Manual

TC3 SMS/SMTP

TwinCAT

Version 1.0
Date 2015-06-18
Order No. TF6350

BECKHOFF

Table of contents

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

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.
For that reason the documentation is not in every case checked for consistency with performance data, standards or other characteristics.
In the event that it contains technical or editorial errors, we retain the right to make alterations at any time and without warning.
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®, EtherCAT®, Safety over EtherCAT®, TwinSAFE®, XFC®and XTS® 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, DE102004044764, DE102007017835
with corresponding applications or registrations in various other countries.

The TwinCAT Technology is covered, including but not limited to the following patent applications and patents:
EP0851348, US6167425 with corresponding applications or registrations in various other countries.

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

Copyright

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.

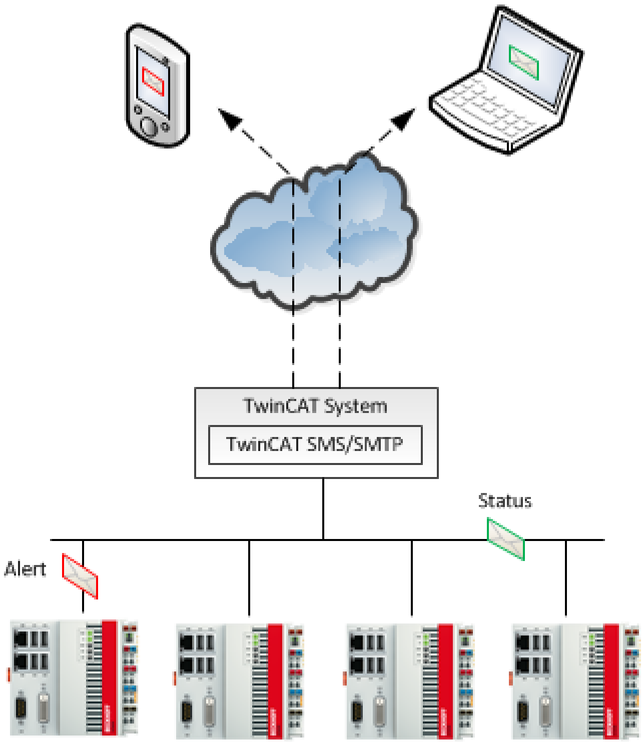
Description of symbols

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.
! Attention	**Damage to the environment or devices** Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.
i Note	**Tip or pointer** This symbol indicates information that contributes to better understanding.

2 Overview

The function provides the possibility to send alerts and stati directly from TwinCAT:



SMTP Server:

The TwinCAT SMTP Server is used to send E-Mail messages with TwinCAT.

TwinCAT SMTP Overview [▶ 17]

SMS Library:

The TwinCAT SMS Library is used to send SMS messages with TwinCAT via GSM modem.

TwinCAT SMS Overview [▶ 42]

3 Installation

3.1 System requirements

Technische Daten	TF6350 TwinCAT3 SMS/SMTP
Zielsystem	Windows NT/2000/XP/Vista/7/8 PC (x86-compatibel)
Min. TwinCAT-Version	3.0.0
Min. TwinCAT-Level	TC1200 TC3 | PLC

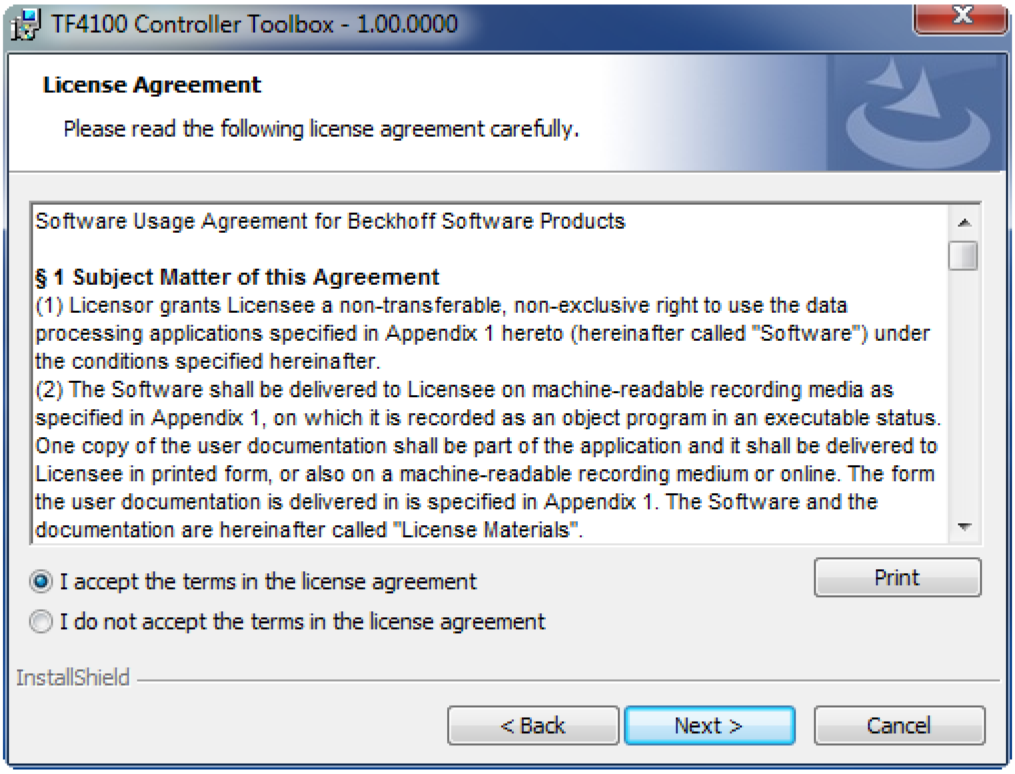
Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp, Tc2_Sms

3.2 Installation

Description of the installation procedure of a TwinCAT 3 Function for Windows-based operating Systems.

1. Double-click the downloaded setup file "TFxxxx".
   Please note: Under Windows 32-bit/64-bit, please start the installation with "Run as Administrator" by right-clicking the setup file and selecting the corresponding option in the context menu.
2. Click on "Next" and accept the license Agreement.



TF4100 Controller Toolbox - 1.00.0000

License Agreement

Please read the following license agreement carefully.

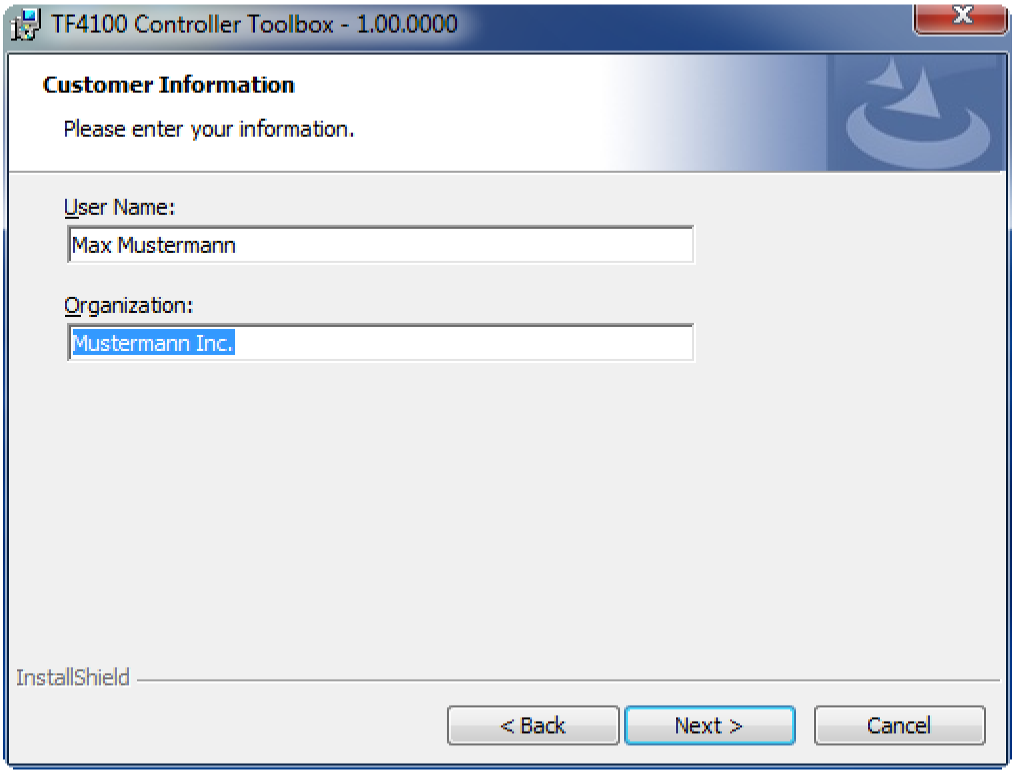
Software Usage Agreement for Beckhoff Software Products

§ 1 Subject Matter of this Agreement
(1) Licensor grants Licensee a non-transferable, non-exclusive right to use the data processing applications specified in Appendix 1 hereto (hereinafter called "Software") under the conditions specified hereinafter.
(2) The Software shall be delivered to Licensee on machine-readable recording media as specified in Appendix 1, on which it is recorded as an object program in an executable status. One copy of the user documentation shall be part of the application and it shall be delivered to Licensee in printed form, or also on a machine-readable recording medium or online. The form the user documentation is delivered in is specified in Appendix 1. The Software and the documentation are hereinafter called "License Materials".

● I accept the terms in the license agreement

○ I do not accept the terms in the license agreement

3. Enter your user information in the specified area.



TF4100 Controller Toolbox - 1.00.0000

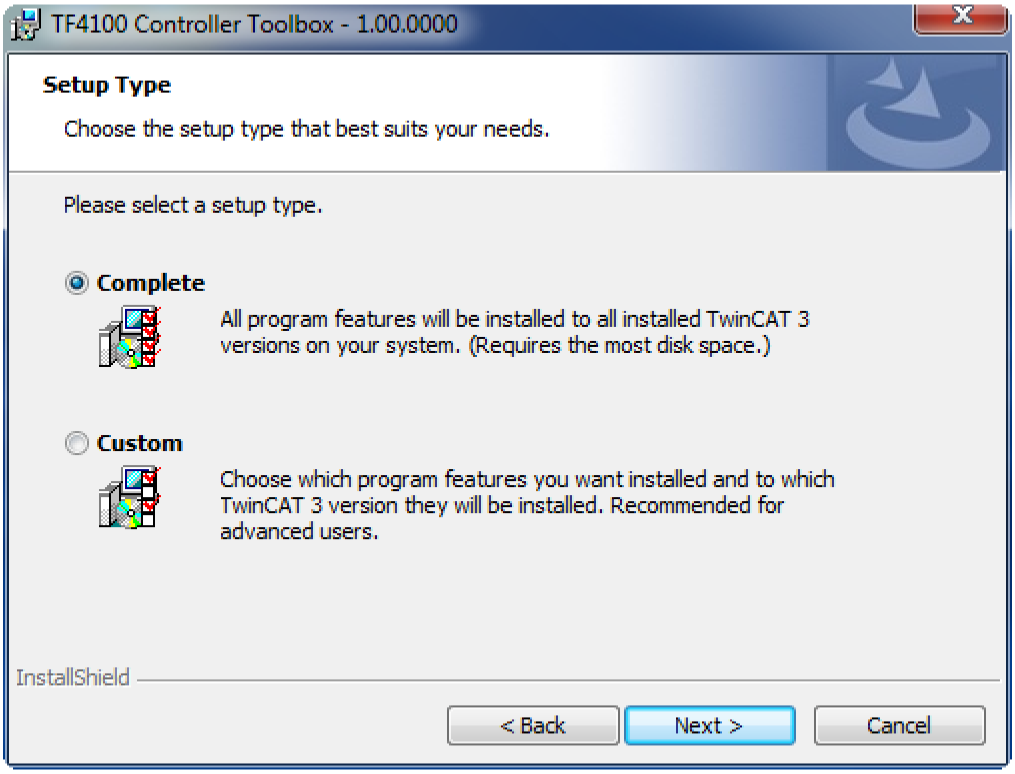
Customer Information

Please enter your information.

User Name:
Max Mustermann

Organization:
Mustermann Inc.

4. To install the full product, including all sub-components, please choose **"Complete"** as the Setup Type.Alternatively you can also install each component seperately by choosing **"Custom"**.



TF4100 Controller Toolbox - 1.00.0000

Setup Type

Choose the setup type that best suits your needs.

Please select a setup type.

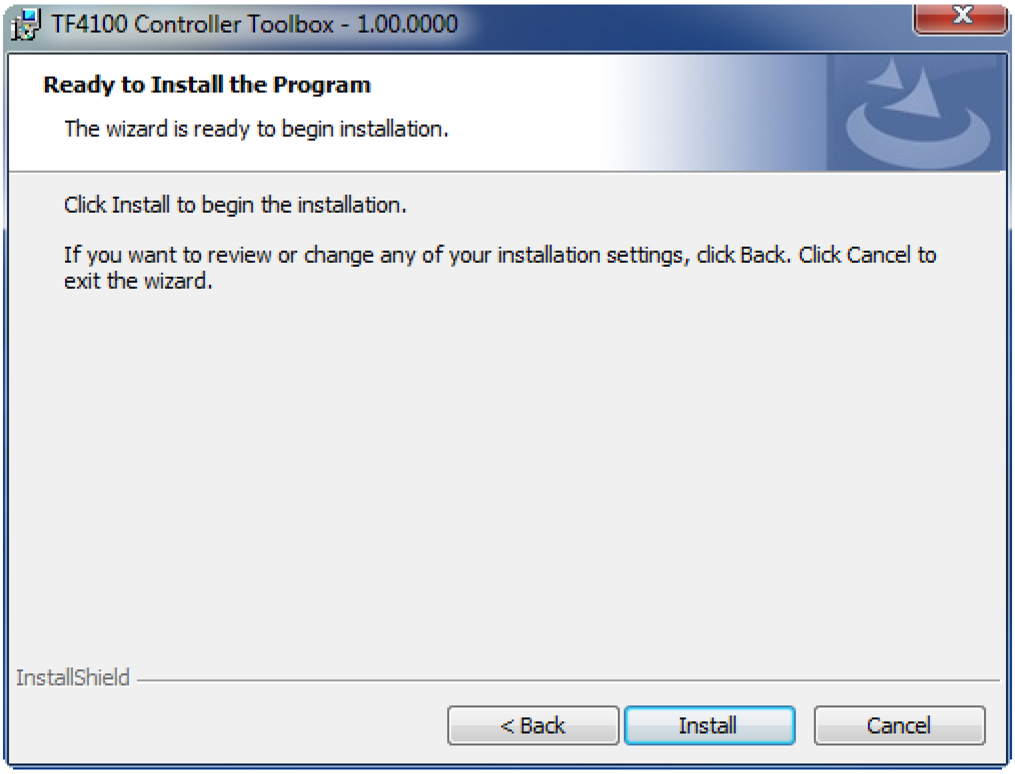
● **Complete**

All program features will be installed to all installed TwinCAT 3 versions on your system. (Requires the most disk space.)

○ **Custom**

Choose which program features you want installed and to which TwinCAT 3 version they will be installed. Recommended for advanced users.

