

TwinCAT 3 Connectivity



Manual

TC3 FTP Client

TwinCAT 3

Version 1.0
Date 2014-06-17
Order No. TF6300

BECKHOFF

Table of contents

1	Foreword	5
1.1	Notes on the documentation	5
1.2	Safety instructions	6
2	Product description	7
3	Installation	9
3.1	System requirements.....	9
3.2	Installation.....	9
3.3	Licensing.....	12
4	Configuration	17
4.1	Introduction to File Transfer Protocol (FTP)	17
4.2	Configure Data-Ports for active FTP.....	18
4.3	Activating an error logfile	19
5	PLC libraries	21
5.1	Tc2_FTP	21
5.2	Function Blocks	22
5.2.1	FB_FTP_HostResolve	22
5.2.2	FB_FTP_Open.....	23
5.2.3	FB_FTP_OpenEX.....	24
5.2.4	FB_FTP_Close	25
5.2.5	FB_FTP_CloseAll	26
5.2.6	FB_FTP_Info.....	27
5.2.7	FB_FTP_FileUpload	28
5.2.8	FB_FTP_FileUploadEx	29
5.2.9	FB_FTP_FileDownload.....	30
5.2.10	FB_FTP_FileDownloadEx.....	32
5.2.11	FB_FTP_DirCreate	33
5.2.12	FB_FTP_DirRemove.....	34
5.2.13	FB_FTP_FileList	35
5.2.14	FB_FTP_FileListEx	36
5.2.15	FB_FTP_FileExist.....	38
5.2.16	FB_FTP_FileRemove	39
5.2.17	FB_FTP_FileRename	40
5.2.18	FB_GetStateTcFTPClient	41
5.3	Functions [obsolete].....	42
5.3.1	F_GetVersionTcFTPClient.....	42
5.4	Data Types	43
5.4.1	T_HFTP	43
5.4.2	ST_FTP_ConnInfo	43
5.4.3	ST_FTP_FileDetails.....	44
5.4.4	E_FTP_ConnMode	44
5.5	Constants.....	45
5.5.1	Konstanten.....	45
6	Samples	46
6.1	Samples.....	46
6.2	TwinCAT FTP Client: Upload of a file to a FTP Server.....	46
6.3	TwinCAT FTP Client: Downloading a file from a FTP Server to an ADS device	47
6.4	TwinCAT FTP Client: Getting connection information with the FB_FTP_Info	49
6.5	TwinCAT FTP Client: Removing a file from the FTP Server.....	50
6.6	TwinCAT FTP Client: Reading of a filelist from the FTP Server	51

7 Appendix	54
7.1 Return Codes.....	54
7.1.1 Overview of the TwinCAT FTP Client Error Codes.....	54
7.1.2 ADS Return Codes	54
7.1.3 FTP Client Return Codes.....	59
7.2 Troubleshooting	60
7.2.1 Troubleshooting	60
7.2.2 Contact Beckhoff Support	60

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®

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

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

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

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.

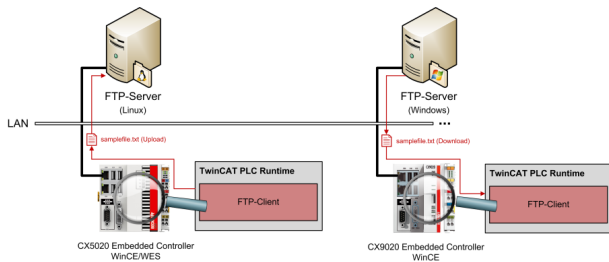
Description of symbols

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	<p>Serious risk of injury! Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.</p>
 WARNING	<p>Risk of injury! Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.</p>
 CAUTION	<p>Personal injuries! Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.</p>
 Attention	<p>Damage to the environment or devices Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.</p>
 Note	<p>Tip or pointer This symbol indicates information that contributes to better understanding.</p>

2 Product description

The TwinCAT 3 Function TF6300 FTP provides features to implement an FTP-Client in the TwinCAT PLC in order to communicate with an FTP-Server via the standardized File Transfer Protocol (FTP). FTP functionalities, e.g. file upload/download, can be accessed via Function Blocks directly from within a PLC program.



The following functions are provided:

Connection establishment

- Authenticating to an FTP-Server
- Connecting to an FTP-Server using active FTP (since version 1.0.8)
- Connecting to an FTP-Server using passive FTP

File transfer functions: Uploading files to an FTP-Server

- Downloading files from an FTP-Server

File functions:

- Rename files on an FTP-Server
- Removing files from an FTP-Server
- Searching files on an FTP-Server
- Creating folders on an FTP-Server
- Removing folders from an FTP-Server

All functionalities can be used out of the PLC with the help of function blocks. You can create connections to different FTP Servers, which can be distinguished and referenced via so-called "handles". Because of these "handles" you don't need to specify connection-related parameters (e.g. IP-Address and port of FTP-Server, authentication data, etc.) every time you want to transfer files.

We recommend to read the following articles of this documentation:

Topic	Content
System requirements [▶ 9]	Describes the system requirements of TwinCAT FTP Client.
Installation	Installation manual of TwinCAT FTP Client.
Licensing	Licensing manual of TwinCAT FTP Client.
Introduction to File Transfer Protocol (FTP) [▶ 17]	Important article about FTP basics. Describes the design of FTP and its specification.
Configure Data-Ports for active FTP [▶ 18]	Describes a feature to configure the used Data-Ports of TwinCAT FTP Client when using active FTP.
Activate an error logfile [▶ 19]	Describes how to activate an error logfile for diagnostics.
Overview about function blocks [▶ 21]	Provides an overview about all PLC Function Blocks of TwinCAT FTP Client.
Samples [▶ 46]	Overview about all available samples. Also provides sample download.

In addition we also recommend to visit our "**Samples**" chapter, in which you can find multiple PLC programs which show how to use TwinCAT FTP Client.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

3 Installation

3.1 System requirements

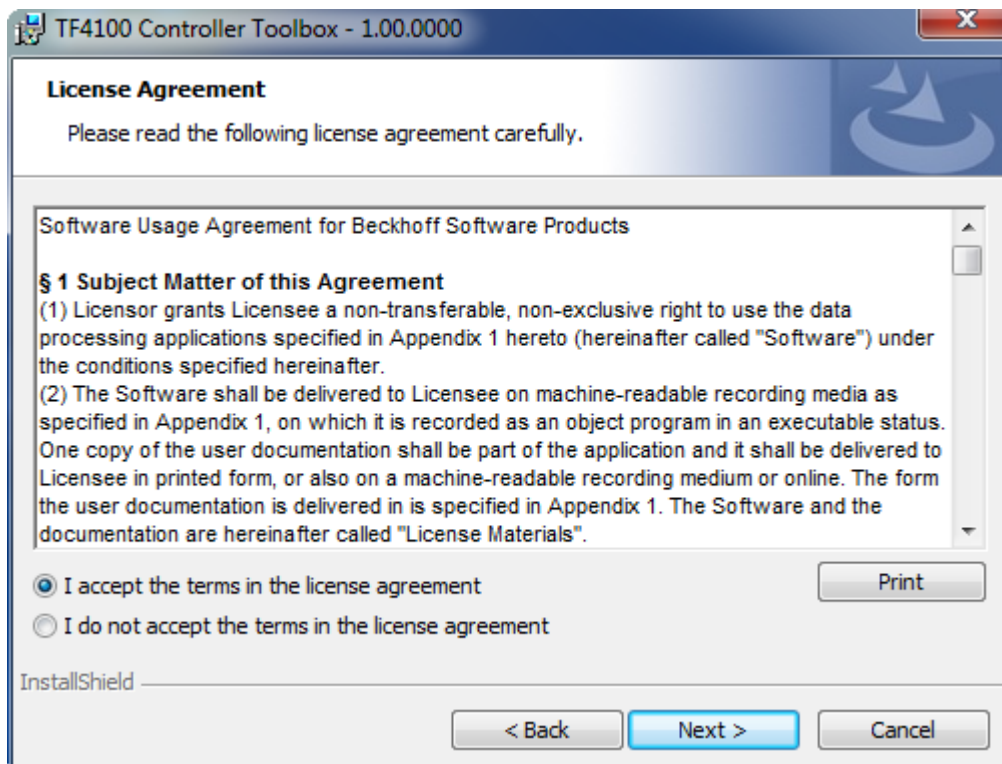
The TwinCAT 3 Function TF6300 FTP is available for WinXP-based Operating Systems (Windows XP, Windows 7, ...) and their Embedded parts (WES2009, WES7, ...).

