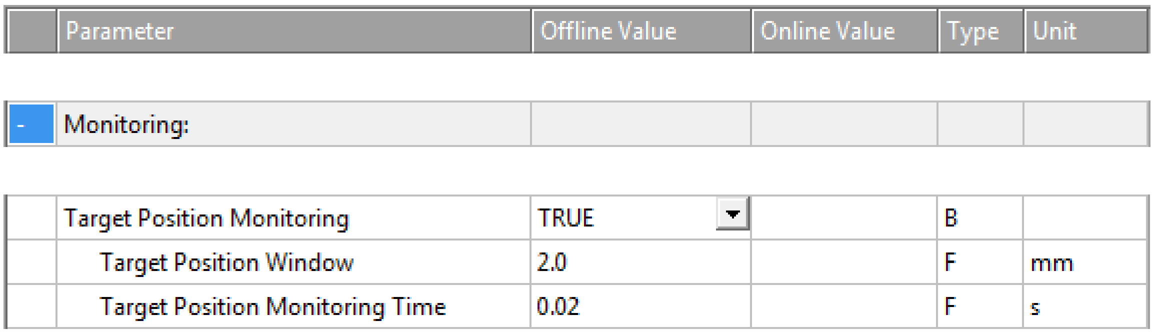
The value of variable `Axis.Status.InPositionArea` corresponds to the state of the checkbox “In Pos. Range” within the group box “Status (phys.)” of the NC-Online dialog. If the variable `Axis.Status.InPositionArea` is set on `TRUE`, the checkbox “In Pos. Range” is checked.

Graphic example



[1]	• Nominal value of the target position.
[2]	• Position Range Window.
[3]	• Position Range Window.
[4]	Variable `Axis.Status.InPositionArea`:
Position Range Monitoring	• If the parameter "Position Range Monitoring" is set to `TRUE` and ... • ... if the actual position is in this range [4], • then the variable `Axis.Status.InPositionArea` is set to `TRUE`.

Target Position Monitoring



	Parameter	Offline Value	Online Value	Type	Unit
-	Monitoring:				
	Target Position Monitoring	TRUE		B	
	Target Position Window	2.0		F	mm
	Target Position Monitoring Time	0.02		F	s

Target Position Monitoring monitors whether the actual position of the NC axis reaches a window around the target position and also remains in this window for a minimum time. After that the status `Flag Axis.Status.InTargetPosition` is set to `TRUE`.

`TRUE`: Target Position Monitoring is enabled.

`FALSE`: Target Position Monitoring is not enabled.

Target Position Window

The Target Position Window specifies the tolerance of the actual position of the NC axis in relation to the target position, which is to be taken into account in Target Position Monitoring.

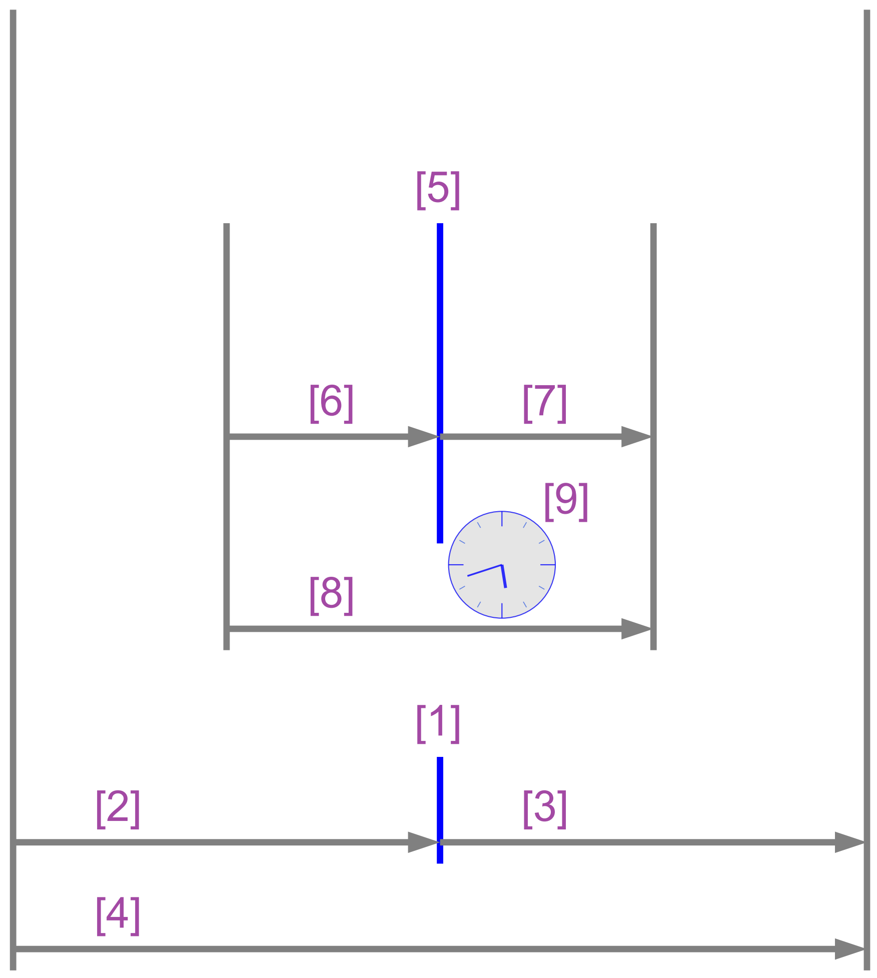
Target Position Monitoring Time

The Target Position Monitoring Time specifies the time in which the actual position of the NC axis must be at least within the tolerance range of the target position (Target Position Window) so that the status flag `Axis.Status.InTargetPosition` is set to `TRUE`.

NC-Online: "In Target Pos." – `Axis.Status.InTargetPosition`

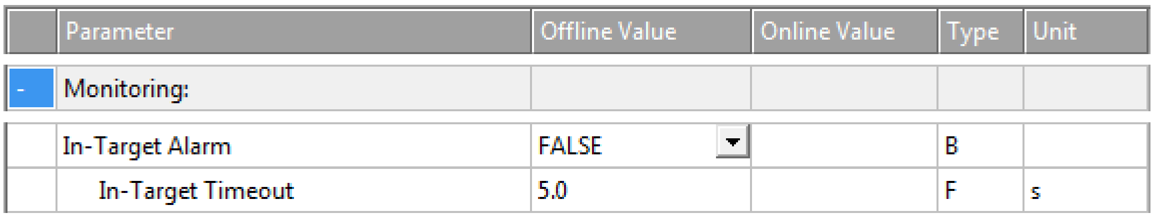
The value of the variable `Axis.Status.InTargetPosition` corresponds to the state of the checkbox "In Target Pos." within the group box "Status (phys.)" of the NC-Online dialog. If the variable `Axis.Status.InTargetPosition` is set on `TRUE`, the checkbox "In Target Pos." is checked.

Graphic example



[5]	• Nominal value of the target position.
[6]	• Target Position Window.
[7]	• Target Position Window.
[8], [9]	Target position:
Target Position Monitoring	• If the parameter "Target Position Monitoring" is set to `TRUE` and ... • ... if the actual position is in this range [8] at least for the duration"Target Position Monitoring Time" [9] without interruption up to the actual time, • then the variable `Axis.Status.InTargetPosition` is set to `TRUE`.

In-Target Alarm



	Parameter	Offline Value	Online Value	Type	Unit
-	Monitoring:				
	In-Target Alarm	FALSE		B	
	In-Target Timeout	5.0		F	s

The In-Target Alarm monitors whether the axis reaches the Target Position Window within the In-Target Timeout.

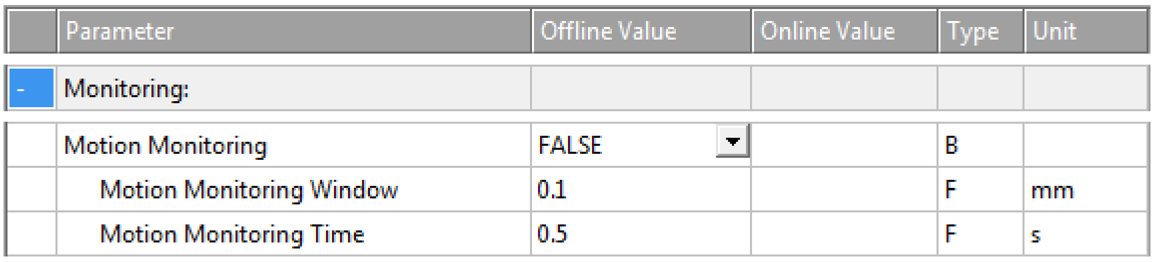
`TRUE`: The In-Target alarm is enabled.

`FALSE`: The In-Target alarm is not enabled.

In-Target Timeout

If the NC axis does not reach the Target Position Window within the In-Target Timeout, the Nc axis reports the error `0x435C`. The time measurement is started when the set position of the axis has reached its nominal position.

Motion Monitoring



	Parameter	Offline Value	Online Value	Type	Unit
-	Monitoring:				
	Motion Monitoring	FALSE		B	
	Motion Monitoring Window	0.1		F	mm
	Motion Monitoring Time	0.5		F	s

Motion Monitoring checks whether an axis is actually moving while it is executing a motion command. This makes it possible, for example, to detect the mechanical blocking of an axis at an early stage.

`TRUE`: Motion Monitoring is enabled.

`FALSE`: Motion Monitoring is not enabled.

Motion Monitoring Window

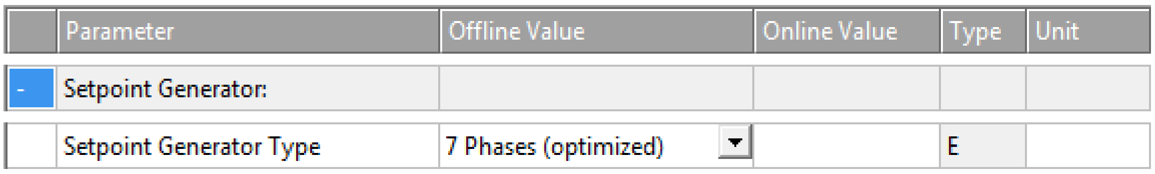
The Motion Monitoring Window defines the distance that the encoder (actual position) should be expected to travel during one cycle of the NC SAF task. Here a value/distance/length of some encoder increments can be set.

Motion Monitoring Time

Monitoring starts as soon as the axis executes a motion command and ends when the axis comes to a logical standstill. If its actual position does not change by more than the Motion Monitoring Window in at least one NC cycle during the Motion Monitoring Time, the NC axis outputs the error `0x435D`.

4.2.3.6 Setpoint Generator

Setpoint Generator Type

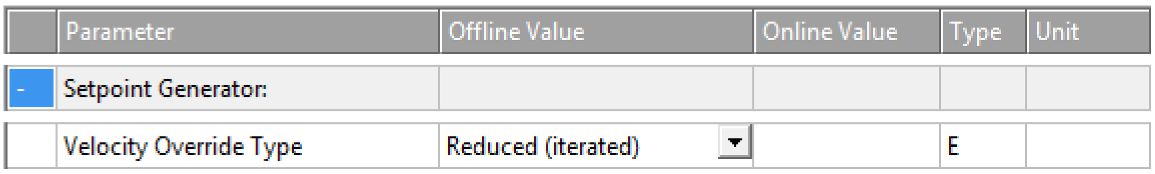


	Parameter	Offline Value	Online Value	Type	Unit
-	Setpoint Generator:				
	Setpoint Generator Type	7 Phases (optimized)		E	

`7 Phase (optimized)`

Only an optimized 7-phase setpoint generator is supported.

Velocity Override Type



	Parameter	Offline Value	Online Value	Type	Unit
-	Setpoint Generator:				
	Velocity Override Type	Reduced (iterated)		E	

The NC axis supports a speed override. This means that changing the override creates a new velocity, but does not affect the ramps (acceleration or jerk). The used override types only differ in terms of reference velocity.

More information about override types can also be found in Path Override (Interpreter Override Types).

`Reduced (iterated)`

The override refers to the maximum velocity of the profile calculated by the setpoint generator.

Example: A motion command with 1000 mm/s with a short travel distance is assigned. This velocity cannot be reached on this route and a travel profile with 700 mm/s at 100% override is calculated. With a smaller override value, the actual travel velocity is further reduced.

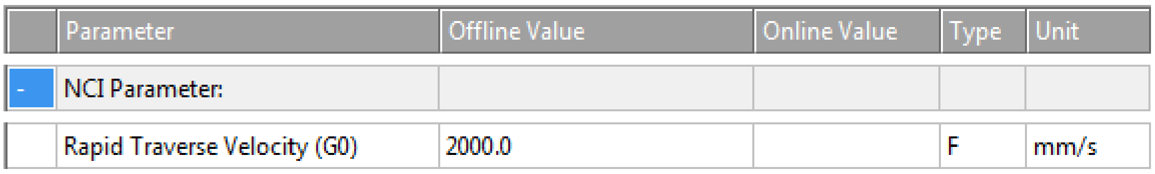
`Original (iterated)`

The override refers to the parameterized velocity of the executed motion command.

Example: A motion command with 1000 mm/s with a short travel distance is assigned. This velocity cannot be reached on this route and a travel profile with 700 mm/s at 100% override is calculated. Since the override refers to the velocity of the motion command, the actual velocity is only reduced here below an override value of 70 %.

4.2.3.7 NCI Parameters

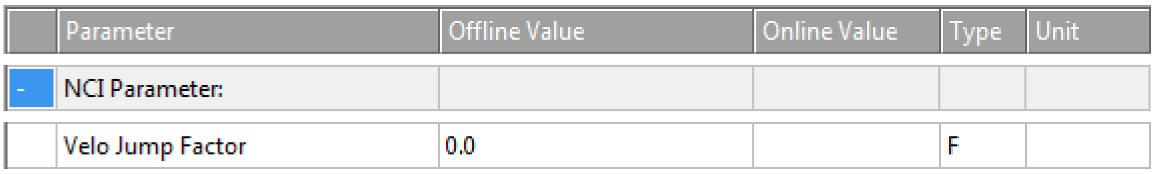
Rapid Traverse Velocity (`G0`)



	Parameter	Offline Value	Online Value	Type	Unit
-	NCI Parameter:				
	Rapid Traverse Velocity (G0)	2000.0		F	mm/s

The Rapid Traverse Velocity is used when an interpreter command `G0` is executed. See section Rapid traverse velocity for a brief description of the interpreter command `G0`.

Velo Jump Factor



	Parameter	Offline Value	Online Value	Type	Unit
-	NCI Parameter:				
	Velo Jump Factor	0.0		F	

The reduction factor `C0[i]` is the Velo Jump Factor.

Background information

Segment transitions

Segments are geometric objects. We consider them as curves in the sense of differential geometry, parameterized by means of their length `arc`.

A segment transition from a segment `S_in` to a segment `S_out` is called of geometric type `Ck`, where `k` is a natural number (including `0`) describing `k` continuous `arc` length differentials for each segment and the corresponding `k^th` derivatives at the transition point.

`C0` transitions: Have a knee-point at the transition point.

`C1` transitions: Appear smooth, but are not smooth in dynamic terms. At the segment transition point, there is a jump in acceleration.

`C2` transitions: Are dynamically smooth and their smoothness is only jerk-limited.

`Ck` transitions: are dynamically smooth.

Segment dynamics

Velocity `v`: The segment set velocity `v` changes at the segment transition from `v_in` to `v_out`. At the segment transition the set velocity is always reduced to the lower of the two values.

Acceleration `a`: At the segment transition, the current path acceleration is always reduced to zero.

Jerk `j`: At the segment transition, the jerk changes according to the geometry of the segment transition. This jerk change can cause a noticeable dynamic jump.

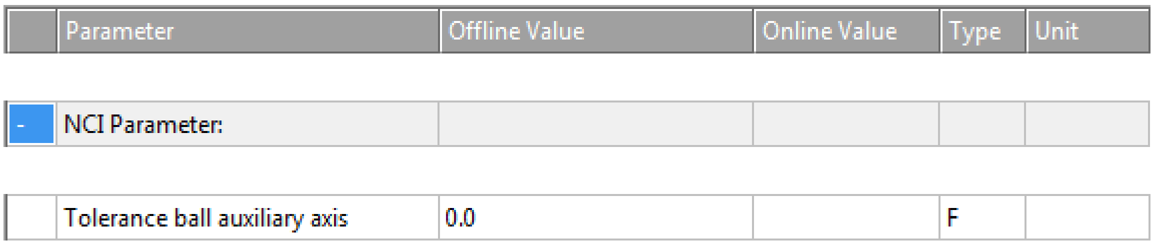
Velocity reduction modes for C0 transitions

Several reduction methods are available for `C0` transitions. One of them is the reduction method VELOJUMP. The reduction method VELOJUMP reduces the velocity after permitted velocity jumps for each axis.

The VELOJUMP reduction method for C0 transitions

Basically `v_link = min(v_in, v_out)` applies. For the axis `[i]`, the allowed absolute velocity jump is `v_jump[i] = C0[i] * min(A+[i], -A-[i]) * T`, where `C0[i]` is the reduction factor, `A+[i], A-[i]` is the acceleration or deceleration limits for the axis `[i]`, and `T` is the cycle time. The VELOJUMP reduction method ensures that the path velocity is reduced at the segment transition `v_link` until the absolute step change in the set axis velocity of axis `[i]` is at most `v_jump[i]`. However, `v_min` has priority: if `v_link` is smaller than `v_min`, then `v_link` is set to `v_min`. In the case of motion reversal with no programmed stop, there will be a step change in axis velocity.

Tolerance ball auxiliary axis

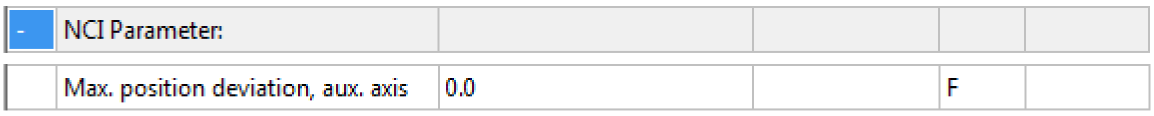


	Parameter	Offline Value	Online Value	Type	Unit
-	NCI Parameter:				
	Tolerance ball auxiliary axis	0.0		F	

See section Tolerance Ball for further information.

Max. position deviation, aux. axis

	Parameter	Offline Value	Online Value	Type	Unit

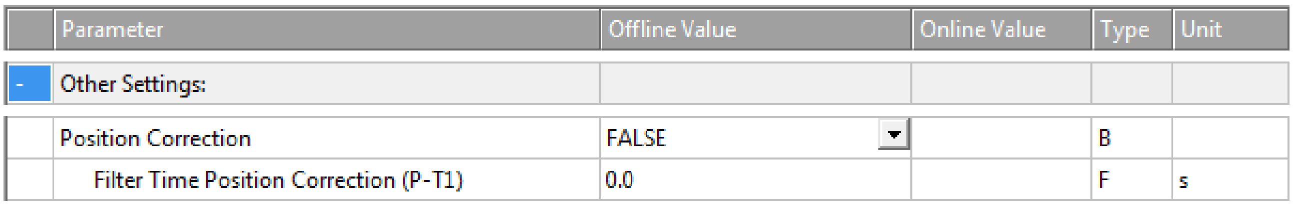


-	NCI Parameter:				
	Max. position deviation, aux. axis	0.0		F	

Introduced for future extensions.

4.2.3.8 Other Settings

Position Correction



	Parameter	Offline Value	Online Value	Type	Unit
-	Other Settings:				
	Position Correction	FALSE		B	
	Filter Time Position Correction (P-T1)	0.0		F	s

The Position Correction can be activated under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Parameter.

Alternatively, the Position Correction can be activated under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Enc | Parameter.

Similar for other identifiers.

`FALSE`: The Position Correction is disabled.

`TRUE`: The Position Correction is enabled.

The variable `axis.PlcToNc.PositionCorrection` is of data type `LREAL` and belongs to the structure `PLCTONC_AXIS_REF`. If Position Correction is enabled, this variable adds an additional offset to the target position. It should be noted that this correction does not affect the software end positions.

Filter Time Position Correction (P-T1)

The filter time for the PT-1 filter, which filters fluctuations within the Actual Position Correction with the filter time set here. See section PT1 Filter for further information on the PT1 filter.

See also:

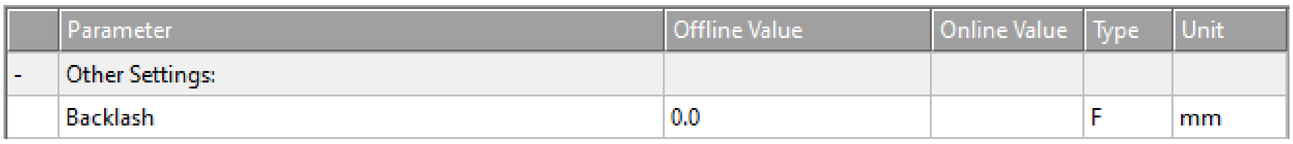
`MC_PositionCorrectionLimiter`

- TwinCAT 3 PLC Lib: Tc2_MC2

The function block `MC_PositionCorrectionLimiter` adds the correction value `PositionCorrectionValue` to the actual position value of the axis. Depending on the `CorrectionMode` the position correction value is either written directly or filtered.

To use the `MC_PositionCorrectionLimiter` function block successfully the Position Correction has to be enabled by setting the parameter Position Correction `TRUE`.

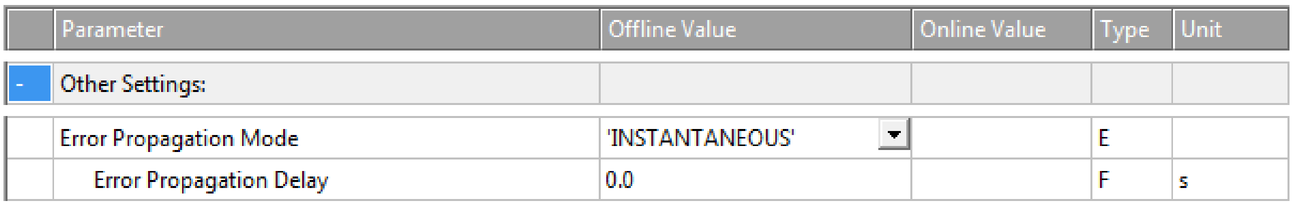
Backlash



	Parameter	Offline Value	Online Value	Type	Unit
-	Other Settings:				
	Backlash	0.0		F	mm

This parameter is only present for compatibility reasons. For more information, visit NC Backlash Compensation.

Error Propagation Mode



	Parameter	Offline Value	Online Value	Type	Unit
-	Other Settings:				
	Error Propagation Mode	'INSTANTANEOUS'		E	
	Error Propagation Delay	0.0		F	s

For the slave axis the error transmission can be delayed.

`'INSTANTANEOUS'`: The error transmission is not delayed.