5. Click on **"Install"**after pressing the **"Next"** to start the Installation.



TF4100 Controller Toolbox - 1.00.0000

Ready to Install the Program

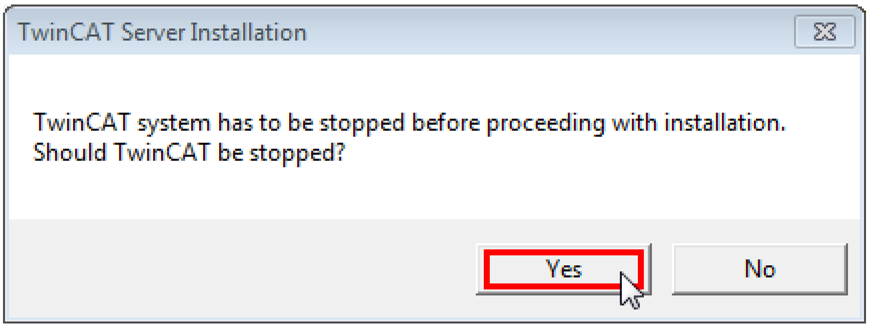
The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

The TwinCAT system has to be stopped before proceeding with installation

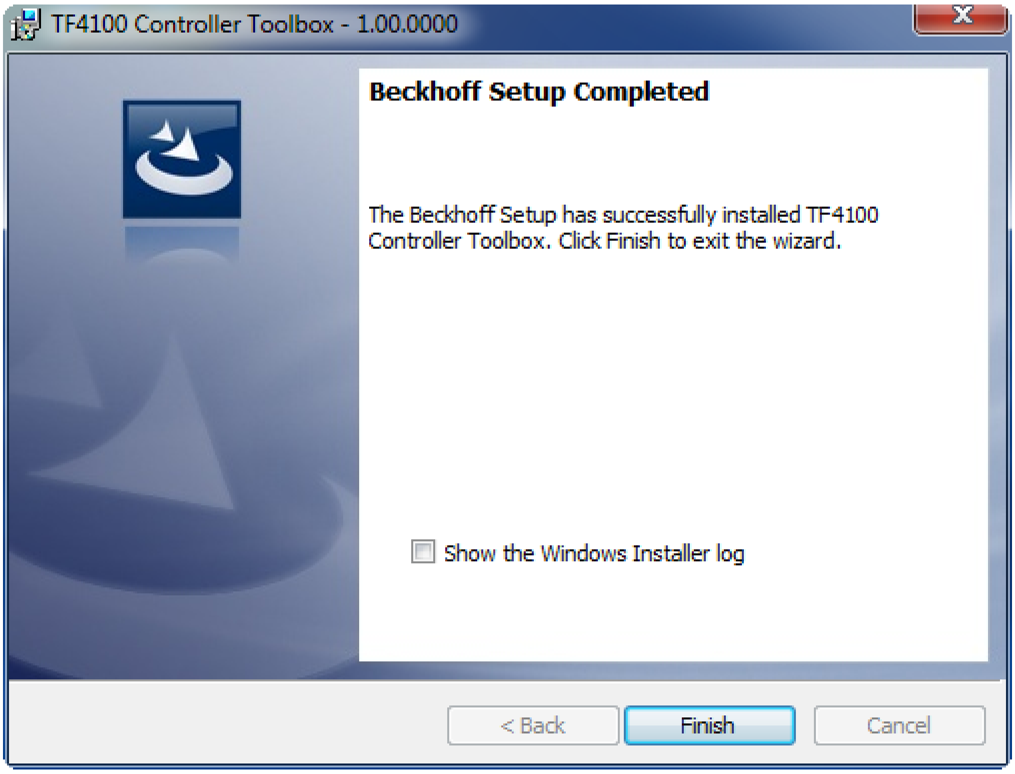
6. Confirm the Dialog with **"Yes"**.



TwinCAT Server Installation

TwinCAT system has to be stopped before proceeding with installation.
Should TwinCAT be stopped?

7. Select **"Finish"** to end the installation process.



TF4100 Controller Toolbox - 1.00.0000

Beckhoff Setup Completed

The Beckhoff Setup has successfully installed TF4100 Controller Toolbox. Click Finish to exit the wizard.

□ Show the Windows Installer log

⇨ The installation is complete now.

After a successful installation the TC 3Function needs to be licensed [▸ 12].

3.3 Installation Windows CE

This part of the documentation describes, how you can install the TwinCAT 3 Function TF6350 SMS SMTP on a Beckhoff Embedded PC Controller based on Windows CE.

The setup process consists of four steps:

- Downloading the setup file
- Installation on a host computer
- Transfering the executable to the Windows CE deviceSoftware installation
- Software upgrade

Hint: The Installation is only necessary to send mails via TwinCAT SMTP.

Downloading the setup file

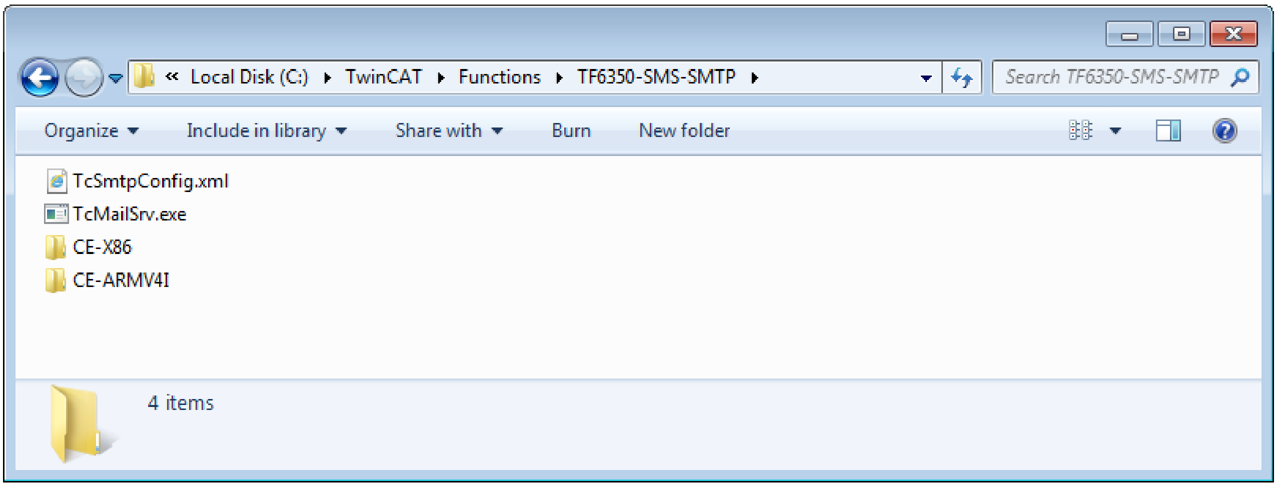
The CAB installation files for Windows CE are part of the TF6350 SMS SMTP setup. Therefore you only need to download one setup file from www.beckhoff.com which contains binaries for Windows XP, Windows 7 and Windows CE (x86 and ARM).

The installation procedure of the TF6350 SMS SMTP setup is described in the regular installation article.

Installation on a host computer

After installation, the install folder (e.g. C:\TwinCAT\Functions\TF6350-SMS-SMTP) contains three directories - each one for a different hardware platform:

- **CE-ARM:** ARM-based Embedded Controllers running Windows CE, e.g. CX8090, CX9020
- **CE-X86:** X86-based Embedded Controllers running Windows CE, e.g. CX50xx. CX20x0
- **Win32:** Embedded Controllers running Windows XP, Windows 7 or Windows Embedded Standard



TcSmtpConfig.xml
TcMailSrv.exe
CE-X86
CE-ARMV4I

4 items

The CE-ARM and CE-X86 folders contain the TF6350 CAB-File for Windows CE - corresponding to the hardware platform of your Windows CE device. This file needs to be transfered to the Windows CE device, see next chapter.

Transfering the executable to the Windows CE device

Transfer the corresponding CAB-File to you Windows CE device. This can be done via one of the following ways:

- via a Shared Folder
- via the integrated FTP-Server
- via a USB Stick, CF card or SD-Card

Software installation

After the file has been transfered via one of the above methods, you need to execute the file and acknowledge the following dialog with "Ok". Restart your Windows CE device after the installation has finished.

After the restart has been completed, the TF6350 SMTP server and configuration will be automatically started in background and is now available to use.

The software will be installed in the following directory on the CE device: **\Hard Disk\TwinCAT\Functions\TF6350-SMS-SMTP**

Software upgrade

If you already have a version of TF6350 installed on your Windows CE device, you need to perform the following steps on the Windows CE device to upgrade to a newer version:

- Open the CE Explorer by clicking on Start --> Run and entering "explorer"
- Navigate to \Hard Disk\TwinCAT\Functions\TF6350-SMS-SMTP\Server
- Rename TcMailSrv.exe
- Restart the Windows CE device
- Transfer the new CAB-File to the CE device
- Execute the CAB-File and install the new version of TF6350
- Delete the old (renamed) files
- Restart the Windows CE device

After the restart is complete, the new version is active.

After a successful installation the TC3 Function needs to be licensed. [▸ 12]

Also see about this

🗎 Licensing [▷ 12]

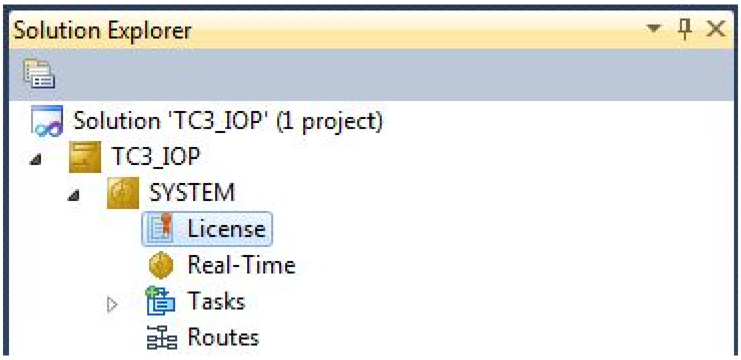
3.4 Licensing

The TwinCAT 3 functions are available both as a full and as a 7-Day trial version. Both license types can be activated via TwinCAT XAE.For more information about TwinCAT 3 licensing, please consult the TwinCAT 3 Help System.The following document describes both licensing scenarios for a TwinCAT 3 function on TwinCAT 3 and is divided into the following sections:

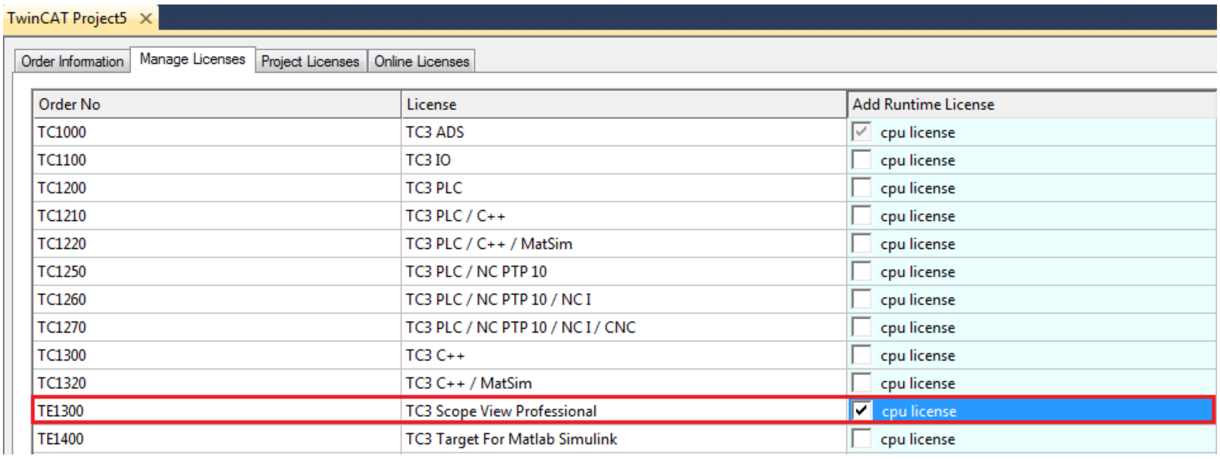
- Licensing a 7-Day trial version [▷ 12]
- Licensing a full version [▷ 13]

Licensing a 7-Day trial version

1. Start TwinCAT XAE
2. Open an existing TwinCAT 3 project or create a new project
3. In “Solution Explorer”, please navigate to the entry **“System\License”**

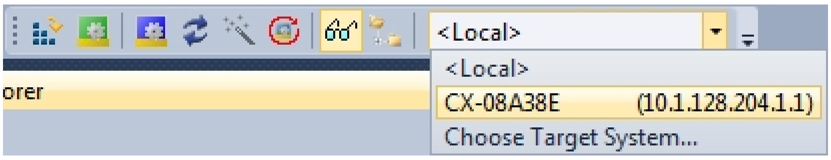


4. Open the tab **"Manage Licenses"** and add a **"Runtime License"** for your product (in this screenshot “TE1300: TC3 Scope View Professional”)

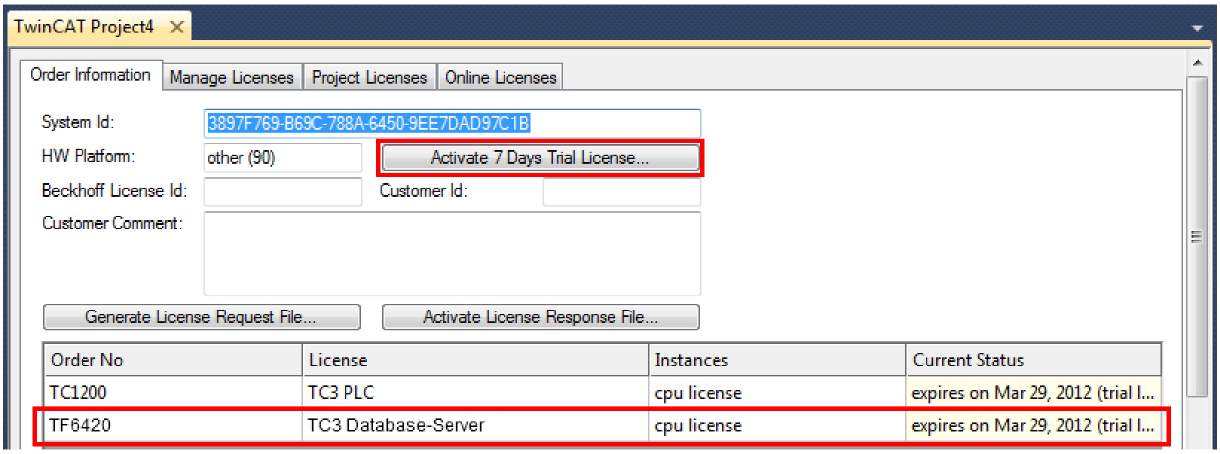


Order No	License	Add Runtime License
TC1000	TC3 ADS	☑ cpu license
TC1100	TC3 IO	□ cpu license
TC1200	TC3 PLC	□ cpu license
TC1210	TC3 PLC / C++	□ cpu license
TC1220	TC3 PLC / C++ / MatSim	□ cpu license
TC1250	TC3 PLC / NC PTP 10	□ cpu license
TC1260	TC3 PLC / NC PTP 10 / NC I	□ cpu license
TC1270	TC3 PLC / NC PTP 10 / NC I / CNC	□ cpu license
TC1300	TC3 C++	□ cpu license
TC1320	TC3 C++ / MatSim	□ cpu license
TE1300	TC3 Scope View Professional	☑ cpu license
TE1400	TC3 Target For Matlab Simulink	□ cpu license

5. **Optional**: If you would like to add a license for a remote device, you first need to connect to the remote device via TwinCAT XAE toolbar



6. Switch to the tab **"Order Information"** and click the button **"Activate 7 Days Trial License..."**to activate a test version



TwinCAT Project4
Order Information
Manage Licenses
Project Licenses
Online Licenses
Activate 7 Days Trial License...
Generate License Request File...
Activate License Response File...