- **Supported hardware platforms:** 32-bit
- **Supported Operating Systems:** Windows XP, Windows XP Embedded, Windows Embedded Standard 2009, Windows 7 Pro, Windows Embedded Standard 7
- **Supported TwinCAT 3 versions:** since TwinCAT 3.0 Build 3102 (XAE/XAR)
- **Needed TwinCAT 3 licenses:** TC1200 PLC and TF6300 FTP. Alternatively 7-Day trial version. See licensing documentation.
- **.NET Framework version:** 2.0

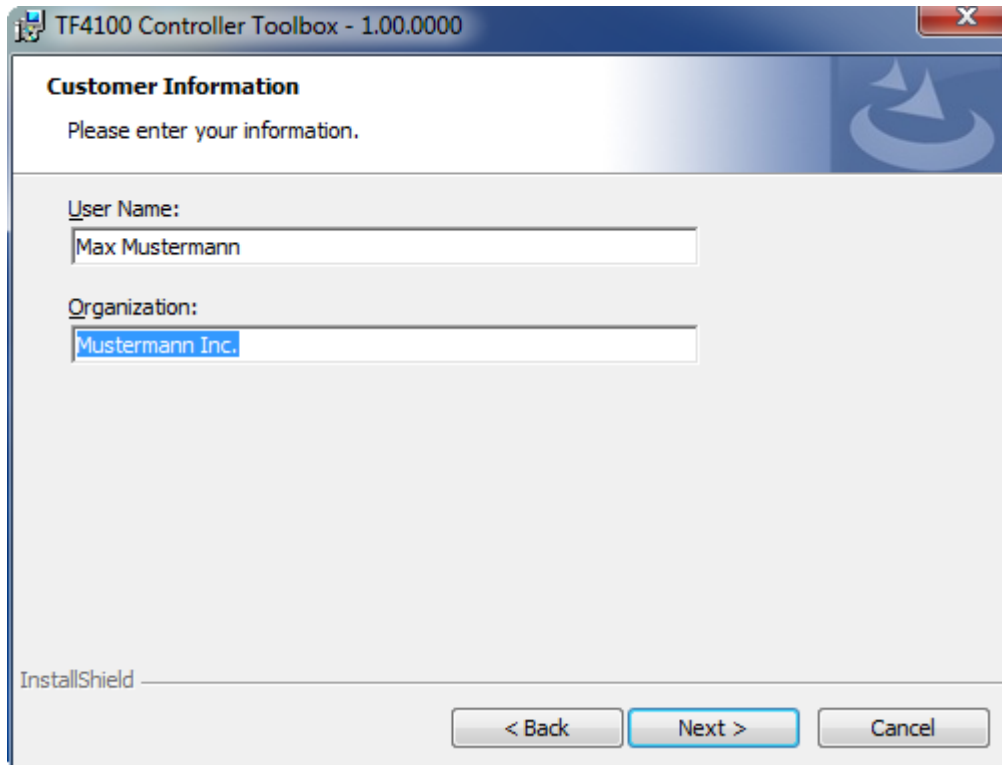
3.2 Installation

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

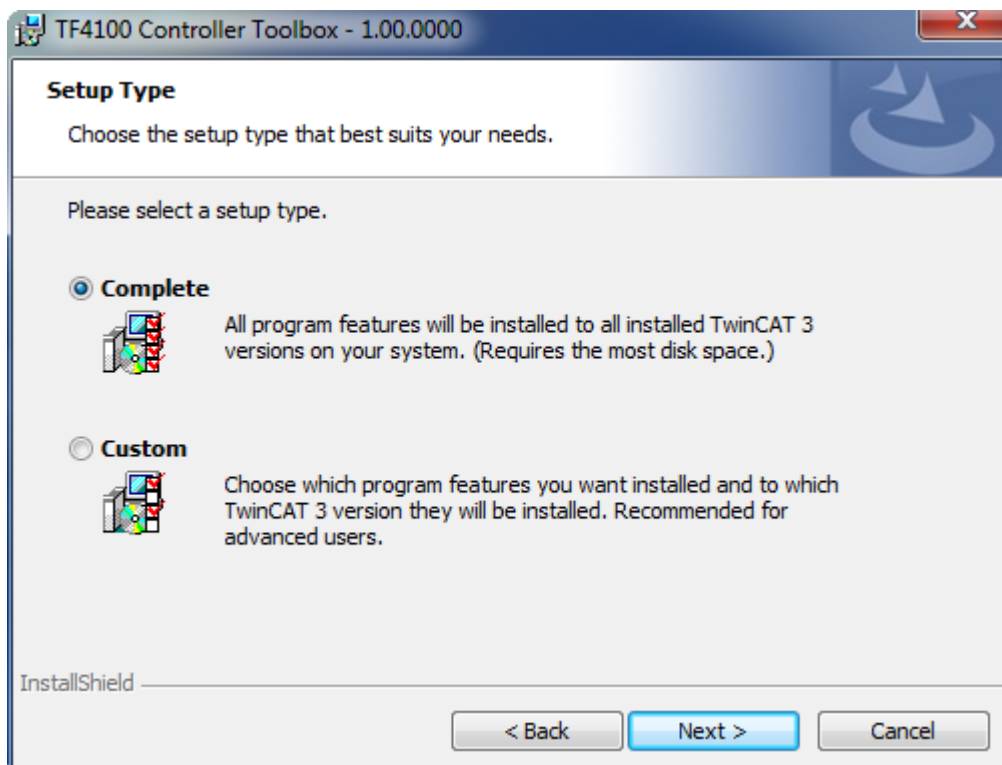
1. Double-click the downloaded setup file "TFxxxx".
Please note: Under Windows 32-bit/64-bit, please start the installation with "Run as Administrator" by right-clicking the setup file and selecting the corresponding option in the context menu.
2. Click on "Next" and accept the license Agreement.



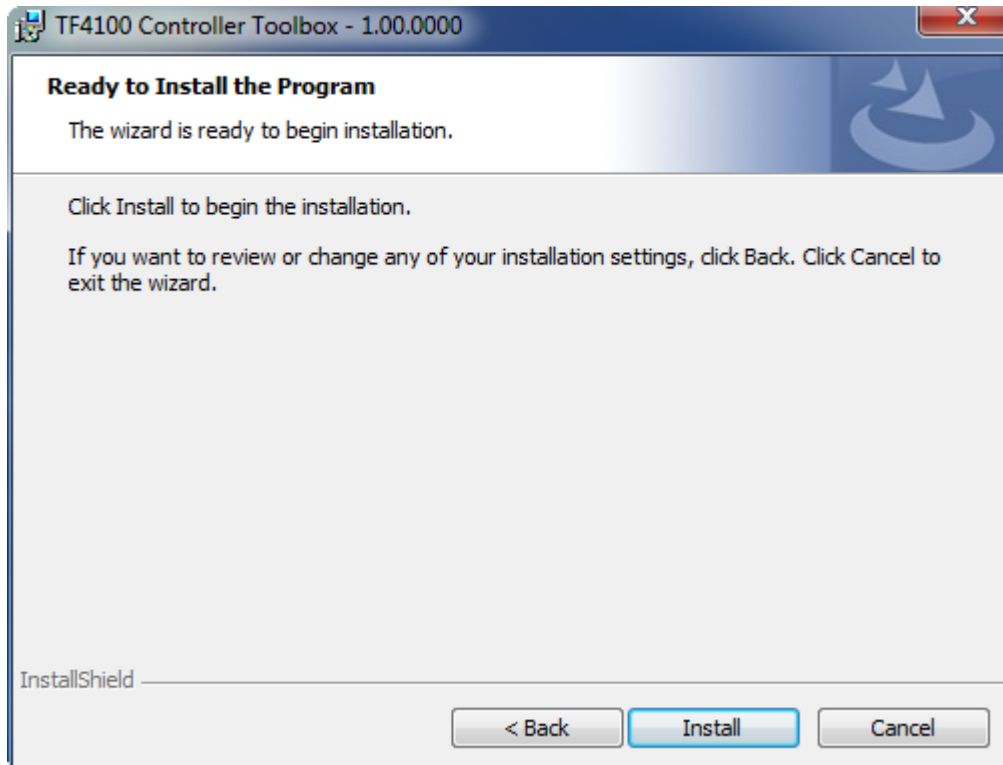
3. Enter your user information in the specified area.