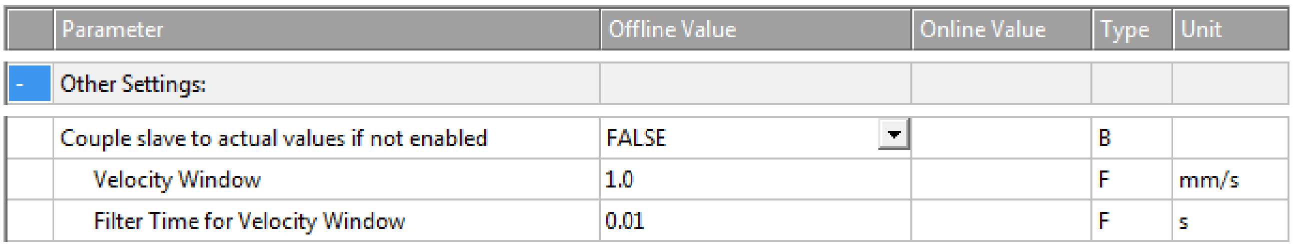
`'DELAYED'`: The error transmission is delayed by the Error Propagation Delay.

Error Propagation Delay

The delay time by which error propagation for the slave axis is delayed when Error Propagation Mode `'DELAYED'` is selected.

If an error occurs at a slave axis during runtime, the corresponding master axis is not set to error state until the time assigned here has elapsed. A state of interest of the slave axis, especially its error state, can be observed by PLC code. In this way, the faulty slave axis can be safely decoupled to safely prevent the entire axis combination from entering the error state.

Couple slave to actual values if not enabled



	Parameter	Offline Value	Online Value	Type	Unit
-	Other Settings:				
	Couple slave to actual values if not enabled	FALSE		B	
	Velocity Window	1.0		F	mm/s
	Filter Time for Velocity Window	0.01		F	s

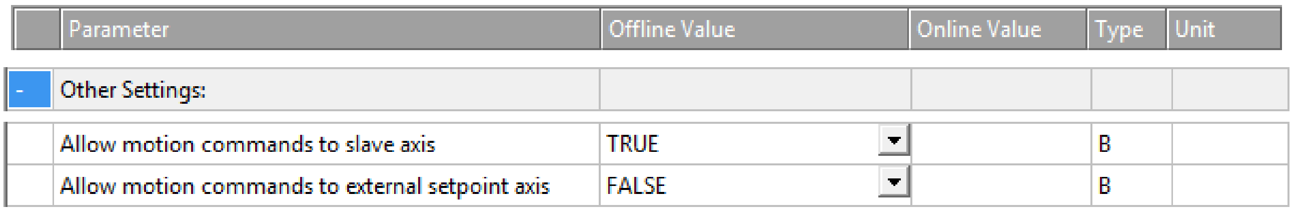
`FALSE`: Not coupled.

`TRUE`: Coupled. The slave axis follows the master actual position while and also when the master is disabled.

Velocity Window und Filter Time for Velocity Window

The coupled slave axis follows the master axis within the `Velocity Window`. If velocity deviations exceed `Filter Time for Velocity Window` beyond `Velocity Window`, an error is output.

Allow motion commands



	Parameter	Offline Value	Online Value	Type	Unit
-	Other Settings:				
	Allow motion commands to slave axis	TRUE		B	
	Allow motion commands to external setpoint axis	FALSE		B	

Allow motion commands to slave axis

In general terms, an axis is in PTP mode all the time. This is about indirectly converting a slave axis into a master axis. Thus it is implicitly decoupled without the need to call on `MC_GearOut` from the PLC code.

`TRUE`: A PTP command can be triggered to the slave axis without first setting the axis to PTP mode.

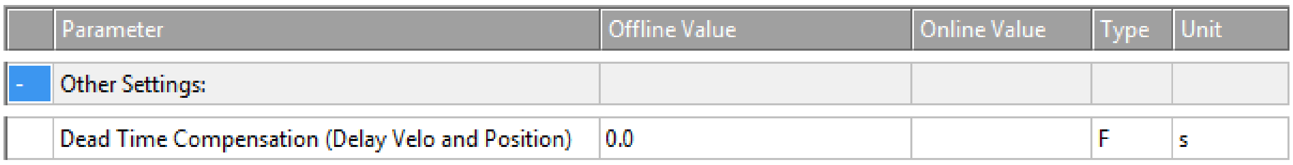
`FALSE`: Before a PTP command can be triggered to the slave axis, the slave axis must be set to PTP mode.

Allow motion commands to external setpoint axis

FALSE: Before a PTP command can be triggered to the external setpoint axis, the external setpoint axis must be set to PTP mode.

TRUE: A PTP command can be triggered to the external setpoint axis without first setting the axis to PTP mode.

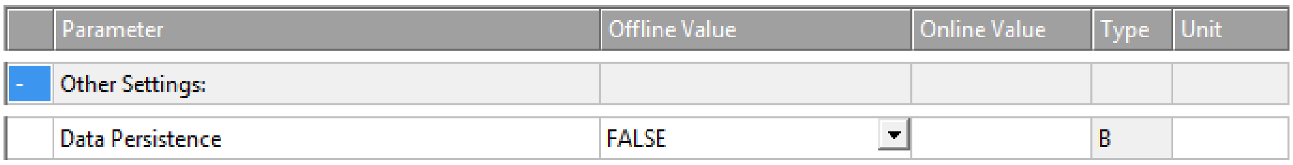
Dead Time Compensation (Delay Velo and Position)



	Parameter	Offline Value	Online Value	Type	Unit
-	Other Settings:				
	Dead Time Compensation (Delay Velo and Position)	0.0		F	s

This parameter is only present for compatibility reasons. Do not use it on new projects.

Data Persistence



	Parameter	Offline Value	Online Value	Type	Unit
-	Other Settings:				
	Data Persistence	FALSE		B	

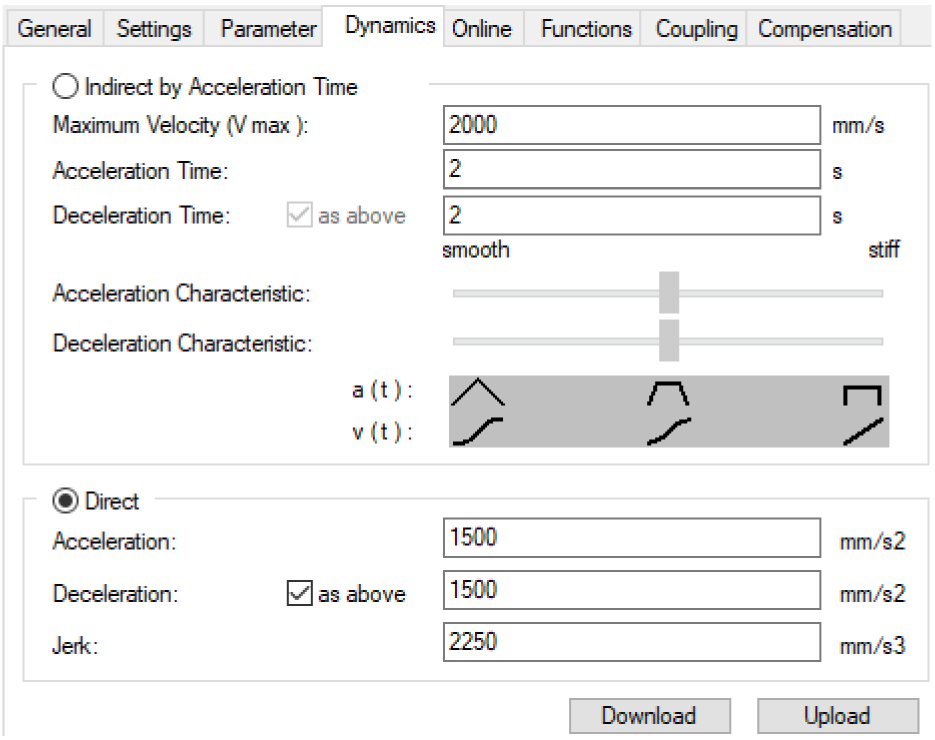
Data Persistence is used for special encoder problems.

FALSE: Data Persistence is not enabled.

TRUE: Data Persistence is enabled.

4.2.4 Dynamics

The axis dynamics can be configured via the **Parameter** tab as well as via the **Dynamics** tab.



( ) Indirect by Acceleration Time

Maximum Velocity (V max):	2000	mm/s
Acceleration Time:	2	s
Deceleration Time: ☑ as above	2	s
Acceleration Characteristic:		
Deceleration Characteristic:		

(●) Direct

Acceleration:	1500	mm/s2
Deceleration: ☑ as above	1500	mm/s2
Jerk:	2250	mm/s3

In this dialog the default axis dynamics is set. The default values are used if no explicit dynamics data are given to a travel command.

The dynamics data Acceleration, Deceleration and Jerk can be entered here either directly or they are determined indirectly via an acceleration time to the maximum velocity. In addition to the acceleration time, the slider is used to determine the ratio of jerk and acceleration, thus selecting a rather stiff or smooth setting.

4.2.5 Online

The **Online** tab is the main dialog for online axis operation. Here the enable can be set and after this the corresponding axis can be moved manually. In addition, the most important axis states are displayed when the configuration is active.

⚠ DANGER

Risk of injury due to movement of axes!

The commissioning results in a movement of axes.

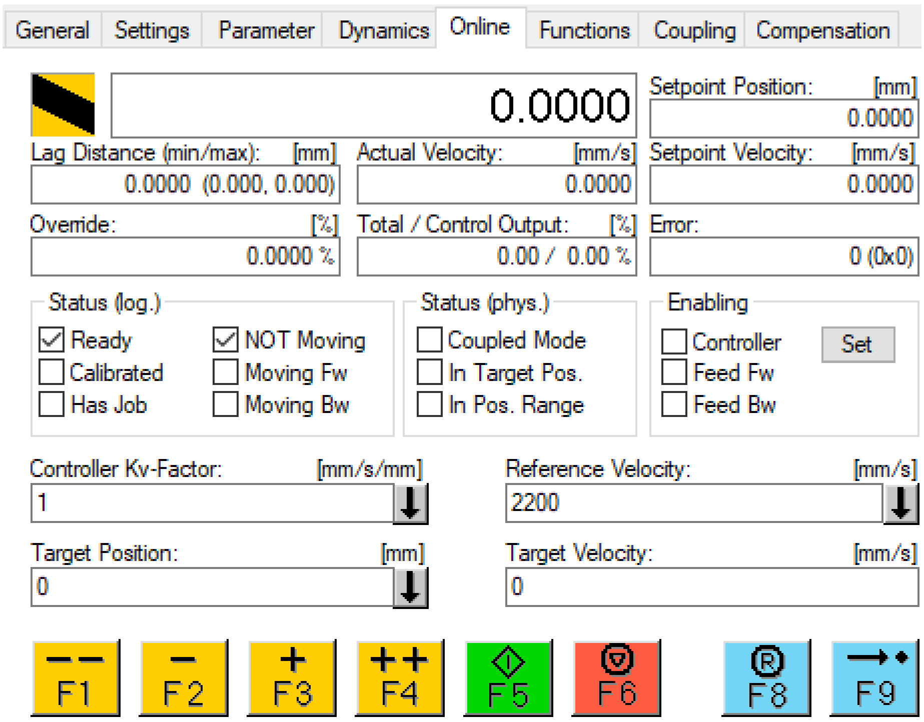
- Make sure that neither you nor others are harmed by the movement, e.g. by maintaining a suitable safety distance.
- Do not perform any action whose consequences you cannot estimate

⚠ WARNING

Incorrect axis position during initial commissioning

Without referencing / calibrating the axis position, the displayed axis position may deviate from the actual axis position.

- Perform a homing to determine the correct actual position using a reference signal.



0.0000

Setpoint Position: [mm] 0.0000

Lag Distance (min/max): [mm] 0.0000 (0.000, 0.000)

Actual Velocity: [mm/s] 0.0000

Setpoint Velocity: [mm/s] 0.0000

Override: [%] 0.0000 %

Total / Control Output: [%] 0.00 / 0.00 %

Error: 0 (0x0)

Status (log.): [✓] Ready; [ ] Calibrated; [ ] Has Job; [✓] NOT Moving; [ ] Moving Fw; [ ] Moving Bw

Status (phys.): [ ] Coupled Mode; [ ] In Target Pos.; [ ] In Pos. Range

Enabling: [ ] Controller; [ ] Feed Fw; [ ] Feed Bw

Controller Kv-Factor: [mm/s/mm] 1

Reference Velocity: [mm/s] 2200

Target Position: [mm] 0

Target Velocity: [mm/s] 0

The dialog is divided into the display of the most important axis states, the setting of the axis enable, input fields as well as function keys for controller settings and travel commands. The units of the values depend on the set "Base" unit.

Display

The actual position of the axis is determined from the feedback of the encoder system and displayed in the large unlabeled field.

Setpoint Position	The target position is specified by the NC during a movement.
Lag Distance	The lag error is the difference between the nominal and actual position.
Actual Velocity	The actual velocity of the axis is determined by deriving the actual position.
Setpoint Velocity	The target velocity is calculated by the NC during a movement.
Override	The override set by the user (0..100%) is displayed here.
Total/Control Output	Total output of the NC axis to the drive in % and position control portion of the output in %.
Error	Axle error code. An axis error can be deleted with the reset key.

Set enables

Via the button **Set** the controller enable, the feed enables and the override of the axis can be set.

Input fields for position controller setting

If the NC axis is operated in velocity mode (CSV), the position is controlled by the NC. At this point the two most important parameters, the gain factor Kv and the reference velocity, can be set. Further setting options can be found in the parameters of the axis.

Input fields for travel commands

Target Position: Target position for a subsequent travel command (F5)

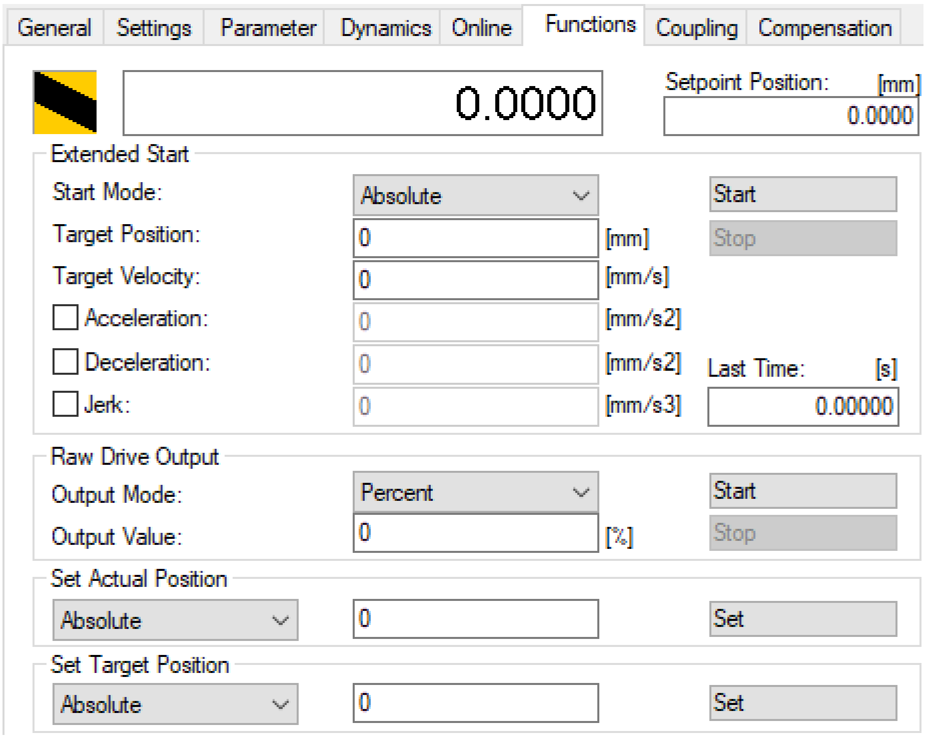
Target Velocity: Velocity of a subsequent travel command (F5)

Function button

Button	Key	Description
– – F1	F1	Reverse travel with Manual Velocity (Fast)
– F2	F2	Reverse travel with "Manual Velocity (Slow)"
+ F3	F3	Forward travel with "Manual Velocity (Slow)"
++ F4	F4	Forward travel with "Manual Velocity (Fast)"
◇ F5	F5	Start, with the values set in the input fields and the set dynamics.
⊚ F6	F6	Stop
Ⓡ F8	F8	Reset
→• F9	F9	Calibrate with the values set in the "Global" menu. ***Notice* The signal source of the referencing cam can be set in the encoder parameters (Homing Sensor Source). In the default setting, the referencing cam signal must be mapped into the axis data structure (Axis.PlcToNc.ControlDword.5) by the PLC so that the sequence triggered with F9 can react to the cam.**

4.2.6 Functions

In the **Functions** tab the most important commands for commissioning can be given to the axis.



0.0000

Setpoint Position: [mm] 0.0000

Extended Start

Start Mode:	Absolute		Start
Target Position:	0	[mm]	Stop
Target Velocity:	0	[mm/s]	
☐ Acceleration:	0	[mm/s2]	
☐ Deceleration:	0	[mm/s2]	Last Time: [s]
☐ Jerk:	0	[mm/s3]	0.00000

Raw Drive Output

Output Mode:	Percent		Start
Output Value:	0	[%]	Stop

Set Actual Position

Absolute	0	Set

Set Target Position

Absolute	0	Set

Different travel commands like absolute start, relative start or endless travel can be selected in the *Extended Start* frame . The necessary parameters below will adjust according to the selection of the start type. Thus, in reverse mode, there are two target positions between which the axis moves back and forth.

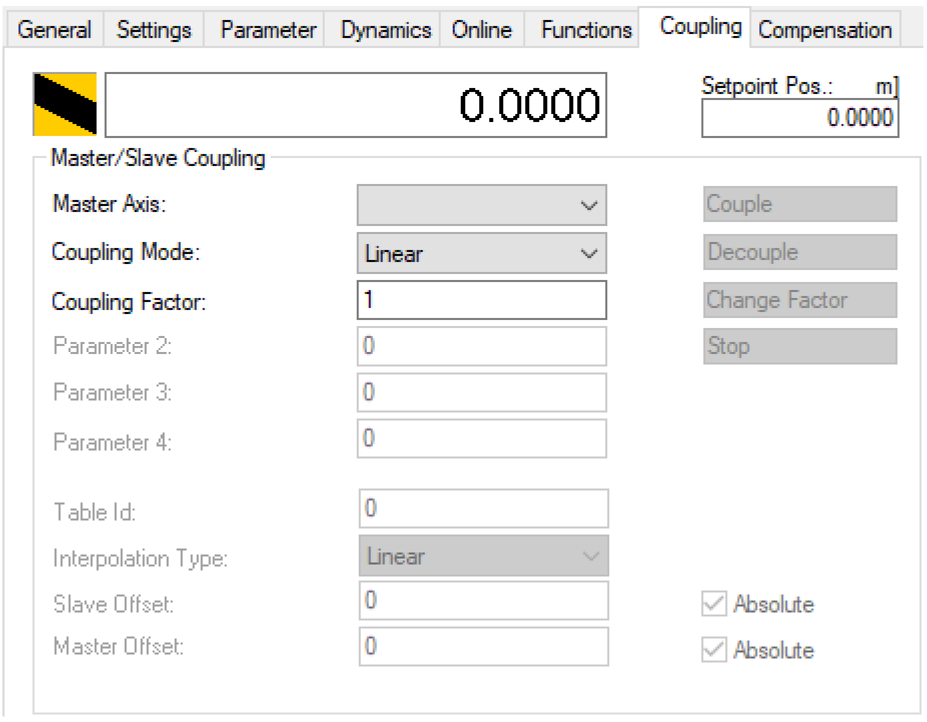
The drive output can be set to a fixed value for commissioning in the *Raw Drive Output* frame. This setting must be used very carefully.

The axis position can be set to a new value in the *Set Actual Position* frame.

Set Target Position changes the target position of the axis during a travel command.

4.2.7 Coupling

Two axes can be coupled with each other in the **Coupling** tab. To do this, first select the master axis and the desired coupling method, e.g. linear coupling. Afterwards, necessary parameters, such as the gear ratio, can be entered. The *Couple* button executes the coupling, *Decouple* decouples the axes again.



General | Settings | Parameter | Dynamics | Online | Functions | Coupling | Compensation

0.0000

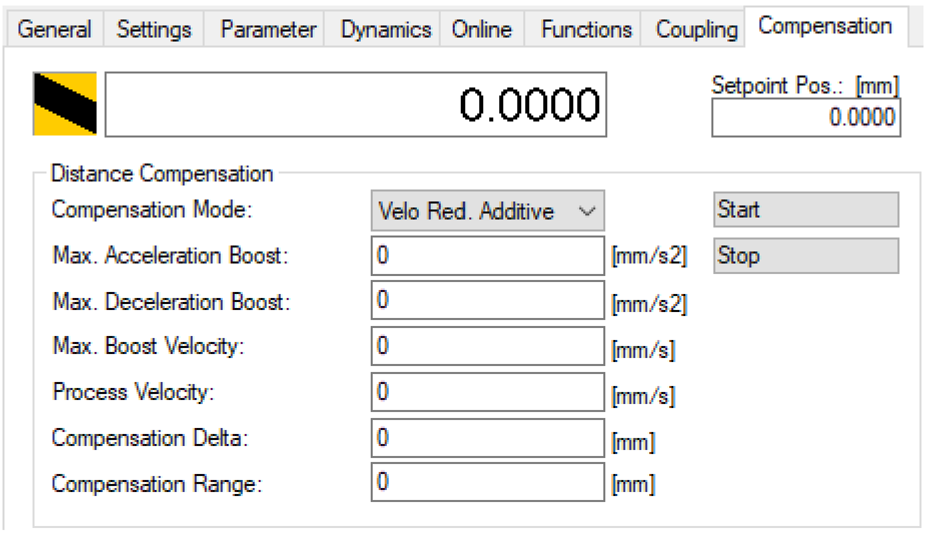
Setpoint Pos.: m] 0.0000

Master/Slave Coupling

Master Axis:		Couple
Coupling Mode:	Linear	Decouple
Coupling Factor:	1	Change Factor
Parameter 2:	0	Stop
Parameter 3:	0	
Parameter 4:	0	
Table Id:	0	
Interpolation Type:	Linear	
Slave Offset:	0	✓ Absolute
Master Offset:	0	✓ Absolute

4.2.8 Compensation

A superimposed travel command can be triggered during the travel of an axis in the **Compensation** tab (position compensation or superposition).