System Id: 3897F769-B69C-788A-6450-9EE7DAD97C1B

HW Platform: other (90)

Beckhoff License Id:

Customer Id:

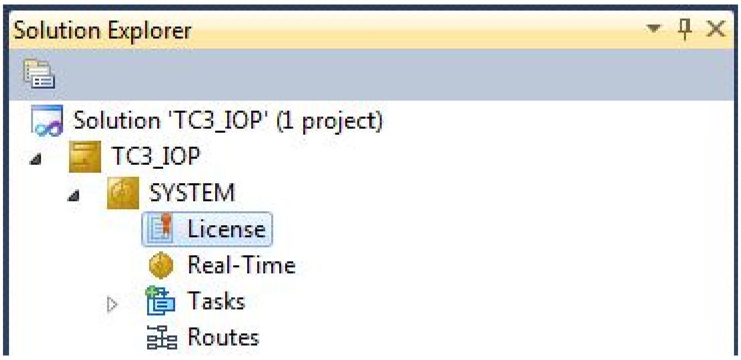
Customer Comment:

Order No	License	Instances	Current Status
TC1200	TC3 PLC	cpu license	expires on Mar 29, 2012 (trial l...
TF6420	TC3 Database-Server	cpu license	expires on Mar 29, 2012 (trial l...

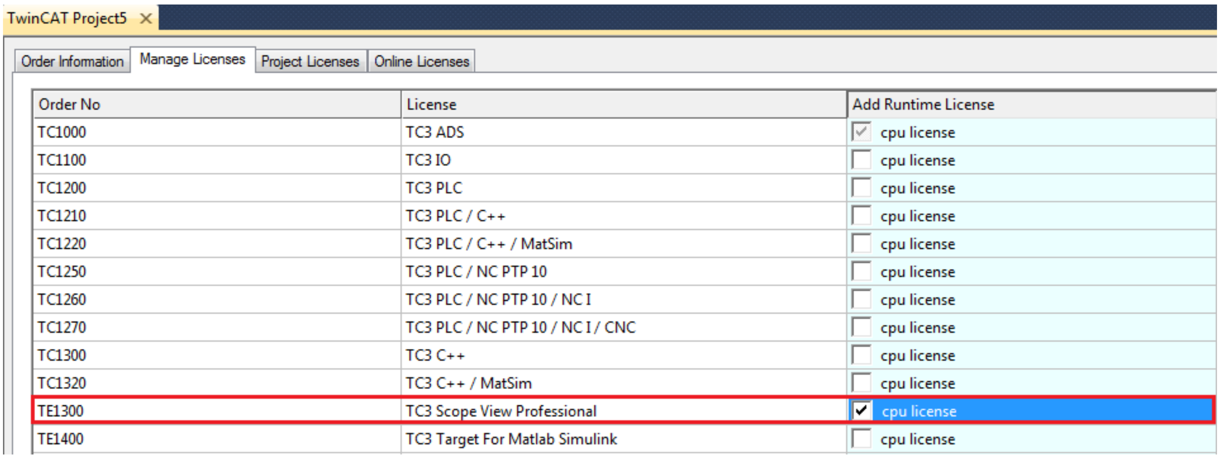
7. Please restart TwinCAT 3 afterwards.

Licensing a full version

8. Start TwinCAT XAE
9. Open an existing TwinCAT 3 project or create a new project
10. In "Solution Explorer", please navigate to the entry **"SYSTEM\License"**

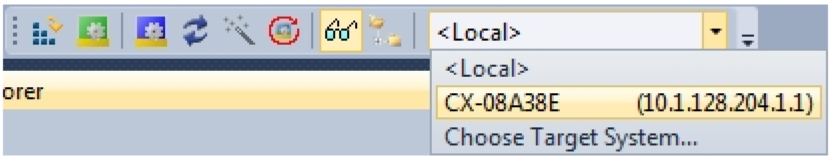


11. Open the tab **"Manage Licenses"** and add a **"Runtime License"** for your product (in this screenshot " TE1300: TC3 Scope View Professional”).

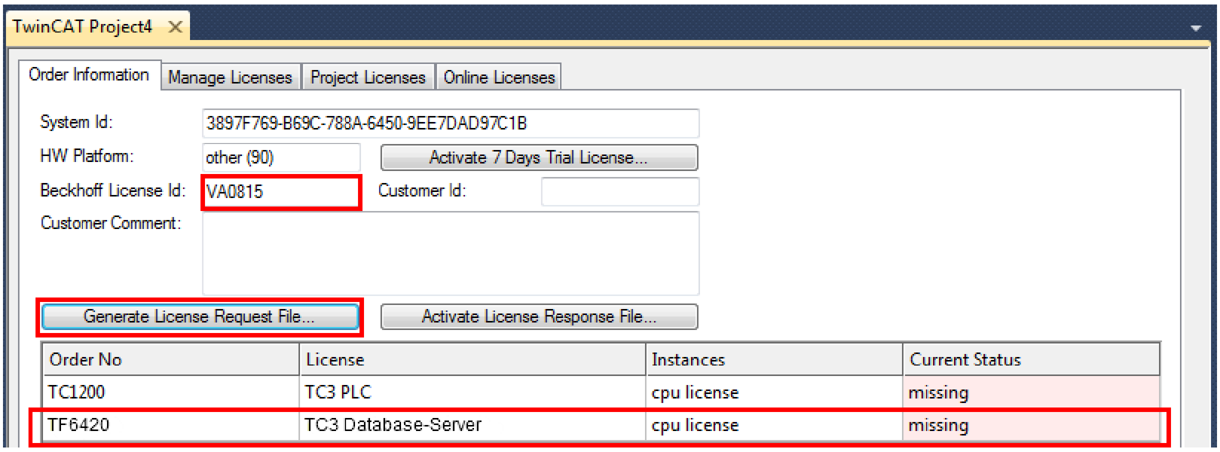


Order No	License	Add Runtime License
TC1000	TC3 ADS	☑ cpu license
TC1100	TC3 IO	□ cpu license
TC1200	TC3 PLC	□ cpu license
TC1210	TC3 PLC / C++	□ cpu license
TC1220	TC3 PLC / C++ / MatSim	□ cpu license
TC1250	TC3 PLC / NC PTP 10	□ cpu license
TC1260	TC3 PLC / NC PTP 10 / NC I	□ cpu license
TC1270	TC3 PLC / NC PTP 10 / NC I / CNC	□ cpu license
TC1300	TC3 C++	□ cpu license
TC1320	TC3 C++ / MatSim	□ cpu license
TE1300	TC3 Scope View Professional	☑ cpu license
TE1400	TC3 Target For Matlab Simulink	□ cpu license

12. **Optional:**If you would like to add a license for a remote device, you first need to connect to the remote device via TwinCAT XAE toolbar



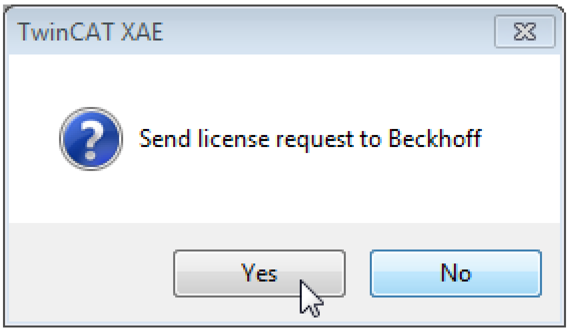
13. Navigate to the **"Order Information"** tab
    The fields "System-ID" and "HW Platform" cannot be changed and just describe the platform for the licensing process in general a TwinCAT 3 license is always bound to these two identifiers:
    the "System-ID" uniquely identifies your system.
    The "HW Platform" is an indicator for the performance of the device.
14. Optionally, you may also enter an own order number and description for your convenience


TwinCAT Project4
Order Information
Manage Licenses
Project Licenses
Online Licenses
Activate 7 Days Trial License...
Generate License Request File...
Activate License Response File...

System Id: 3897F769-B69C-788A-6450-9EE7DAD97C1B
HW Platform: other (90)
Beckhoff License Id: VA0815
Customer Id:
Customer Comment:

Order No	License	Instances	Current Status
TC1200	TC3 PLC	cpu license	missing
TF6420	TC3 Database-Server	cpu license	missing

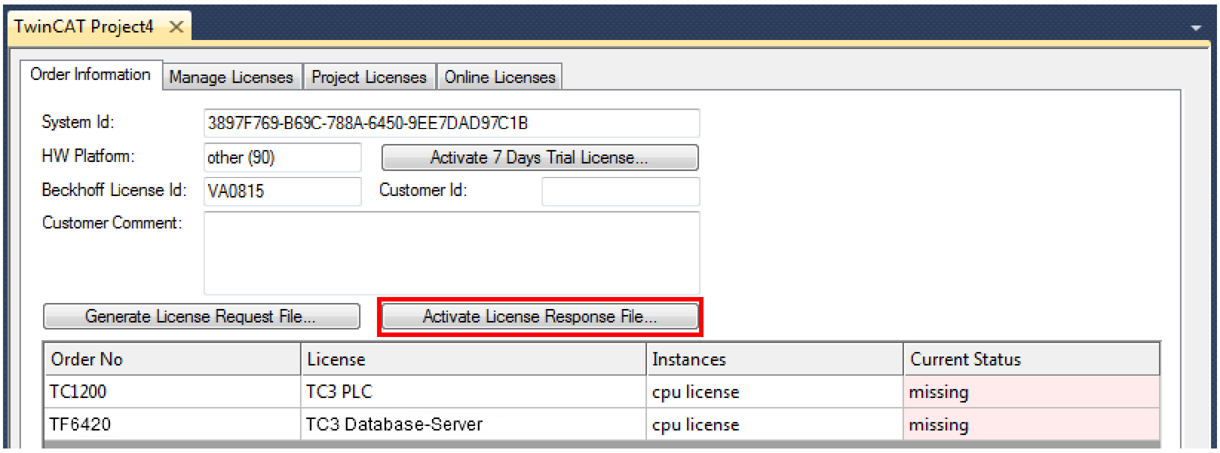
15. enter the "Beckhoff License ID" and click on **"Generate License Request File..."**. If you are not aware of your **"Beckhoff License ID"** please contact your local sales representative.
16. After the license request file has been saved, the system asks whether to send this file via E-Mail to the Beckhoff Activation Server



Send license request to Beckhoff

17. After clicking "Yes", the standard E-Mail client opens and creates a new E-Mail message to "tclicense@beckhoff.com" which contains the "License Request File"
18. Send this Activation Request to Beckhoff
    NOTE! The "License Response File" will be sent to the same E-Mail address used for sending out the "License Request File"

19. After receiving the activation file, please click on the button "Activate License Response File..."in the TwinCAT XAE license Interface.



TwinCAT Project4
Order Information
Manage Licenses
Project Licenses
Online Licenses
Activate 7 Days Trial License...
Generate License Request File...
Activate License Response File...

System Id: 3897F769-B69C-788A-6450-9EE7DAD97C1B

HW Platform: other (90)

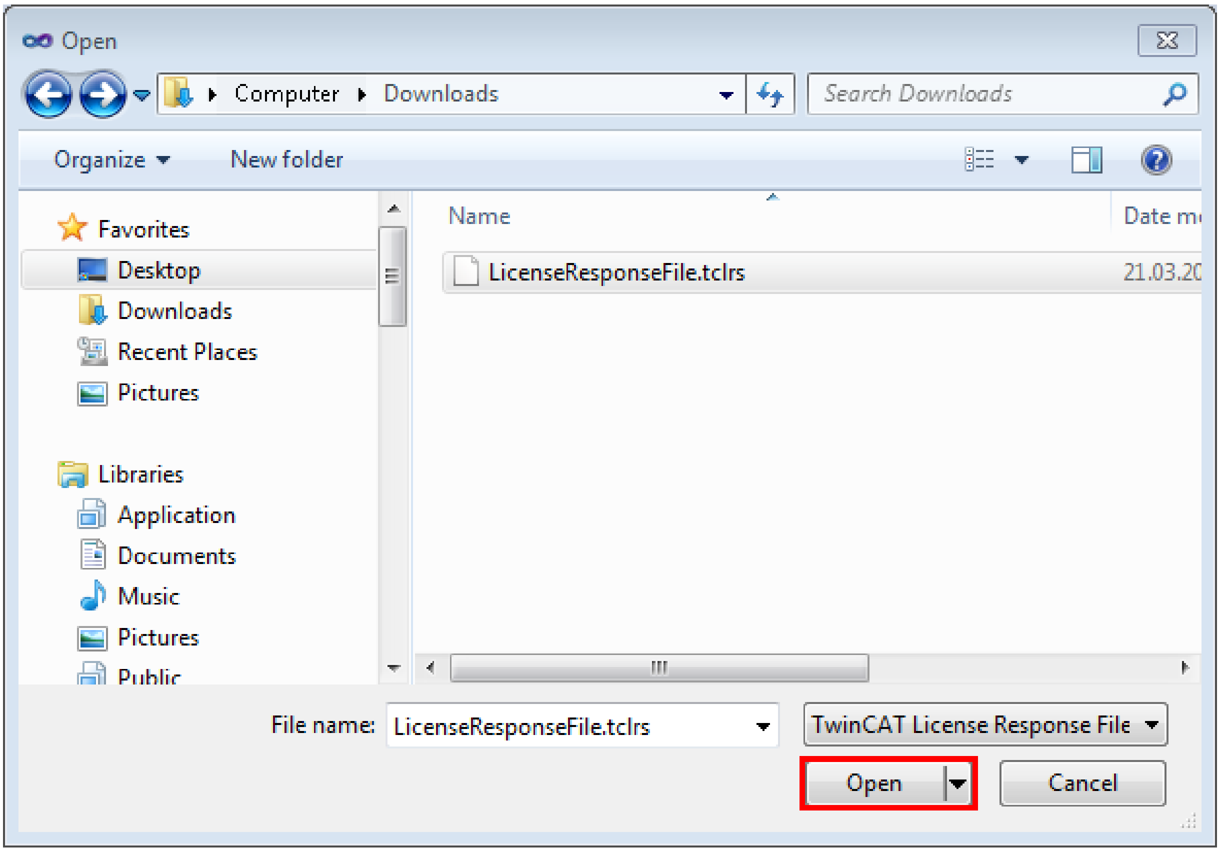
Beckhoff License Id: VA0815

Customer Id:

Customer Comment:

Order No	License	Instances	Current Status
TC1200	TC3 PLC	cpu license	missing
TF6420	TC3 Database-Server	cpu license	missing

20. Select the received "Licnse response file" and click on "Open"

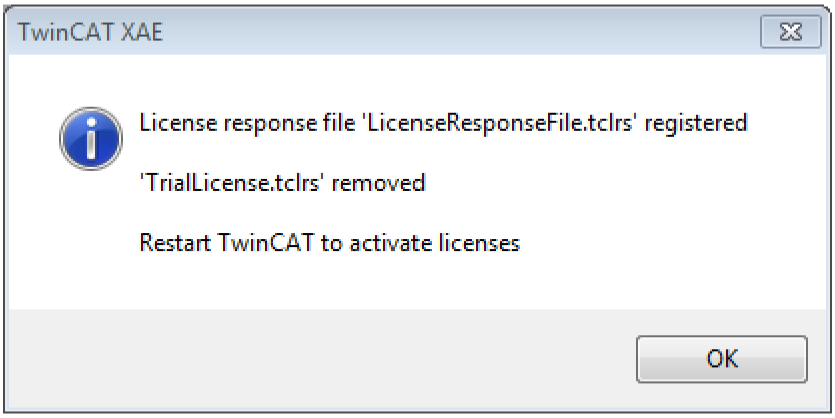


File name: LicenseResponseFile.tclrs

TwinCAT License Response File

21. The "License Response File" will be imported and all included licenses will be activated. If there have been any trial licenses, these will be removed accordingly.

