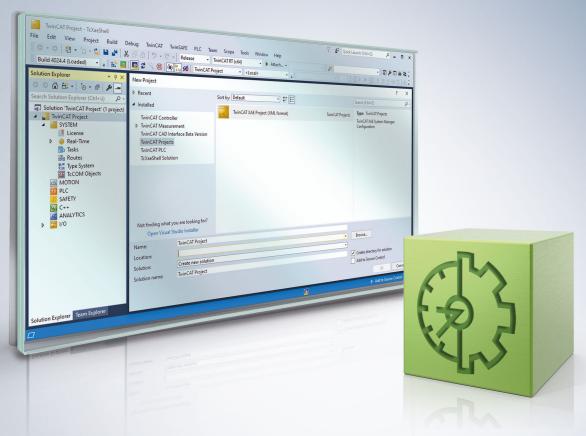
BECKHOFF New Automation Technology

Manual | EN TwinCAT 3 Type system



1 Foreword

1.1 Notes on the documentation

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

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

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

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

Disclaimer

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

We reserve the right to revise and change the documentation at any time and without prior announcement. No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

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

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

Patent Pending

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

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702 with corresponding applications or registrations in various other countries.

Ether CAT.

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

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

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

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of symbols

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

▲ DANGER

Serious risk of injury!

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

A WARNING

Risk of injury!

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

Personal injuries!

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

NOTE

Damage to the environment or devices

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



Tip or pointer

This symbol indicates information that contributes to better understanding.

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 <u>https://www.beckhoff.com/secquide</u>.

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 <u>https://www.beckhoff.com/secinfo</u>.

2 Overview

TwinCAT 3 provides a type system for the management of data types. The type system consists of system basic types and can be extended by custom data types through the customer project.

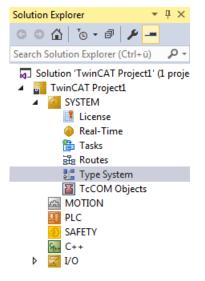
This documentation describes the TwinCAT 3 type system and the management of data types. The TMC editor, with which the data types are created and described, is described in the documentation entitled "C++" in the "TwinCAT Module Class Editor (TMC)" section.

3 Project-based type system

The TwinCAT 3 type system is project-specific; i.e. it is a fixed component of a TwinCAT 3 project in a Visual Studio solution.

Data types can be defined at various points and transferred if necessary to the TwinCAT 3 type system. Thus, local data types can also exist that don't exist in the TwinCAT 3 type system.

You will find the type system in the TwinCAT 3 project tree as an object in the SYSTEM subtree.



4 Data types

The TwinCAT 3 type system displays the data types in an editor on four different tabs. The editor is opened by double-clicking on the "Type System" object in the TwinCAT 3 project tree.

Solution Explorer 🔹 푸 ×	TwinCAT Project1 💠 🗙			
	Data Types Interfaces Functions Event C	lasses		
Search Solution Explorer (Ctrl+ü) Solution 'TwinCAT Project1' (1 proje TwinCAT Project1 SYSTEM License Real-Time Tasks	Name ADMSYNC_COPYINFO AdsAddInitCommand AMSADDR AMSHEAD AMSNETID	NS	GUID 18071995-00 F6F369BF-57 18071995-00 18071995-00 18071995-00	Size 32 40 8 32 6
語 Routes 計 Type System	EcNcTrafoParameter		D400B256-8F	4
TcCOM Objects	ETcIotMqttClientState ETHERNET_ADDRESS		DF915CC7-0 CC07E0A0-F	-
DLC SAFETY SAFCY	ETYPE_VLAN_HEADER INTERFACE_TYPE		478C4436-6F ACAD4AA7	-
▶ Z 1/0			F0F5BE0A-A	4

The following data types (TMC editor: "Specifications") are displayed on the Data Types tab:

- Alias: these data types are simply synonyms for other data types. For example, a time range (duration) can be defined in a specific project as UINT.
- Struct: these data types are structures of other data types, which in turn can also be structures.
- Enum: these data types describe enumerations.
- Array: these data types are arrays with a defined number of dimensions as well as the respective length.

The interfaces are displayed on the **Interfaces** tab. This data type describes an interface that can be provided or used by different components such as function blocks or TcCOM modules. An interface consists of methods that have a respective signature.

The **Functions** tab shows PLC functions and PLC function blocks whose definition was read from in a TMC/ TML file.

The Event Classes tab defines event classes that are used for the TwinCAT 3 EventLogger.