General | Settings | Parameter | Dynamics | Online | Functions | Coupling | Compensation

0.0000

Setpoint Pos.: [mm] 0.0000

Distance Compensation

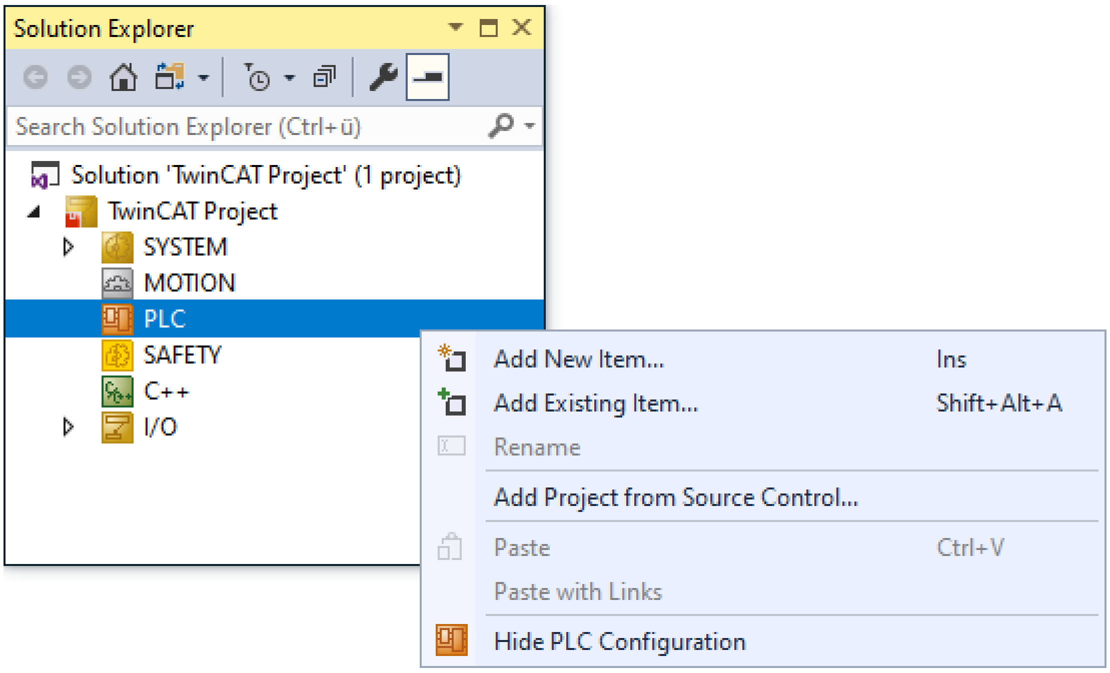
Compensation Mode:	Velo Red. Additive		Start
Max. Acceleration Boost:	0	[mm/s2]	Stop
Max. Deceleration Boost:	0	[mm/s2]	
Max. Boost Velocity:	0	[mm/s]	
Process Velocity:	0	[mm/s]	
Compensation Delta:	0	[mm]	
Compensation Range:	0	[mm]	

5 PLC

Function: Configuration and programming of PLC projects

Call: Select **Add New Item...** in the context menu and create a new PLC project.

Context menu


Solution Explorer
Search Solution Explorer (Ctrl+ü)
Solution 'TwinCAT Project' (1 project)
TwinCAT Project
SYSTEM
MOTION
PLC
SAFETY
C++
I/O
Add New Item... Ins
Add Existing Item... Shift+Alt+A
Rename
Add Project from Source Control...
Paste Ctrl+V
Paste with Links
Hide PLC Configuration

Add New Item…	Create new PLC project.
Add Existing Item…	Add existing PLC project.
Rename	Rename project.
Add Project from Source Control…	Add existing PLC project from source control.
Paste	Paste PLC project.
Paste with Links	Paste PLC project with links.
Hide PLC Configuration	Hide PLC configuration in Solution Explorer.

For detailed information on PLC configuration and programming, refer to the PLC documentation in the Beckhoff Information System:

- PLC

5.1 Your first TwinCAT 3 PLC project

Contents of your first project

In this tutorial, you will program a simple refrigerator control.

- As with a conventional refrigerator, the set temperature is set by the user via a control knob.
- The refrigerator detects the actual temperature via a sensor. If this is too high, the refrigerator starts the compressor with an adjustable delay.
- The compressor cools until the set temperature minus a hysteresis of 1 degree is reached. The hysteresis is intended to prevent the actual temperature from oscillating too much around the set temperature, and to prevent the compressor from continuously switching on and off.
- When the door is open, a lamp lights up inside the refrigerator.
- If the door is open for too long, a timed acoustic signal will sound.

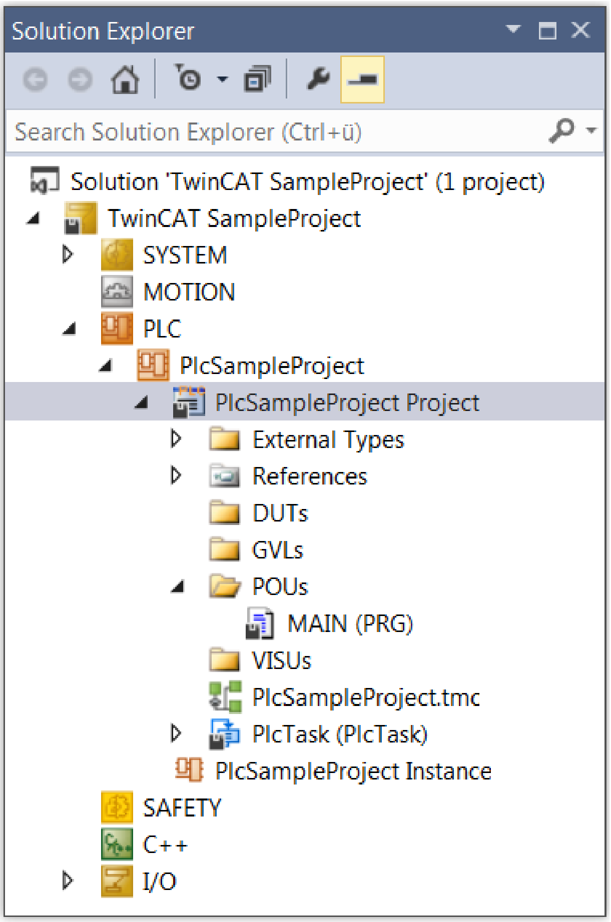
- If the compressor does not reach the set temperature for an extended period of time despite the motor's activity, the beeper emits a continuous acoustic signal.

Project planning:

The cooling activity is controlled in the main program of the PLC project, the signal management take place in another program block. The required standard function blocks are available in the Tc2_Standard library. Since no real temperature sensors and no real actuators are connected in this example project, you will also write a program for simulating the rise and fall in temperature. This allows you to monitor the operation of the refrigerator control unit in online mode. You define variables that are to be used by all function blocks in a global variable list.

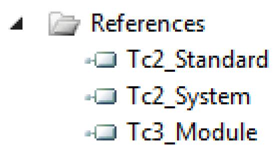
Creating the PLC project

1. In the **File** menu, select **New > Project** to create a new TwinCAT project file.
   ⇨ A new Solution with the TwinCAT project tree opens in the **Solution Explorer**.
2. In the **Solution Explorer** select the **PLC** node and the command **Add New Item**, in order to add a PLC project to the TwinCAT project.
   ⇨ The dialog **Add New Item – TwinCAT <project name>** opens.
3. In the category **Plc Templates** select the **Standard PLC project** template.
4. Enter a name and storage location for the project and click the **Add** button.
   ⇨ The selected template automatically creates a MAIN program, which is called by a task. "Structured Text (ST)" is automatically selected as the programming language.



The Library Manager with some important default libraries is automatically selected under

References. The Tc2_Standard library contains all the functions and function blocks described by the IEC 61131-3 standard.


References
Tc2_Standard
Tc2_System
Tc3_Module

Declaring global variables

First, declare the variables that you want to use throughout the PLC project. To do this, you create a global variable list:

1. Select the subfolder **GVLs** in the PLC project tree.
2. In the context menu select the command **Add > Global Variable List**.
3. Change the automatically entered name "GVL" to "GVL_Var".
4. Confirm with **Open**.

   ⇨ The object “GVL_Var” ( ) appears in the PLC project tree in the subfolder **GVLs**. The GVL editor opens.

   ⇨ When the textual view appears, the keywords VAR_GLOBAL and END_VAR are already included.

5. Activate the tabular view for the example by clicking on the button in the right sidebar of the editor.

   ⇨ An empty row appears. The cursor is located in the column **Name**.

6. Enter “fTempActual” in the **Name** field.

   ⇨ At the same time, the scope VAR_GLOBAL and the data type BOOL are automatically entered in the row.

7. Double-click the field in the **Data type** column.

   ⇨ The field is now editable, and the button appears.

8. Click the button and select **Input Assistant**.

   ⇨ The **Input Assistant** dialog opens.

9. Select the data type REAL and click **OK**.
10. Enter a numerical value in the **Initialization** column, for example “8.0”.

⇨ Declare the following variables in the same way:

Name	Data type	Initialization	Comment
fTempActual	REAL	1.0	Actual temperature
fTempSet	REAL	8.0	Set temperature
bDoorOpen	BOOL	FALSE	Door status
tImAlarmThreshold	TIME	T#30s	Compressor running time after which an alarm sounds.
tDoorOpenThreshold	TIME	T#10s	Time from door opening after which an alarm sounds.
bCompressor	BOOL	FALSE	Control signal
bSignal	BOOL	FALSE	Control signal
bLamp	BOOL	FALSE	Status message

Creating the main program for cooling control in the CFC editor

In the MAIN program block created by default, you describe the main function of the PLC program: The compressor becomes active and cools when the actual temperature is higher than the set temperature plus a hysteresis. The compressor is switched off when as the actual temperature is lower than the set temperature minus the hysteresis.

Perform the following steps to describe this functionality in the implementation language "Continuous Function Chart (CFC)":

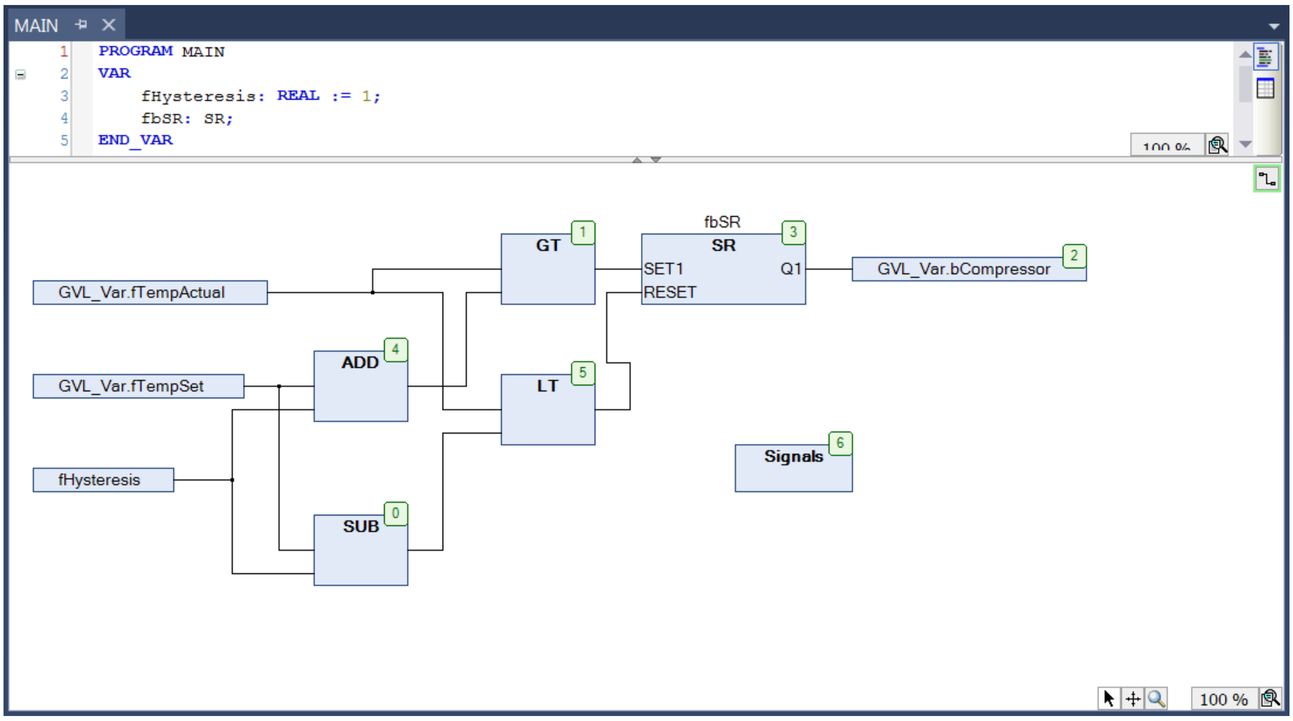
✓ Since the automatically created MAIN program block is created by default in the implementation language "Structured Text (ST)", you must first delete this program. With the context menu command **Add > POU...** you create a new MAIN program in the implementation language "Continuous Function Chart (CFC)".

1. Then double-click on the **MAIN** program in the PLC project tree (subfolder **POUs**).
   ⇨ The CFC Editor opens with the **MAIN** tab. The Declaration editor in textual or tabular representation is displayed above the graphical editor area. On the right is the **Toolbox** view. If the toolbox does not appear, you can use the command **Toolbox** in the **View** menu to place it on the desktop.
2. In the **Toolbox** view, click the **Input** element and drag it with the mouse to a position in the CFC Editor.
   ⇨ The nameless input **???** has been inserted.
3. In the CFC Editor, click the input **???** and open the **Input Assistant** by clicking ... .
4. Select the variable `fTempActual` from the category **Variables** at **Project > GVLs**.
5. Confirm the dialog with **OK** to reference the global variable `fTempActual`.
6. As in step 3, create a further input with the name of the global variable `rTempSet`.
7. Create another input.
8. Click **???** and replace it with the name `fHysteresis`.
   ⇨ Since this is not the name of an already known variable, the **Auto Declare** dialog appears. The name is already included in the dialog.
9. Complete the fields in the **Auto Declare** dialog with the data type REAL and the initialization value "1".
10. Click the **OK** button.
    ⇨ The variable `fHysteresis` appears in the declaration editor.
11. Now add an addition function block: In the **Toolbox** view, click the **Function block** element and drag it with the mouse to a position in the CFC Editor.
    ⇨ The function block appears in the CFC Editor.
12. Replace **???** with ADD.
    ⇨ The ADD (Addition) function block adds all inputs that are connected to it.
13. Connect the input GVL_Var.fTempSet with the ADD function block: To do this, click on the output connector of the input and drag it to the upper input of the ADD function block.
14. Connect the `fHysteresis` input to the lower input of the ADD function block in the same way.
    ⇨ The two inputs `fHysteresis` and `fTempSet` are now added by ADD.
15. If you want to move an element in the editor, click on a free space in the element or on the frame so that the element is selected (red frame, shaded red).
16. Keep the mouse button pressed and drag the element to the desired position.
17. Create another function block to the right of the ADD function block.
    ⇨ It is intended to compare "GVL_Var.fTempActual" with the sum of "GVL_Var.fTempSet" and fHysteresis.
18. Assign the function GT (Greater Than) to the function block.
    ⇨ The GT function block works as follows:
    ```
    IF (oberer Eingang > unterer Eingang) THEN Ausgang := TRUE;
    ```
19. Connect the input "GVL_Var.fTempActual" with the upper input of the GT function block.
20. Connect the output of the ADD function block with the lower input of the GT function block.
21. Now create a function block to the right of the GT function block that starts or stops the cooling compressor depending on the input condition (Set - Reset).
22. Enter the name "SR" in the **???** field.
23. Close the open input field above the function block (SR_0) with the Enter key.
    ⇨ The **Auto Declare** dialog appears.
24. Declare the variable with the name "fbSR" and the data type SR.

25. Click the **OK** button.
    ⇨ The SR function block, which is also defined in the library Tc2_Standard, determines the THEN at the output of the GT function block. The inputs SET1 and RESET appear.
26. Connect the output connection on the right of the GT function block to the SET1 input of the fbSR function block.
    ⇨ SR can reset a Boolean variable from FALSE to TRUE. If the condition applies at input SET1, the Boolean variable is set to TRUE. If the condition applies to RESET, the variable is reset. In our example, the Boolean (global) variable is "GVL_Var.bCompressor".
27. Create an **Output** element and assign the global variable "GVL_Var.bCompressor" to it. Draw a connecting line between "GVL_Var.bCompressor" and the output connection from Q1 to SR.

Now enter the condition under which the compressor should switch off again, i.e. under which the RESET input of the SR function block receives a TRUE signal. To do this, formulate the opposite condition as described above. Use the function block SUB (Subtract) and LT (Less Than) for this purpose.

The following CFC plan is created:


```
PROGRAM MAIN
VAR
    fHysteresis: REAL := 1;
    fbSR: SR;
END_VAR
```

Creating a program block for signal management in the Ladder Diagram editor

Now implement the signal management for the alarm buzzer and for switching the lamp on and off in a further program block. The implementation language "Ladder Diagram (LD)" is suitable for this purpose.

Use separate networks for the following signals:

- A continuous acoustic signal sounds if the compressor runs too long because the temperature is too high.
- A pulsed signal sounds if the door is open too long.
- The light is on as long as the door is open.

1. In the PLC project tree (subfolder **POUs**) create a POU object of type **Program** with the implementation language "Ladder Diagram (LD)".
2. Name the POU object "Signals**"**.
    ⇨ "Signals" appears in the PLC project tree below MAIN. The Ladder Diagram editor opens with the **Signals** tab. The declaration editor appears in the upper part of the screen, the **Toolbox** view on the right. The LD contains an empty network.
3. In the network, you program that an acoustic signal sounds if the cooling compressor runs for too long without reaching the set temperature. To do this, insert a TON timer function block.

⇨ It switches a Boolean TRUE signal to TRUE after a specified time.

4. Select a TON in the **Toolbox** view under **Function blocks** and drag it into the empty network to the **Start here** rectangle, which appears.
5. Release the mouse button when the field turns green.

   ⇨ The function block appears as rectangle with inputs and outputs and is automatically assigned the instance name TON_0. The line editor is open and the cursor is blinking.
6. Confirm the instance name with Enter key.

   ⇨ The **Auto Declare** dialog opens.
7. Declare the variable with the name "fbTimer1" and the data type TON.
8. Click the **OK** button.

If you want to read the help for the function block, mark the complete name of the function block with the cursor and press [F1].

9. To program that the function block is activated as soon as the cooling compressor starts running, name the contact at the upper input of the function block "GVL_Var.bCompressor".

   ⇨ You have already defined this Boolean variable in the global variable list "GVL_Var".

When you start entering a variable name at the input position, you will always automatically receive a list of all variables whose names begin with the characters entered and which can be used at this point. This feature is a default setting in the TwinCAT options for Smart Coding.

10. Insert the signal you want to activate: To do this, drag a **Coil** from the **Toolbox** view, category **Ladder Diagram elements** to output Q of the TON function block. Name the coil "GVL_Var.bSignal".
11. Add the variable "GVL_Var.tImAlarmThreshold" to the PT input of "fbTimer1" to define the time from activation of the TON device after which the signal should sound. To do this, click on the rectangle with a border to the right of the input connection.
12. Enter the variable name.
13. Click the TON function block and in the context menu select the command **Remove unused FB call parameters**.

    ⇨ The unused output ET was removed.
14. In the second LD network, you program that the signal should sound intermittently when the door is opened too long.
15. First, create a new POU object of type **Function Block** in the implementation language "Function Block Diagram (FBD)" in the PLC project tree, subfolder **POUs**.
16. Name it "FB_Blinker".

    ⇨ The function block FB_Blinker appears in the PLC project tree above MAIN. The FBD editor opens with the **FB_Blinker** tab. The declaration editor appears in the upper part of the screen, the **Toolbox** view on the right. The FBD contains an empty network.

    ⇨ The flasher is to be realized by an AND operator and two TON function blocks.
17. Select a **function block** in the **Toolbox** view under **General** and drag it into the empty network to the **Start here** rectangle, which appears.
18. Release the mouse button when the field turns green.

    ⇨ The function block appears as a rectangle with inputs and outputs.
19. Click **???** within the function block and enter the keyword AND in the field, which is now editable.
20. Confirm with Enter key.

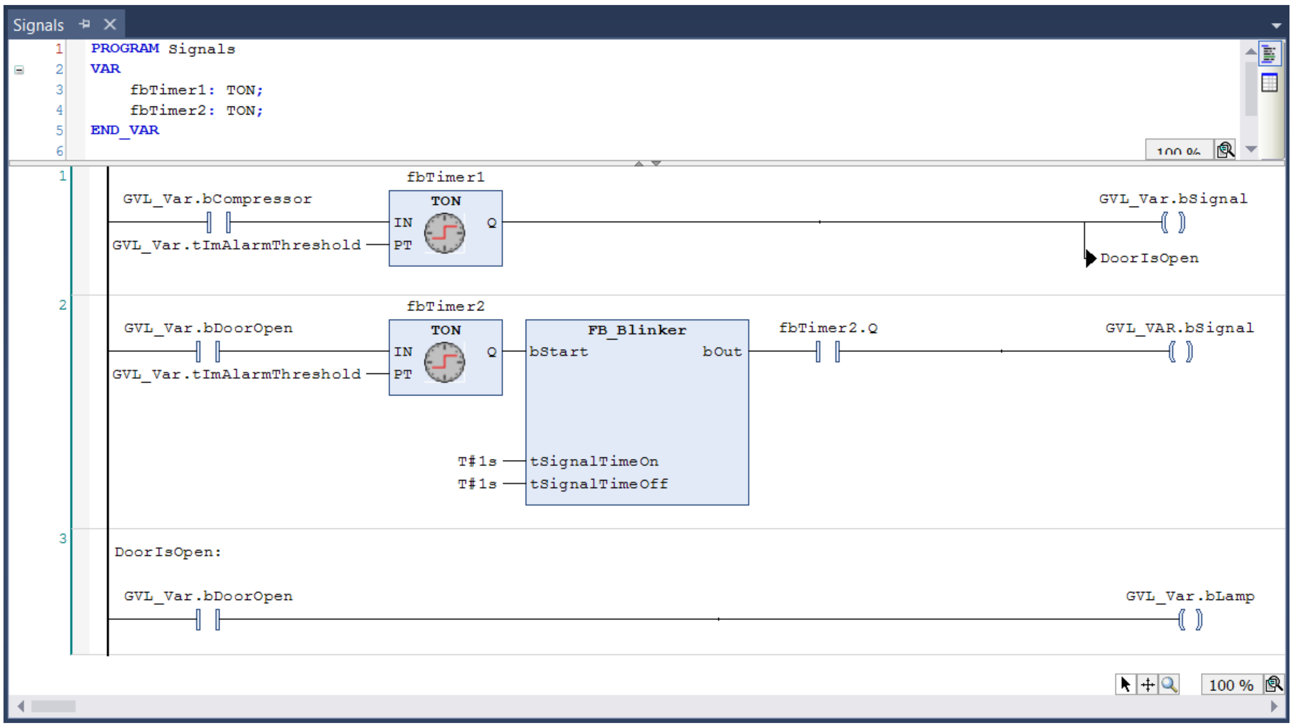
    ⇨ Since it is a function, no instantiation is required.
21. Select a TON function block in the **Toolbox** view under **Function blocks** and drag it into the network to the output of the AND function block.
22. Release the mouse button when the field turns green.