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

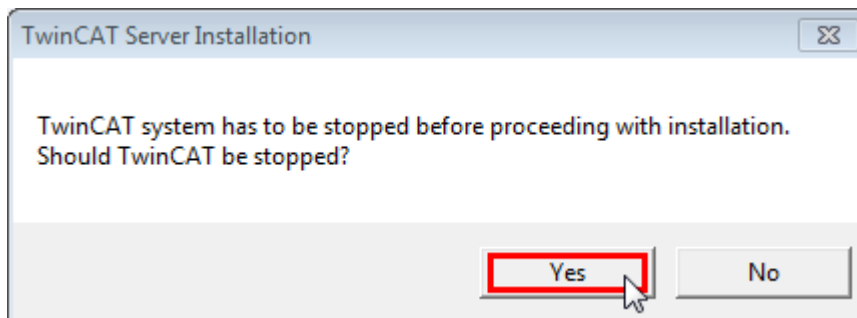


5. Click on **"Install"** after pressing the **"Next"** to start the Installation.

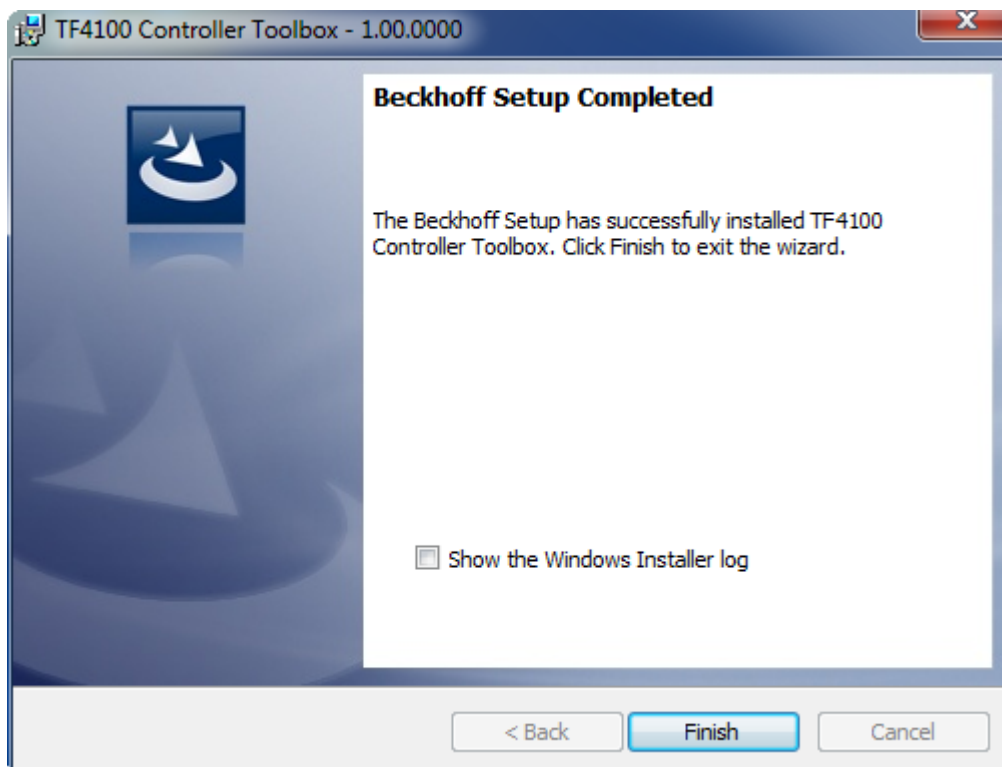


The TwinCAT system has to be stopped before proceeding with installation

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



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



⇒ The installation is complete now.

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

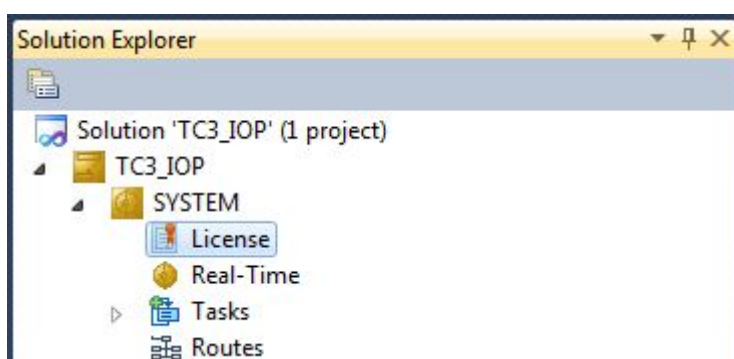
3.3 Licensing

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

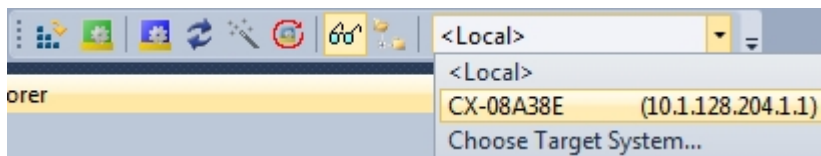
1. Start TwinCAT XAE
2. Open an existing TwinCAT 3 project or create a new project
3. In "Solution Explorer", please navigate to the entry **"SystemLicense"**



- 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	<input checked="" type="checkbox"/> cpu license
TC1100	TC3 IO	<input type="checkbox"/> cpu license
TC1200	TC3 PLC	<input type="checkbox"/> cpu license
TC1210	TC3 PLC / C++	<input type="checkbox"/> cpu license
TC1220	TC3 PLC / C++ / MatSim	<input type="checkbox"/> cpu license
TC1250	TC3 PLC / NC PTP 10	<input type="checkbox"/> cpu license
TC1260	TC3 PLC / NC PTP 10 / NC I	<input type="checkbox"/> cpu license
TC1270	TC3 PLC / NC PTP 10 / NC I / CNC	<input type="checkbox"/> cpu license
TC1300	TC3 C++	<input type="checkbox"/> cpu license
TC1320	TC3 C++ / MatSim	<input type="checkbox"/> cpu license
TE1300	TC3 Scope View Professional	<input checked="" type="checkbox"/> cpu license
TE1400	TC3 Target For Matlab Simulink	<input type="checkbox"/> cpu license

- 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



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

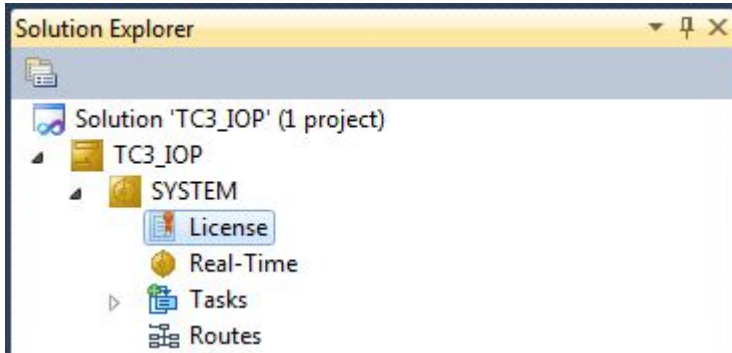
Order No	License	Instances	Current Status
TC1200	TC3 PLC	cpu license	expires on Mar 29, 2012 (trial I...
TF6420	TC3 Database-Server	cpu license	expires on Mar 29, 2012 (trial I...

- Please restart TwinCAT 3 afterwards.

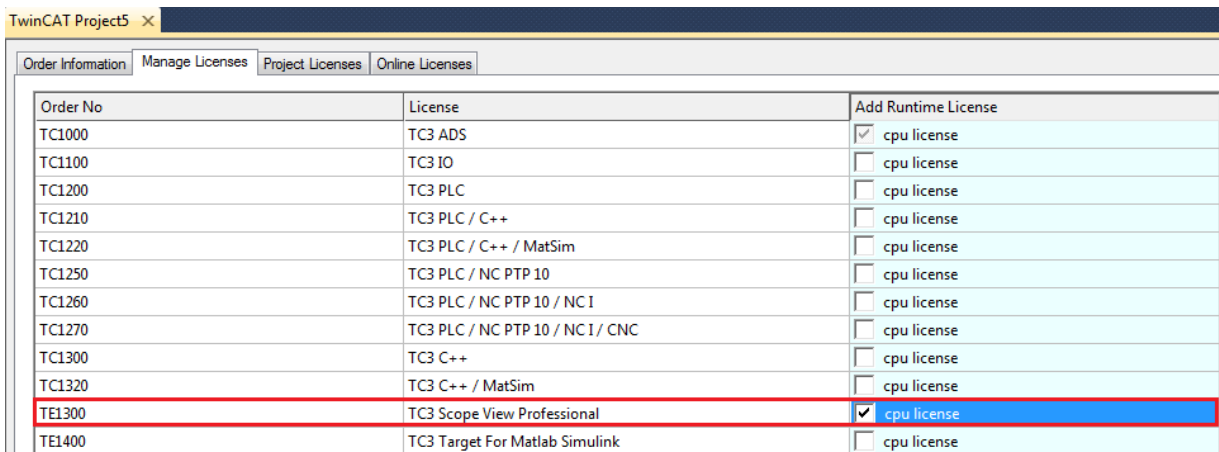
Licensing a full version

- Start TwinCAT XAE
- Open an existing TwinCAT 3 project or create a new project

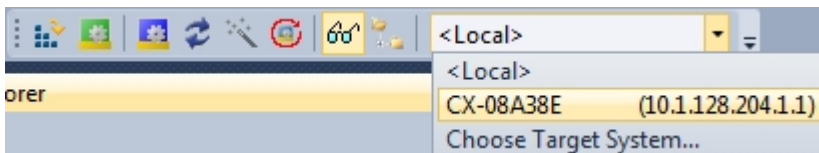
10. In "Solution Explorer", please navigate to the entry **"SYSTEMLicense"**