22. Please restart TwinCAT to activate licenses..



License response file 'LicenseResponseFile.tclrs' registered

'TrialLicense.tclrs' removed

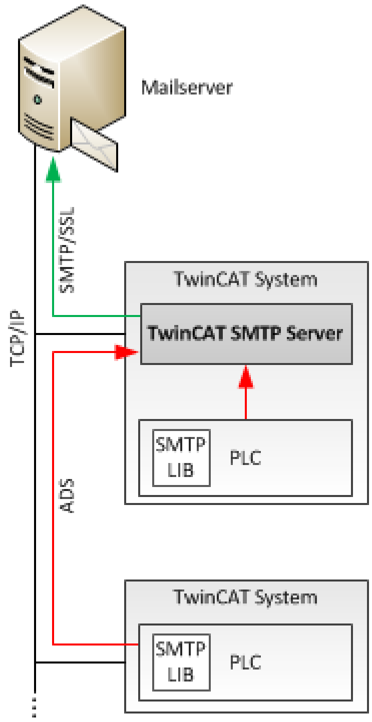
Restart TwinCAT to activate licenses

🛈 **NOTE! The license file will be automatically copied to "..\TwinCAT\3.1\Target\License" on the local device.**

4 TwinCAT SMTP

The TwinCAT SMTP Server enables to send eMails directly out of the PLC. The PLC SMTP library and the TwinCAT SMTP Server will be provided by the setup.

Mails can be send from a local system or remote system which is connected via network (TCP/IP). Therefore several machines can be observed and provide information (e.g. status, alerts, attached measured values) via mail worldwide.



The TwinCAT SMTP Server can communicate with a local mailserver (e.g. Exchange, SendMail, Notes) or external mail providers (e.g. GoogleMail, Hotmail, GMX). SMTP and encrypted communication via SSL/STARTTLS is supported.

4.1 Configuration

TC SMTP Server uses a XML-based configuration, which is located in the installation folder of the supplement (\TwinCAT\Functions\TF6350-SMS-SMTP).

The default configuration is:

```
<TcSmtpConfig>
     <!-- EnableLogFile: 0 (Disabled), 1 (Enabled), 2 (Verbose) -->
     <EnableLogFile>0</EnableLogFile>
     <!-- LogSize: in Byte, 0 = use Default -->
     <LogSize>20000</LogSize>
     <!-- Authentication: 0 (NONE), 1 (AUTO), 2 (LOGIN), 3 (NTLM), 4 (PLAIN) -->
     <Authentication>1</Authentication>
     <!-- Port: 0 (use default ports) -->
     <Port>0</Port>
     <!-- ContentEncoding: 0 (7BIT), 1 (8BIT), 2 (BINARY), 3 (BASE64), 4 (QUOTED_PRINTABLE) -->
     <ContentEncoding>0</ContentEncoding>
     <!-- Timeout for the socket connection -->
     <Timeout>15000</Timeout>
     <!-- Charset for the message content -->
     <Charset>iso-8859-1</Charset>
     <Reconnects>5</Reconnects>
</TcSmtpConfig>
```

Notes regarding the XML configuration file

EnableLogFile: Enable logging only for diagnose issues.

Authentication: Method for authenticate. Use option 1.

Port: Option 0 uses the default ports.

ContentEncoding: Defines the content encoding.

Timeout: Timeout for mail delivery in ms.

Charset: Defines the character set.

Reconnects: Number of retries.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2 PLC Libraries

Name	Beschreibung
FB_SmtpV3 [▸ 19]	Function block to send emails to the TwinCAT SMTP Server
FB_SmtpV3_Full [▸ 21]	Function block with additional properties (e.g. send attachments, set priorities, send HTML messages)

Note

Obsolete function blocks

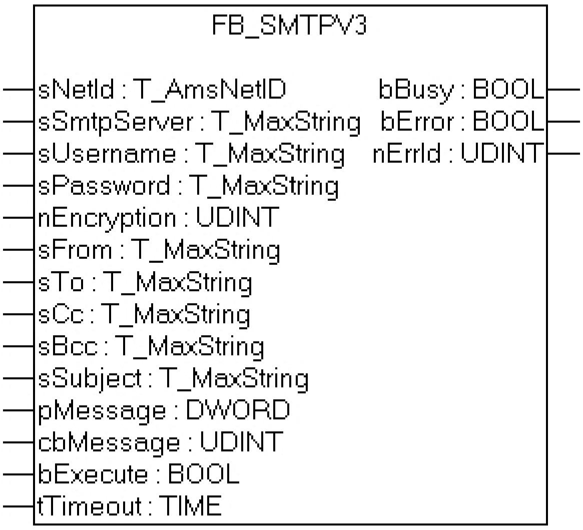
The function blocks, which are described in the chapter obsolete are obsolete. Their usage is depricated.
Please use the function blocks **FB_SmtpV3** and **FB_SmtpFull** which should provide the same functionality.

Requirements

Development environment	Target system	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2.1 Function blocks

4.2.1.1 FB_SmtpV3



The block sends a byte stream to a remote ADS device via ADS. The TwinCAT ADS Smtp service must be running on the remote ADS device, so that the byte stream can be received and processed into an e-mail. Once the byte stream has been processed the e-mail is sent.

VAR_INPUT

```
VAR_INPUT
sNetId     : T_AmsNetID; (* AmsNetID *)
sSmtpServer : T_MaxString; (* Smtp Server addres ( IP or Name) *)
sUsername     : T_MaxString;  (* Smtp Username *)
sPassword     : T_MaxString;  (* Smtp Password *)
nEncryption : UDINT;      (* 0=NONE, 1=STARTTLS, 2=SSL *)
sFrom     : T_MaxString; (* Sender stzring *)
sTo     : T_MaxString;  (* To recipient string *)
sCc     : T_MaxString;  (* Cc recipient string *)
sBcc     : T_MaxString; (* Bcc recipient string *)
sSubject     : T_MaxString;  (* Subject string *)
pMessage     : DWORD;          (* Pointer to the message *)
cbMessage     : UDINT;  (* Message lenght in byte to send *)
bExecute     : BOOL;
tTimeout     : TIME := T#20s;
END_VAR
```

sNetId: AmsNetID on which the TwinCAT Smtp server runs.

sSmtpServer: Name or IP of the Smtp server.

sUsername: Username for the Smtp Server.

sPassword: Password for the Smtp Server.

nEncryption: Smtp encryption type:
0 = NONE
1 = STARTTLS
2 = SSL

sFrom: A string containing the e-mail address of the sender. A sender must be specified. The string is limited to 255 characters.

sTo: A string containing the e-mail address of the recipient. Several addresses can be specified, separated by semicolon. At least one recipient has to be specified. The string is limited to 255 characters.

sCc: A string containing an e-mail address of a further recipient (cc=carbon copy). This string can also be empty. A copy of the e-mail is sent to this recipient. The e-mail address of this recipient is **visible** to other recipients. It is possible to enter multiple recipient addresses separated by semicolons. The string is limited to 255 characters.

sBcc: A string containing the e-mail address of a further recipient (Bcc = blind carbon copy). This string can also be empty. A copy of the e-mail is sent to this\these recipient\s. The e-mail address of this recipient is not visible to other recipients. It is possible to enter multiple recipient addresses separated by semicolons. The string is limited to 255 characters.

sSubject: A string containing the subject line for the e-mail. The e-mail may be sent without subject, in which case the name of the sending computer is automatically entered in the subject line (e.g. "Mail sent from: CX_00762C"). The string for the subject line is limited to 255 characters.

pMessage: The address (a pointer) to a null-terminated string containing the e-mail text. The e-mail may be sent without body text, in which case the date and time are entered automatically (e.g. "Mail sent at: Thu, 23 Mar 2006 02:31:44 -0800"). The address of the string can be determined with the ADR operator.

cbMessage: Length of the e-mail text. The length can be determined through the LEN operator.

bExecute: The function block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the command.

VAR_OUTPUT

```
VAR_OUTPUT
bBusy : BOOL;
bError : BOOL;
nErrId : UDINT;
END_VAR
```

bBusy: This output remains TRUE until the block has executed a command, but at the longest for the duration supplied to the tTimeOut input.

bError : This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in iErrorId.

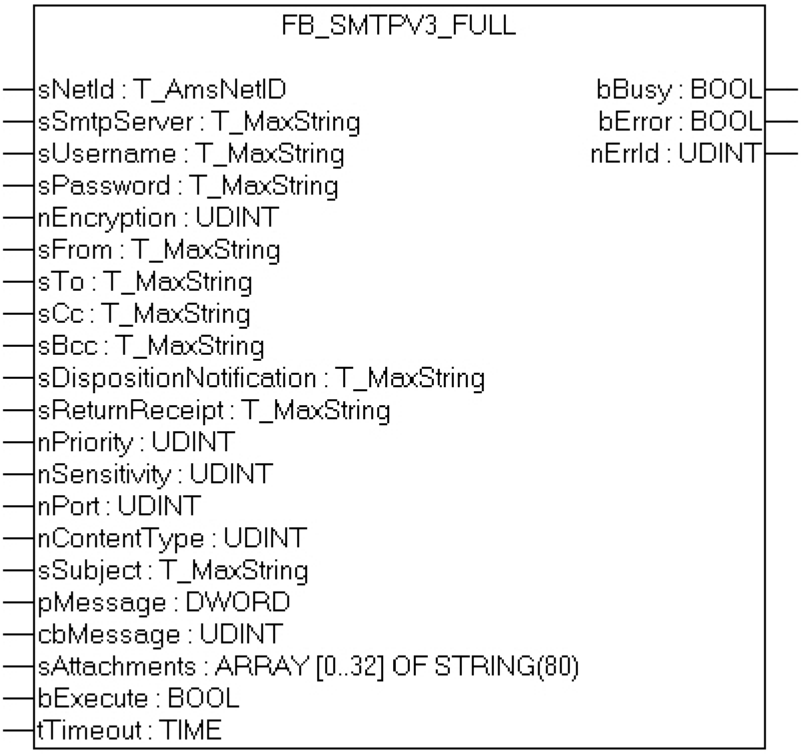
nErrId: Contains the command-specific error code of the most recently executed command (see table [▸ 35]).

Notes: Make sure, that you don't use \o within byte-arrays. Otherwise the Message will be cut.
The maximum amount of characters, that can be used in a message, is 510.725 - you have 1275 characters for From, To, Cc, Bcc and Subject.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2.1.2 FB_SmtpV3_Full



This function block communicates over ADS with the TwinCAT SMTP Server. It offers a wide range of mail functionalities as for example the prioritization of emails out of the PLC. The individual parameters will be descriped in detail in this documentation.

VAR_INPUT

```
VAR_INPUT
sNetId         : T_AmsNetID;     (* AmsNetID *)
sSmtpServer        : T_MaxString;     (* Smtp Server addres ( IP or Name) *)
sUsername         : T_MaxString;     (* Smtp Username *)
sPassword         : T_MaxString;     (* Smtp Password *)
nEncryption        : UDINT;         (* 0=NONE, 1=TLS, 2=SSL*)
sFrom         : T_MaxString;     (* Sender string *)
sTo            : T_MaxString;     (* To recipient string *)
sCc            : T_MaxString;     (* Cc recipient string *)
sBcc         : T_MaxString;     (* Bcc recipient string *)
sDispositionNotification : T_MaxString;    (* Disposition notification recipent string *)
sReturnReceipt     : T_MaxString;     (* Return recipent string *)
nPriority         : UDINT;         (* Priority value *)
nSensitivity     : UDINT;         (* Sensitivity value *)
nPort         : UDINT;         (* Communication port *)
nContentType     : UDINT;         (* Content type *)
sSubject         : T_MaxString;     (* Subject string *)
pMessage         : DWORD;         (* Pointer to the message *)
cbMessage         : UDINT;         (* Messagelenght in byte to send *)
sAttachments     : ARRAY [0..32] OF STRING; (* Different attachments *)
bExecute         : BOOL;         (* Trigger flag *)
tTimeout         : TIME := T#20s;     (* Communication timeout *)
END_VAR
```

sNetId: AmsNetID on which the TwinCAT SMTP server runs.

sSmtpServer: Name or IP of the SMTP server.

sUsername: Username for the SMTP server.

sPassword: Password for the SMTP server.

nEncryption: Smtp encryption type:
0 = NONE
1 = STARTTLS
2 = SSL

sFrom: A string containing the email address of the sender. A sender must be specified. The string is limited to 255 characters.

sTo: A string containing the email address of the recipient. Several addresses can be specified, separated by semicolon. At least one recipient has to be specified. The string is limited to 255 characters.

sCc: A string containing an email address of a further recipient (cc=carbon copy). This string can also be empty. A copy of the email is sent to this recipient. The email address of this recipient is **visible** to other recipients. It is possible to enter multiple recipient addresses separated by semicolons. The string is limited to 255 characters.

sBcc: A string containing the email address of a further recipient (Bcc = blind carbon copy). This string can also be empty. A copy of the email is sent to this\these recipient\s. The email address of this recipient is not visible to other recipients. It is possible to enter multiple recipient addresses separated by semicolons. The string is limited to 255 characters.

sDispositionNotification: The mail address which is given to this parameter receives an return receipt of the recipients under sTo and sCc. The condition precedent is that the return receipt will be send by the recipients.

sReturnReceipt: An acknowledgment of transfer will be send to this mail address.

nPriority: With this parameter you can set the priority of the mail:
1 = Highest
2 = not used
3 = Normal
4 = not used
5 = Lowest

nSensitivity: With this parameter you can set the confidentiality of the message:
0 = Private
1 = Personal
2 = Normal
3 = Confidential

nPort: You can choose the communication-port here. If you do not enter an own port it will be accessed to the default-port 25.

nContentType: With this parameter it is possible to make a HTML-code which is given per pointer (pMessage) and size (cbMessage) to a string variable readable in the mail**.**

sSubject: A string containing the subject line for the e-mail. The email may be sent without subject, in which case the name of the sending computer is automatically entered in the subject line (e.g. "Mail sent from: CX_00762C"). The string for the subject line is limited to 255 characters.

pMessage: The address (a pointer) to a null-terminated string containing the email text. The email may be sent without body text, in which case the date and time are entered automatically (e.g. "Mail sent at: Thu, 23 Mar 2006 02:31:44 -0800"). The address of the string can be determined with the ADR operator.

cbMessage: Length of the email text. The length can be determined through the LEN operator.

bExecute: The function block is activated by a rising edge at this input.

sAttachments: Array of filenames

tTimeout: Maximum time allowed for the execution of the command.