    ⇨ The function block appears as rectangle with inputs and outputs and is automatically assigned the instance name TON_0.

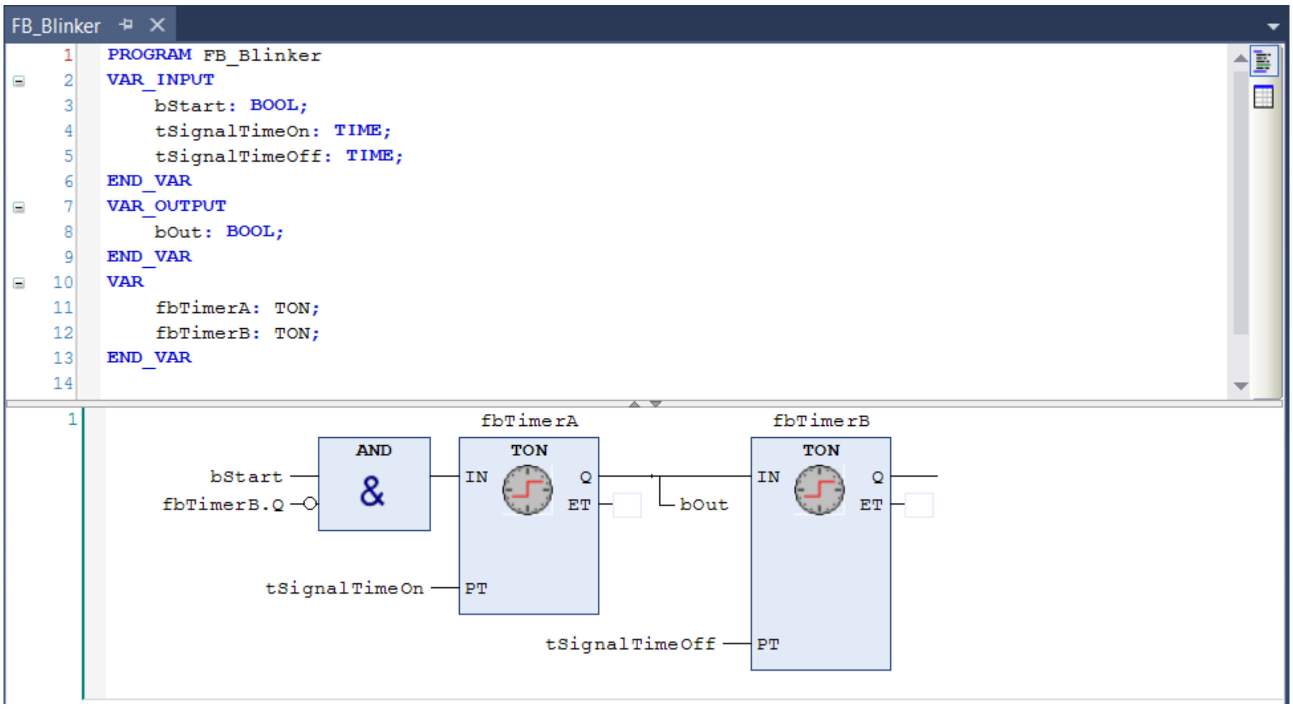
23. Close the open input field above the function block (TON_0) with the Enter key.
    ⇨ The **Auto Declare** dialog opens.
24. Declare the variable with the name “fbTimerA” and the data type TON.
25. Click the **OK** button.
26. Select a further TON function block in the **Toolbox** view under **Function blocks** and drag it into the network to the output of the TON function block "fbTimerA".
27. Release the mouse button when the field turns green.
    ⇨ The function block appears as rectangle with inputs and outputs and is automatically assigned the instance name "TON_0".
28. Close the open input field above the function block (TON_0) with the Enter key.
    ⇨ The **Auto Declare** dialog opens.
29. Declare the variable with the name “fbTimerB” and the data type TON. Click the **OK** button.
    ⇨ The first input of the AND function block is to be connected to a Boolean variable "bStart".
30. Click **???** and enter the variable name.
31. Confirm with Enter key.
    ⇨ The **Auto Declare** dialog opens. The name and the data type are recognized automatically.
32. Select the entry VAR_INPUT as **Scope**.
33. Confirm the dialog with **OK**.
    ⇨ The second input of the AND function block is to be connected with the output Q of the second TON function block "fbTimerB".
34. Click **???** and open the **Input Assistant** via … .
35. In the category **Variables** select the function block "fbTimerB" and the output Q.
36. Confirm the dialog with **OK**.
    ⇨ The variable "fbTimerB.Q" is added at the second input.
37. Select the second input and open the context menu with a right-click.
38. Select the command **Negation** to negate the input.
    ⇨ A circle appears at the corresponding input.
39. Set the time until output Q is set via the PT inputs of the TON function blocks.
40. Declare the input variables "tSignalTimeOn" and "tSignalTimeoff" for the TON function blocks "fbTimerA" and "fbTimerB" via the **Auto Declare** dialog.
41. Select the entry VAR_INPUT as **Scope**.
    ⇨ The generated clock pulse is to be output at output Q of the TON function block "fbTimerA".
42. To do this, select an **assignment** in the **Toolbox** view under **General** and drag it into the network to the output of the TON function block "fbTimerA".
43. Release the mouse button when the field turns green.
    ⇨ The assignment is added between the function blocks "fbTimerA" and "fbTimerB".
44. Click **???** and enter the variable name "bOut".
45. Confirm with Enter key.
    ⇨ The **Auto Declare** dialog opens.
46. Select the **Scope** VAR_OUTPUT and the data type BOOL.
47. Confirm the dialog.
48. Finally, remove the **???** at the unused inputs and outputs of the function blocks.
    ⇨ The completed function block FB_Blinker can now be instantiated and called.
49. Open the program "Signals" in the FBD/LD/IL editor.
50. Click below the first network in the editor window.
51. In the context menu, select the command **Insert network**.
    ⇨ An empty network with number 2 appears.

52. As in the first network, implement a TON function block for time-controlled activation of the signal, this time triggered by the global variable "GVL_Var.bDoorOpen" at input IN.
53. Add the global variable "GVL_Var.tImDoorOpenThreshold" at the input PT.
54. In addition, add the function block FB_Blinker to output Q of the TON function block in this network.
    ⇨ The function block FB_Blinker clocks the signal forwarding Q and therefore "GVL_Var.bSignal".
55. To do this, drag a **Contact** element from the **Toolbox** view to the OUT output of the function block.
56. Assign the variable "fbTimer2.Q" to the contact.
57. Insert an element **Coil** after the contact and assign the global variable "GVL_Var.bSignal" to it.
58. Assign the value T#1s to the two input variables "tSignalTimeOn" and "tSignalTimeOff" of function block FB_Blinker.
    ⇨ The cycle time is then 1 second for TRUE and 1 second for FALSE.
59. Click on the TON function block.
60. Select the command **Remove unused FB call parameters** in the context menu.
    ⇨ The unused output ET was removed.
    ⇨ In the third network of the LD, program the lamp to light up while the door is open.
61. To do this, add another network and at the left a contact "GVL_Var.bDoorOpen", which leads directly to an inserted coil "GVL_Var.bLamp".
    ⇨ TwinCAT processes the networks of an LD in succession.
62. To ensure that only network 1 or only network 2 is executed, add a jump to network 3 at the end of network 1.
63. Select network 3 by clicking into the network or into the field with the network number.
64. From the context menu select the command **Insert label**.
65. Replace the text **Label:** of the label in the upper left section of the network with "DoorIsOpen".
66. Select network 1.
67. Drag the **Jump** element into network from the **Toolbox** view, **General** category.
68. Position it on the **Output** rectangle that appears, or at **Insert jump here**.
    ⇨ The jump element appears. The jump destination is still shown as **???**.
69. Selecting **???** and click ... .
70. Select “DoorIsOpen” from the possible label identifiers.
71. Confirm with **OK**.
    ⇨ The label to network 3 is implemented.

⇨ The LD program now looks as follows:



```
PROGRAM Signals
VAR
    fbTimer1: TON;
    fbTimer2: TON;
END_VAR
```



```
PROGRAM FB_Blinker
VAR_INPUT
    bStart: BOOL;
    tSignalTimeOn: TIME;
    tSignalTimeOff: TIME;
END_VAR
VAR_OUTPUT
    bOut: BOOL;
END_VAR
VAR
    fbTimerA: TON;
    fbTimerB: TON;
END_VAR
```

Calling the “Signals” program in the main program

In our sample program, the MAIN program should call the “Signals” program for signal processing.

1. In the PLC project tree double-click on the MAIN program.
   ⇨ The MAIN program opens in the editor.
2. Drag a **Box** element from the **Toolbox** view into the editor of MAIN.
3. Use the **Input Assistant** to add the "Signals" program call to this function block from the **Module Calls** category.

Creating an ST program block for a simulation

Since this sample project is not linked to real sensors and actuators, write a program for simulating the rise and fall in temperature. This allows you to monitor the operation of the refrigerator control unit in online mode.

Use "Structured Text (ST)" to create the simulation program.

The program increases the temperature until the MAIN program detects that the set temperature has been exceeded and activates the cooling compressor. The simulation program then lowers the temperature again until the main program deactivates the compressor.

1. Add a POU function block called "Simulation" of type **Program** and written in the implementation language "Structured Text (ST)" to the PLC project tree.
2. Implement the following in the ST editor:

```
PROGRAM Simulation
VAR
    fbT1                  : TON;            //
The temperature is decreased on a time delay, when the compressor has been activated
    tCooling              : TIME := T#500MS;
    bReduceTemp           : BOOL;           //Signal for dereasing the temperature
    fbT2                  : TON;            //
The temperature is increased on a time delay, when the compressor has been activated
    tEnvironment          : TIME := T#2S;   //Delay time when the door is closed
    tEnvironmentDoorOpen  : TIME := T#1s;   //Delay time when the door is open
    bRaiseTemp            : BOOL;           //Signal for increasing the temperature
    tImTemp               : TIME;           //Delay time
    nCounter              : INT;
END_VAR

// After the compressor has been activated due to fTempActual being too high, the temperature de
creases.
// The temperature is decremented by 0.1°C per cycle after a delay of P_Cooling
IF GVL_VAR.bCompressor THEN
    fbT1(IN:= GVL_Var.bCompressor, PT:= tCooling, Q=>bReduceTemp);
    IF bReduceTemp THEN
        GVL_Var.fTempActual := GVL_Var.fTempActual-0.1;
        fbT1(IN:=FALSE);
    END_IF
END_IF

//If the door is open, the warming will occur faster; SEL selects tEnvironmentDoorOpen
tImTemp:=SEL(GVL_Var.bDoorOpen, tEnvironment, tEnvironmentDoorOpen);

//If the compressor is not in operation, then the cooling chamber will become warmer.
//The  temperature is incremented by 0.1°C per cycle after a delay of tImTemp
fbT2(IN:= TRUE, PT:= tImTemp, Q=>bRaiseTemp);
IF bRaiseTemp THEN
    GVL_Var.fTempActual := GVL_Var.fTempActual + 0.1;
    fbT2(IN:=FALSE);
END_IF

nCounter := nCounter+1; // No function, just for demonstration purposes.
```

Visualization

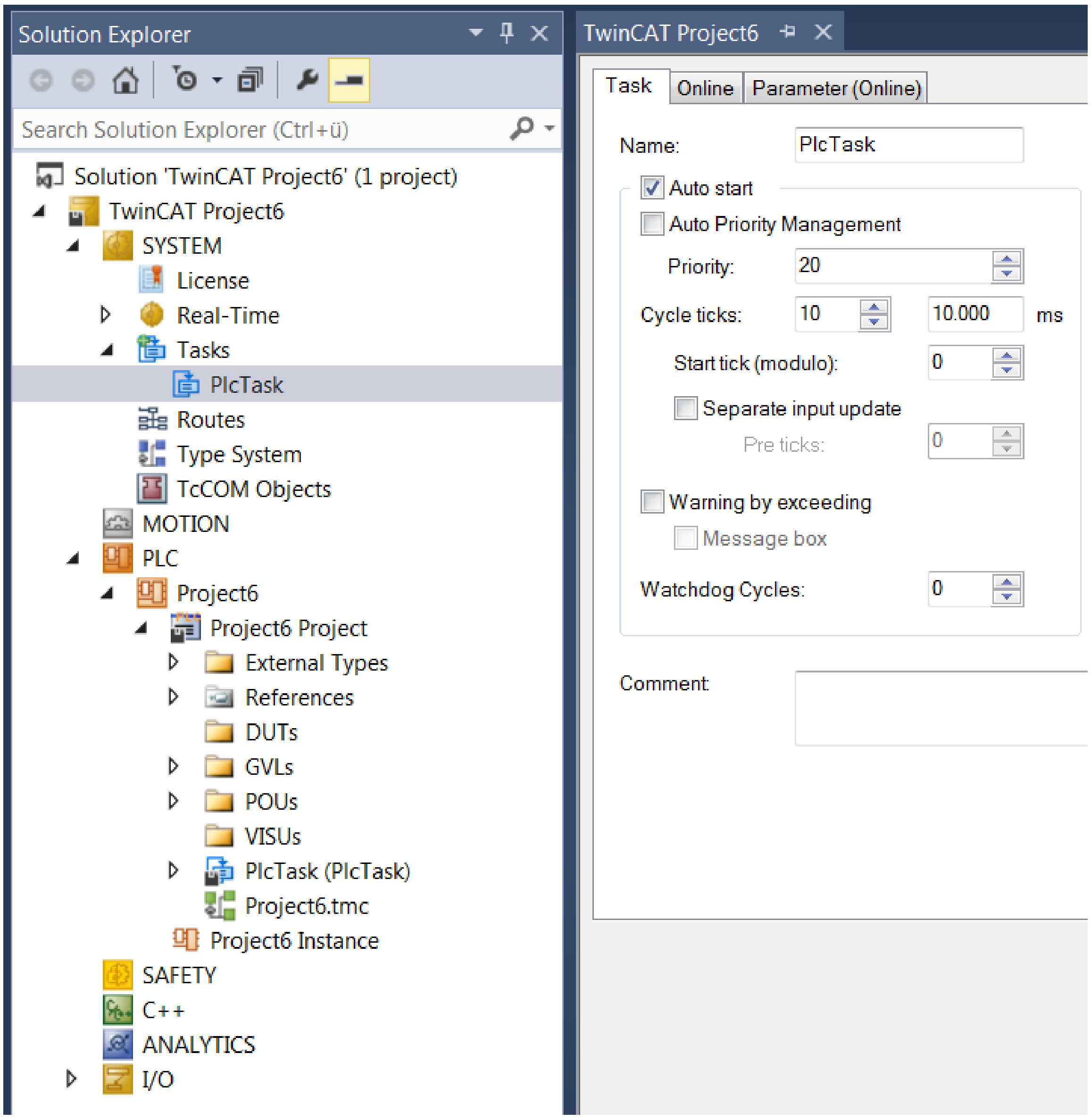
For convenient operation and monitoring of the entire control program, a visualization can be used that displays the refrigerator and shows the operation of the simulation program. In TwinCAT, you can also program a visualization in the PLC area. When the project is started on the controller, the visualization starts without you having to make an entry. Depending on the programming, you can trigger opening and closing of the door by clicking an on/off switch, for example, or by adjusting the temperature preselection using the needle of a control knob. The process of creating a visualization is not described here.

Defining the programs to be executed in the task configuration

The preset task configuration contains the call for the MAIN program. For our sample project you have to add the call for the program "Simulation".

1. In the PLC project tree, drag the "Simulation" entry to the task reference (PlcTask).
   ⇨ The "Simulation" program is added to the task configuration.
2. To view the task configuration, double-click the **PlcTask** entry in the PLC project tree.
   ⇨ In the **Solution Explorer** the referenced task (**PlcTask**) is enabled under **SYSTEM > Tasks**.
3. Double-click the task to open the task's configuration in an editor.

⇨ In the PLC project tree at **PlcTask** you see the POUs that are called by the task: MAIN (entered by default) and Simulation. The editor of the **PlcTask** node in the SYSTEM area shows the corresponding cycle time for the referenced PlcTask. In this sample, the interval is 10 milliseconds. In online mode, the task will process the two function blocks once per cycle.



Name: PlcTask
[✓] Auto start
[ ] Auto Priority Management
Priority: 20
Cycle ticks: 10 10.000 ms
Start tick (modulo): 0
[ ] Separate input update
Pre ticks: 0
[ ] Warning by exceeding
[ ] Message box
Watchdog Cycles: 0
Comment:

Defining the active PLC project

A TwinCAT controller can run multiple PLC projects. The first entry in the **Active PLC Project** drop-down list in the **TwinCAT PLC Toolbar Options** shows the currently active PLC project. If several PLC projects are present, you can select a PLC project via the drop-down list.

Checking the PLC project for errors

While entering code, TwinCAT immediately alerts you to syntax errors by means of a squiggly red line.

1. To get a syntax check over the whole PLC project, select the PLC project object "<PLC project name> Project".
2. Choose the command **Check all objects** from the context menu or from the **Build** menu.
   ⇨ The results of the check are displayed in the **Error List** view.
3. If necessary, open the **Error List** view with the command **Error List** from the **View** menu.
4. You can then double-click the message to jump to the corresponding code position.

The command **Check all objects** can be used to check and compile all function blocks in the PLC node. An error is generated if the compilation reveals function blocks in the tree that are uncompilable, for example because they may only have been included for testing purposes. It is therefore advisable to run the command **Check all objects**, particularly for checking library function blocks.

The commands **Build** or **Rebuild** can be used to limit checking and compiling to function blocks that are actually used in the PLC project.

Further checks of the PLC project are performed when it is loaded onto the controller.

Only error-free PLC projects can be loaded onto the controller.

Compiling the PLC module

The commands **Build** or **Rebuild** can be used to compile your code used in the PLC project and check for syntactic accuracy.

Choose Target System

Now select the target device for your control program from the **Choose Target System** drop-down list in the **TwinCAT XAE Base Toolbar Options**:

- Select **<Local>** to load the control code directly into the local runtime of your programming device. (Select this options for the present sample.)
- If you want to select another target device, select **Choose Target System** from the drop-down list. Then select a preconfigured target device or browse the network for a target device, configure it and then select it.

Activation of the configuration

1. Click in the **TwinCAT XAE Base Toolbar Options**.
   ⇨ A dialog appears asking whether you want to activate the configuration.
2. Click on **OK**.
   ⇨ A dialog appears asking whether TwinCAT should be restarted in Run mode.
3. Click on **OK**.

⇨ The configuration is activated, and TwinCAT is set to Run mode. In the taskbar shows the current status: . Activation also transfers the PLC project to the controller.

Loading the PLC project onto the PLC

✓ The PLC project was compiled without errors. See step Checking the PLC program for errors [▶ 96],

1. Select the **Login** command in the **PLC** menu or click on in the **TwinCAT PLC Toolbar Options**.
   ⇨ A dialog box appears asking whether the application should be created and loaded.
2. Click **Yes**.

⇨ The PLC project is loaded onto the controller. The engineering environment is now in online mode. The PLC modules are not yet in Run mode. In the **Solution Explorer** the following symbol appears before the PLC project object: . During the loading process, the **Error List** view displays information about the generated code size, the size of the global data and the resulting memory requirements on the controller.

Starting the program

If you have followed the entire tutorial up to this point, you can now use the PLC project on the PLC device.

1. Select the **Start** command in the **PLC** menu or click on in the **TwinCAT PLC Toolbar Options** (**[F5]**).

⇨ The program is running. The PLC modules are in Run mode. In the **Solution Explorer** the following symbol appears before the PLC project object: .

Monitoring and writing variable values once at runtime

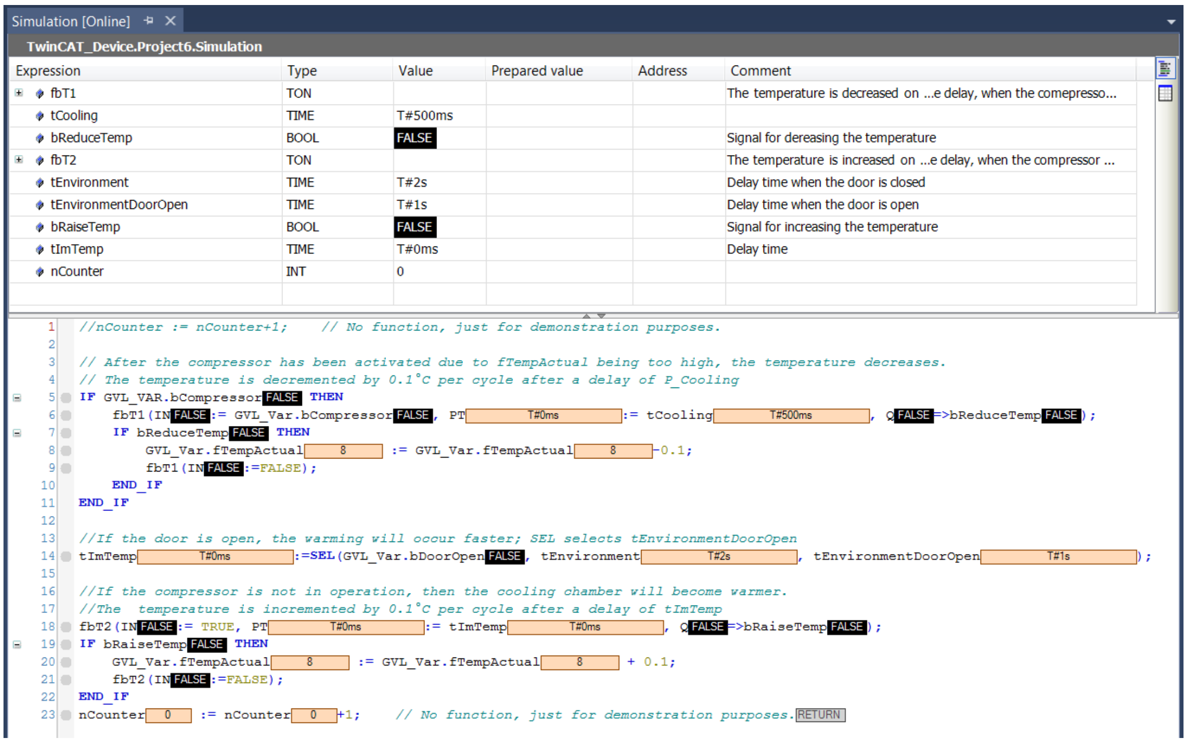
The following section illustrates the monitoring of the variable values in the various program blocks, and you are invited to set a certain variable value on the controller once from TwinCAT.

The actual values of the program variables can be seen in the online views of the function block editors or in watch lists. This sample is deliberately limited to monitoring in the function block editor.

✓ The PLC program runs on the controller.

1. Double-click the objects MAIN, "Signals", "Simulation" and "GVL_Var" in the PLC project tree to open the online views of the editors.