5 Handling of data types

In order to create or modify a data type via the TwinCAT 3 type system, select the **New** or **Edit** command from the context menu of the first table column on the appropriate tab of the type system editor. Both commands open the TMC editor in which you can edit the data type.

Data types from PLC projects

Data types (DUTs) can be created and saved in a PLC project. These data types initially exist locally in the PLC project and are not usable from the point of view of TwinCAT 3. If the data types are used in the input/ output memory map (%l* / %Q*), they are imported into the TwinCAT 3 type system so that they can also be linked through the mapping.

With the **Convert to Global Type** command in the context menu of a DUT in the PLC project tree you can transfer the DUT to the type system of the higher-level TwinCAT project. Thereafter the data type is usable in the PLC via the external types and is managed in the TwinCAT 3 type system.

To transfer a data type from the TwinCAT 3 type system to a PLC project, you can use the source code in the "Data Types" dialog.

Data types from C++ projects

In C++ projects the data types are defined in the TMC editor in parallel with the modules. Like the internal DUTs in the PLC project, these data types are local and thus invisible in the TwinCAT 3 type system.

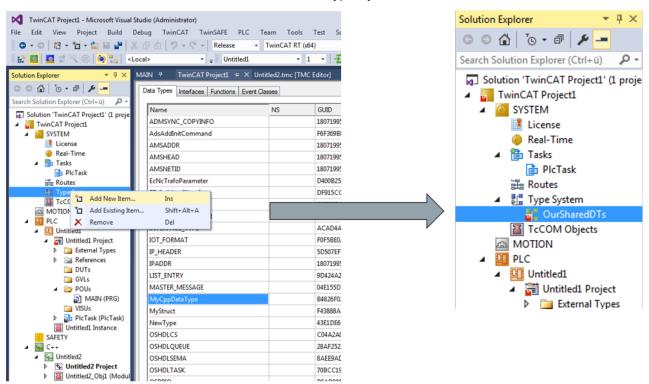
Through the use of the data types in a C++/Matlab module, which has also been instanced, the data types are inserted into the TwinCAT 3 type system.

You can also insert a data type into the TwinCAT 3 type system without using the data type in an instanced C++ module by activating the **Persistent (even if unused)** check box.

Use of data types in several projects

In some cases it may be useful to use data types in several projects. In particular for EAP/network variables it can be useful to use the same data type on both the publisher and subscriber side.

You can create individual TMC files for this under the "Type System" node.



A check box appears in front of every data type in the editor window of the TMC files. Using the check box you can specify which data type is to be deposited in the respective TMC file.

Solution Explorer 🛛 👻 무 🗙	MAIN 7 Tw	inCAT Project1 🗢 🗙 Untitled2.tmc [TM	C Editor]	Module1.cp	p Mo	odule1.h
◎ ◎ 🏠 🧿 + 🗗 🖊 💻	Data Types Inte	erfaces Functions Event Classes				
Search Solution Explorer (Ctrl+ü) 🛛 🔎 🗸		Name	NS	GUID	Size	Туре
Solution 'TwinCAT Project1' (1 proje TwinCAT Project1		EcNcTrafoParameter	145	D400B256-8		Enum
▲ SYSTEM		ETcIotMqttClientState		DF915CC7	4	Enum
License		ETYPE_VLAN_HEADER		478C4436-6	4	Alias
Real-Time		MyCppDataType		B4826F02-B	2	Alias
▲ 🎁 Tasks 💼 PicTask		MyStruct		F43888AD	12	Struct
a Routes		NewType		43E1DE6B-4	1	Alias
🔺 🚛 Type System		TcIotMqttQos		76342E07-4	1	Enum
UurSharedDTs		TcJsonLevelInfo		AD05131C	12	Struct, 64bit
TcCOM Objects		TcNcTrafoParameter		082AF37D	96	Struct, 64bit
		TCPIP_EVENT		7CECC506	4	Enum
Untitled1		E_AX5000_P_0275_ActiveFeedbackAnd	AX5000	A39361D3	1	Enum
Untitled1 Project		ST_AX5000_P_0275	AX5000	A0791761-1	2	Struct

The data types are additionally deposited in the TMC files. You can use these files on different computers and in different projects, for example, by means of file exchange or version control. However, the file itself must not be used by different projects at the same time, so that these are normally stored in the project directory and this project is then available as a copy on different computers, e.g. via version control.