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

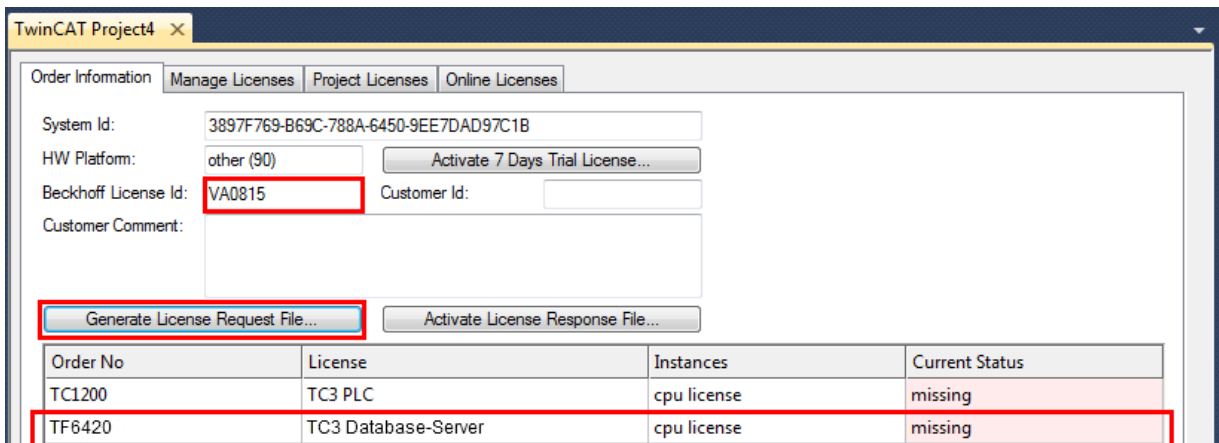


12. **Optional:** If you would like to add a license for a remote device, you first need to connect to the remote device via TwinCAT XAE toolbar

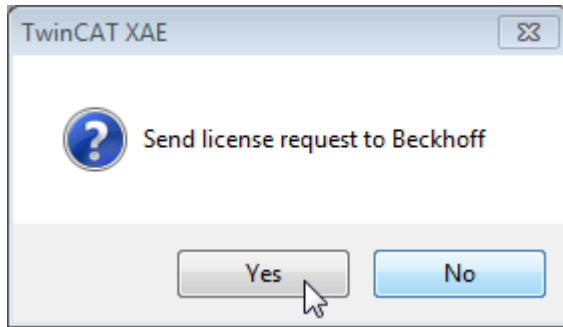


13. Navigate to the **"Order Information"** tab
 The fields "System-ID" and "HW Platform" cannot be changed and just describe the platform for the licensing process in general a TwinCAT 3 license is always bound to these two identifiers:
 the "System-ID" uniquely identifies your system.
 The "HW Platform"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

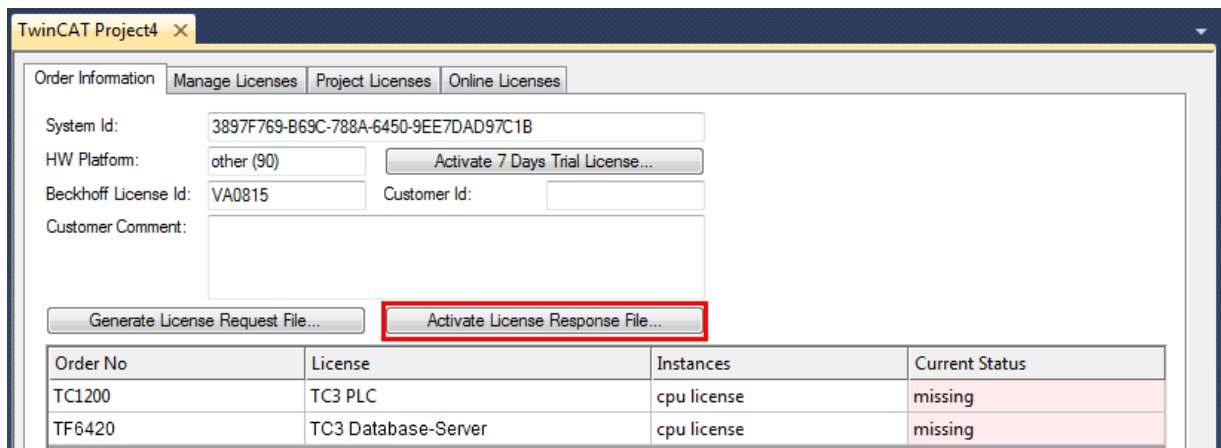
14. Optionally, you may also enter an own order number and description for your convenience



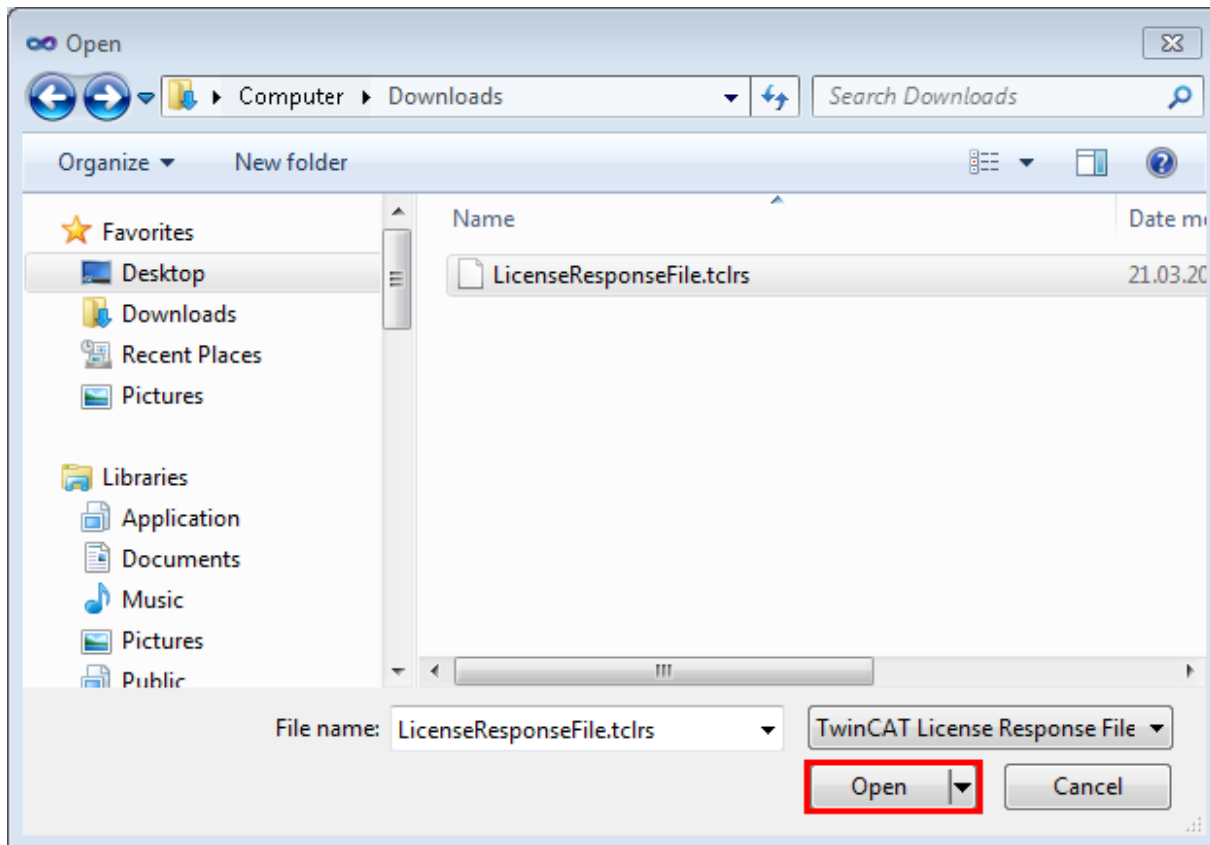
15. enter the "Beckhoff License ID" and click on **"Generate License Request File..."**. If you are not aware of your **"Beckhoff License ID"** please contact your local sales representative.
16. After the license request file has been saved, the system asks whether to send this file via E-Mail to the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



17. After clicking "Yes", the standard E-Mail client opens and creates a new E-Mail message to ["tclicense@beckhoff.com"](mailto: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.

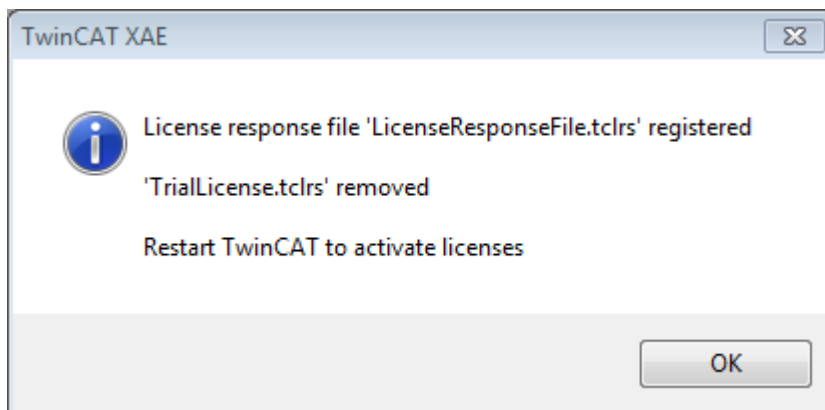


20. Select the received "Licnse response file" and click on "Open"



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..