   ⇨ In the declaration part of each view, the table of expressions in the **Value** column shows the actual value of the variables on the controller.



Simulation [Online]

TwinCAT_Device.Project6.Simulation

Expression	Type	Value	Prepared value	Address	Comment
fbT1	TON				The temperature is decreased on ...e delay, when the comepresso...
tCooling	TIME	T#500ms			
bReduceTemp	BOOL	FALSE			Signal for dereasing the temperature
fbT2	TON				The temperature is increased on ...e delay, when the compressor ...
tEnvironment	TIME	T#2s			Delay time when the door is closed
tEnvironmentDoorOpen	TIME	T#1s			Delay time when the door is open
bRaiseTemp	BOOL	FALSE			Signal for increasing the temperature
tImTemp	TIME	T#0ms			Delay time
nCounter	INT	0			

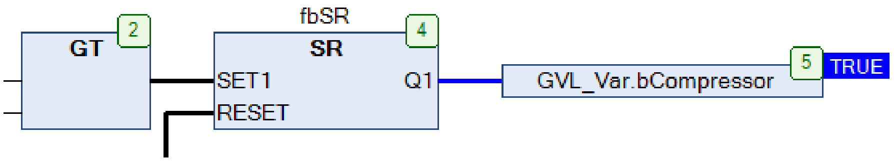
```
//nCounter := nCounter+1;    // No function, just for demonstration purposes.

// After the compressor has been activated due to fTempActual being too high, the temperature decreases.
// The temperature is decremented by 0.1°C per cycle after a delay of P_Cooling
IF GVL_VAR.bCompressor FALSE THEN
    fbT1(IN FALSE := GVL_Var.bCompressor FALSE , PT T#0ms := tCooling T#500ms , Q FALSE =>bReduceTemp FALSE );
    IF bReduceTemp FALSE THEN
        GVL_Var.fTempActual 8 := GVL_Var.fTempActual 8 -0.1;
        fbT1(IN FALSE :=FALSE);
    END_IF
END_IF

//If the door is open, the warming will occur faster; SEL selects tEnvironmentDoorOpen
tImTemp T#0ms :=SEL(GVL_Var.bDoorOpen FALSE , tEnvironment T#2s , tEnvironmentDoorOpen T#1s );

//If the compressor is not in operation, then the cooling chamber will become warmer.
//The  temperature is incremented by 0.1°C per cycle after a delay of tImTemp
fbT2(IN FALSE := TRUE, PT T#0ms := tImTemp T#0ms , Q FALSE =>bRaiseTemp FALSE );
IF bRaiseTemp FALSE THEN
    GVL_Var.fTempActual 8 := GVL_Var.fTempActual 8 + 0.1;
    fbT2(IN FALSE :=FALSE);
END_IF
nCounter 0 := nCounter 0 +1;     // No function, just for demonstration purposes. RETURN
```

   ⇨ The monitoring in the implementation part depends on the implementation language: For non-Boolean variables, the value is always in a rectangular field to the right of the identifier. In the ST editor, this also applies to Boolean variables. This display is named "Inline Monitoring". In the graphical editors, the value of a Boolean variable is indicated by the color of the output link line: black for FALSE, blue for TRUE:



⇨ Note how the variable values in the different function blocks change. For example, the global variable list "GVL_Var" shows how the values of "fTempActual" and "bCompressor" change when the simulation program is executed.

One-time setting of variable values on the controller:

1. Set the focus to the online view of the global variable list "GVL_Var**"**.
2. To specify a new setpoint, double-click the column **Prepared value** under the expression "fTempSet".

   ⇨ An input field opens.

3. Enter the value "9" and exit the input field.
4. To set the door to open, click once in the **Prepared value** field of the "bDoorOpen" expression.
   ⇨ The value TRUE is entered.
5. Click three more times to see that you can use this to set the prepared value to FALSE, then to empty again and then to TRUE again.
6. To write the prepared value TRUE once to the variable, select the command **Write values** in the **PLC** menu or click in the **TwinCAT PLC Toolbar Options**.

⇨ The two values are transferred to the **Value** column. The variable "bDoorOpen" no longer changes its value and the set temperature is now 9 degrees. The variable "tImTemp" changes to the value 1 s, because the refrigerator door is now "open" and thus the heating by simulation should be faster than before (2 s).

Setting breakpoints and step-by-step execution at runtime

Debugging: For troubleshooting, you want to check the variable values at certain code positions. To do this, you can define breakpoints for processing and initiate step-by-step execution of the instructions.

✓ The PLC program is loaded onto the controller and is running.

1. Double-click the "Simulation" object to open the program in the editor.
2. Place the cursor in the code line `nCounter:=nCoutner+1;` and press **[F9]**.
   ⇨ The symbol ● appears before the code line. It indicates that a breakpoint is set at this line. The symbol immediately changes to . The yellow arrow always points to the next instruction to be executed.
3. Consider the value of the variable "nCounter" in the inline monitoring or in the declaration part of the **Simulation** program.
   ⇨ The variable value no longer changes. The execution was stopped at the breakpoint.
4. Press **[F5]** to restart the execution.
   ⇨ The program stops again at the breakpoint after one cycle. "nCounter" was incremented by 1.
5. Press **[F11]** to execute the next step
   ⇨ RETURN at the end of the line `nCounter:=nCounter+1;` the instruction is highlighted in yellow
6. Press **[F11]** again to execute the next step.
   ⇨ The execution jumps to the editor of MAIN. Pressing **[F11]** repeatedly shows how the program is executed step-by-step. The instruction to be executed is again marked with a yellow arrow.
7. To disable the breakpoint and return to normal execution, move the cursor into the code line again and press **[F9]**. Then press **[F5]** to restart the program execution.

⇨ You can run through the program step-by-step and check variable values at certain code positions.

Executing a single cycle at runtime

✓ The PLC program is loaded onto the controller and is running.

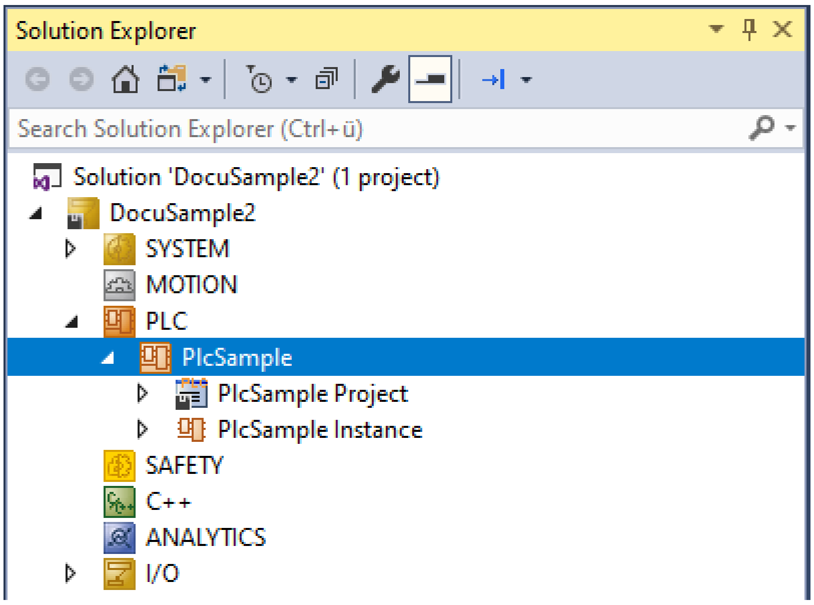
1. Observe the line `nCounter:=nCounter+1;` again in the **Simulation** program.
2. Click in the **TwinCAT PLC Toolbar Options** to execute a single cycle.
   ⇨ The execution runs through a cycle and stops again at the breakpoint. "nCounter" was incremented by 1.
3. Click the button three more times to see the individual cycles. Then press **[F5]** again.

⇨ The program runs again without stop and without forced values. The variable "tImTemp" has the value 1 s again.

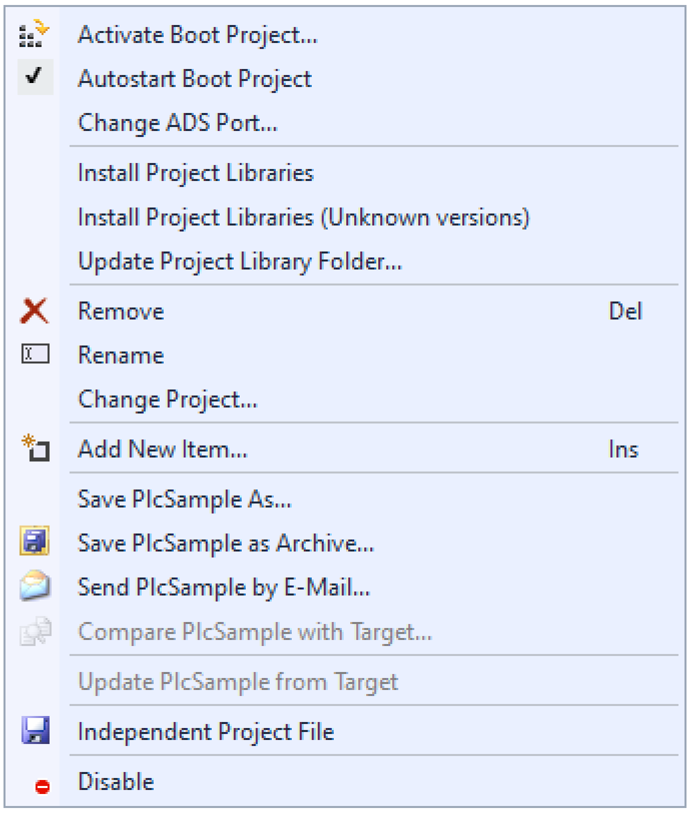
5.2 PLC project node

Function: This node encloses the PLC project and contains both the source code and the instances

Call: Right click on the node of the PLC project.



Context menu

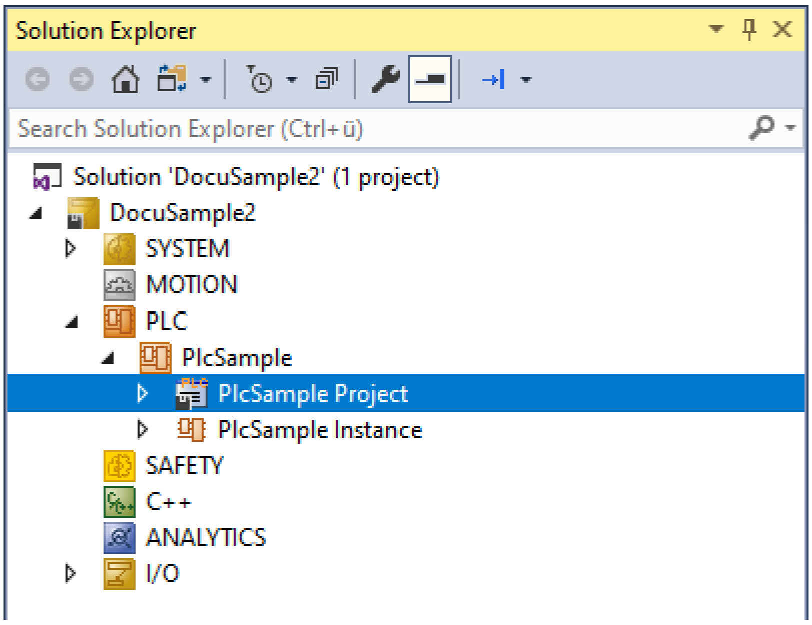

Activate Boot Project...
Autostart Boot Project
Change ADS Port...
Install Project Libraries
Install Project Libraries (Unknown versions)
Update Project Library Folder...
Remove Del
Rename
Change Project...
Add New Item... Ins
Save PlcSample As...
Save PlcSample as Archive...
Send PlcSample by E-Mail...
Compare PlcSample with Target...
Update PlcSample from Target
Independent Project File
Disable

Activate Boot Project…	Activate the boot project. (Writing the generated code to the target system.)
Autostart Boot Project…	Enable/disable the option for an automatic start of the boot project.
Change ADS Port…	Change the ADS port of the PLC project.
Install Project Libraries	Installing the libraries stored in the project.
Install Project Libraries (Unknown Versions)	Installs only the versions of the libraries used in the project that are unknown to the engineering system.
Update Project Libraries Folder…	Updating the libraries stored in the project.
Remove	Remove the PLC project.
Rename	Rename the PLC project.
Change Project…	Change the PLC project used in the project.
Add New Item…	Add another instance of the PLC project. (Only possible if this is already compiled)
Save <PlcSample> As…	Saves the contents of the PLC project to a new folder.
Save <PlcSample> as Archive…	Saves the content of the PLC project to a *.zip archive.
Save <PlcSample> by E-Mail…	Saves the content of the PLC project to a *.zip archive and opens an e-mail in the default e-mail program with the generated archive as attachment.
Compare <PlcSample> with Target…	Compares the PLC project with the project stored on the target system. (Provided that storing the source code on the target system is enabled)
Update <PlcSample> from Target	Loads the state of the project stored on the target system and thus replaces the local state of the PLC project.
Git	Opens the Git sub-context menu. (This entry is only present when Git is selected as the Source Control system)
Independent Project File	Saves the PLC project as an independent project file. (All information concerning the PLC project is stored in a *.xti file, which in turn is referenced in the TwinCAT project)
Disable	Disables the PLC project.

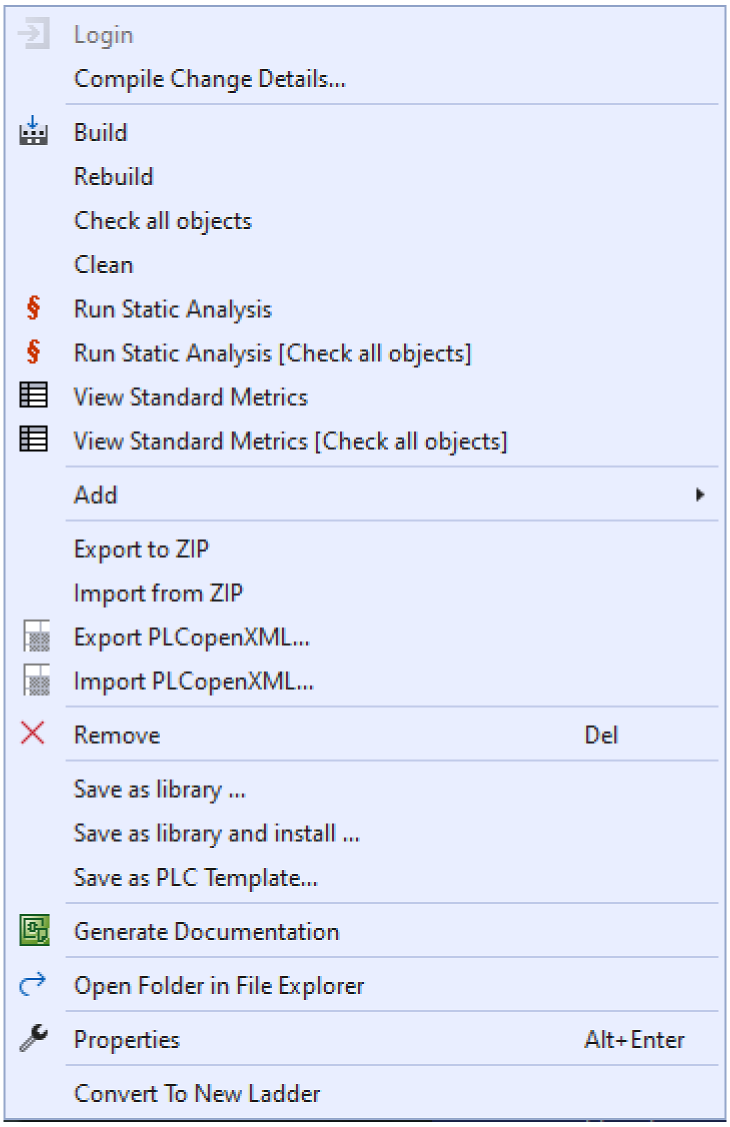
5.3 PLC project source code node

Function: This node contains the source code of the PLC project.

Call: Right-click on the <PlcSample> Project node.



Context menu

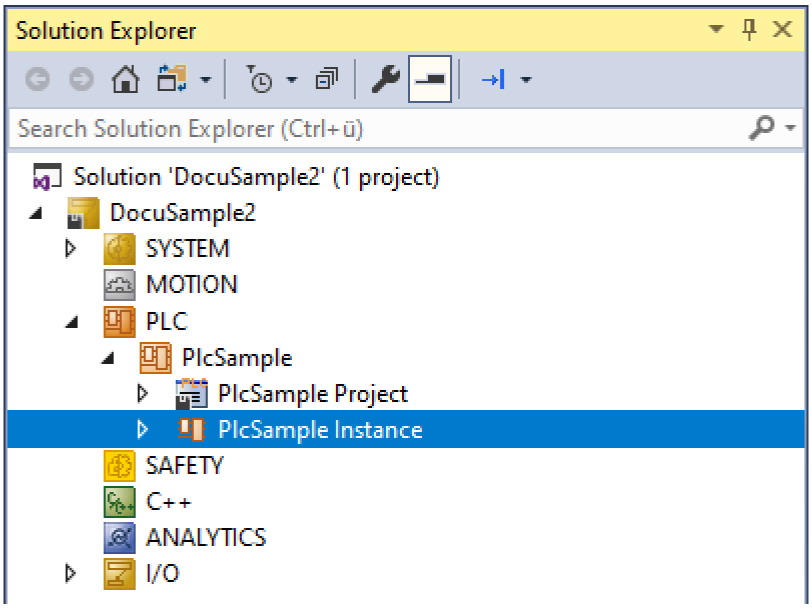


Login	Login (go online) with the PLC editors. If there are changes in the PLC code, an Online Change or download is offered. (You can find more information in the PLC documentation in chapter Loading program code, logging in and starting PLC.)
Compile Change Details…	Detail information about changes after compilation.
Build	Create/build the PLC project.
Rebuild	New build of the PLC project.
Check all objects	Check all objects.
Clean	Cleaning
Run Static Analysis	Run the static code analysis. (See TE1200 Tc3 PLC Static Analysis)
Run Static Analysis [Check all objects]	Run static code analysis on all objects. (See TE1200 Tc3 PLC Static Analysis)
View Standard Metrics	View standard metrics. (See TE1200 Tc3 PLC Static Analysis)
View Standard Metrics [Check all objects]	Display standard metrics considering all objects. (See TE1200 Tc3 PLC Static Analysis)
Add	Adding PLC objects.
Export to ZIP	Export selected objects to a *.zip file.
Import from ZIP	Import objects from a *.zip file.
Export PLCopen XML…	Export selected objects to a PLCopen xml file.
Import PLCopen XML…	Importing objects from a PLCopen file.
Remove	Remove the PLC project source code node. (The instance of the PLC is preserved)
Save as Library	Save the PLC project as a library.
Save as Library and install…	Save the PLC project as a library and install the created library.
Save as PLC Template	Save the PLC project as a template.
Generate Documentation	Generates documentation of the source code using markups and the project structure. (License of TE1030 Documentation Generation required.)
Open Folden in File Explorer	Opening a Windows Explorer in the path of the PLC project
Properties	Properties of the PLC project

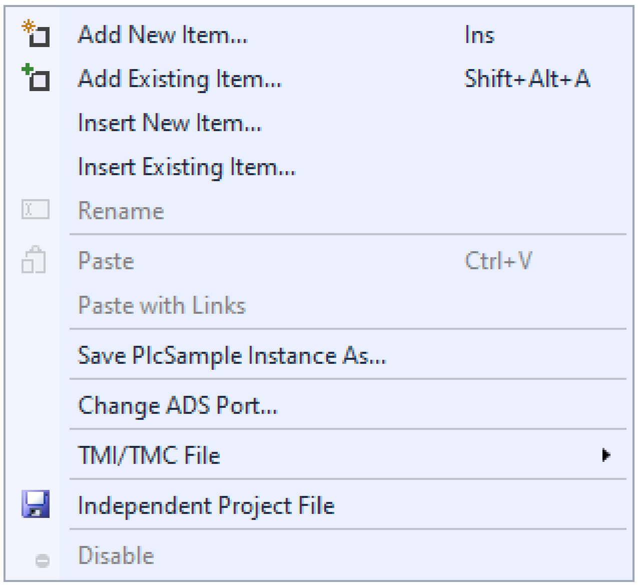
5.4 PLC instance node

Function: The PLC instance node represents the runtime module (the instance) that is generated from the PLC project. On the individual tabs described in the following you can see the information about the PLC runtime module divided into categories as well as set options and parameters and thus influence the behavior of the runtime module.

Call: Double-click on the node **<PLC name> Instance**.



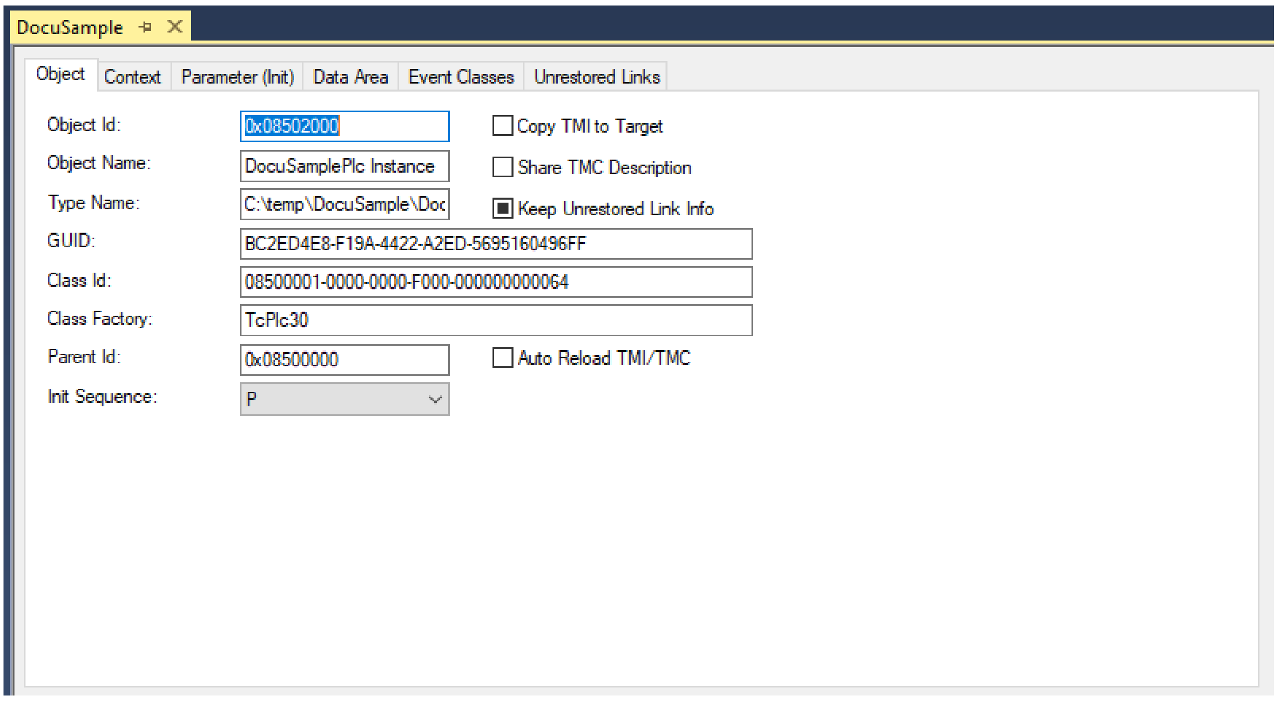
Context menu



Add New Item…	Create new PLC instance.
Add Existing Item…	Add existing PLC instance.
Insert New Item…	Inserting a new TcCOM module below the PLC instance.
Insert Existing Item…	Insertion of an existing TcCOM module instance below the PLC instance node.
Rename	Rename the instance name.
Paste	Paste
Paste with Links	Paste with linking information.
Change ADS Port…	Change the ADS port of the instance.
TMI/TMC File	Submenu for handling the TMI/TMC information.
Independent Project File	Saving the instance as an independent project file (*.xti file).
Disable	Disable the PLC instance.