Solution Explorer 🔹 👎 🗙	MAIN 7 TwinCAT Project1 -> X Module1.	cpp Module1.ł	1	
G O 🟠 🐻 - 🗗 🗡 🗕	"In the sectors from the law			
Search Solution Explorer (Ctrl+ü) 🛛 🔎 🝷	Computer Lokaler Datentr	räger (D:) N TwinCAT	Project1	▼ 4 ₂
👦 Solution 'TwinCAT Project1' (1 proje		lager (D.) V TWINCAT	Floject	• •7
 TwinCAT Project1 SYSTEM 	Organisieren 🔻 📄 Öffnen 🛛 Brennen	Neuer Ordner		
License	Name	Änderungsdatum	Тур	Größe
🧼 Real-Time	\mu TwinCAT Project1	2/3/2017 10:52 AM	Dateiordner	
▲ 🏥 Tasks i PicTask	OurSharedDTs.tmc	2/3/2017 11:39 AM	TMC-Datei	3 KB
Routes	sharedTMC.tmc	1/31/2017 8:41 AM	TMC-Datei	1 KB
🔺 📲 Type System	TwinCAT Project1.project.~u	2/3/2017 11:42 AM	~U-Datei	1 KB
📲 OurSharedDTs	😭 TwinCAT Project1.sdf	2/3/2017 10:53 AM	SQL Server Comp	320 KB
sharedTMCOuter TcCOM Objects	TwinCAT Project1.sln	2/3/2017 8:51 AM	Microsoft Visual S	3 KB

Since the GUID is used to identify the data type, the type system recognizes this double deposition automatically.

When using data types after they have been integrated in several projects, make sure that changes to the data types are made as far as possible only in one place. Otherwise the different variants can no longer be merged to a common version.

See also:

Management and identification of data types [11]

6

Management and identification of data types

Data types in the TwinCAT 3 type system are fundamentally identified on the basis of their GUID. Thus, several data types can exist with the same name. The same applies to different versions of a data type. Each version of a data type is assigned a new GUID.

NewType [TMC Editor] 😐 🗙 TwinC	T Project1							
C 🗚 💽 🔍								
▲ 5 TMC Module Classes ▷ ☆ Translations ▲ * Data Types ③ NewType	Edit the properties of the Data Type.							
- NewType	General properties							
	Name NewType							
	Guid {1d3c4c03-0ad9-46e5-b9f6-376c7c0404f6}							
	Specification Alias							
	Choose data type							
	Select BYTE							
	Type Information Namespace Guid {18071995-0000-0000-00000000001}							

At the same time, each data type has a list of data types that it keeps hidden ("Datatype Hides").

Datatype Hides
Guid
{1d3c4c03-0ad9-46e5-b9f6-376c7c0404f6}
{b976392c-ea80-4ea4-9080-b71da766f9d0
{5ab5cb59-ec49-4886-8027-48c425a3be7e

This makes it possible to use different versions of a data type in the project at the same time.

The **Update Instances...** command in the context menu of a data type in the type system editor (**Data Types** tab) employs the respectively latest version for selected uses of a data type.

TwinCAT has a so-called reference counter for each data type. This counter can be seen in the **RefCount** column in the editor of the type system. Each use of the data type in a project, and also in an editor and so on, increments the counter. If a counter is at 0, the data type is no longer used and is discarded.

Name	\sim 1	NS	GUID	Size	Туре	Unit	Comment	RefCount	FormatStr	Relations	Properties	
NCDRIVESTRUCT_TWOSP	ED I	MC	8334E88E-C9	40	Struct			[2]		0	0	
NCENCODERSTRUCT_IN2	1	MC	3B4978B1-9A	40	Struct			[2]		1	0	
NCENCODERSTRUCT_IN2	3 1	MC	901C2423-65	40	Struct			[2]		1	0	
NCENCODERSTRUCT_IN3	1	MC	9ED17BA1-B	80	Struct			[2]		0	0	
NCENCODERSTRUCT_OU	2 !	MC	9CC50AB2-5	40	Struct			[2]		1	0	
NCENCODERSTRUCT_OU	3 1	MC	4AA66E19-7	80	Struct			[2]		0	0	
NCTOPLC_AXIS_REF	1	MC	6A65C767-34	256	Struct			[2]		7	1	
NCTOPLC_AXIS_REF_OLD:	1	MC	BCDE0C45-A	256	Struct			[2]		7	1	
NCTOPLC_NCICHANNEL	REF I	MC	56354211-98	160	Struct			[2]		1	0	
NewType			12E1DE6B-41	1	Alias			(0)		0	0	
OSHDLCS Edit)4A2A0C-C	8	Alias, 64bit			[8]		0	0	
OSHDLQUEUE			AF2522-C	8	Alias, 64bit			[6]		0	0	
OSHDESEIVIA	Delete (if unuse		EE9AD4-0	8	Alias, 64bit			[6]		0	0	
	stent (even if uni	-	BCC193-B	8	Alias, 64bit			[3]		0	0	
OSPRIO Hide	types with same	name	AD8059-2	4	Alias			[2]		0	0	
OSQITEM Sear	ch References		44A185-6	8	Alias, 64bit			[5]		0	0	

If the **Persistent (even if unused)** setting in the context menu of a data type is activated, the data type description will be saved in the TwinCAT project file (*.tsproj) even if the data type is not used in the TwinCAT project. The setting is activated by default with data types that are newly created directly via the type system editor. This ensures that the data types are not directly deleted if the TwinCAT project is saved before the new data types are used.

