BECKHOFF New Automation Technology

Manual | EN

COM Object

TwinCAT 2 | System

2022-07-19 | Version: 1.0

Table of contents

BECKHOFF

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.

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.

⚠ WARNING
Risk of injury!
Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION
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 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   General

The TwinCAT system can be started or stopped from a central location. This is usually done manually by the system startup engineer or developer, or from within the program due to external influences such as voltage failure or Windows shutdown. A COM object has been introduced within the TwinCAT system service, in order to enable greater influence of an application software upon the TwinCAT system.

Deployment

TwinCAT can be started or stopped via the TwinCAT system object. By using COM/DCOM, this can also happen via a network. Furthermore, the state (RUN or STOP) can be queried via this COM object, and COM events for signalling a start or stop of the TwinCAT system can also be triggered. Now it is possible, for example, for an application to be informed about a change in state of a remote TwinCAT system.

3 API

3.1 TcSystemServer (CoClass)

The COM interface to the TwinCAT system is implemented in the TcSystemServer class. In addition to ADS, TcSystemServer offers the opportunity to control the TwinCAT system via COM/DCOM. In contrast to ADS communication, COM offers the opportunity to start a stopped TwinCAT system, and to receive events relating to the TwinCAT start or stop. DCOM also makes all functions available via a network.

The default **TcSystemServer** interface is called ITcSystemServer [▶ 9]. **TcSystemServer** is implemented as an "out of process server" within the EXE file "TCATSysSrv.exe".

3.2 ITcSystemServer

The TwinCAT system can be controlled via the ITcSystemServer interface. ITcSystemServer has been designed as an OLE automation interface, so that it can also be used by script languages such as Visual Basic Script.

Methods in Vtable sequence

IUnknown methods	Description
QueryInterface	Returns a pointer to the interface requested.
AddRef	Increments the reference counter.
Release	Decrements the reference counter.
IDispatch methods	Description
GetTypeInfoCount	Supplies the number of "type information" interfaces, that an object offers (0 or 1).
GetTypeInfo	Gets the type information for an object.
GetIDsOfNames	Connects names of methods with optional arguments with a corresponding set of DISPIDs.
Invoke	Offers access to properties and methods of an object.

ITcSystemServer methods	Description
StartSystem [▶ 9]	Starts the TwinCAT system.
StopSystem [▶ 10]	Stops the TwinCAT system.
RestartSystem [▶ 10]	Stop with subsequent start of the TwinCAT system.
SaveServerCfg [▶ 11]	Requests all TwinCAT COM servers to save their configuration in the TwinCAT project file (*.tps).
SystemState [▶ 11]	Property displaying the current state of the TwinCAT system.
AmsNetId [▶ 11]	Property, with which the AmsNetId of a TwinCAT system can be queried.
RtcTimeDiff [▶ 12]	Property, with which the current time difference between the Windows NT system time and the real-time clock of the PC is displayed.
AdjustSystemTimeToRtcTime [▶ 12]	Synchronises the Windows NT system time with the real-time clock of the PC.

3.2.1 StartSystem

Method for starting of the TwinCAT system via COM/DCOM.

```
HRESULT StartSystem(
  [in,optional] VARIANT startParm,
);
```

Parameters

startParm   [in,optional]
                                        Optional start parameter which is currently not being evaluated.

Return Values

HRESULT == S_OK   TwinCAT start successful.
HRESULT == E_FAIL   Unable to execute start.

Comments

StartSystem starts the TwinCAT system on the computer on which the TcSystemServer object was instantiated. All messages are registered in the Windows events display of that computer.

3.2.2   StopSystem

Method for stopping the TwinCAT system via COM/DCOM.

```
HRESULT StartSystem(
  [in,optional] VARIANT stopParm,
);
```

Parameters

stopParm   [in,optional]
                                        Optional stop parameter which is currently not being evaluated.

Return Values

HRESULT == S_OK   TwinCAT halt successful.
HRESULT == E_FAIL   Error during TwinCAT system halt.

Comments

StopSystem stops the TwinCAT system on the computer on which the TcSystemServer object was instantiated. All messages are registered in the Windows events display of that computer. The TwinCAT system will be stopped, even if StopSystem returns an error.

3.2.3   RestartSystem

Method for stopping with subsequent start of the TwinCAT system via COM/DCOM.

```
HRESULT RestartSystem(
  [in,optional] VARIANT restParm,
);
```

Parameters

restParm   [in,optional]
                                        Optional parameter which is currently not being evaluated.

Return Values

HRESULT == S_OK   TwinCAT restart successful.
HRESULT == E_FAIL   Error during TwinCAT system restart.

Comments

RestartSystem stops and starts the TwinCAT system on the computer on which the TcSystemServer object was instantiated. All messages are registered in the Windows events display of that computer. If RestartSystem returns an error, the TwinCAT system in question will be in stop state.

3.2.4   SaveServerConfig

This method causes all COM/DCOM servers logged on to TwinCAT to save their configuration.

```
HRESULT SaveServerConfig(
  [in,optional] VARIANT saveParm,
);
```

Parameters

saveParm   [in,optional]
                            If only a certain COM server is required to save its data, the relevant class ID
                            can be specified as a BSTR via "saveParm".