VAR_OUTPUT

```
VAR_OUTPUT
bBusy : BOOL;
bError : BOOL;
nErrId : UDINT;
END_VAR
```

bBusy: This output remains TRUE until the block has executed a command, but at the longest for the duration supplied to the tTimeOut input.

bError : This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in iErrorId.

nErrId: Contains the command-specific error code of the most recently executed command (see table [▸ 35]).

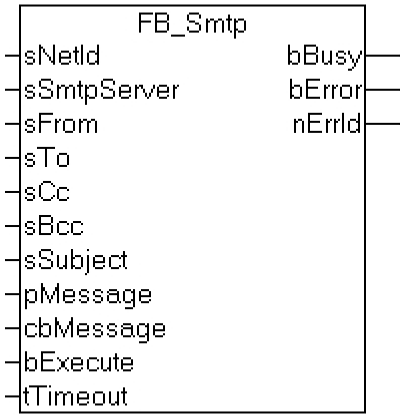
Notes: Make sure, that you don't use \o within byte-arrays. Otherwise the Message will be cut.
The maximum amount of characters, that can be used in a message, is 510.725 - you have 1275 characters for From, To, Cc, Bcc and Subject.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2.1.3 [obsolete functions]

4.2.1.3.1 FB_Smtp



The block sends a byte stream to a remote ADS device via ADS. The TwinCAT ADS Smtp service must be running on the remote ADS device, so that the byte stream can be received and processed into an e-mail. Once the byte stream has been processed the e-mail is sent.
Note that password checking must be disabled on the SMTP server, since the TwinCAT ADS Smtp service does not register on the server via password checking.

VAR_INPUT

```
VAR_INPUT
     sNetId     : T_AmsNetID; (* AmsNetID *)
     sSmtpServer : T_MaxString; (* Smtp-Server address (IP or Name) *)
     sFrom     : T_MaxString; (* Sender string *)
     sTo     : T_MaxString; (* To recipient string *)
     sCc     : T_MaxString; (* Cc recipient string *)
     sBcc     : T_MaxString; (* Bcc recipient string *)
     sSubject     : T_MaxString; (* Subject string *)
     pMessage     : DWORD;     (* Pointer to the message *)
cbMessage     : UDINT;     (* Messagelenght to send *)
bExecute     : BOOL;
     tTimeout     : TIME := T#20s;
END_VAR
```

sNetId: AmsNetID on which the TwinCAT SMS server runs.

sSmtpServer: Name or IP of the Smtp server.

sFrom: A string containing the e-mail address of the sender. A sender must be specified. The string is limited to 255 characters.

sTo: A string containing the e-mail address of the recipient. Several addresses can be specified, separated by semicolon. At least one recipient has to be specified. The string is limited to 255 characters.

sCc: A string containing an e-mail address of a further recipient (cc=carbon copy). This string can also be empty. A copy of the e-mail is sent to this recipient. The e-mail address of this recipient is **visible** to other recipients. It is possible to enter multiple recipient addresses separated by semicolons. The string is limited to 255 characters.

sBcc: A string containing the e-mail address of a further recipient (Bcc = blind carbon copy). This string can also be empty. A copy of the e-mail is sent to this\these recipient\s. The e-mail address of this recipient is not visible to other recipients. It is possible to enter multiple recipient addresses separated by semicolons. The string is limited to 255 characters.

sSubject: A string containing the subject line for the e-mail. The e-mail may be sent without subject, in which case the name of the sending computer is automatically entered in the subject line (e.g. "Mail sent from: CX_00762C"). The string for the subject line is limited to 255 characters.

pMessage: The address (a pointer) to a null-terminated string containing the e-mail text. The e-mail may be sent without body text, in which case the date and time are entered automatically (e.g. "Mail sent at: Thu, 23 Mar 2006 02:31:44 -0800"). The address of the string can be determined with the ADR operator.

cbMessage: Length of the e-mail text. The length can be determined through the LEN operator.

bExecute: The function block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the command.

VAR_OUTPUT

```
VAR_OUTPUT
     bBusy : BOOL;
     bError : BOOL;
     nErrId : UDINT;
END_VAR
```

bBusy: This output remains TRUE until the block has executed a command, but at the longest for the duration supplied to the tTimeOut input.

bError : This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in iErrorId.

nErrId: Contains the command-specific error code of the most recently executed command (see table [▸ 35]).

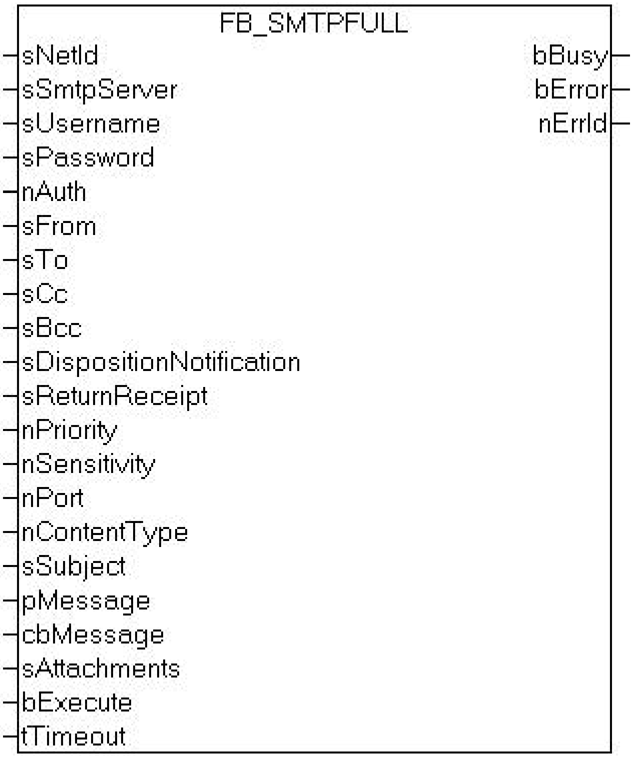
Notes:

Make sure, that you don't use \o within byte-arrays. Otherwise the Message will be cut.
The maximum amount of characters, that can be used in a message, is 510.725 - you have 1275 characters for From, To, Cc, Bcc and Subject.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2.1.3.2 FB_SmtpFull



This function block communicates over ADS with the TwinCAT SMTP Server. It offers a wide range of mail functionalities as for example the prioritization of emails out of the PLC. The individual parameters will be described in detail in this documentation.

VAR_INPUT

```
VAR_INPUT
     sNetId         : T_AmsNetID;     (* AmsNetID *)
     sSmtpServer        : T_MaxString;      (* Smtp Server addres ( IP or Name) *)
     sUsername         : T_MaxString;     (* Smtp Username *)
     sPassword         : T_MaxString;     (* Smtp Password *)
     nAuth         : UDINT;         (* Smtp Auth Type*)
     sFrom         : T_MaxString;     (* Sender stzring *)
     sTo            : T_MaxString;     (* To recipient string *)
     sCc            : T_MaxString;     (* Cc recipient string *)
     sBcc         : T_MaxString;     (* Bcc recipient string *)
     sDispositionNotification : T_MaxString;      (* Disposition notification recipent string *)
     sReturnReceipt     : T_MaxString;      (* Return recipent string *)
     nPriority         : UDINT;         (* Priority value *)
     nSensitivity     : UDINT;         (* Sensitivity value *)
     nPort         : UDINT;         (* Communication port *)
     nContentType     : UDINT;         (* Content type *)
     sSubject         : T_MaxString;     (* Subject string *)
     pMessage         : DWORD;          (* Pointer to the message *)
     cbMessage         : UDINT;              (* Messagelenght in byte to send *)
     sAttachments     : ARRAY [0..32] OF STRING; (* Different attachments *)
     bExecute         : BOOL;          (* Trigger flag *)
     tTimeout         : TIME := T#20s;      (* Communication timeout *)
END_VAR
```

sNetId: AmsNetID on which the TwinCAT SMTP server runs.

sSmtpServer: Name or IP of the SMTP server.

sUsername: Username for the SMTP server.

sPassword: Password for the SMTP server.

nAuth: Smtp Auth Type:
0 = AUTH NONE
1 = RESERVED
2 = AUTH LOGIN
3 = AUTH NTLM
4 = AUTH PLAIN

sFrom: A string containing the email address of the sender. A sender must be specified. The string is limited to 255 characters.

sTo: A string containing the email address of the recipient. Several addresses can be specified, separated by semicolon. At least one recipient has to be specified. The string is limited to 255 characters.

sCc: A string containing an email address of a further recipient (cc=carbon copy). This string can also be empty. A copy of the email is sent to this recipient. The email address of this recipient is **visible** to other recipients. It is possible to enter multiple recipient addresses separated by semicolons. The string is limited to 255 characters.

sBcc: A string containing the email address of a further recipient (Bcc = blind carbon copy). This string can also be empty. A copy of the email is sent to this\these recipient\s. The email address of this recipient is not visible to other recipients. It is possible to enter multiple recipient addresses separated by semicolons. The string is limited to 255 characters.

sDispositionNotification: The mail address which is given to this parameter receives an return receipt of the recipients under sTo and sCc. The condition precedent is that the return receipt will be send by the recipients.

sReturnReceipt: An acknowledgment of transfer will be send to this mail address.

nPriority: With this parameter you can set the priority of the mail:
1 = Highest
2 = not used
3 = Normal
4 = not used
5 = Lowest

nSensitivity: With this parameter you can set the confidentiality of the message:
0 = Private
1 = Personal
2 = Normal
3 = Confidential

nPort: You can choose the communication-port here. If you do not enter an own port it will be accessed to the default-port 25.

nContentType: With this parameter it is possible to make a HTML-code which is given per pointer (pMessage) and size (cbMessage) to a string variable readable in the mail**.**

sSubject: A string containing the subject line for the e-mail. The email may be sent without subject, in which case the name of the sending computer is automatically entered in the subject line (e.g. "Mail sent from: CX_00762C"). The string for the subject line is limited to 255 characters.

pMessage: The address (a pointer) to a null-terminated string containing the email text. The email may be sent without body text, in which case the date and time are entered automatically (e.g. "Mail sent at: Thu, 23 Mar 2006 02:31:44 -0800"). The address of the string can be determined with the ADR operator.

cbMessage: Length of the email text. The length can be determined through the LEN operator.

bExecute: The function block is activated by a rising edge at this input.

sAttachments: Array of filenames

tTimeout: Maximum time allowed for the execution of the command.

VAR_OUTPUT

```
VAR_OUTPUT
     bBusy : BOOL;
     bError : BOOL;
     nErrId : UDINT;
END_VAR
```

bBusy: This output remains TRUE until the block has executed a command, but at the longest for the duration supplied to the tTimeOut input.

bError : This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in iErrorId.

nErrId: Contains the command-specific error code of the most recently executed command (see table [▸ 35]).

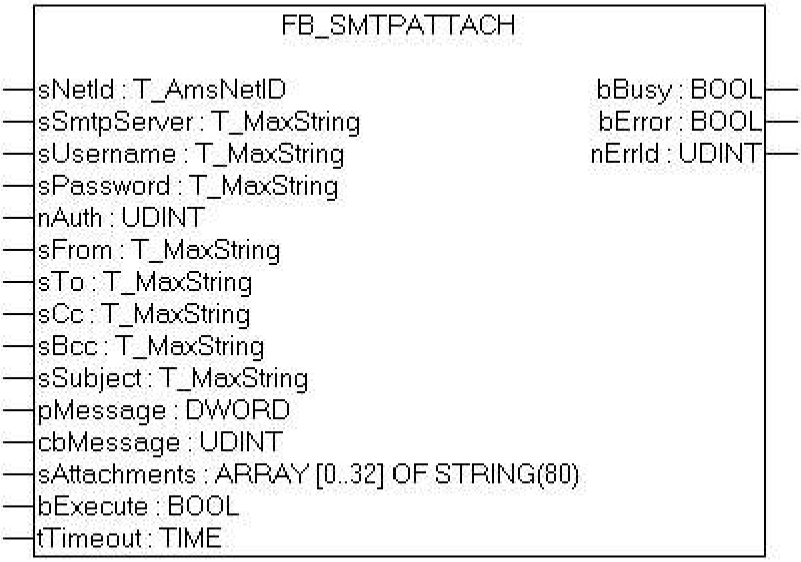
Notes:

Make sure, that you don't use \o within byte-arrays. Otherwise the Message will be cut.
The maximum amount of characters, that can be used in a message, is 510.725 - you have 1275 characters for From, To, Cc, Bcc and Subject.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2.1.3.3 FB_SmtpAttach



The block sends a byte stream to a remote ADS device via ADS. The TwinCAT ADS Smtp service must be running on the remote ADS device, so that the byte stream can be received and processed into an e-mail. Once the byte stream has been processed the e-mail is sent.

VAR_INPUT

```
VAR_INPUT
sNetId     : T_AmsNetID;     (* AmsNetID *)
sSmtpServer : T_MaxString;      (* Smtp Server addres (IP or Name)*)
sUsername     : T_MaxString;     (* Smtp Username *)
sPassword     : T_MaxString;     (* Smtp Password *)
nAuth     : UDINT;         (* Smtp Auth Type *)
sFrom     : T_MaxString;     (* Sender stzring *)
sTo     : T_MaxString;     (* To recipient string *)
sCc     : T_MaxString;     (* Cc recipient string *)
sBcc     : T_MaxString;     (* Bcc recipient string *)
sSubject     : T_MaxString;     (* Subject string *)
pMessage     : DWORD;         (* Pointer to the message *)
cbMessage     : UDINT;         (* Messagelenght in byte to send *)
sAttachments : ARRAY [0..32] OF STRING;
bExecute     : BOOL;
tTimeout     : TIME := T#20s;
END_VAR
```

sNetId: AmsNetID on which the TwinCAT SMS server runs.

sSmtpServer: Name or IP of the Smtp server.

sUsername: Username for the Smtp Server.

sPassword: Password for the Smtp Server.

nAuth: Smtp Auth Type:
0 = AUTH NONE
1 = RESERVED
2 = AUTH LOGIN
3 = AUTH NTLM
4 = AUTH PLAIN

sFrom: A string containing the e-mail address of the sender. A sender must be specified. The string is limited to 255 characters.

sTo: A string containing the e-mail address of the recipient. Several addresses can be specified, separated by semicolon. At least one recipient has to be specified. The string is limited to 255 characters.

sCc: A string containing an e-mail address of a further recipient (cc=carbon copy). This string can also be empty. A copy of the e-mail is sent to this recipient. The e-mail address of this recipient is **visible** to other recipients. It is possible to enter multiple recipient addresses separated by semicolons. The string is limited to 255 characters.

sBcc: A string containing the e-mail address of a further recipient (Bcc = blind carbon copy). This string can also be empty. A copy of the e-mail is sent to this\these recipient\s. The e-mail address of this recipient is not visible to other recipients. It is possible to enter multiple recipient addresses separated by semicolons. The string is limited to 255 characters.

sSubject: A string containing the subject line for the e-mail. The e-mail may be sent without subject, in which case the name of the sending computer is automatically entered in the subject line (e.g. "Mail sent from: CX_00762C"). The string for the subject line is limited to 255 characters.

pMessage: The address (a pointer) to a null-terminated string containing the e-mail text. The e-mail may be sent without body text, in which case the date and time are entered automatically (e.g. "Mail sent at: Thu, 23 Mar 2006 02:31:44 -0800"). The address of the string can be determined with the ADR operator.

cbMessage: Length of the e-mail text. The length can be determined through the LEN operator.

bExecute: The function block is activated by a rising edge at this input.

sAttachments: Array containing filenames

tTimeout: Maximum time allowed for the execution of the command.