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

4 Configuration

4.1 Introduction to File Transfer Protocol (FTP)

This chapter of the documentation gives some basic information about the File Transfer Protocol and provides links to other, in-depth articles.

General

The File Transfer Protocol (FTP) is based only on TCP connections and specifies two TCP ports that are important for transferring data.

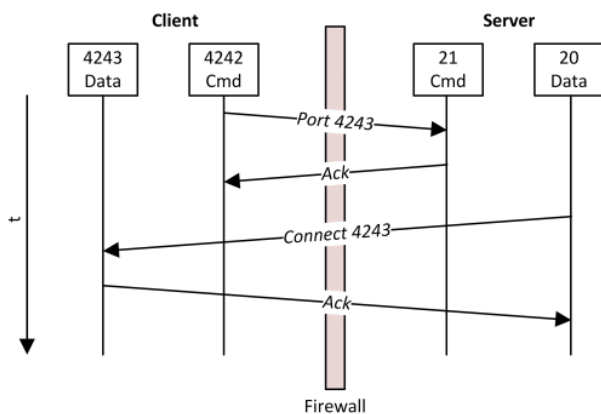
- Port 20/tcp: This port is also known as **Data-Port** and is used for Sending/Receiving files and directory information.
- Port 21/tcp: This port is also known as **Command-Port** and is used to transmit status information between Client and Server.

To send and receive files (Data-Port) and to transmit commands (Command-Port), a separate TCP connection is being used. FTP specifies two connection modes: "**Active FTP**" and "**Passive FTP**". Depending on the connection mode, the ports mentioned above will be initiated differently, which should be discussed in the following.

Please note: TwinCAT FTP Client supports both connection modes since version 1.0.8. Older versions only support "Passive FTP". The connection mode can be selected e.g. via the input parameter eMode of data type E_FTP_ConnMode [▶ 44] in the function block FB_FTP_OpenEx [▶ 24].

Active FTP

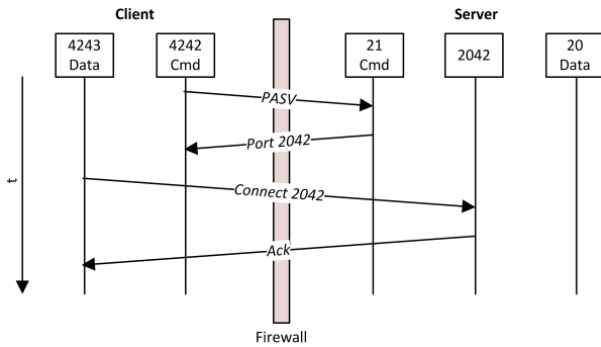
When using the connection mode "Active FTP", the Client connects to the Command-Port of the FTP-Server by using a randomly selected source port N, e.g. 4242/tcp. Subsequently the Client binds itself to port N+1 and notifies the Server of this port. The Server then connects to this port N+1 and uses its Data-Port as the source port.



One of the problems when using active FTP connections is, that the Client itself does not initiate a connection to the Server's Data-Port. Instead, the Client notifies the Server of its own Data-Port (N+1) and then the Server actively connects to this port. If the Client is located behind a Firewall or NAT-device, this could involve additional configuration on the Client's Firewall because the Client's Data-Port must be reachable for the Server (cf. picture "Connect 4243"). The Data-Ports used by TwinCAT FTP-Client can be permanently configured, which makes the Firewall configuration easier. For this special Registry Keys exists to Configure Data-Ports for active FTP [▶ 18].

Passive FTP

This connection mode is often used if the Client is not directly reachable for the Server, e.g. if the Client is located behind a Firewall. When using passive FTP, the Client initiates a connection via two randomly selected TCP ports N (Command-Port) and N+1 (Data-Port). The first port is used to connect to the Server's Command-Port. However, instead of notifying the Server about the Data-Port N+1, so that the Server may actively connect to it (cf. active FTP) the Client transmit the FTP command "PASV". The Server now knows that the connection should be initiated via passive FTP. As a result, the Server opens a (randomly) selected port P as its Data-Port and notifies the Client of this port. The Client then initiates a connection with port P, using its own Data-Port N+1. This connection is then used to transmit files.



You can clearly see, that, on closer examination, the Firewall problem of passive FTP is reversed when compared with active FTP. The Server's Firewall needs to be configured so that the Server's Data-Port is reachable for the Client. Many FTP Servers provide possibilities to configure the used Data-Port, e.g. the Microsoft IIS as shown in the corresponding [KnowledgeBase article about passive FTP in IIS](#).

Sources

Wikipedia: File Transfer Protocol URL: http://de.wikipedia.org/wiki/File_Transfer_Protocol10. April 2011

Microsoft Technet: About Active and Passive FTP Connections: [http://technet.microsoft.com/en-us/library/cc973739\(v=bts.10\).aspx](http://technet.microsoft.com/en-us/library/cc973739(v=bts.10).aspx)04.01.2012

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

4.2 Configure Data-Ports for active FTP

If you use the active FTP mode for connecting to the FTP Server, the Client will select its own Data-Port for the connection establishment and the Server will connect to this port. If the TwinCAT FTP Client should be used in active FTP mode, a port range from which Data-Ports will be chosen can be defined. The TwinCAT FTP Client will choose a free port out of the range to create a connection to the FTP Server.

The port range will be configure with the following three registrykeys.

"HKEY_LOCAL_MACHINE\\Software\\Beckhoff\\TwinCAT FTP Client\\Configuration\\ACTIVE_PortRangeEnabled"

"HKEY_LOCAL_MACHINE\\Software\\Beckhoff\\TwinCAT FTP Client\\Configuration\\ACTIVE_MaxPort"

"HKEY_LOCAL_MACHINE\\Software\\Beckhoff\\TwinCAT FTP Client\\Configuration\\ACTIVE_MinPort"

To activate the port range you first have to set the registry key "ACTIVE_PortRangeEnabled" to 1. Otherwise the TwinCAT FTP Client will use any free port for the connection establishment.

With the registrykeys "ACTIVE_MaxPort" and "ACTIVE_MinPort" it is possible to declare the lower and upper bound of the port range.

Please note: To allow multiple FTP connections, the port range has to be defined with an appropriate size of ports. For example, if the TwinCAT FTP Client connects to three FTP-Servers simultaneously and starts one file transfer on each connection, the port range should at least include three ports.