Return Values

HRESULT == S_OK   Saving successful.
HRESULT != S_OK   Error during saving.

Comments

Depending on the configuration, calls of this method can be quite slow and should therefore only be executed if necessary.

3.2.5   SystemState

Property displaying the current state of the TwinCAT system.

```
HRESULT SystemState(
  [out,retval] long* pVal,
);
```

Parameters

pVal   [out, retval]
                            pVal shows the current state of the TwinCAT system. The currently
                            implemented states are ADSSTATE_RUN (5) and ADSSTATE_STOP (6).

Return Values

HRESULT == S_OK   Reading successful.
HRESULT != S_OK   Error during reading of the status.

Comments

Calling of **SystemState** returns the current state of the TwinCAT system.

3.2.6   AmsNetId

Property returning the AmsNetId of the TwinCAT system in question as a character string.

```
HRESULT AmsNetId(
  [out,retval] BSTR* pVal,
);
```

Parameters

pVal	[out, retval]
	pVal returns the 6 byte AmsNetId as a basic string (BSTR).

Return Values

HRESULT == S_OK	Reading successful.
HRESULT != S_OK	Error during reading of AmsNetId.

Comments

After AmsNetId has been called, AmsNetId is available as a character string. The memory for the character string is provided by the system and does not have to be allocated by the calling application. If this method is called from C or C++ without "wrapper" functions, the calling application is responsible for releasing the memory via **SysFreeString**.

3.2.7   RtcTimeDiff

Property, in which the current time difference (in seconds) between the Windows NT system time and the real-time clock of the PC is displayed.

```
HRESULT RtcTimeDiff(
  [out,retval] LONG* pVal,
);
```

Parameters

pVal	[out, retval]
	pVal returns the current time difference in seconds as a long value.

Return Values

HRESULT == S_OK	Reading successful.
HRESULT != S_OK	Error during reading of the time difference.

Comments

Calling of **RtcTimeDiff** causes the current real-time clock values (RTC or CMOS CLOCK) to be read and compared with the Windows NT system time. The difference is the NT system time minus the RTC time. Because the maximum daily deviation of the system time from the RTC time is 1 second, frequent calls of this function are not required.

3.2.8   AdjustSystemTimeToRtcTime

ITcSystemServer::AdjustSystemTimeToRtcTime

Method for adapting the Windows NT system time to the real-time clock of the PC.

```
HRESULT AdjustSystemTimeToRtcTime(
  [in,optional] VARIANT parm,
);
```

Parameters

parm	[in,optional]
	Optional boolean value which determines whether the TwinCAT system time should also be adjusted. Parm equals TRUE means the TwinCAT system time will be adjusted, FALSE or no parameter means that only the Windows

NT system time will be adjusted. If C++ is called, please note that the parameter transferred is of type VT_BOOL, and TRUE corresponds to the value 0xFFFF.

Return Values

HRESULT == S_OK   Adaptation executed successfully.
HRESULT == E_FAIL   Unable to execute start.

Comments

AdjustSystemTimeToRtcTime compares the real-time clock with the Windows NT system time and corrects the Windows NT system time by the difference. Time zones and summertime are considered. If the optional parameter contains TRUE, the TwinCAT system time will also be corrected. Please note that the correction can lead to time jumps during measurements or logbook entries.

3.3   _ITcSystemEvents

An application can sign on to the events listed below via the _ITcSystemEvents event interface. The event interface is implemented as an OLE connection point and can therefore easily be integrated with Visual Basic. Examples for the integration with other programming languages such as C++ are provided in the MSDN library.

Methods in Vtable sequence

IUnknown methods	Description
QueryInterface	Returns a pointer to the interface requested.
AddRef	Increments the reference counter.
Release	Decrements the reference counter.

IDispatch methods	Description
GetTypeInfoCount	Supplies the number of "type information" interfaces, that an object offers (0 or 1).
GetTypeInfo	Gets the type information for an object.
GetIDsOfNames	Connects names of methods with optional arguments with a corresponding set of DISPIDs.
Invoke	Offers access to properties and methods of an object.

Requirements

_ITcSystemEvents methods	Description
OnSystemStart [▶ 13]	Signals the starting of a TwinCAT system.
OnSystemStop [▶ 14]	Signals the stopping of a TwinCAT system.

3.3.1   OnSystemStart

Method which is called after the TwinCAT system has been started.

```
HRESULT OnSystemStart(
   [in,optional] VARIANT startParm,
);
```

Parameters

startParm   [in,optional]
                 Optional start parameter which is currently not being evaluated.

Return Values

The return values are not evaluated.

Comments

OnSystemStart is called after the successful start of TwinCAT. All TwinCAT components have been started once this event is being signaled.

3.3.2   OnSystemStop

Method which is called when the TwinCAT system is stopped.

```
HRESULT OnSystemStop(
  [in,optional] VARIANT stopParm,
);
```

Parameters

stopParm   [in,optional]
                Optional parameter which is currently not being evaluated.

Return Values

The return values are not evaluated.

Comments

OnSystemStop is called prior to the final halt of TwinCAT.

More Information:
www.beckhoff.com/automation