5.4.1 Object

Information about the PLC runtime module is displayed in the **Object** tab of the PLC instance. In addition, a few common settings concerning the PLC runtime module can also be set.



Object Id: 0x08502000
Object Name: DocuSamplePlc Instance
Type Name: C:\temp\DocuSample\Doc
GUID: BC2ED4E8-F19A-4422-A2ED-5695160496FF
Class Id: 08500001-0000-0000-F000-000000000064
Class Factory: TcPlc30
Parent Id: 0x08500000
Init Sequence: P
☐ Copy TMI to Target
☐ Share TMC Description
▣ Keep Unrestored Link Info
☐ Auto Reload TMI/TMC

Information concerning the PLC runtime module:

Object Id	TwinCAT Object ID of the PLC runtime module
Object Name	Name of the PLC runtime module/the PLC instance
Type Name	Type name of the runtime module (Corresponds to the project name in the PLC.)
GUID	GUID of the PLC runtime module
Class Id	Class ID of the PLC
Class Factory	Class Factory of the PLC
Parent Id	The ID of the parent runtime module. For the PLC this is a PLC Control instance that manages the PLC runtimes.

In addition, some settings can also be made:

Init Sequence	Defines the startup behavior of a runtime module.
Copy TMI to Target	Defines whether the TMI file is copied to the target.
Share TMC Description	Defines whether the TMC file is copied to the target.
Keep Unrestored Link Info	Enables the **Unrestored Links** option (see Unrestored Links [▶ 109]).
Auto Reload TMI/TMC	Defines whether the TMC file should be reloaded when a change is made externally.

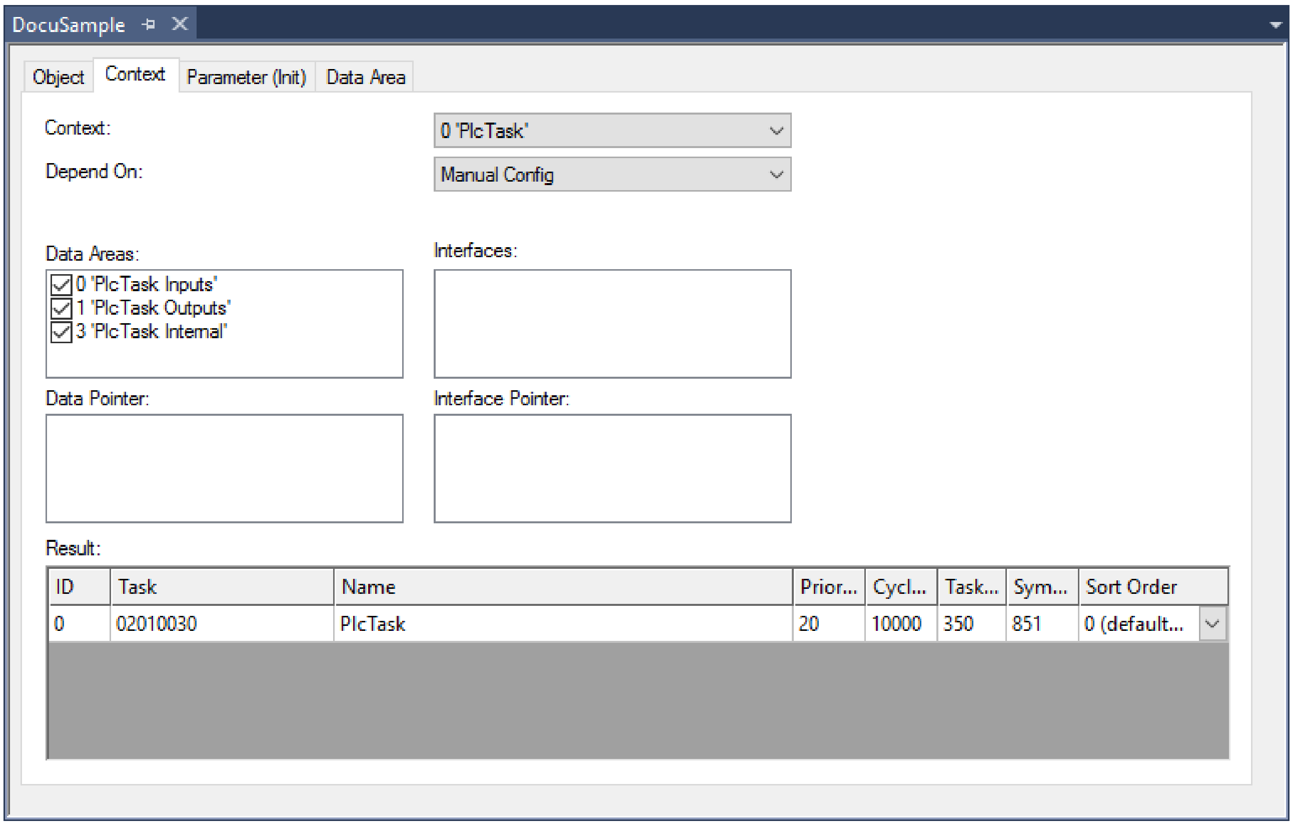
The option **Keep Unrestored Link Info** is an option with 3 states:

Checked	Link information is retained.
Not checked	Link information is discarded.
Intermediate (filled with a rectangle)	It will ask each time if the link information should be kept.

The option **Auto Reload TMI/TMC** defines whether the TMC file should be re-read in case of a change from outside. This is important when using the stand-alone PLC (see Use stand-alone PLC project), because then the TMC file is recreated by recompiling the PLC in another TwinCAT project.

5.4.2 Context

The (time) contexts of the PLC module are listed in the **Context** tab. These are the tasks used in the PLC, which are shown in this tab in table form with their settings.



Context: 0 'PlcTask'

Depend On: Manual Config

Data Areas:
- [✓] 0 'PlcTask Inputs'
- [✓] 1 'PlcTask Outputs'
- [✓] 3 'PlcTask Internal'

Interfaces:

Data Pointer:

Interface Pointer:

Result:

ID	Task	Name	Prior...	Cycl...	Task...	Sym...	Sort Order
0	02010030	PlcTask	20	10000	350	851	0 (default...

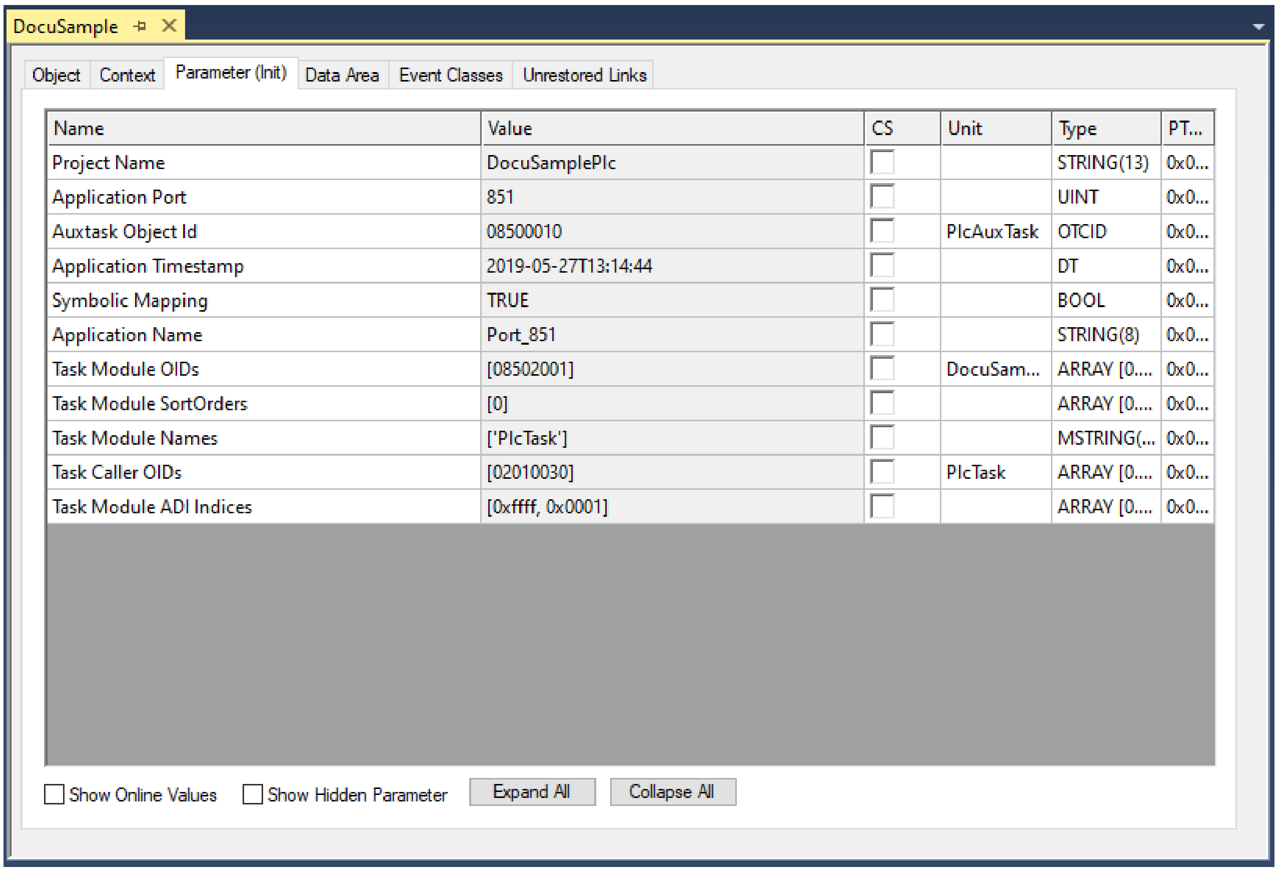
Context	Depending on the context selected here, the Data Area displays which **Data Areas** are updated by the selected context.
Depends On	Defines the dependencies of a context (task). For the PLC this is "Manual Config".
Data Areas	Existing Data Areas of the PLC Instance
Interfaces	Interfaces of a TcCOM module (irrelevant for a PLC instance)
Data Pointer	Data Pointer of a TcCOM module (irrelevant for a PLC instance)
Interface Pointer	Interface Pointer of a TcCOM module (irrelevant for a PLC instance)

The contexts are listed sorted by their ID in the table **Result**.

ID	Context ID of a task
Task	Here a TwinCAT task can be assigned to a Referenced Task in the PLC. (Happens automatically, but can be adjusted manually here)
Name	Name of the assigned TwinCAT tasks
Priority	Priority of the assigned TwinCAT tasks
Cycle Time (µs)	Cycle time of the assigned TwinCAT tasks
Task Port	Task Port of the assigned TwinCAT tasks
Symbol Port	Symbol Port of the assigned TwinCAT tasks
Sort Order	Sequence position at which the PLC is to be called by this task. The lower, the more likely the module will be called. This option is only needed if multiple runtime modules are called by this task. By default, modules are executed in the order they are added to a task.

5.4.3 Parameter (Init)

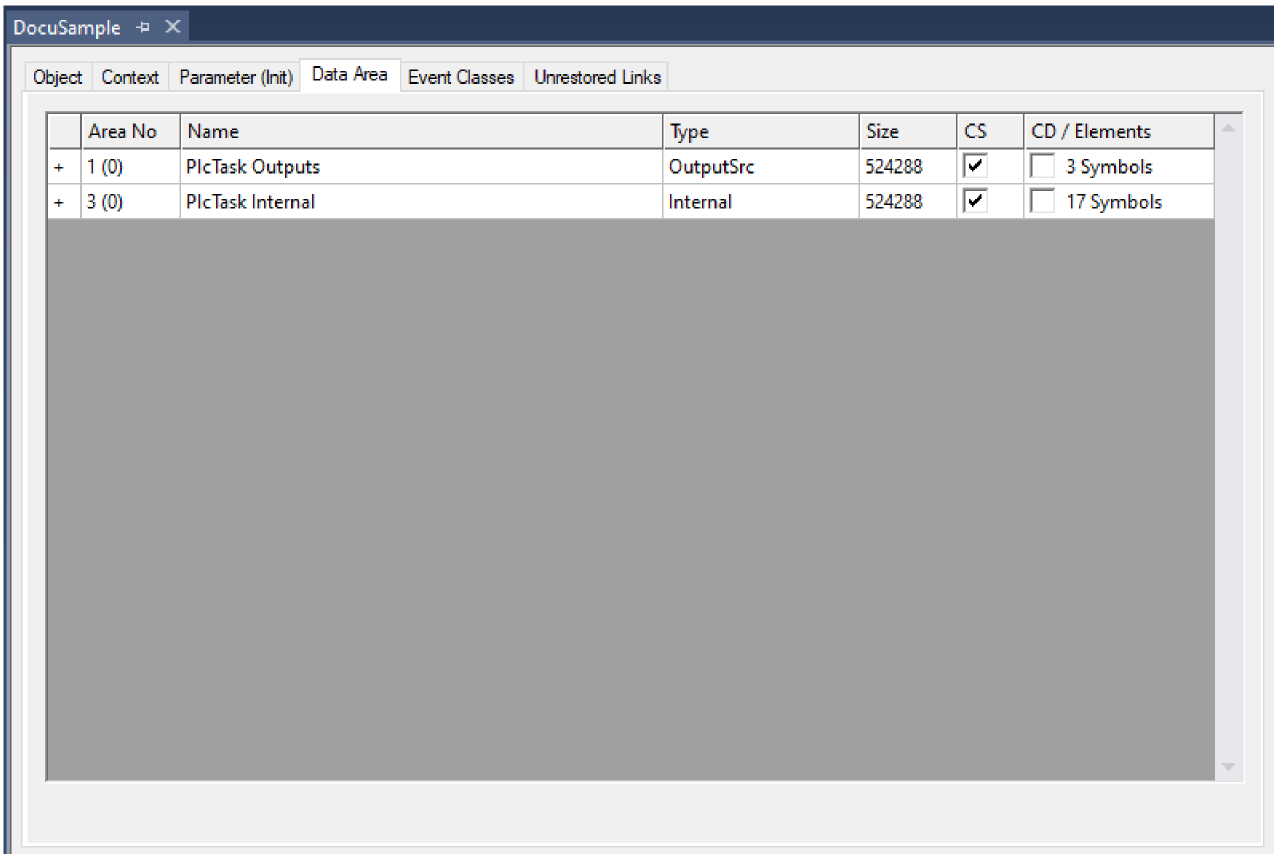
The **Parameter (Init)** tab shows the initialization parameters of the PLC runtime module. Checking the box in the column **CS** (Create Symbol) will create an ADS symbol for the selected parameter.



Name	Value	CS	Unit	Type	PT...
Project Name	DocuSamplePlc	☐		STRING(13)	0x0...
Application Port	851	☐		UINT	0x0...
Auxtask Object Id	08500010	☐	PlcAuxTask	OTCID	0x0...
Application Timestamp	2019-05-27T13:14:44	☐		DT	0x0...
Symbolic Mapping	TRUE	☐		BOOL	0x0...
Application Name	Port_851	☐		STRING(8)	0x0...
Task Module OIDs	[08502001]	☐	DocuSam...	ARRAY [0....	0x0...
Task Module SortOrders	[0]	☐		ARRAY [0....	0x0...
Task Module Names	['PlcTask']	☐		MSTRING(...	0x0...
Task Caller OIDs	[02010030]	☐	PlcTask	ARRAY [0....	0x0...
Task Module ADI Indices	[0xffff, 0x0001]	☐		ARRAY [0....	0x0...

5.4.4 Data Area

The **Data Area** tab shows the data areas of the PLC runtime module.



	Area No	Name	Type	Size	CS	CD / Elements
+	1 (0)	PlcTask Outputs	OutputSrc	524288	✓	☐ 3 Symbols
+	3 (0)	PlcTask Internal	Internal	524288	✓	☐ 17 Symbols

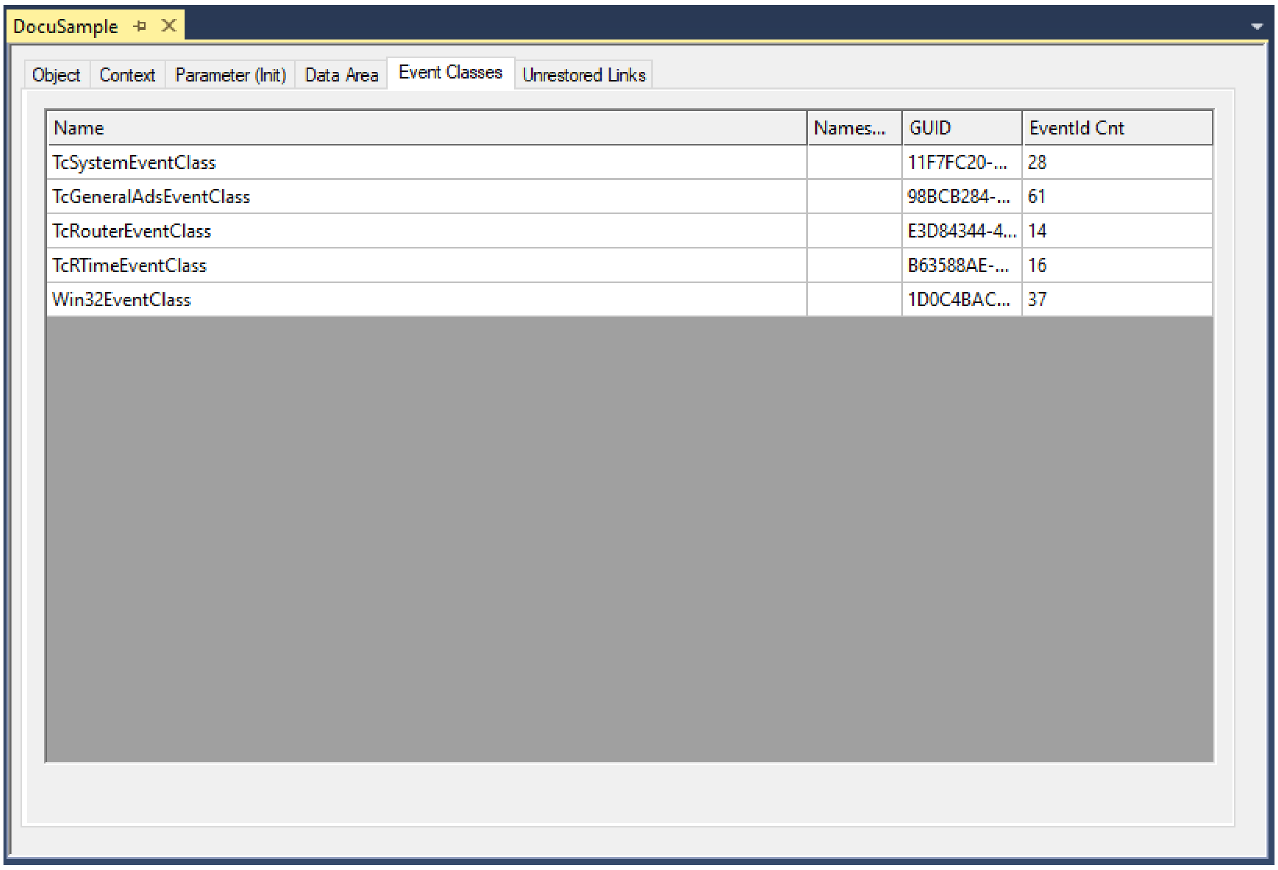
5.4.5 Symbol Initialization

All initialization parameters defined in the PLC project are displayed in a table in the **Symbol Initialization** tab. (See also Attribute 'TcInitSymbol'.)

5.4.6 Event Classes

Event classes are groups of events (possibly for a theme) and, in the sense of the TwinCAT type system, data types that can be used in different modules. For this reason they are created as data types in the TwinCAT type system (System > Type System > Event Classes).

All known event classes are listed on the Event Classes tab in the TwinCAT type system. The TMC editor, in which the event classes can be defined and edited, can be opened via the context menu commands **Edit** and **New**.