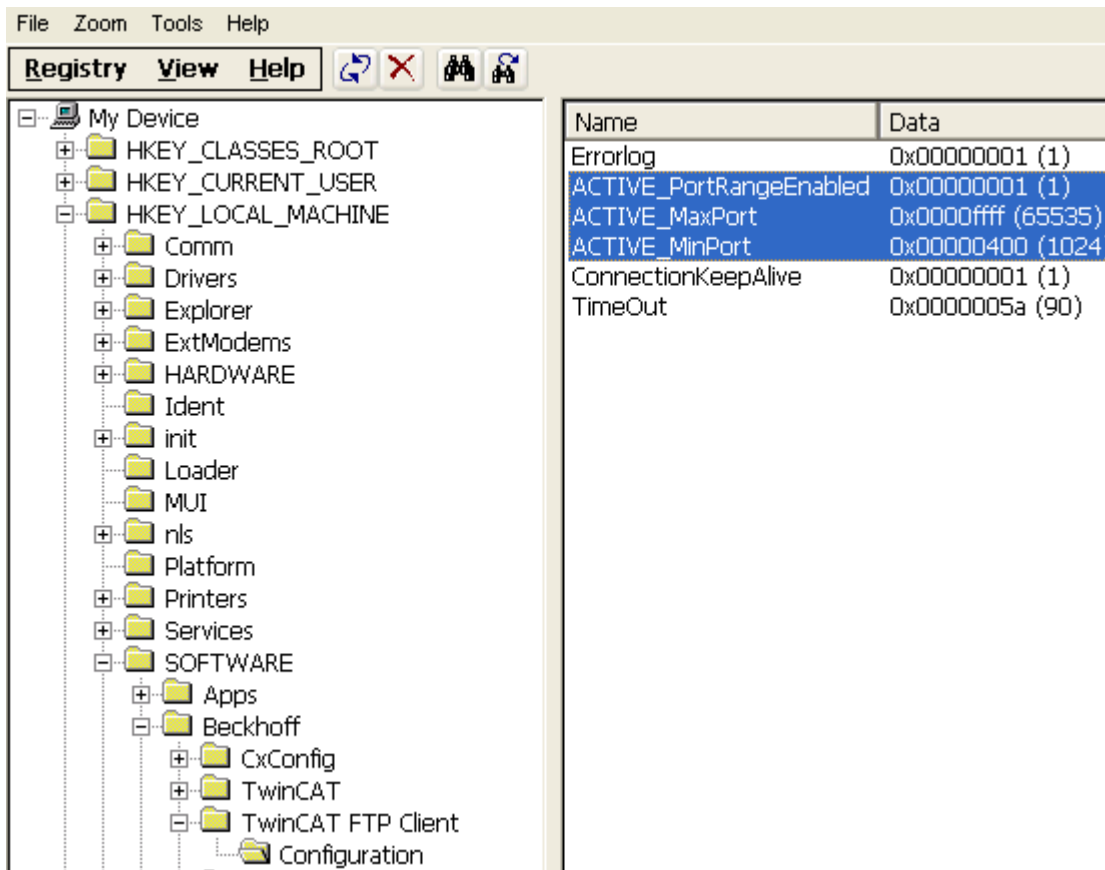


Fig. 1: RegKey2

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

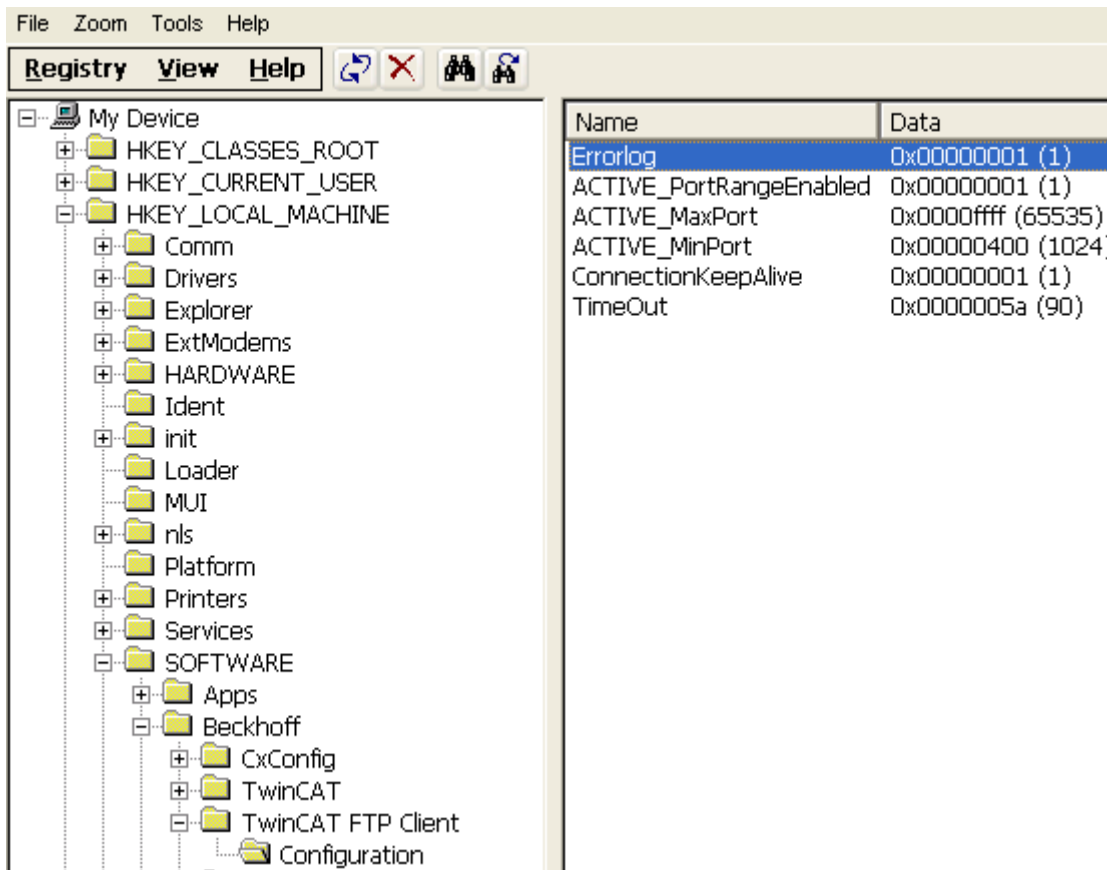
4.3 Activating an error logfile

To localize occurred errors and to get a better description of these errors the TwinCAT FTP Client has the possibility to protocol these errors in a text file.

The text file is called "**TcFTPErrorLog.txt**" and will be created in the folder where the "EXE" of the TwinCAT FTP Client is located.

You are able to activate the functionality by creating the following Registry Key:

"HKEY_LOCAL_MACHINE\\Software\\Beckhoff\\TwinCAT FTP Client\\Configuration\\ErrorLog"




By setting the value 0, the error log will be deactivated after a restart of the TwinCAT System.

By setting the value 1, the error log will be activated and a simple error logging will be performed after a restart of the TwinCAT System

By setting the value 2, the error log will be activated and an extended error logging will be performed after a restart of the TwinCAT System

 Attention	<p>Damage of storage medium</p> <p>To many write cycles to the Compact Flash Card can shorten its service life.</p>
---	--

 Attention	<p>Damage of storage medium</p> <p>Use the function of the Errorlogfile only if you do tests!</p>
---	--

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5 PLC libraries

5.1 Tc2_FTP

Overview

The Tc2_FTP library contains function blocks to control and configure the TwinCAT FTP Client.

Function Blocks

Name	Description
FB_GetStateTcFTPClient [▶ 41]	Call state information.
FB_FTP_HostResolve [▶ 22]	Resolves the given host name to an IPv4 address
FB_FTP_Open [▶ 23]	Open a connection to a FTP Server. (Passive)
FB_FTP_OpenEx [▶ 24]	Open a connection to a FTP Server. (Passive / Active)
FB_FTP_Close [▶ 25]	Close a connection to a FTP Server.
FB_FTP_CloseAll [▶ 26]	Close all connections to a FTP Server
FB_FTP_Info [▶ 27]	Get information from all opened connections
FB_FTP_FileUpload [▶ 28]	Copy a file to a FTP Server. Use an existing connection.
FB_FTP_FileUploadEx [▶ 29]	Copy a file to a FTP Server.
FB_FTP_FileDownload [▶ 30]	Copy a file from a FTP Server to an ADS device. Use an existing connection.
FB_FTP_FileDownloadEx [▶ 32]	Copy a file from a FTP Server to an ADS device.
FB_FTP_DirCreate [▶ 33]	Create a directory on the FTP Server
FB_FTP_DirRemove [▶ 34]	Remove a directory on the FTP Server
FB_FTP_FileList [▶ 35]	Select all files and folder which pass the searchmask.
FB_FTP_FileListEx [▶ 36]	Select all files and folder with additional information which pass the searchmask.
FB_FTP_FileExist [▶ 38]	Find out if the searched file exist on the FTP Server.
FB_FTP_FileRemove [▶ 39]	Remove a file from the FTP Server.
FB_FTP_FileRename [▶ 40]	Rename a file on the FTP Server.

Functions

Name	Description
F_GetVersionTcFTPClient [▶ 42]	Call version information.

Data Types

Name	Description
T_HFTP [▶ 43]	Handle of the FTP Server.
ST_FTP_ConnInfo [▶ 43]	Information of existing FTP connections.
ST_FTP_FileDetails [▶ 44]	Returning file information of the FTP "LIST" command
E_FTP_ConnMode [▶ 44]	FTP connection modes (Passiv / Aktiv).

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.2 Function Blocks

5.2.1 FB_FTP_HostResolve



With the function block FB_FTP_HostResolve it is possible to resolve hostnames to it specified IPv4 address.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId := '';
  sHostname   : T_MaxString := '';
  bExecute    : BOOL;
  tTimeout    : TIME       := T#15s;
END_VAR
```

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

sHostname : Is a string containing the name of the host which will be resolved.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrID     : UDINT;
  sIPv4Addr  : T_IPv4Addr;
END_VAR
```

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

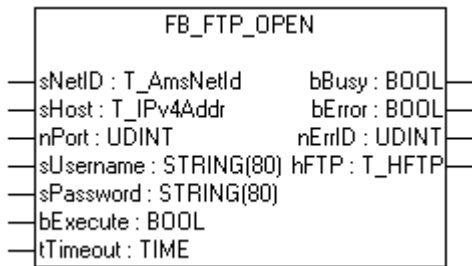
nErrID : Supplies the ErrorCode [▶ 54] when the bError output is set.

sIPv4Addr : Returns the IPv4 address of the given host name.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.2.2 FB_FTP_Open



You can open a connection to a FTP-Server with this function block FB_FTP_Open. You can use the returned handle for further actions at the FTP-Server. The connection mode "passive FTP" will be used.