VAR_OUTPUT

```
VAR_OUTPUT
     bBusy : BOOL;
     bError : BOOL;
     nErrId : UDINT;
END_VAR
```

bBusy: This output remains TRUE until the block has executed a command, but at the longest for the duration supplied to the tTimeOut input.

bError : This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in iErrorId.

nErrId: Contains the command-specific error code of the most recently executed command (see table [▸ 35]).

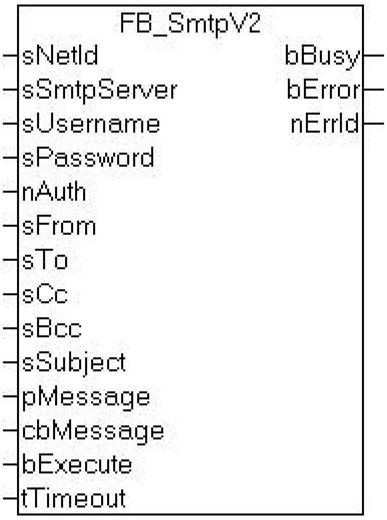
Notes:

Make sure, that you don't use \o within byte-arrays. Otherwise the Message will be cut.
The maximum amount of characters, that can be used in a message, is 510.725 - you have 1275 characters for From, To, Cc, Bcc and Subject.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2.1.3.4 FB_SmtpV2



The block sends a byte stream to a remote ADS device via ADS. The TwinCAT ADS Smtp service must be running on the remote ADS device, so that the byte stream can be received and processed into an e-mail. Once the byte stream has been processed the e-mail is sent.

VAR_INPUT

```
VAR_INPUT
     sNetId     : T_AmsNetID; (* AmsNetID *)
     sSmtpServer : T_MaxString; (* Smtp Server addres ( IP or Name) *)
     sUsername     : T_MaxString; (* Smtp Username *)
     sPassword     : T_MaxString; (* Smtp Password *)
     nAuth     : UDINT;     (* Smtp Auth Type *)
     sFrom     : T_MaxString; (* Sender stzring *)
     sTo     : T_MaxString; (* To recipient string *)
     sCc     : T_MaxString; (* Cc recipient string *)
     sBcc     : T_MaxString; (* Bcc recipient string *)
     sSubject     : T_MaxString; (* Subject string *)
     pMessage     : DWORD;      (* Pointer to the message *)
     cbMessage     : UDINT;      (* Messagelenght in byte to send *)
     bExecute     : BOOL;
     tTimeout     : TIME := T#20s;
END_VAR
```

sNetId: AmsNetID on which the TwinCAT Smtp server runs.

sSmtpServer: Name or IP of the Smtp server.

sUsername: Username for the Smtp Server.

sPassword: Password for the Smtp Server.

nAuth: Smtp Auth Type:
0 = AUTH NONE
1 = RESERVED
2 = AUTH LOGIN
3 = AUTH NTLM
4 = AUTH PLAIN

sFrom: A string containing the e-mail address of the sender. A sender must be specified. The string is limited to 255 characters.

sTo: A string containing the e-mail address of the recipient. Several addresses can be specified, separated by semicolon. At least one recipient has to be specified. The string is limited to 255 characters.

sCc: A string containing an e-mail address of a further recipient (cc=carbon copy). This string can also be empty. A copy of the e-mail is sent to this recipient. The e-mail address of this recipient is **visible** to other recipients. It is possible to enter multiple recipient addresses separated by semicolons. The string is limited to 255 characters.

sBcc: A string containing the e-mail address of a further recipient (Bcc = blind carbon copy). This string can also be empty. A copy of the e-mail is sent to this\these recipient\s. The e-mail address of this recipient is not visible to other recipients. It is possible to enter multiple recipient addresses separated by semicolons. The string is limited to 255 characters.

sSubject: A string containing the subject line for the e-mail. The e-mail may be sent without subject, in which case the name of the sending computer is automatically entered in the subject line (e.g. "Mail sent from: CX_00762C"). The string for the subject line is limited to 255 characters.

pMessage: The address (a pointer) to a null-terminated string containing the e-mail text. The e-mail may be sent without body text, in which case the date and time are entered automatically (e.g. "Mail sent at: Thu, 23 Mar 2006 02:31:44 -0800"). The address of the string can be determined with the ADR operator.

cbMessage: Length of the e-mail text. The length can be determined through the LEN operator.

bExecute: The function block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the command.

VAR_OUTPUT

```
VAR_OUTPUT
     bBusy : BOOL;
     bError : BOOL;
     nErrId : UDINT;
END_VAR
```

bBusy: This output remains TRUE until the block has executed a command, but at the longest for the duration supplied to the tTimeOut input.

bError : This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in iErrorId.

nErrId: Contains the command-specific error code of the most recently executed command (see table [▸ 35]).

Notes:

Make sure, that you don't use \o within byte-arrays. Otherwise the Message will be cut.
The maximum amount of characters, that can be used in a message, is 510.725 - you have 1275 characters for From, To, Cc, Bcc and Subject.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2.2 Samples

4.2.2.1 Send mail sample

The function block FB_SmtpV3 enables to send mails from your PLC.

Description

A message will be sent if the variable bStart is triggered.

ToDo: Configure mailserver and credentials.

Download Sample1 TC3 project

Program-variables

```
PROGRAM MAIN
VAR
SendMail: FB_SmtpV3;
sMessage: STRING := 'Hello Beckhoff';
R_Edge: R_TRIG;
bStart: BOOL;
bBusy: BOOL;
bError: BOOL;
nErrID: UDINT;
bSend: BOOL;
nErr: UDINT;
nMailCounter: UDINT;
END_VAR
```

Program-code

```

fbSendMail(
sNetId:= '',
sSmtpServer:= 'mail.company.com',
sUsername:= 'TestUser',
sPassword:= 'TestPwd',
sFrom:= 'TestUser@company.com',
sTo:= 'service@company.com',
sSubject:= 'Email from your Beckhoff PLC',
pMessage:= ADR(sMessage),
cbMessage:= SIZEOF(sMessage),
bExecute:= bStart,
bError=> bError,
bBusy=> bBusy,
nErrId=> nErrId);

IF NOT bBusy AND NOT bError AND bStart THEN
bStart := FALSE;
END_IF
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.2.2.2 SmtpFull sample with features

With the function block FB_SmtpV3_Full more email functionalities are available.

Description

In this example the following possibilities will be descriped:

- The email text can provide **HTML** code, which offers new formatting possibilities.
- Furthermore it is possible to set the **priority** of an email

- Several files can be sent as **attachment**

A message will be sent if the variable bStart is triggered.

ToDo: Configure mailserver address, credentials and adjust file paths of the attachments.

Download Sample2 TC3 project

Program-variables

```
VAR
fbSmtpFull    : FB_SmtpV3_Full;
sMessage_HTML : STRING(500) := '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"><html><head><title>HTML-Test</title><body><h3>Dear SMTP-
User,</h3>this e-mail was sent in HTML from <u>your PLC!</u><br/><br/>Best regards,<br/>your Beck-
hoff-Team</body></html>';
nPriority     : UDINT;
R_Edge      : R_TRIG;
bStart      : BOOL;
bError      : BOOL;
udErrId      : UDINT;
nMailCounter : UDINT := 0;
sFiles      : ARRAY [0..32] OF STRING(80) := ['c:\Temperatures.txt','c:\Log.txt']; (* TODO: Adjust
file paths*)END_VAR
```

Program-code

```

fbSmtpFull(
sNetId:= '',
sSmtpServer:= 'mail.company.com',
sUsername:= 'TestUser',
sPassword:= 'TestPwd',
sFrom:= 'TestUser@company.com',
sTo:= 'service@company.com',
sSubject:= 'Email from your Beckhoff PLC',
nContentType:= 2, (* 2 = HTML *)
nPriority:= 1, (* 1 = HIGH *)
sAttachments:= sFiles,
pMessage:= ADR(sMessage_HTML),
cbMessage:= SIZEOF(sMessage_HTML),
bExecute:= bStart,
bError=> bError,
bBusy=>bBusy,
nErrId=> nErrId);

IF NOT bBusy AND NOT bError AND bStart THEN
nState := 0;
bStart := FALSE;
END_IF
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.3 Appendix

4.3.1 Support and Service

The following list provides some basic help in case any errors should occur and should be read *before* contacting our support department.

1. **Check if one of the PLC function blocks returns an errorcode:**

   Please consult the list of error codes:

   SMTP Error codes [▸ 35]

   ADS Return codes

Win Socket Error codes [▶ 37]

2. **Activate the logging option in the TcSmtpConfig.xml**
   The TcSmtpConfig.xml is located in \TwinCAT\Functions\TF6350-SMS-SMTP

   Enable verbose logging by setting the EnableLogFile = 2 and restart TwinCAT.

   <TcSmtpConfig> <!-- EnableLogFile: 0 (Disabled), 1 (Enabled), 2 (Verbose) --> <EnableLogFile>2</EnableLogFile> ... <Reconnects>5</Reconnects> </TcSmtpConfig>

   The log file will be generate in \TwinCAT\Functions\TF6350-SMS-SMTP .

Should the troubleshooting checklist does not help, please contact our support department and provide the following information:

General system information

- What kind of hardware is being used on the computer running TF6350 SMS/SMTP?
  - Beckhoff IPC or Embedded PC: Which product number does the PC have?
  - Which Operating System image version is currently installed on that computer?

Product-related system information

- Which version of TF6350 SMS/STMP is being used?
- Which function blocks of the Tc2_SMTP library are being used in the PLC program?
- Which SMTP Server is being used?
- Microsoft Exchange Server
- Axigen
- PostFix
- WebMail Provider (e.g. GMAIL, Hotmail, GMX)
- Please provide the SMTP logfile (see 2. of the troubleshooting list)
- Please provide an exact description of the environment in which the product TF6350 SMS/SMTP is being used
  - Where is the computer running TF6350 SMS/SMTP located?
  - Where is the SMTP-Server located? (Local network, Internet)
  - Which encryption is in use? (NON, STARTTLS, SSL)
  - What are the IP settings of the Mail-Server and the computer running TF6350 SMS/SMTP? (IP address, subnet mask, Ports)
  - In case that there is a firewall between both computers:

What kind of Firewall system is being used (please provide vendor information)?

Has the Firewall been configured to allow SMTP connections?

TF6350 TC3 SMS/SMTP

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages:
http://www.beckhoff.com

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49(0)5246/963-157
Fax:	+49(0)5246/963-9157
e-mail:	support@beckhoff.com

4.3.2 Error Codes

4.3.2.1 Error Codes

This list contains error codes of the TwinCAT supplement product SMTP Server. If you miss some error codes, please look under ADS return codes [▸ 46] or WinSockErrorCodes [▸ 37].

Hex	Dec	Description
0x8000	32778	ADS return code
0x800A	32778	Not connected
0x800B	32779	Sender expected
0x800C	32780	Recipients expected
0x800D	32781	Send FROM command failed
0x800E	32782	Send DATA command failed
0x800F	32783	Send mail header failed
0x8010	32784	Send mail body failed
0x8011	32785	Send "end of mail indicator" failed
0x8012	32786	Send "RCPT" command failed
0x8013	32787	Server Response got no username request
0x8014	32788	Server Response got no password request
0x8015	32789	Unable to create socket connection
0x8016	32790	Authentication type not supported by smtp server
0x8017	32791	Wrong username or password
0x8018	32792	Not supported
0x8019	32793	Invalid hostname
0x801A	32794	Unable to send attachment
0x801B	32795	File not found
0x801C	32796	Invalid Version (New SMTP Server with old SMTP PLC library)
0x801D	32797	Unable to connect (Connection error => sometimes wrong port or wrong server)
0x801E	32798	Unable to create socket
0x801F	32799	WSA startup failed
0x8020	32800	Invalid hostname
0x8021	32801	Unecpected response from server
0x8022	32802	Error while receiving data
0x8023	32803	No supported authentication methods found
0x8024	32804	Invalid parameter
0x80A0	32928	Security interface not found
0x80A1	32929	Unable to call security interface
0x80A2	32930	Security initialization failed
0x80A4	32932	Unable to create credentials
0x80A5	32933	SSL-handshake failed
0x80A6	32934	Invalid server credentials
0x80A7	32935	Unable to verify server
0x80A8	32936	Unable to encrypt message
0x80A9	32937	Unable to decrypt message

The following errors can occur in older versions of the server (< 1.0.14)

Hex	Dec	Description
0x000A	10	Not connected
0x000B	11	Sender expected
0x000C	12	Recipients expected
0x000D	13	Send FROM command failed
0x000E	14	Send DATA command failed
0x000F	15	Send mail header failed
0x0010	16	Send mail body failed
0x0011	17	Send "end of mail indicator" failed
0x0012	18	Send "RCPT" command failed
0x0064	100	General error
0x0065	101	Invalid parameter
0x0066	102	Funtion not loaded
0x0067	103	Dll not loaded
0x0068	104	TcSmtpDll.dll cannot load. Check the installation from the TcSmtpDll.dll.
0x80D3	211	System status, or system help reply
0x80D6	214	Help message [Information on how to use the receiver or the meaning of a particular non-standard command; this reply is useful only to the human user]
0x80FB	251	User not local; will forward to <forward-path>
0x8163	354	Start mail input; end with <CRLF>.<CRLF>
0x81A5	421	<domain> Service not available, closing transmission channel [This may be a reply to any command if the service knows it must shut down]
0x81C2	450	Requested mail action not taken: mailbox unavailable [E.g., mailbox busy]
0x81C3	451	Requested action aborted: error in processing
0x81C4	452	Requested action not taken: insufficient system storage
0x81F4	500	Syntax error, command unrecognized [This may include errors such as command line too long]
0x81F5	501	Syntax error in parameters or arguments.
0x81F6	502	Command not implemented.
0x81F7	503	Bad sequence of commands.
0x8504	504	Command parameter not implemented
0x8226	550	Requested action not taken: mailbox unavailable [E.g., mailbox not found, no access]
0x8227	551	User not local; please try <forward-path>
0x8228	552	Requested mail action aborted: exceeded storage allocation
0x8229	553	Requested action not taken: mailbox name not allowed [E.g., mailbox syntax incorrect]
0x8224	554	Transaction failed

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

4.3.2.2 Windows Socket Error Codes

The following table describes the possible error codes, returned by the WSAGetLastError function. The errors are sorted in alphabetical order. Some error codes that are defined in Winsock2.h are not returned. They are not included in the list.