If a SharedTMC is used underneath the **Type System** object in the TwinCAT project tree, the setting should not be activated for data types in this file as the data types are saved both in the project and in the SharedTMC. The setting is deactivated by default with data types that are newly created directly via a SharedTMC editor.

The **Auto Delete (if unused)** setting should not be manually changed, but is shown for the sake of completeness. Data types for which this setting is activated are hidden for PLC projects and cannot be used there. The setting should not be used, for example, to automatically clean the type system. Unused data types are not automatically saved in the TwinCAT project and are then no longer in the type system after reloading the TwinCAT project.

7 Alignment of data types

The memory layout of a data type is determined by the alignment. Further information on the alignment can be found in the "Alignment" section in the documentation entitled "PLC".

With the default alignment of 8 bytes it can be ensured that the access to data types functions optimally in terms of runtime and access on different platforms. Deviation from this should only take place in exceptional cases.

The TwinCAT 3 type system marks data types in color.

• Yellow if the length of the data type is not a multiple of the largest internal field (max. 8 bytes). As a result, the alignment no longer obeys the rules in the case of an array of such a data type.

Авалицинссоптини		10130301 371E 4000 0230 0	VT	JUNC
AlignmentMismatch		035500DD-FE07-4D6D-B195	10	Struct
DV KDUC CTATE	10	07000053 0D43 4044 0516 1		e .

• Red if the alignment within the data type no longer obeys the rules.

OUTIERS	1100000 2002 TOAN ALCO	-	- Cuius
OuterType	566B0D7D-3403-4C85-8BEC	6	Struct
DTIME CTATERIACC	2500007 4050 4045 0102	2	A 1*

The TMC editor offers the possibility to specify the memory layout of a data type for a selected alignment.

2	Add, remove and reorder Symbols.						
- 4			•				
Name	Specification	2 Byte	be S	Set data layout for selected symbols using current alignment (A	Alt+A)		
Value	Alias	4 Byte					
Status	Alias	8 Byte 16 Byte					
Data	Alias	32 Byte					

Alternatively, the layout can be manually specified using offsets.

TMC Module Classes	Optional symb	ol settings
 Data Types Modules CModule1 Implemented Interfaces 	Offset [Bits] Size [Bits]	x64 specific x64 specific
 Parameters Data Areas Inputs Symbols Value 	Unit Comment	x64 specific

If the size of a data type that is used in another data type is changed, then this data type must also be adjusted. The TMC editor offers an appropriate recursive function for this at the data type overview level.

2 😵 D	Module Classes ata Types OuterType	•	d, remove and reorder Data Types.
	Sub Items	i 🤹 –8 –	- 🖡 🕆 🔍 📲 2 Byte - 🚼 🛃 🖳
	Lo SubItem2	Name	Guid Set the data layout for selected and depending data types using current alignment (Alt+A)
A 👔	InnerType	OuterType	{ee2b5052-91b9-4d6a-aff5-4e53a8950ebf} Struct 6.0
✓ La Sub Items La SubItem1 La SubItem2	InnerType	{05e03cd4-5206-473b-9c8a-bb015d5fe9cd} Struct 4.0	

8 Files in connection with the type system

The TwinCAT 3 type system is formulated entirely in XML.

Depending on the field of application there are different files that contain the data types:

- .tsproj file TwinCAT project This file contains the entire TwinCAT project, including the complete TwinCAT 3 type system.
- .tmc files TwinCAT Module Class files
 These files are used to describe the TcCOM modules themselves. They include module class
 descriptions and the data types used. At the same time, these files are used to realize the exchange of
 data types between projects, as described above.
- .tmi files TwinCAT Module Instance files
 These files describe the instance of a class. They are deposited on the destination by the
 TwinCAT 3 Engineering in order to describe an instance of a class. In addition, instance information
 can also be transferred from one project to another using a .tmi file.

More Information: www.beckhoff.com/te1000

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