VAR_INPUT

```

VAR_INPUT
  sNetID      : T_AmsNetId := '';
  sHost       : T_IPv4Addr := '127.0.0.1';
  nPort       : UDINT      := 21;
  sUsername   : STRING     := '';
  sPassword   : STRING     := '';
  bExecute    : BOOL;
  tTimeout    : TIME       := T#15s;
END_VAR
    
```

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

sHost : Is a string containing the IPv4 address of the FTP-Server.

nPort : FTP port (default 21).

sUsername : Username for the FTP Server authentication

sPassword : Password for the FTP Server authentication

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrID    : UDINT;
  hFTP      : T_HFTP;
END_VAR
```

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [► 54] when the bError output is set.

hFTP : Handle of a FTP Server connection.

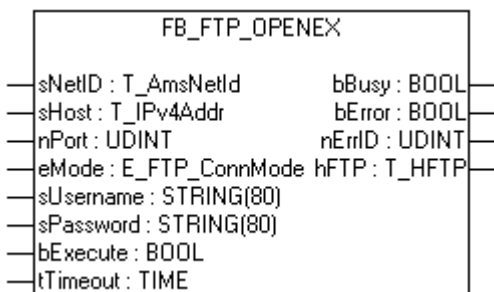
Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

Also see about this

📄 T_HFTP [► 43]

5.2.3 FB_FTP_OpenEX



You can open a connection to a FTP-Server with this function block FB_FTP_OpenEx. You can use the returned handle for further actions at the FTP-Server. You can choose between two connection modes like active and passive FTP.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId := '';
  sHost       : T_IPv4Addr := '127.0.0.1';
  nPort       : UDINT      := 21;
  eMode       : E_FTP_ConnMode := eConnMode_PASSIVE;
  sUsername   : STRING     := '';
  sPassword   : STRING     := '';
  bExecute    : BOOL;
  tTimeout    : TIME       := T#15s;
END_VAR
```

E_FTP_ConnMode [► 44]

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

sHost : Is a string containing the IPv4 address of the FTP-Server.

nPort : FTP port (default 21).

eMode : FTP connection mode (active / passive).

sUsername : Username for the FTP Server authentication

sPassword : Password for the FTP Server authentication

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrID    : UDINT;
  hFTP      : T_HFTP;
END_VAR
```

T_HFTP [▶ 43]

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

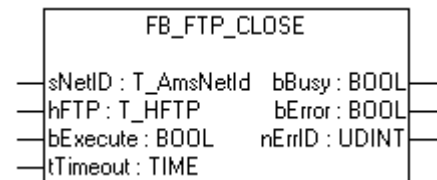
nErrID : Supplies the ErrorCode [▶ 54] when the bError output is set.

hFTP : Handle of a FTP Server connection.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.2.4 FB_FTP_Close



You can close existing connections to a FTP-Server with this function block FB_FTP_Close

VAR_INPUT

```
VAR_INPUT
  sNetID : T_AmsNetId := '';
  hFTP   : T_HFTP;
  bExecute : BOOL;
```

```

    tTimeout      : TIME      := T#15s;
END_VAR

```

T_HFTP [▶ 43]

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

hFTP : Handle to a FTP Server.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```

VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrID     : UDINT;
END_VAR

```

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

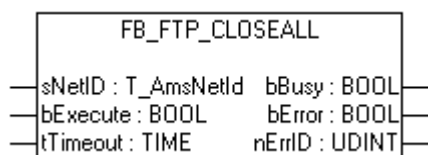
bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [▶ 54] when the bError output is set.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.2.5 FB_FTP_CloseAll



You can close all existing connections to a FTP-Server with this function block FB_FTP_CloseAll.

VAR_INPUT

```

VAR_INPUT
    sNetID      : T_AmsNetId := '';
    bExecute     : BOOL;
    tTimeout     : TIME      := T#15s;
END_VAR

```

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

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

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

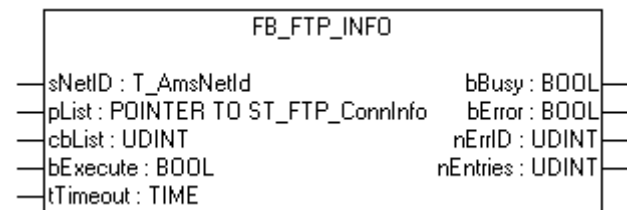
bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [▶ 54] when the bError output is set.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.2.6 FB_FTP_Info



You can get information about the existing connections to the FTP-Server with this function block FB_FTP_Info.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId      := '';
  pList       : POINTER TO ST_FTP_ConnInfo;
  cbList      : UDINT           := 0;
  bExecute    : BOOL;
  tTimeout    : TIME           := T#15s;
END_VAR
```

ST_FTP_ConnInfo [▶ 43]

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

pList : Pointer address to a variable of the type ST_FTP_ConnInfo. This variable can also be an array of the type ST_FTPConnInfo.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrID     : UDINT;
  nEntries   : UDINT;
END_VAR
```

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

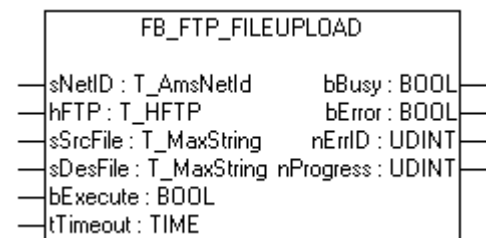
nErrID : Supplies the ErrorCode [► 54] when the bError output is set.

nEntries : Number of existing connections.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.2.7 FB_FTP_FileUpload



You can upload files to a FTP-Server with this function block FB_FTP_FileUpload. This function block uses an existing connection to a FTP-Server.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId := '';
  hFTP       : T_HFTP;
  sSrcFile    : T_MaxString := '';
  sDesFile    : T_MaxString := '';
  bExecute    : BOOL;
  tTimeout    : TIME      := T#15s;
END_VAR
```

T_HFTP [► 43]

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

hFTP : Handle to a FTP Server.

sSrcFile : Source file which will be copied to the FTP Server.

sDesFile : Path and name of the destination file on the FTP Server.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrID    : UDINT;
  nProgress  : UDINT;
END_VAR
```

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

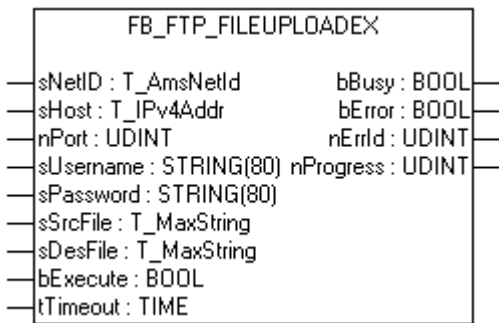
nErrID : Supplies the ErrorCode [▶ 54] when the bError output is set.

nProgress : Shows the current status of the data transmission in percent.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.2.8 FB_FTP_FileUploadEx



You can upload files to a FTP-Server with this function block FB_FTP_FileUpload. The connection mode "passive FTP" will be used.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId := '';
  sHost       : T_IPv4Addr := '127.0.0.1';
  nPort       : UDINT      := 21;
  sUsername   : STRING     := '';
  sPassword   : STRING     := '';
  sSrcFile    : T_MaxString := '';
  sDesFile    : T_MaxString := '';
  bExecute    : BOOL;
  tTimeout    : TIME       := T#15s;
END_VAR
```

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

sHost : Is a string containing the IPv4 address of the FTP-Server.

nPort : FTP port (default 21).

sUsername : Username for the FTP Server authentication

sPassword : Password for the FTP Server authentication

sSrcFile : Source file which will be copied to the FTP Server.

sDesFile : Path and name of the destination file on the FTP Server.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```

VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrID    : UDINT;
  nProgress  : UDINT;
END_VAR
    
```

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

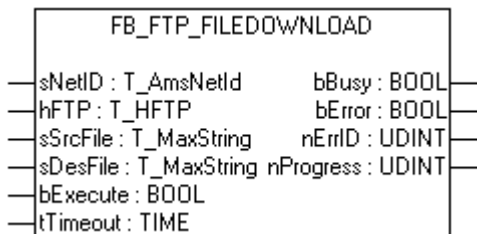
nErrID : Supplies the ErrorCode [▶ 54] when the bError output is set.

nProgress : Shows the current status of the data transmission in percent.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.2.9 FB_FTP_FileDownload



You can download files from a FTP-Server with this function block FB_FTP_FileDownload. This function block uses an existing connection to a FTP-Server.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId := '';
  hFTP        : T_HFTP
  sSrcFile    : T_MaxString := '';
  sDesFile    : T_MaxString := '';
  bExecute    : BOOL;
  tTimeout    : TIME      := T#15s;
END_VAR
```

T_HFTP [► 43]

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

hFTP : Handle to a FTP Server.

sSrcFile : Source file which will be copied from the FTP Server.

sDesFile : Path and name of the destination file on the ADS device.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  nErrID      : UDINT;
  nProgress   : UDINT;
END_VAR
```

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

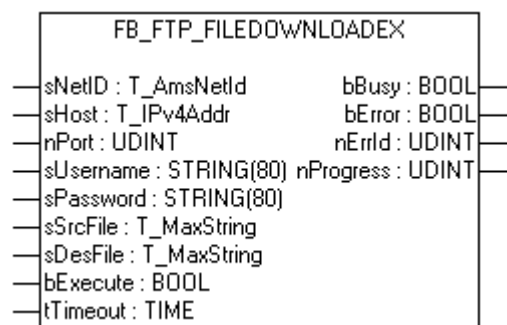
nErrID : Supplies the ErrorCode [► 54] when the bError output is set.