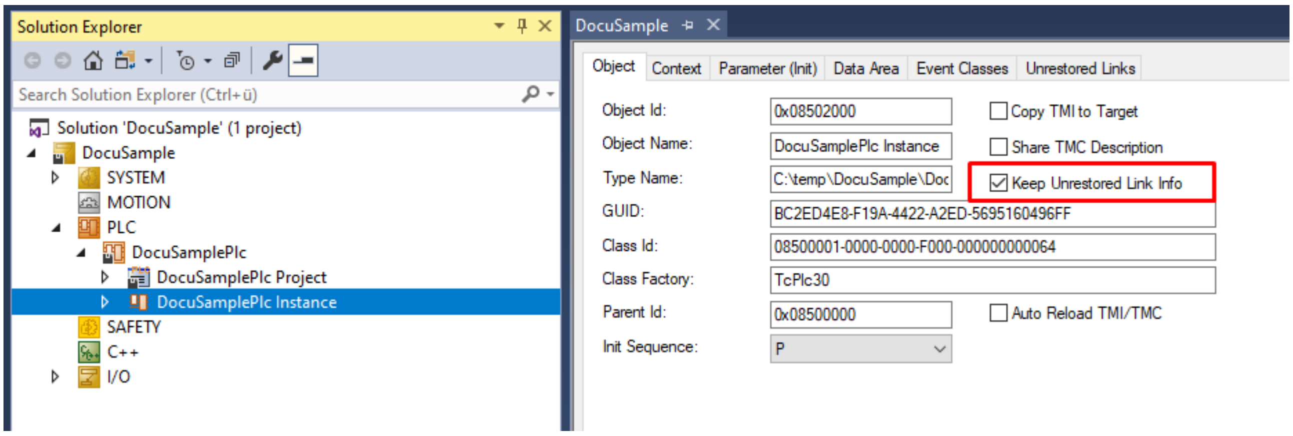


Name	Names...	GUID	EventId Cnt
TcSystemEventClass		11F7FC20-...	28
TcGeneralAdsEventClass		98BCB284-...	61
TcRouterEventClass		E3D84344-4...	14
TcRTimeEventClass		B63588AE-...	16
Win32EventClass		1D0C4BAC...	37

5.4.7 Unrestored Links

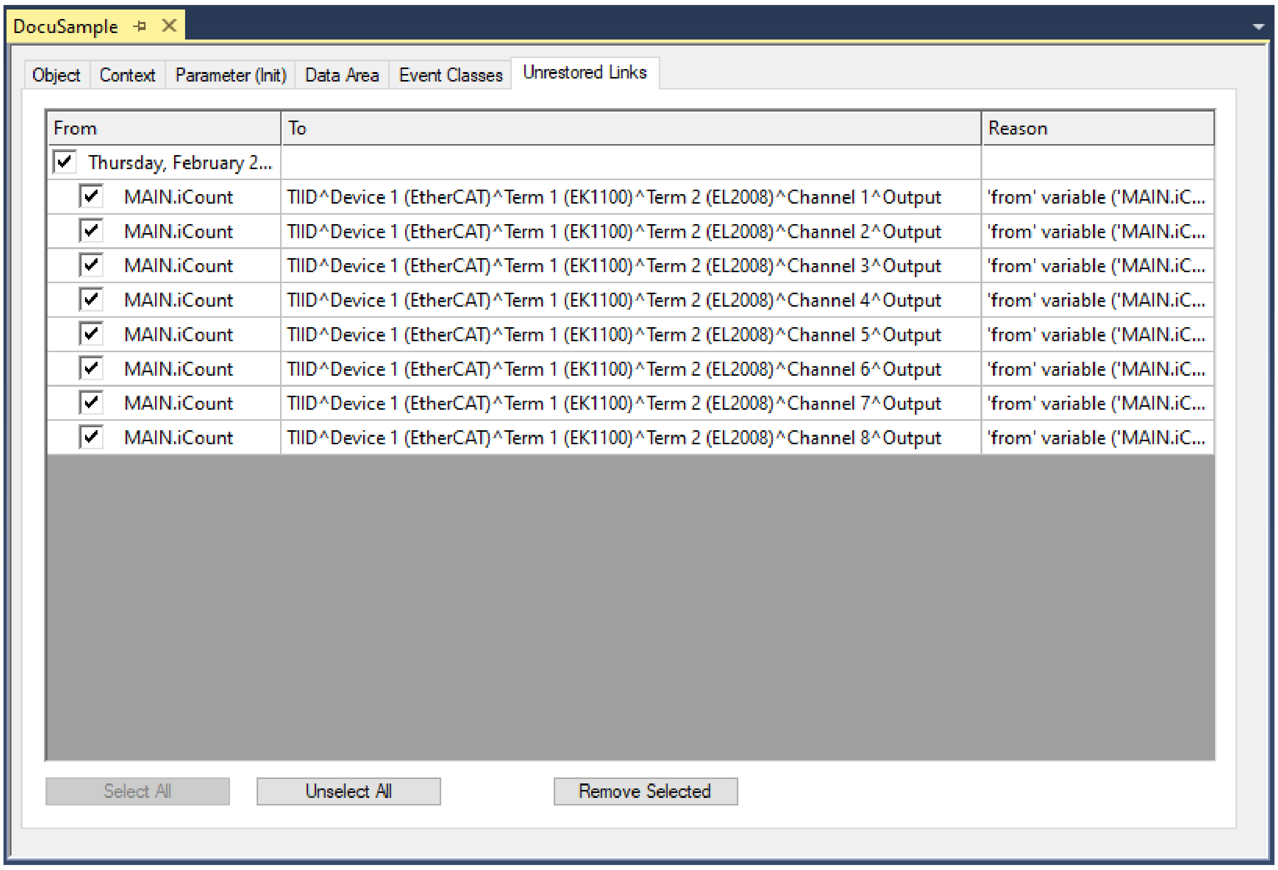
Keep Unrestored Link Info

If not set otherwise, TwinCAT only saves the required information so that the project repository is not unnecessarily loaded. In association with this, TwinCAT also checks whether links are still valid. Any invalid links found are automatically deleted. This mechanism is obstructive when merging projects, since only the code and the links can be merged, but the updated process image is only available after recompiling the code that has now been merged. It is thus possible for the link information to be newer than the process image and the automatic optimization function would delete all links to the new variables in the process image. With the option **Keep Unrestored Link Info**, the link information marked for deletion is retained and automatically restored as soon as the variables show up in the process image.



Object Id:	0x08502000	☐ Copy TMI to Target
Object Name:	DocuSamplePlc Instance	☐ Share TMC Description
Type Name:	C:\temp\DocuSample\Doc	☑ Keep Unrestored Link Info
GUID:	BC2ED4E8-F19A-4422-A2ED-5695160496FF	
Class Id:	08500001-0000-0000-F000-000000000064	
Class Factory:	TcPlc30	
Parent Id:	0x08500000	☐ Auto Reload TMI/TMC
Init Sequence:	P	

The unrestored links are displayed in the **Unrestored Links** tab. They are grouped there according to the date of "disappearance".



From	To	Reason
✓ Thursday, February 2...		
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 1^Output	'from' variable ('MAIN.iC...
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 2^Output	'from' variable ('MAIN.iC...
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 3^Output	'from' variable ('MAIN.iC...
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 4^Output	'from' variable ('MAIN.iC...
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 5^Output	'from' variable ('MAIN.iC...
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 6^Output	'from' variable ('MAIN.iC...
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 7^Output	'from' variable ('MAIN.iC...
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 8^Output	'from' variable ('MAIN.iC...

Delete unrestored links permanently

If unrestored links are not needed, they can be permanently deleted. Proceed as follows:

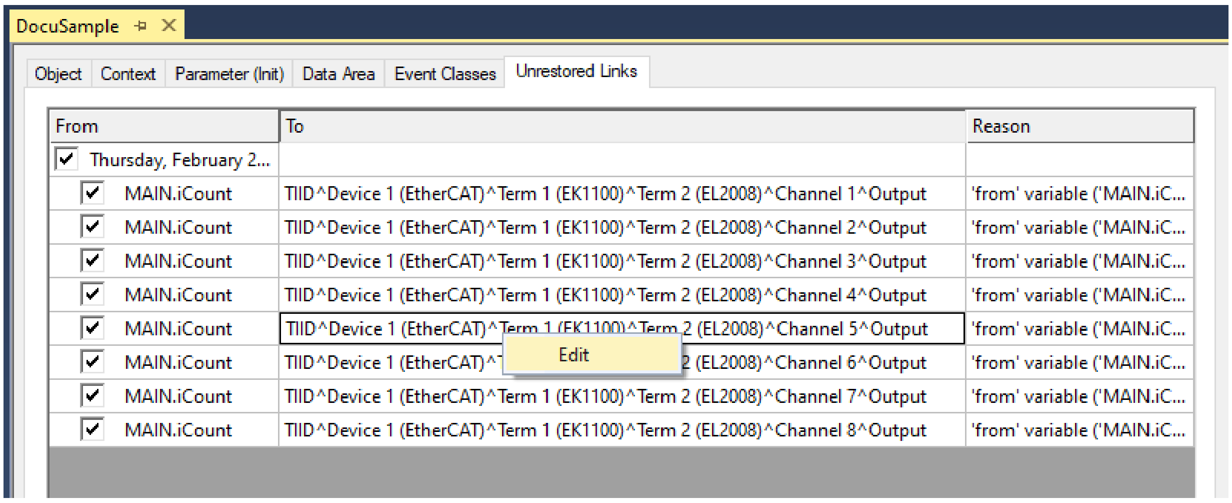
1. Select the links you want to delete.
2. This selection can be done individually by placing a check mark in front of the variable name.
3. You can also select a whole group by checking the box in front of the "Delete" date or select all deleted links by pressing the **Select All** button.
4. Use the **Remove Selected** button to delete the selected links.

⇨ The selected links will be permanently deleted.

Edit unrestored links

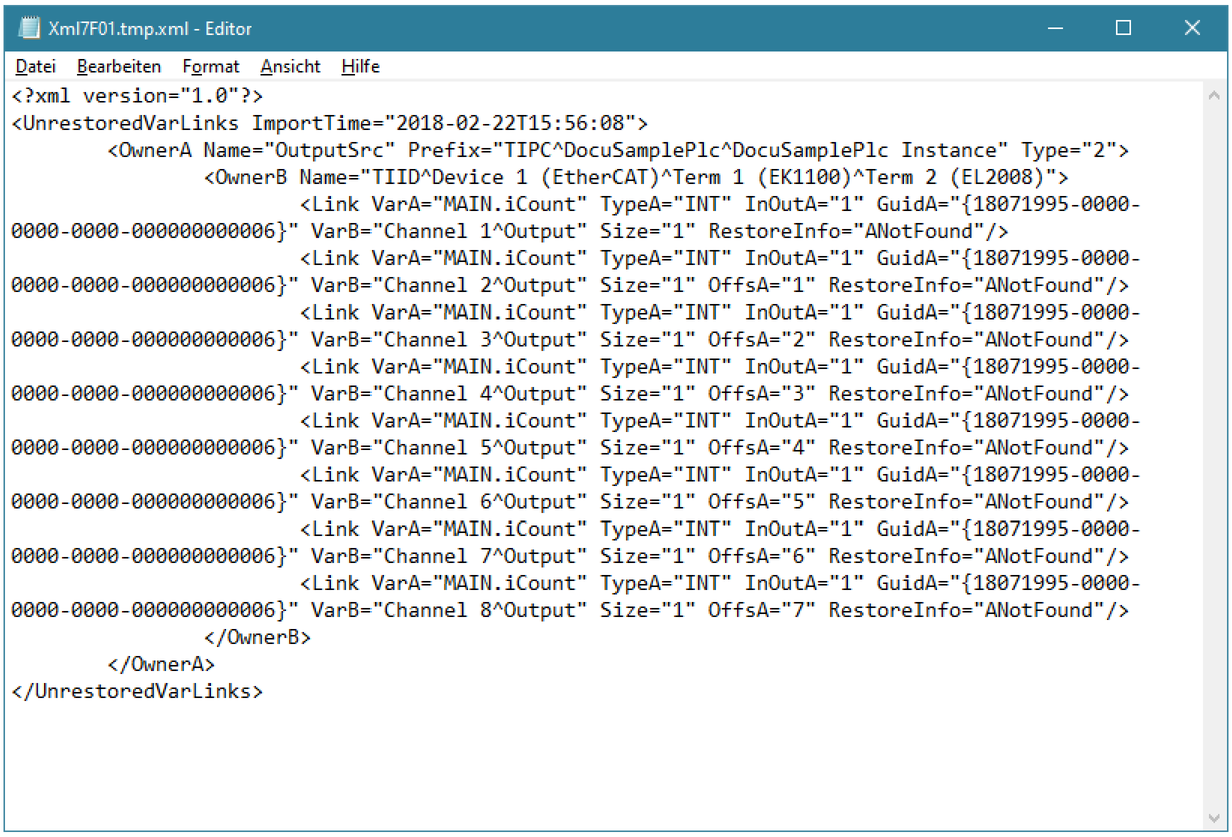
If the links could not be restored due to renaming, you can edit the unrestored links. Proceed as follows:

1. Use the context menu entry **Edit** to open the unrestored links in the Windows text editor.



From	To	Reason
✓ Thursday, February 2...		
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 1^Output	'from' variable ('MAIN.iC...
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 2^Output	'from' variable ('MAIN.iC...
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 3^Output	'from' variable ('MAIN.iC...
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 4^Output	'from' variable ('MAIN.iC...
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 5^Output	'from' variable ('MAIN.iC...
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^ 2 (EL2008)^Channel 6^Output	'from' variable ('MAIN.iC...
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 7^Output	'from' variable ('MAIN.iC...
✓ MAIN.iCount	TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)^Channel 8^Output	'from' variable ('MAIN.iC...

2. Use the commands **Search** and **Replace** to perform renaming operations within the file.



```
<?xml version="1.0"?>
<UnrestoredVarLinks ImportTime="2018-02-22T15:56:08">
        <OwnerA Name="OutputSrc" Prefix="TIPC^DocuSamplePlc^DocuSamplePlc Instance" Type="2">
                <OwnerB Name="TIID^Device 1 (EtherCAT)^Term 1 (EK1100)^Term 2 (EL2008)">
                        <Link VarA="MAIN.iCount" TypeA="INT" InOutA="1" GuidA="{18071995-0000-
0000-0000-000000000006}" VarB="Channel 1^Output" Size="1" RestoreInfo="ANotFound"/>
                        <Link VarA="MAIN.iCount" TypeA="INT" InOutA="1" GuidA="{18071995-0000-
0000-0000-000000000006}" VarB="Channel 2^Output" Size="1" OffsA="1" RestoreInfo="ANotFound"/>
                        <Link VarA="MAIN.iCount" TypeA="INT" InOutA="1" GuidA="{18071995-0000-
0000-0000-000000000006}" VarB="Channel 3^Output" Size="1" OffsA="2" RestoreInfo="ANotFound"/>
                        <Link VarA="MAIN.iCount" TypeA="INT" InOutA="1" GuidA="{18071995-0000-
0000-0000-000000000006}" VarB="Channel 4^Output" Size="1" OffsA="3" RestoreInfo="ANotFound"/>
                        <Link VarA="MAIN.iCount" TypeA="INT" InOutA="1" GuidA="{18071995-0000-
0000-0000-000000000006}" VarB="Channel 5^Output" Size="1" OffsA="4" RestoreInfo="ANotFound"/>
                        <Link VarA="MAIN.iCount" TypeA="INT" InOutA="1" GuidA="{18071995-0000-
0000-0000-000000000006}" VarB="Channel 6^Output" Size="1" OffsA="5" RestoreInfo="ANotFound"/>
                        <Link VarA="MAIN.iCount" TypeA="INT" InOutA="1" GuidA="{18071995-0000-
0000-0000-000000000006}" VarB="Channel 7^Output" Size="1" OffsA="6" RestoreInfo="ANotFound"/>
                        <Link VarA="MAIN.iCount" TypeA="INT" InOutA="1" GuidA="{18071995-0000-
0000-0000-000000000006}" VarB="Channel 8^Output" Size="1" OffsA="7" RestoreInfo="ANotFound"/>
                </OwnerB>
        </OwnerA>
</UnrestoredVarLinks>
```

3. Save the changes and close the text editor.

⇨ The links will be restored during the next build process.

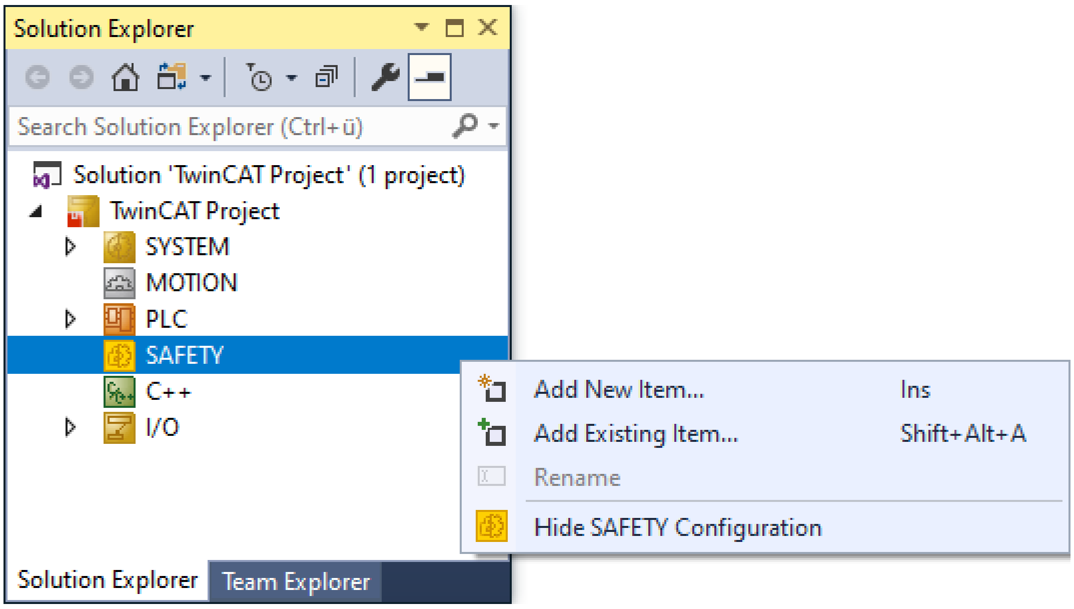
6 SAFETY

Function: The TwinCAT Safety PLC realizes a safety-related runtime environment on a standard industrial PC. Currently only Beckhoff IPCs can be used. Further information on permitted configurations can be found in the document "List of approved system configurations" on the Beckhoff website.

The safety-related logic can be created in Safety C, in future also via the graphical TwinSAFE Editor.

Call: Select **Add New Item...** in the context menu and create a new safety configuration.

Context menu


Solution Explorer
Search Solution Explorer (Ctrl+ü)
Solution 'TwinCAT Project' (1 project)
TwinCAT Project
SYSTEM
MOTION
PLC
SAFETY
C++
I/O
Add New Item... Ins
Add Existing Item... Shift+Alt+A
Rename
Hide SAFETY Configuration
Solution Explorer
Team Explorer

Add New Item…	Add new item.
Add Existing Item…	Adding an existing item.
Rename	Rename configuration.
Hide SAFETY Configuration	Hide configuration.

More information on the Safety topic can be found here:

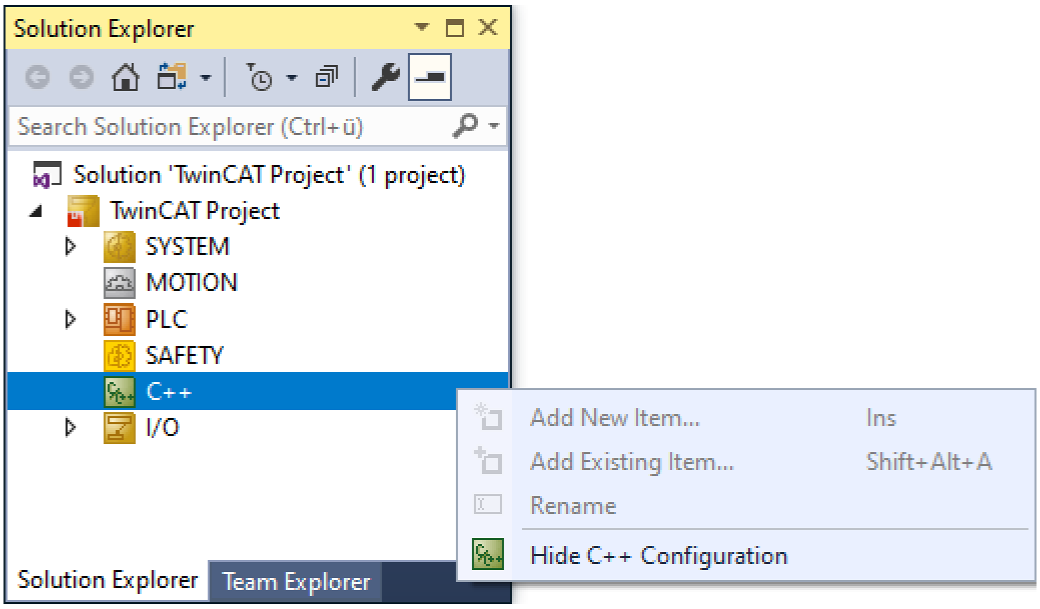
- TwinSAFE
- TwinCAT 3 Safety

7 C++

Function: TwinCAT 3 C++ implements a real-time execution of C++ code on an industrial PC. For programming, the widely used programming language C++ is supported, which is connected to the real-time via the TwinCAT SDK and CRT.

Call: Select **Add New Item...** in the context menu and create a new C++ configuration.

Context menu



Add New Item…	Add new item.
Add Existing Item…	Adding an existing item.
Rename	Rename configuration.
Hide C++ Configuration	Hide configuration.

Further information can be found in the Beckhoff Information System:

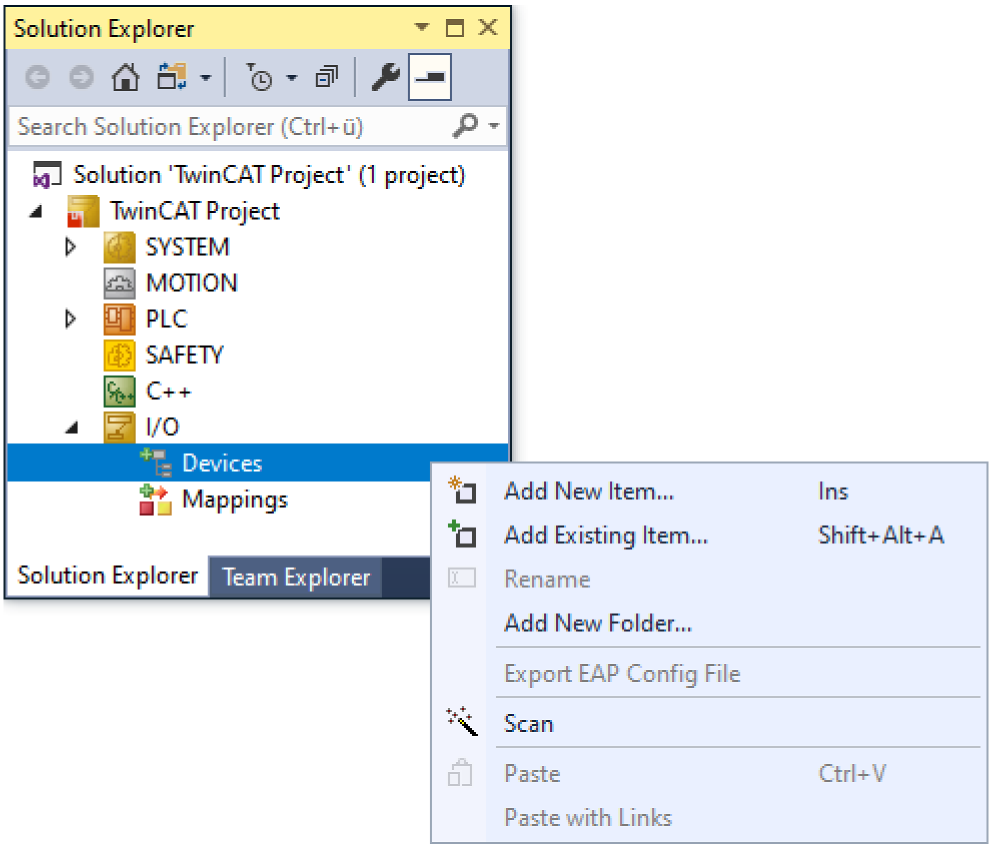
- TwinCAT 3 C/C++

8 I/O

Function: Using TwinCAT 3 I/O, cyclic data can be collected by different fieldbuses in process images. Cyclic tasks drive the corresponding fieldbuses. Various fieldbuses can be operated with different cycle times on one CPU. Applications can directly access the process image. The fieldbuses and the process images are configured in TwinCAT 3 Engineering.