Return Value	Description
WSAEINTR10004	Interrupted function call.blocking operation was interrupted by a call to WSACancelBlockingCall.
WSAEACCES 10013	Permission denied.An attempt was made to access a socket in a way forbidden by its access permissions. An example is using a broadcast address for sendto without broadcast permission being set using setsockopt(SO_BROADCAST). Another possible reason for the WSAEACCES error is that when the bind function is called (on Windows NT 4 SP4 or later), another application, service, or kernel mode driver is bound to the same address with exclusive access. Such exclusive access is a new feature of Windows NT 4 SP4 and later, and is implemented by using the SO_EXCLUSIVEADDRUSE option.
WSAEFAULT 10014	Bad address.The system detected an invalid pointer address in attempting to use a pointer argument of a call. This error occurs if an application passes an invalid pointer value, or if the length of the buffer is too small. For instance, if the length of an argument, which is a sockaddr structure, is smaller than the sizeof(sockaddr).
WSAEINVAL 10022	Invalid argument.Some invalid argument was supplied (for example, specifying an invalid level to the setsockopt function). In some instances, it also refers to the current state of the socket—for instance, calling accept on a socket that is not listening.
WSAEMFILE 10024	Too many open files.Too many open sockets. Each implementation may have a maximum number of socket handles available, either globally, per process, or per thread.
WSAEWOULDBLOCK 10035	Resource temporarily unavailable.This error is returned from operations on nonblocking sockets that cannot be completed immediately, for example recv when no data is queued to be read from the socket. It is a nonfatal error, and the operation should be retried later. It is normal for WSAEWOULDBLOCK to be reported as the result from calling connect on a nonblocking SOCK_STREAM socket, since some time must elapse for the connection to be established.
WSAEINPROGRESS 10036	Operation now in progress.A blocking operation is currently executing. Windows Sockets only allows a single blocking operation—per- task or thread—to be outstanding, and if any other function call is made (whether or not it references that or any other socket) the function fails with the WSAEINPROGRESS error.
WSAEALREADY 10037	Operation already in progress.An operation was attempted on a nonblocking socket with an operation already in progress—that is, calling connect a second time on a nonblocking socket that is already connecting, or canceling an asynchronous request (WSAAsyncGetXbyY) that has already been canceled or completed.
WSAENOTSOCK 10038	Socket operation on nonsocket.An operation was attempted on something that is not a socket. Either the socket handle parameter did not reference a valid socket, or for select, a member of an fd_set was not valid.
WSAEDESTADDRREQ 10039	Destination address required.A required address was omitted from an operation on a socket. For example, this error is returned if sendto is called with the remote address of ADDR_ANY.
WSAEMSGSIZE 10040	Message too long.A message sent on a datagram socket was larger than the internal message buffer or some other network limit, or the buffer used to receive a datagram was smaller than the datagram itself.
WSAEPROTOTYPE 10041	Protocol wrong type for socket.A protocol was specified in the socket function call that does not support the semantics of the socket type requested. For example, the ARPA Internet UDP protocol cannot be specified with a socket type of SOCK_STREAM.
WSAENOPROTOOPT 10042	Bad protocol option.An unknown, invalid or unsupported option or level was specified in a getsockopt or setsockopt call.
WSAEPROTONOSUPPORT 10043	Protocol not supported.The requested protocol has not been configured into the system, or no implementation for it exists. For example, a socket call requests a SOCK_DGRAM socket, but specifies a stream protocol.
WSAESOCKTNOSUPPORT 10044	Socket type not supported.The support for the specified socket type does not exist in this address family. For example, the optional type SOCK_RAW might be selected in a socket call, and the implementation does not support SOCK_RAW sockets at all.

Return Value	Description
WSAEOPNOTSUPP 10045	Operation not supported.The attempted operation is not supported for the type of object referenced. Usually this occurs when a socket descriptor to a socket that cannot support this operation is trying to accept a connection on a datagram socket.
WSAEPFNOSUPPORT 10046	Protocol family not supported.The protocol family has not been configured into the system or no implementation for it exists. This message has a slightly different meaning from WSAEAFNOSUPPORT. However, it is interchangeable in most cases, and all Windows Sockets functions that return one of these messages also specify WSAEAFNOSUPPORT.
WSAEAFNOSUPPORT 10047	Address family not supported by protocol family.An address incompatible with the requested protocol was used. All sockets are created with an associated address family (that is, AF_INET for Internet Protocols) and a generic protocol type (that is, SOCK_STREAM). This error is returned if an incorrect protocol is explicitly requested in the socket call, or if an address of the wrong family is used for a socket, for example, in sendto.
WSAEADDRINUSE 10048	Address already in use.Typically, only one usage of each socket address (protocol/IP address/port) is permitted. This error occurs if an application attempts to bind a socket to an IP address/port that has already been used for an existing socket, or a socket that was not closed properly, or one that is still in the process of closing. For server applications that need to bind multiple sockets to the same port number, consider using setsockopt (SO_REUSEADDR). Client applications usually need not call bind at all— connect chooses an unused port automatically. When bind is called with a wildcard address (involving ADDR_ANY), a WSAEADDRINUSE error could be delayed until the specific address is committed. This could happen with a call to another function later, including connect, listen, WSAConnect, or WSAJoinLeaf.
WSAEADDRNOTAVAIL 10049	Cannot assign requested address.The requested address is not valid in its context. This normally results from an attempt to bind to an address that is not valid for the local computer. This can also result from connect, sendto, WSAConnect, WSAJoinLeaf, or WSASendTo when the remote address or port is not valid for a remote computer (for example, address or port 0).
WSAENETDOWN 10050	Network is down.A socket operation encountered a dead network. This could indicate a serious failure of the network system (that is, the protocol stack that the Windows Sockets DLL runs over), the network interface, or the local network itself.
WSAENETUNREACH 10051	Network is unreachable.A socket operation was attempted to an unreachable network. This usually means the local software knows no route to reach the remote host.
WSAENETRESET 10052	Network dropped connection on reset.The connection has been broken due to keep-alive activity detecting a failure while the operation was in progress. It can also be returned by setsockopt if an attempt is made to set SO_KEEPALIVE on a connection that has already failed.
WSAECONNABORTED 10053	Software caused connection abort.An established connection was aborted by the software in your host computer, possibly due to a data transmission time-out or protocol error.
WSAECONNRESET 10054	Connection reset by peer.An existing connection was forcibly closed by the remote host. This normally results if the peer application on the remote host is suddenly stopped, the host is rebooted, the host or remote network interface is disabled, or the remote host uses a hard close (see setsockopt for more information on the SO_LINGER option on the remote socket). This error may also result if a connection was broken due to keep-alive activity detecting a failure while one or more operations are in progress. Operations that were in progress fail with WSAENETRESET. Subsequent operations fail with WSAECONNRESET.
WSAENOBUFS 10055	No buffer space available.An operation on a socket could not be performed because the system lacked sufficient buffer space or because a queue was full.
WSAEISCONN 10056	Socket is already connected.A connect request was made on an already-connected socket. Some implementations also return this error if sendto is called on a connected SOCK_DGRAM socket (for SOCK_STREAM sockets, the to parameter in sendto is ignored) although other implementations treat this as a legal occurrence.

Return Value	Description
WSAENOTCONN 10057	Socket is not connected.A request to send or receive data was disallowed because the socket is not connected and (when sending on a datagram socket using sendto) no address was supplied. Any other type of operation might also return this error—for example, setsockopt setting SO_KEEPALIVE if the connection has been reset.
WSAESHUTDOWN 10058	Cannot send after socket shutdown.A request to send or receive data was disallowed because the socket had already been shut down in that direction with a previous shutdown call. By calling shutdown a partial close of a socket is requested, which is a signal that sending or receiving, or both have been discontinued.
WSAETIMEDOUT 10060	Connection timed out.A connection attempt failed because the connected party did not properly respond after a period of time, or the established connection failed because the connected host has failed to respond.
WSAECONNREFUSED 10061	Connection refused.No connection could be made because the target computer actively refused it. This usually results from trying to connect to a service that is inactive on the foreign host—that is, one with no server application running.
WSAEHOSTDOWN 10064	Host is down.A socket operation failed because the destination host is down. A socket operation encountered a dead host. Networking activity on the local host has not been initiated. These conditions are more likely to be indicated by the error WSAETIMEDOUT.
WSAEHOSTUNREACH 10065	No route to host.A socket operation was attempted to an unreachable host. See WSAENETUNREACH.
WSAEPROCLIM 10067	Too many processes.A Windows Sockets implementation may have a limit on the number of applications that can use it simultaneously.WSAStartup may fail with this error if the limit has been reached.
WSASYSNOTREADY 10091	Network subsystem is unavailable.This error is returned by WSAStartup if the Windows Sockets implementation cannot function at this time because the underlying system it uses to provide network services is currently unavailable. Users should check: • That the appropriate Windows Sockets DLL file is in the current path. • That they are not trying to use more than one Windows Sockets implementation simultaneously. If there is more than one Winsock DLL on your system, be sure the first one in the path is appropriate for the network subsystem currently loaded. • The Windows Sockets implementation documentation to be sure all necessary components are currently installed and configured correctly.
WSAVERNOTSUPPORTED 10092	Winsock.dll version out of range.The current Windows Sockets implementation does not support the Windows Sockets specification version requested by the application. Check that no old Windows Sockets DLL files are being accessed.
WSANOTINITIALISED 10093	Successful WSAStartup not yet performed.Either the application has not called WSAStartup or WSAStartup failed. The application may be accessing a socket that the current active task does not own (that is, trying to share a socket between tasks), or WSACleanup has been called too many times.
WSAEDISCON 10101	Graceful shutdown in progress.Returned by WSARecv and WSARecvFrom to indicate that the remote party has initiated a graceful shutdown sequence.
WSATYPE_NOT_FOUND 10109	Class type not found.The specified class was not found.
WSAHOST_NOT_FOUND 11001	Host not found.No such host is known. The name is not an official host name or alias, or it cannot be found in the database(s) being queried. This error may also be returned for protocol and service queries, and means that the specified name could not be found in the relevant database.
WSATRY_AGAIN 11002	Nonauthoritative host not found.This is usually a temporary error during host name resolution and means that the local server did not receive a response from an authoritative server. A retry at some time later may be successful.

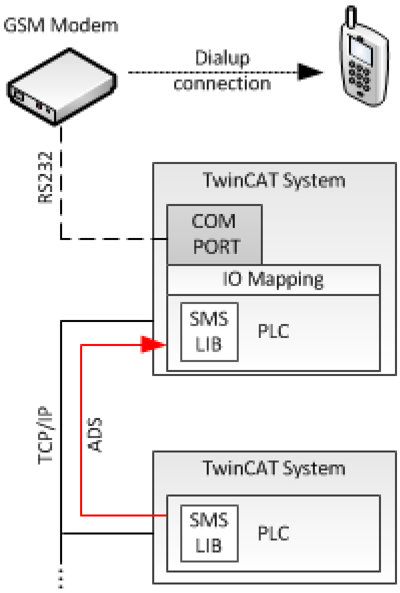
Return Value	Description
WSANO_RECOVERY 11003	This is a nonrecoverable error.This indicates that some sort of nonrecoverable error occurred during a database lookup. This may be because the database files (for example, BSD-compatible HOSTS, SERVICES, or PROTOCOLS files) could not be found, or a DNS request was returned by the server with a severe error.
WSANO_DATA 11004	Valid name, no data record of requested type.The requested name is valid and was found in the database, but it does not have the correct associated data being resolved for. The usual example for this is a host name-to-address translation attempt (using gethostbyname or WSAAsyncGetHostByName) which uses the DNS (Domain Name Server). An MX record is returned but no A record—indicating the host itself exists, but is not directly reachable.
WSA_INVALID_HANDLE OS dependent	Specified event object handle is invalid.An application attempts to use an event object, but the specified handle is not valid.
WSA_INVALID_PARAMETER OS dependent	One or more parameters are invalid.An application used a Windows Sockets function which directly maps to a Windows function. The Windows function is indicating a problem with one or more parameters.
WSA_IO_INCOMPLETE OS dependent	Overlapped I/O event object not in signaled state.The application has tried to determine the status of an overlapped operation which is not yet completed. Applications that use WSAGetOverlappedResult (with the fWait flag set to FALSE) in a polling mode to determine when an overlapped operation has completed, get this error code until the operation is complete.
WSA_IO_PENDING OS dependent	Overlapped operations will complete later. The application has initiated an overlapped operation that cannot be completed immediately. A completion indication will be given later when the operation has been completed.
WSA_NOT_ENOUGH_MEMORY OS dependent	Insufficient memory available.An application used a Windows Sockets function that directly maps to a Windows function. The Windows function is indicating a lack of required memory resources.
WSA_OPERATION_ABORTED OS dependent	Overlapped operation aborted.An overlapped operation was canceled due to the closure of the socket, or the execution of the SIO_FLUSH command in WSAIoctl.
WSAINVALIDPROCTABLE OS dependent	Invalid procedure table from service provider.A service provider returned a bogus procedure table to Ws2_32.dll. (This is usually caused by one or more of the function pointers being null.)
WSAINVALIDPROVIDER OS dependent	Invalid service provider version number.A service provider returned a version number other than 2.0.
WSAPROVIDERFAILEDINIT OS dependent	Unable to initialize a service provider.Either a service provider's DLL could not be loaded (LoadLibrary failed) or the provider's WSPStartup/NSPStartup function failed.
WSASYSCALLFAILURE OS dependent	System call failure.Generic error code, returned under various conditions. Returned when a system call that should never fail does fail. For example, if a call to WaitForMultipleEvents fails or one of the registry functions fails trying to manipulate the protocol/namespace catalogs.Returned when a provider does not return SUCCESS and does not provide an extended error code. Can indicate a service provider implementation error.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Smtp

5 TwinCAT SMS

The TwinCAT SMS library contains library for sending SMS messages directly from the PLC. The SMS library is based on the 'Serial Communication' library, which will be installed, too. This makes it possible to communicate with the PC's serial interface and with the serial terminal (EL6xxx and KL6xxx) in the same way.

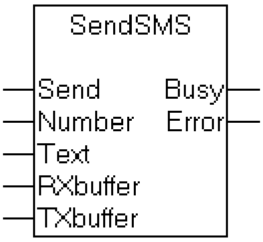


Additionally it is possible to provide status or alerts from other TwinCAT system, which are connected via network.

5.1 PLC libraries

5.1.1 Function blocks

5.1.1.1 SendSMS Function Block



The **SendSMS** function block allows an SMS to be sent via a connected GSM modem. The function block is based on the 'Serial Communication' library.

Because the block only communicates via the **ComBuffer**structure in the 'Serial Communication' library, instances can be formed, and it can be applied to every kind of serial intSmserface.

VAR_INPUT

```
Send     : BOOL;
Number     : String;
Text     : String(160);
```

Send: The function block is activated by a positive edge at this input.

Number: telephone number to be dialled in national format (e.g.: 0170123456)

Text: The SMS message to be sent

VAR_OUTPUT

```
Busy     : BOOL;
Error     : INT;
```

Busy: This output is set when there is a rising edge at the Send input, and remains set until the SMS has been sent to the modem or until an error has occurred.