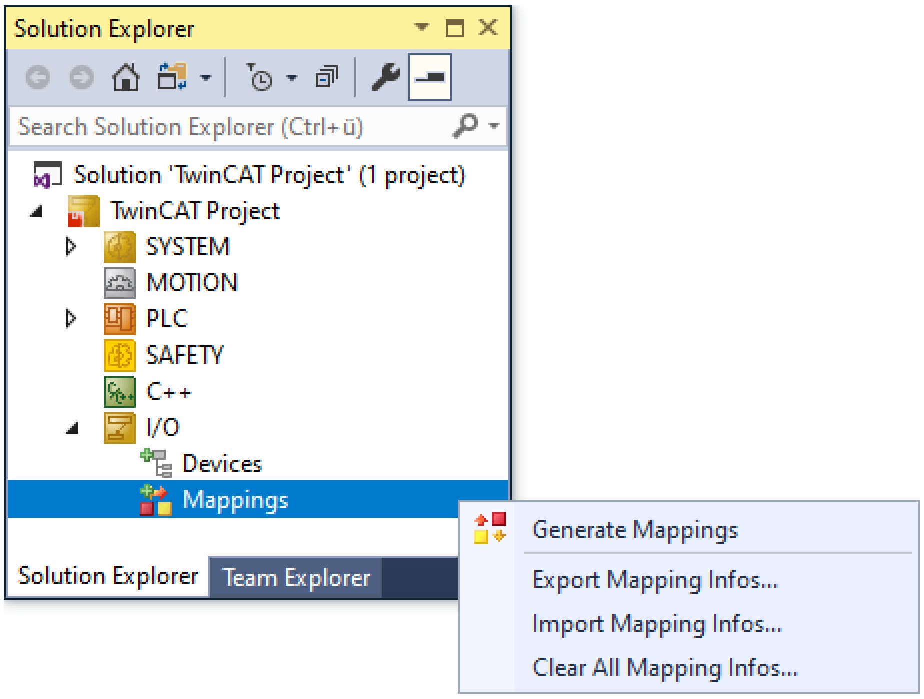
Call: Click on the arrow in front of **I/O**.

Devices context menu

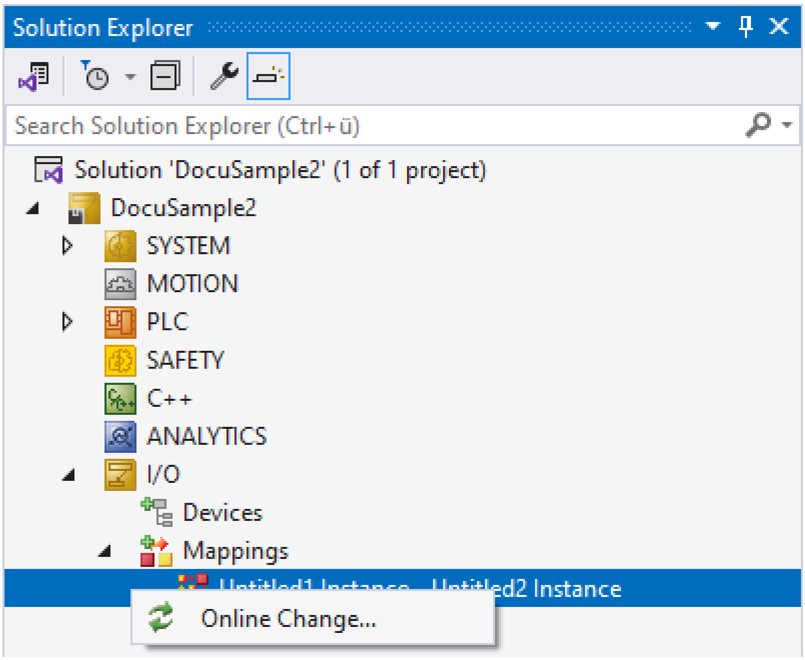


Add New Item…	Add new item.
Add Existing Item…	Adding an existing item.
Rename	Rename configuration.
Add New Folder…	Add new folder.
Export EAP Config File	Export the EtherCAT EAP configuration
Scan	Scan for devices.
Paste	Paste configuration.
Paste with Links	Paste configuration with links.

Mappings context menu



Generate Mappings	Generate mappings.
Export Mapping Infos…	Export mapping information.
Import Mapping Infos…	Import mapping information.
Clear All Mapping Infos…	Clear all mapping information.


Solution Explorer
Search Solution Explorer (Ctrl+ü)
Solution 'DocuSample2' (1 of 1 project)
DocuSample2
SYSTEM
MOTION
PLC
SAFETY
C++
ANALYTICS
I/O
Devices
Mappings
Untitled1 Instance - Untitled2 Instance
Online Change...

Online Change…	Online Change of mapping information is performed

An Online Change of the mapping changes the copy rule of the data. The Online Change can only be performed if the process images themselves do not change. Changing process images always requires reactivating the configuration. An exception to this is the process image of a PLC.

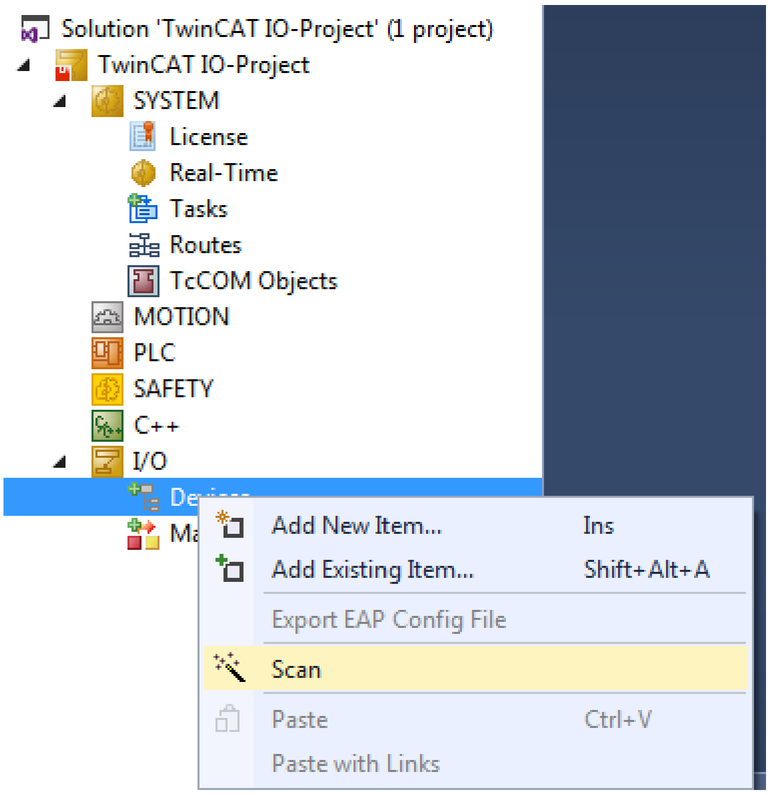
Further information can be found in the TwinCAT 3 I/O documentation.

8.1 Quick start

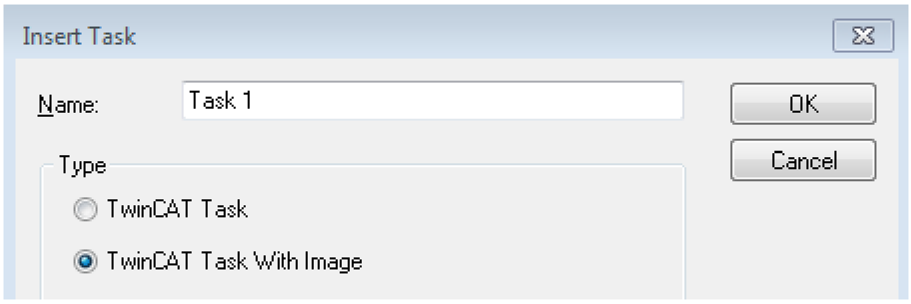
This guide explains the individual steps required to set up access by means of TwinCAT I/O to inputs/outputs that are provided by an appropriate I/O card.

✓ A created solution with TwinCAT 3 project.

1. Configure the hardware in the I/O area.
   When the target system is in Config mode, select the "Scan" mechanism:



2. Add a **TwinCAT Task With Image** via right click on System->Tasks:

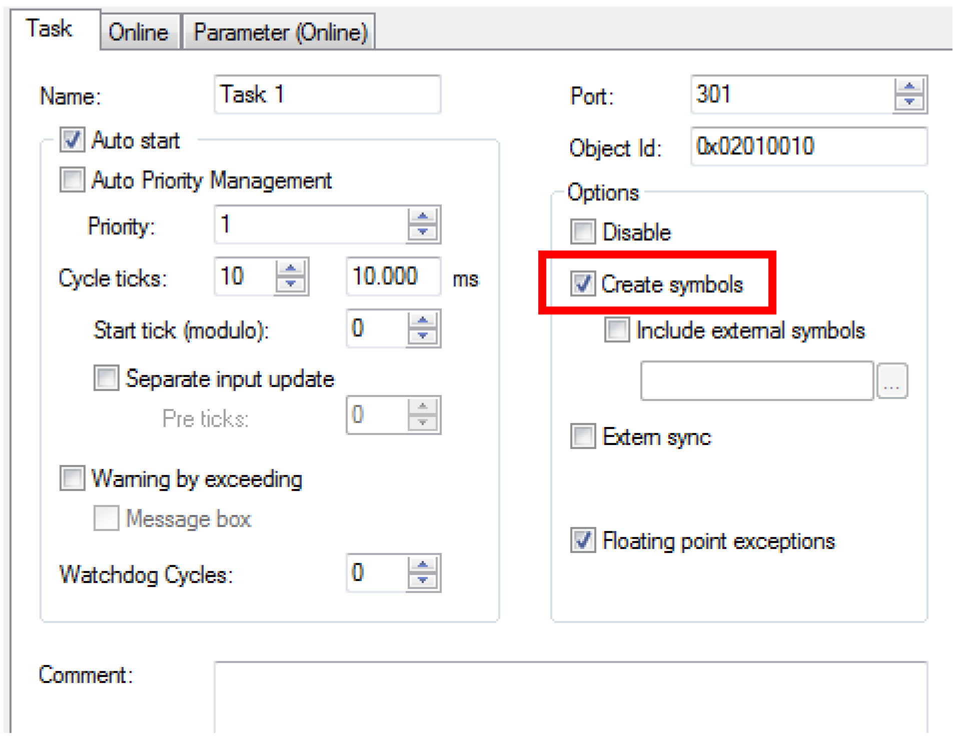

Insert Task

Name: Task 1

Type
TwinCAT Task
[✓ TwinCAT Task With Image]

⇨ The task contains a map for inputs and outputs.

3. Set the **Create Symbol** on the task so that symbolic access is possible:

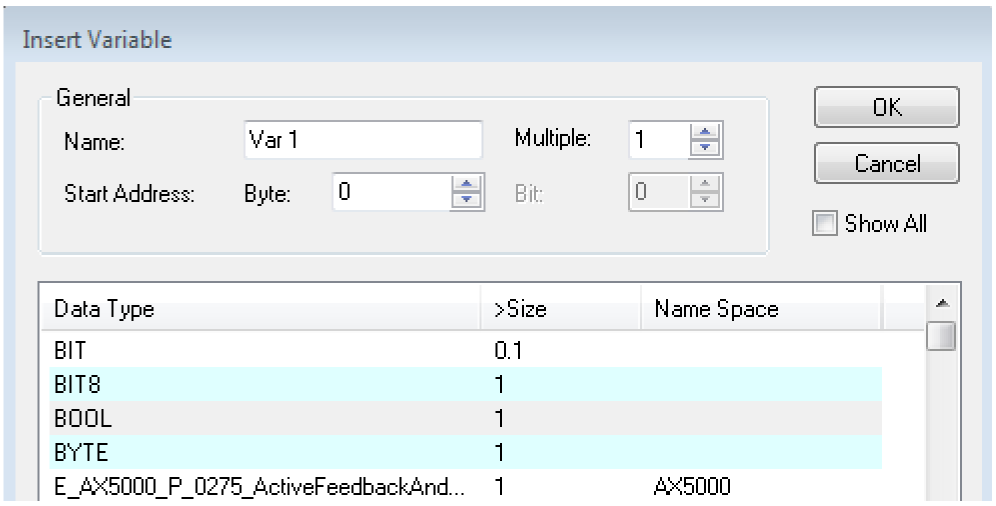


Name: Task 1
[✓ Auto start]
Auto Priority Management
Priority: 1
Cycle ticks: 10 10.000 ms
Start tick (modulo): 0
Separate input update
Pre ticks: 0
Warning by exceeding
Message box
Watchdog Cycles: 0

Port: 301
Object Id: 0x02010010
Options
Disable
[✓ Create symbols]
Include external symbols
Extern sync
[✓ Floating point exceptions]

Comment:

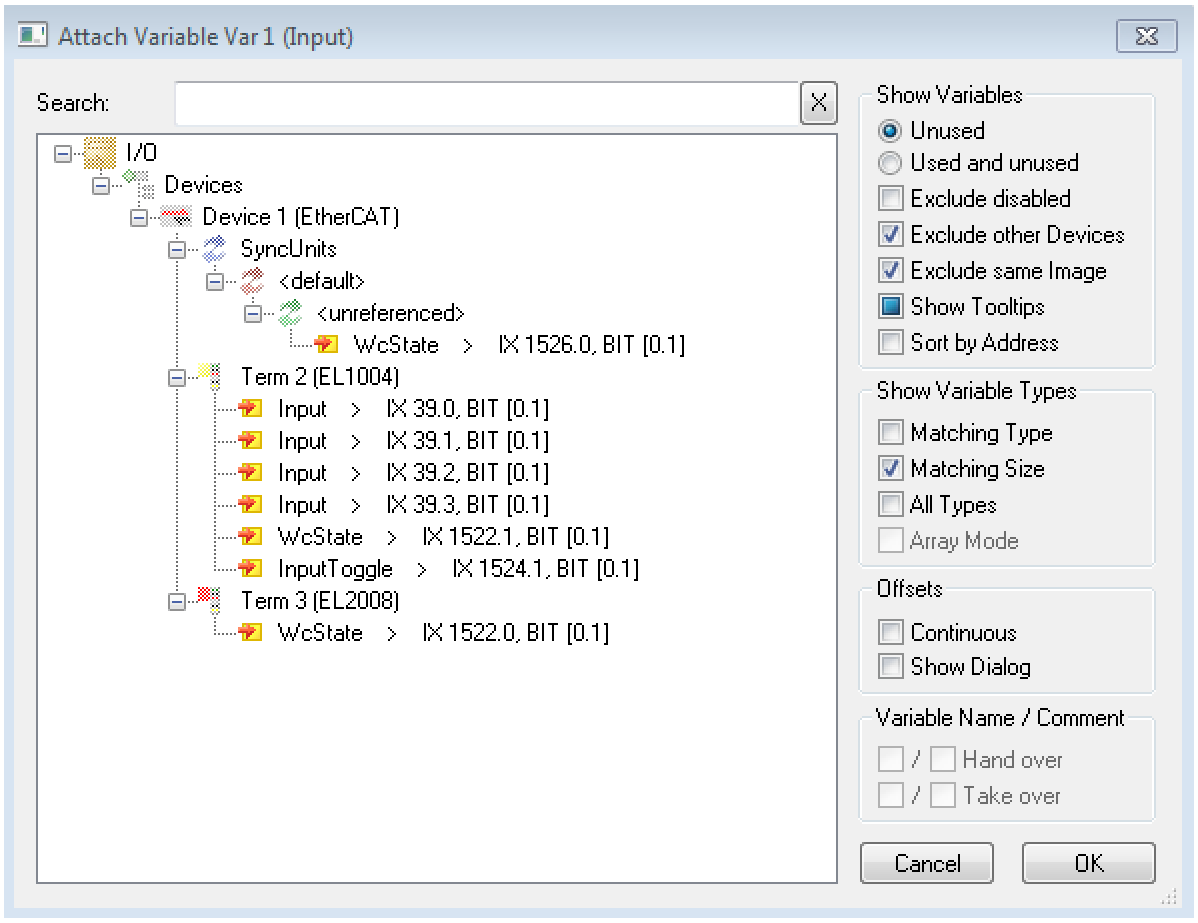
4. Create a symbol with **Add New Item** by right-clicking on Inputs or Outputs. To do this, select the appropriate type. Additionally, you can assign a name.



General
Name: Var 1
Multiple: 1
Start Address: Byte: 0
Bit: 0
Show All

Data Type	>Size	Name Space
BIT	0.1	
BIT8	1	
BOOL	1	
BYTE	1	
E_AX5000_P_0275_ActiveFeedbackAnd...	1	AX5000

5. Open the symbol that you have created by double-clicking and link a variable to the I/O with "Linked To...":



Attach Variable Var 1 (Input)

Search:

```
I/O
  Devices
    Device 1 (EtherCAT)
      SyncUnits
        <default>
          <unreferenced>
            WcState     >   IX 1526.0, BIT [0.1]
      Term 2 (EL1004)
        Input       >   IX 39.0, BIT [0.1]
        Input       >   IX 39.1, BIT [0.1]
        Input       >   IX 39.2, BIT [0.1]
        Input       >   IX 39.3, BIT [0.1]
        WcState     >   IX 1522.1, BIT [0.1]
        InputToggle >   IX 1524.1, BIT [0.1]
      Term 3 (EL2008)
        WcState     >   IX 1522.0, BIT [0.1]
```

Show Variables
- [✓ Unused]
- Used and unused
- ☐ Exclude disabled
- ☑ Exclude other Devices
- ☑ Exclude same Image
- ■ Show Tooltips
- ☐ Sort by Address

Show Variable Types
- ☐ Matching Type
- ☑ Matching Size
- ☐ All Types
- ☐ Array Mode

Offsets
- ☐ Continuous
- ☐ Show Dialog

Variable Name / Comment
- ☐ / ☐ Hand over
- ☐ / ☐ Take over

⇨ Start the system. The variables can be accessed by ADS.

⇨ The required symbol information can be read off from the symbols.

⇨ The "Target Browser" (from Scope -> Target Browser) can be used.

⇨ To do this you must configure the respective port for the display via the **Server Settings Dialog** in the target browser, depending on the ADS port used for the task.

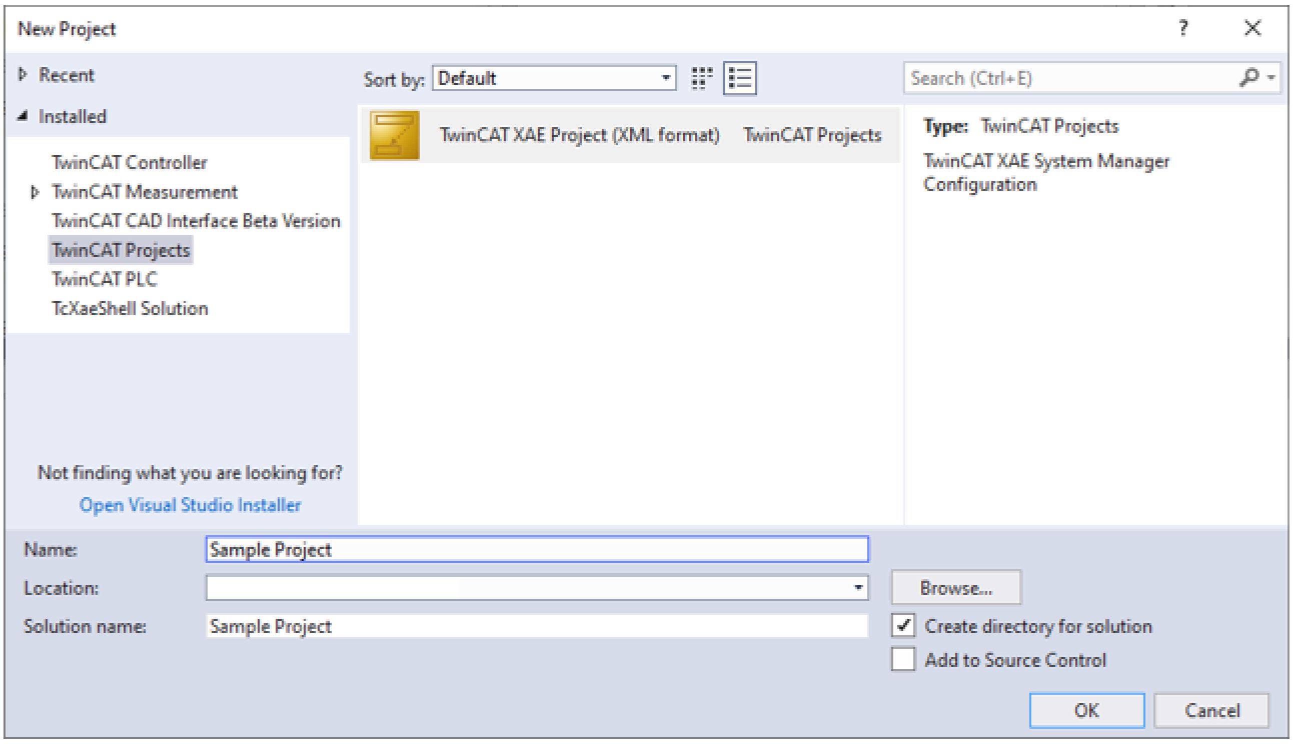
9 Renaming a project

Scenario

An existing project is to be reused for a new machine. For this purpose, the project is to be given the name of the new machine.

Challenge

When creating a project, you will be asked whether a Solution directory should be created. This option is the default behavior. If it is selected, the TwinCAT 3 project will be created in this directory.



Type: TwinCAT Projects
TwinCAT XAE System Manager Configuration

Name: Sample Project
Location:
Solution name: Sample Project
[✓] Create directory for solution
[ ] Add to Source Control

The Solution file (*.sln) for the project contains a reference to all subprojects. If a TwinCAT 3 project is renamed, only the project file is renamed, not the folder. Thus, after renaming the project, the old folder name of the TwinCAT 3 XAE project will still exist on the hard disk, even though it is not located in the solution.

Solution options

- Select generic folder names that have no reference to a project.
- Disable the option **Create directory for solution**, then the solution file is stored next to the project file of the TwinCAT 3 XAE project.
- If the Solution file only contains one TwinCAT 3 project, then the TwinCAT 3 project can be copied manually to another folder without the *.sln file and the folder in which the TwinCAT 3 project is saved. Double-clicking the *.tsproj file also opens the TwinCAT 3 XAE Shell. You can rename the project accordingly in the XAE Shell. When saving the project, you will automatically be asked whether a new Solution file should be created.
- Copy the entire directory and adapt the folder names of the subprojects manually. Subsequently, these folder names must also be manually swapped in the *.sln file.

More Information:
www.beckhoff.com/automation

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