Error: If an error occurs while the SMS is being transferred, the Busy output is reset, and an error code is made available at the Error output. If the Error output is 0, the transfer was successful.

The function block can return the following errors:

Number	Meaning	Cause
1	Communication with the modem is not possible.	Is the terminal correctly configured? Has the appropriate ComLib library been used?
2	Modem reports an error during configuration.	Is a compatible GSM modem connected?
3	Modem can not send SMS.	Is the SIM card working properly? Can the card be used without entering the PIN? Is the modem connected to the network? Is a compatible modem connected?
4	Communication error.	Has the correct transmission speed been set?

VAR_IN_OUT

```
RXbuffer     : ComBuffer;
TXbuffer     : ComBuffer;
```

RXbuffer: Structure for communication with the serial interface. An interface-specific block in the 'Serial Communication' library fills this buffer with the data for the interface.

TXbuffer: Structure for communication with the serial interface. An interface-specific block in the 'Serial Communication' library transfers the data from this buffer to the interface.

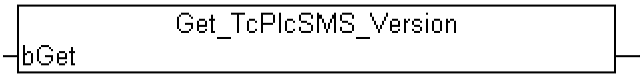
These structures, and their usage, are described in more detail in the documentation for the ‘ Serial Communication' library.The SendSMS block is here connected to a SendString or ReceiveString block.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Sms

5.1.2 Functions

5.1.2.1 FUNCTION Get_TcPlcSMS_Version



The function returns library version info.

FUNCTION Get_TcPlcSMS_Version: STRING(20)

VAR_INPUT

```
bGet : BOOL;
```

bGet: The compiler requires at least one input parameter for functions. You can set this parameter to TRUE or FALSE.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Sms

5.1.3 Global constants

5.1.3.1 Library version

All libraries have a specific version. This version is shown in the PLC library repository too.
A global constant contains the library version information:

Global_Version

```
VAR_GLOBAL CONSTANT
    stLibVersion_Tc2_SMS_SMTP : ST_LibVersion;
END_VAR
```

ST_LibVersion

To compare the existing version to a required version the function F_CmpLibVersion (defined in Tc2_System library) is offered.

Note	**Compare versions** All other possibilities known from TwinCAT2 libraries to query a library version are obsolete!

5.2 Samples

5.2.1 Sending an SMS

It is possible to communicate with the PC's serial interface and with the serial terminal (EL6xxx and KL6xxx) in the same way.

1. Add the Tc2_SMS and Tc2_SerialCom Library to your PLC project
2. Follow the tutorial for the ‘Serial Communication' library to configure the communication
3. Check the documentation of your used GSM modem to disabe the PIN request (the SIM card should not be protected by a PIN)

Supported devices:

- Westermo GS-01 (communication parameters: 9600 baud, 8 data bits, no parity bit, one stop bit)
- Maestro 100 (communication parameters: 9600 baud, 8 data bits, no parity bit, one stop bit)
- For further device tests please contact support@beckhoff.com

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Sms, Tc2_SerialCom Library

5.3 Appendix

5.3.1 Fault Finding

There are a number of reasons why an SMS may fail to be sent with the SendSMS function block or SMS COM Server:

- no connection to the GSM modem
- incorrectly configured communication settings of SMS COM Server
- incorrect call to the ADS service
- the use of an unsupported GSM modem
- incorrectly configured serial terminal (Advanced or Standard, 3 byte / 5 byte, speed, ...)
- incorrect telephone number
- PIN required (the SIM card must not be protected by a PIN)
- Serial terminal not initialised (call KL6Init)
- incorrect in GSM network

A variety of tools are available to look for these errors:

Using the Log File

Keeping records in a log file can be activated with the TwinCAT SMS Server Configurator. Once this has been done, all the messages sent and the errors are written into the TcSmsSrvCfg.xml file. The file can be found in the TwinCAT installation directory.

NT Event Log

Errors when sending messages are also always recorded in the NT Event Log. The Event Log can be opened through the TwinCAT icon on the task bar.

ADS Error Messages

If the call to an ADS Function fails, the error is coded in the function's return value. A list of these error codes can be found under ADS Return Codes. [▸ 46]

Configuration of the Terminal

The serial terminal can be configured in different ways. Terminals that have been differently configured, have to some extent a different representation in the process image (3 byte /5 byte terminals, advanced/standard). It must be noted that the ComLib library must be appropriate for the terminal configuration. See also the documentation for the KL6xxx and the ComLib documentation:

It is also important that the terminal transmission speed be matched to that of the modem in use.

Sending a Test SMS

A test SMS can easily be sent with the Visual Basic example program, to find out whether an error lies with the ADS call or in the configuration of the SMS Server.

Sending a Test SMS using a Mobile Telephone

To find out whether the SIM card is correctly configured, it can be inserted into an ordinary mobile phone and used to send an SMS. It should not be necessary to enter a PIN number here.

Network Selection with the Westermo GS-01

GS-01 has a number of variations for the various networks in Europe and in the USA. The lamp on the front of the modem indicates whether a network is available. The lamp flashes if the modem is connected to a network. If the lamp is continuously illuminated, the fault finding section should be consulted in the Westermo manual.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_

5.3.2 ADS Return Codes

Error codes: 0x000 [▸ 47]..., 0x500 [▸ 47]..., 0x700 [▸ 48]..., 0x1000 [▸ 50]...

Global Error Codes

Hex	Dec	Description
0x0	0	no error
0x1	1	Internal error
0x2	2	No Rtime
0x3	3	Allocation locked memory error
0x4	4	Insert mailbox error
0x5	5	Wrong receive HMSG
0x6	6	target port not found
0x7	7	target machine not found
0x8	8	Unknown command ID
0x9	9	Bad task ID
0xA	10	No IO
0xB	11	Unknown ADS command
0xC	12	Win 32 error
0xD	13	Port not connected
0xE	14	Invalid ADS length
0xF	15	Invalid ADS Net ID
0x10	16	Low Installation level
0x11	17	No debug available
0x12	18	Port disabled
0x13	19	Port already connected
0x14	20	ADS Sync Win32 error
0x15	21	ADS Sync Timeout
0x16	22	ADS Sync AMS error
0x17	23	ADS Sync no index map
0x18	24	Invalid ADS port
0x19	25	No memory
0x1A	26	TCP send error
0x1B	27	Host unreachable
0x1C	28	Invalid AMS fragment

Router Error Codes

Hex	Dec	Name	Description
0x500	1280	ROUTERERR_NOLOCKEDMEMORY	No locked memory can be allocated
0x501	1281	ROUTERERR_RESIZEMEMORY	The size of the router memory could not be changed
0x502	1282	ROUTERERR_MAILBOXFULL	The mailbox has reached the maximum number of possible messages. The current sent message was rejected
0x503	1283	ROUTERERR_DEBUGBOXFULL	The mailbox has reached the maximum number of possible messages. The sent message will not be displayed in the debug monitor
0x504	1284	ROUTERERR_UNKNOWNPORTTYPE	Unknown port type
0x505	1285	ROUTERERR_NOTINITIALIZED	Router is not initialized
0x506	1286	ROUTERERR_PORTALREADYINUSE	The desired port number is already assigned
0x507	1287	ROUTERERR_NOTREGISTERED	Port not registered
0x508	1288	ROUTERERR_NOMOREQUEUES	The maximum number of Ports reached
0x509	1289	ROUTERERR_INVALIDPORT	Invalid port
0x50A	1290	ROUTERERR_NOTACTIVATED	TwinCAT Router not active

General ADS Error Codes

Hex	Dec	Name	Description
0x700	1792	ADSERR_DEVICE_ERROR	error class <device error>
0x701	1793	ADSERR_DEVICE_SRVNOTSUPP	Service is not supported by server
0x702	1794	ADSERR_DEVICE_INVALIDGRP	invalid index group
0x703	1795	ADSERR_DEVICE_INVALIDOFFSET	invalid index offset
0x704	1796	ADSERR_DEVICE_INVALIDACCESS	reading/writing not permitted
0x705	1797	ADSERR_DEVICE_INVALIDSIZE	parameter size not correct
0x706	1798	ADSERR_DEVICE_INVALIDDATA	invalid parameter value(s)
0x707	1799	ADSERR_DEVICE_NOTREADY	device is not in a ready state
0x708	1800	ADSERR_DEVICE_BUSY	device is busy
0x709	1801	ADSERR_DEVICE_INVALIDCONTEXT	invalid context (must be in Windows)
0x70A	1802	ADSERR_DEVICE_NOMEMORY	out of memory
0x70B	1803	ADSERR_DEVICE_INVALIDPARM	invalid parameter value(s)
0x70C	1804	ADSERR_DEVICE_NOTFOUND	not found (files, ...)
0x70D	1805	ADSERR_DEVICE_SYNTAX	syntax error in command or file
0x70E	1806	ADSERR_DEVICE_INCOMPATIBLE	objects do not match
0x70F	1807	ADSERR_DEVICE_EXISTS	object already exists
0x710	1808	ADSERR_DEVICE_SYMBOLNOTFOUND	symbol not found
0x711	1809	ADSERR_DEVICE_SYMBOLVERSIONINVAL	symbol version invalid
0x712	1810	ADSERR_DEVICE_INVALIDSTATE	server is in invalid state
0x713	1811	ADSERR_DEVICE_TRANSMODENOTSUPP	AdsTransMode not supported
0x714	1812	ADSERR_DEVICE_NOTIFYHNDINVALID	Notification handle is invalid
0x715	1813	ADSERR_DEVICE_CLIENTUNKNOWN	Notification client not registered
0x716	1814	ADSERR_DEVICE_NOMOREHDLS	no more notification handles
0x717	1815	ADSERR_DEVICE_INVALIDWATCHSIZE	size for watch too big
0x718	1816	ADSERR_DEVICE_NOTINIT	device not initialized
0x719	1817	ADSERR_DEVICE_TIMEOUT	device has a timeout
0x71A	1818	ADSERR_DEVICE_NOINTERFACE	query interface failed
0x71B	1819	ADSERR_DEVICE_INVALIDINTERFACE	wrong interface required
0x71C	1820	ADSERR_DEVICE_INVALIDCLSID	class ID is invalid
0x71D	1821	ADSERR_DEVICE_INVALIDOBJID	object ID is invalid
0x71E	1822	ADSERR_DEVICE_PENDING	request is pending
0x71F	1823	ADSERR_DEVICE_ABORTED	request is aborted
0x720	1824	ADSERR_DEVICE_WARNING	signal warning
0x721	1825	ADSERR_DEVICE_INVALIDARRAYIDX	invalid array index
0x722	1826	ADSERR_DEVICE_SYMBOLNOTACTIVE	symbol not active
0x723	1827	ADSERR_DEVICE_ACCESSDENIED	access denied
0x724	1828	ADSERR_DEVICE_LICENSENOTFOUND	missing license
0x725	1829	ADSERR_DEVICE_LICENSEEXPIRED	license expired
0x726	1830	ADSERR_DEVICE_LICENSEEXCEEDED	license exceeded
0x727	1831	ADSERR_DEVICE_LICENSEINVALID	license invalid
0x728	1832	ADSERR_DEVICE_LICENSESYSTEMID	license invalid system id
0x729	1833	ADSERR_DEVICE_LICENSENOTIMELIMIT	license not time limited
0x72A	1834	ADSERR_DEVICE_LICENSEFUTUREISSUE	license issue time in the future
0x72B	1835	ADSERR_DEVICE_LICENSETIMETOLONG	license time period to long
0x72c	1836	ADSERR_DEVICE_EXCEPTION	exception occured during system start
0x72D	1837	ADSERR_DEVICE_LICENSEDUPLICATED	License file read twice
0x72E	1838	ADSERR_DEVICE_SIGNATUREINVALID	invalid signature
0x72F	1839	ADSERR_DEVICE_CERTIFICATEINVALID	public key certificate
0x740	1856	ADSERR_CLIENT_ERROR	Error class <client error>
0x741	1857	ADSERR_CLIENT_INVALIDPARM	invalid parameter at service
0x742	1858	ADSERR_CLIENT_LISTEMPTY	polling list is empty
0x743	1859	ADSERR_CLIENT_VARUSED	var connection already in use
0x744	1860	ADSERR_CLIENT_DUPLINVOKEID	invoke ID in use
0x745	1861	ADSERR_CLIENT_SYNCTIMEOUT	timeout elapsed
0x746	1862	ADSERR_CLIENT_W32ERROR	error in win32 subsystem
0x747	1863	ADSERR_CLIENT_TIMEOUTINVALID	Invalid client timeout value

Hex	Dec	Name	Description
0x748	1864	ADSERR_CLIENT_PORTNOTOPEN	ads-port not opened
0x750	1872	ADSERR_CLIENT_NOAMSADDR	internal error in ads sync
0x751	1873	ADSERR_CLIENT_SYNCINTERNAL	hash table overflow
0x752	1874	ADSERR_CLIENT_ADDHASH	key not found in hash
0x753	1875	ADSERR_CLIENT_REMOVEHASH	no more symbols in cache
0x754	1876	ADSERR_CLIENT_NOMORESYM	invalid response received
0x755	1877	ADSERR_CLIENT_SYNCRESINVALID	sync port is locked

RTime Error Codes

Hex	Dec	Name	Description
0x1000	4096	RTERR_INTERNAL	Internal fatal error in the TwinCAT real-time system
0x1001	4097	RTERR_BADTIMERPERIODS	Timer value not vaild
0x1002	4098	RTERR_INVALIDTASKPTR	Task pointer has the invalid value ZERO
0x1003	4099	RTERR_INVALIDSTACKPTR	Task stack pointer has the invalid value ZERO
0x1004	4100	RTERR_PRIOEXISTS	The demand task priority is already assigned
0x1005	4101	RTERR_NOMORETCB	No more free TCB (Task Control Block) available. Maximum number of TCBs is 64
0x1006	4102	RTERR_NOMORESEMAS	No more free semaphores available. Maximum number of semaphores is 64
0x1007	4103	RTERR_NOMOREQUEUES	No more free queue available. Maximum number of queue is 64
0x100D	4109	RTERR_EXTIRQALREADYDEF	An external synchronization interrupt is already applied
0x100E	4110	RTERR_EXTIRQNOTDEF	No external synchronization interrupt applied
0x100F	4111	RTERR_EXTIRQINSTALLFAILED	The apply of the external synchronization interrupt failed
0x1010	4112	RTERR_IRQLNOTLESSOREQUAL	Call of a service function in the wrong context
0x1017	4119	RTERR_VMXNOTSUPPORTED	Intel VT-x extension is not supported
0x1018	4120	RTERR_VMXDISABLED	Intel VT-x extension is not enabled in system BIOS
0x1019	4121	RTERR_VMXCONTROLSMISSING	Missing function in Intel VT-x extension
0x101A	4122	RTERR_VMXENABLEFAILS	Enabling Intel VT-x fails

TCP Winsock Error Codes

Hex	Dec	Description
0x274d	10061	A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.
0x2751	10065	No connection could be made because the target machine actively refused it. This error normally occurs when you try to connect to a service which is inactive on a different host - a service without a server application.
0x274c	10060	No route to a host. A socket operation was attempted to an unreachable host
		Further Winsock error codes: Win32 Error Codes