nProgress : Shows the current status of the data transmission in percent. At **CE FTP Server** no values between 0% and 100% will be returned.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.2.10 FB_FTP_FileDownloadEx



You can download files from a FTP-Server with this function block FB_FTP_FileDownload. The connection mode "passive FTP" will be used.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId := '';
  sHost       : T_IPv4Addr := '127.0.0.1';
  nPort       : UDINT      := 21;
  sUsername   : STRING     := '';
  sPassword   : STRING     := '';
  sSrcFile    : T_MaxString := '';
  sDesFile    : T_MaxString := '';
  bExecute    : BOOL;
  tTimeout    : TIME       := T#15s;
END_VAR
```

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

sHost : Is a string containing the IPv4 address of the FTP-Server.

nPort : FTP port (default 21).

sUsername : Username for the FTP Server authentication

sPassword : Password for the FTP Server authentication

sSrcFile : Source file which will be copied from the FTP Server.

sDesFile : Path an name of the destination file on the ADS device.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrID     : UDINT;
  nProgress  : UDINT;
END_VAR
```

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

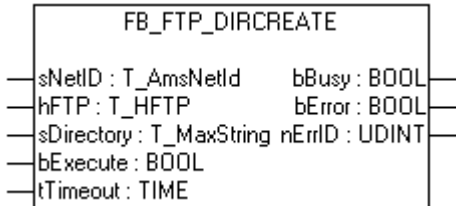
nErrID : Supplies the ErrorCode [► 54] when the bError output is set.

nProgress : Shows the current status of the data transmission in percent. At **CE FTP Server** no values between 0% and 100% will be returned.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.2.11 FB_FTP_DirCreate



You can create folder on e FTP-Server with this function block FB_FTP_DirCreate. This function block uses an existing connection to a FTP-Server.

VAR_INPUT

```

VAR_INPUT
  sNetID      : T_AmsNetId := '';
  hFTP        : T_HFTP     := 0;
  sDirName    : T_MaxString := '';
  bExecute    : BOOL;
  tTimeout    : TIME      := T#15s;
END_VAR
  
```

T_HFTP [▶ 43]

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

hFTP : Handle to a FTP Server.

sDirectory : The name of the new directory.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```

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

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

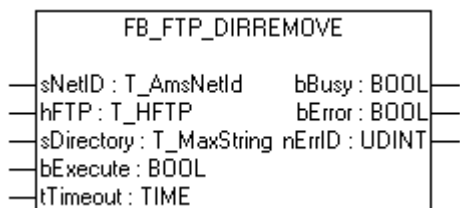
bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [▶ 54] when the bError output is set.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.2.12 FB_FTP_DirRemove



You can remove existing folders from a FTP-Server with this function block FB_FTP_DirRemove. This function block uses an existing connection to a FTP-Server.

VAR_INPUT

```

VAR_INPUT
    sNetID      : T_AmsNetId := '';
    hFTP       : T_HFTP;
    sDirectory  : T_MaxString := '';
    bExecute   : BOOL;
    tTimeout   : TIME      := T#15s;
END_VAR

```

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

hFTP : Handle to a FTP Server.

sDirectory : The name of the directory which will be removed.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```

VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrID    : UDINT;
END_VAR

```

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [► 54] when the bError output is set.

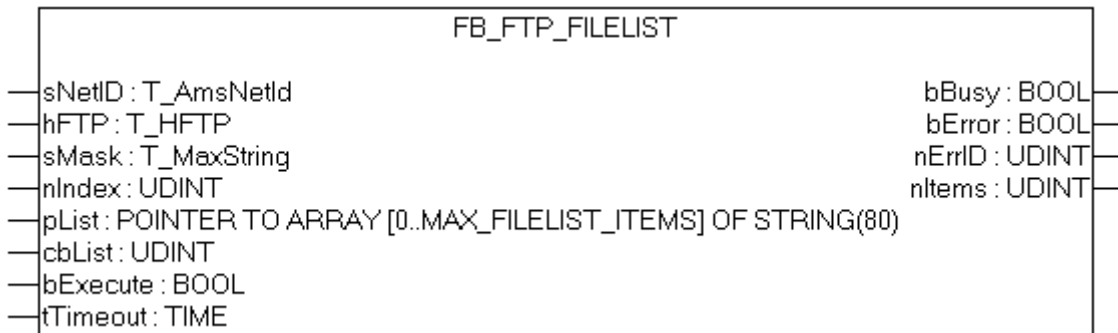
Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

Also see about this

 T_HFTP [▶ 43]

5.2.13 FB_FTP_FileList



You can read a list of file- and foldernames from a FTP-Server with this function block FB_FTP_FileList. This function block uses an existing connection to a FTP-Server.

VAR_INPUT

```

VAR_INPUT
  sNetID      : T_AmsNetId           := '';
  hFTP        : T_HFTP;
  sMask       : T_MaxString          := '';
  nIndex      : UDINT;
  pList       : POINTER TO ARRAY [0..MAX_FILELIST_ITEMS] OF STRING(80);
  cbList      : UDINT;
  bExecute    : BOOL;
  tTimeout    : TIME                 := T#20s;
END_VAR
    
```

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

hFTP : Handle to a FTP Server.

sMask : Searchmask to filter the files.

Beispiel	
**	Read all filenames
*	Read all filenames and all foldernames
*.txt	Read all filenames with the extension "TXT"
\Test*.*	Read all filenames of the subfolder Test

nIndex : Index of the first read filename.

pList : Pointer to a stringarray where the filenames will be stored

cbList : The sizeof the stringarray in bytes.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```

VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrID     : UDINT;
  nItems     : UDINT;
END_VAR

```

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [► 54] when the bError output is set.

nItems : Shows the number of all founded files or folders.

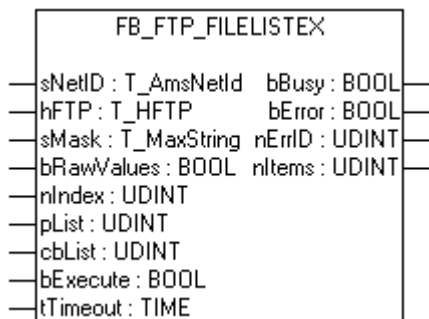
Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

Also see about this

📄 T_HFTP [► 43]

5.2.14 FB_FTP_FileListEx



You can read a list of file- and folder names from a FTP-Server with this function block FB_FTP_FileListEx. This function block uses an existing connection to a FTP-Server. This function block returns in contrast to FB_FTP_FileList more detailed information about the files.

VAR_INPUT

```

VAR_INPUT
  sNetID      : T_AmsNetId := '';
  hFTP       : T_HFTP;
  sMask      : T_MaxString := '';
  bRawValues  : BOOL;
  nIndex     : UDINT;
  pList      : UDINT;
  cbList     : UDINT;
  bExecute   : BOOL;
  tTimeout   : TIME      := T#20s;
END_VAR

```

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

hFTP : Handle to a FTP Server.

sMask : Searchmask to filter the files.

Beispiel	
.	Read all filenames
*	Read all filenames and all foldernames
*.txt	Read all filenames with the extension "TXT"
\Test*.*	Read all filenames of the subfolder Test

bRawValues : The format of the returning filelist.

bRawValues := TRUE information will be returned as an T_MaxString array.

bRawValues := FALSE informationen will be returned as an ST_FTP_FileDetails [▶ 44] array. (not supported for every FTP Server)

nIndex : Index of the first read filename.

pList : Pointer to a stringarray where the filenames will be stored

cbList : The sizeof the stringarray in bytes.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrID    : UDINT;
  nItems    : UDINT;
END_VAR
```

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [▶ 54] when the bError output is set.

nItems : Shows the number of all founded files or folders.

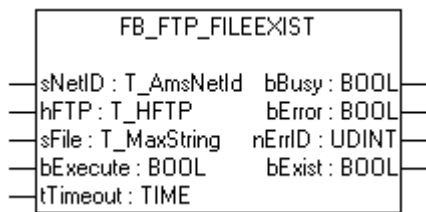
Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

Also see about this

📖 T_HFTP [▶ 43]

5.2.15 FB FTP_FileExist



You can check, if a destined file exist on a FTP-Server, with this function block FB FTP_FileExist. This function block uses an existing connection to a FTP-Server.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId := '';
  hFTP       : T_HFTP;
  sFile      : T_MaxString := '';
  bExecute   : BOOL;
  tTimeout   : TIME      := T#15s;
END_VAR
```

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

hFTP : Handle to a FTP Server.

sFile : The name of the searched file with path.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrID    : UDINT;
  bExist    : BOOL;
END_VAR
```

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [► 54] when the bError output is set.

bExist : Becomes TRUE, if the searched file exist.

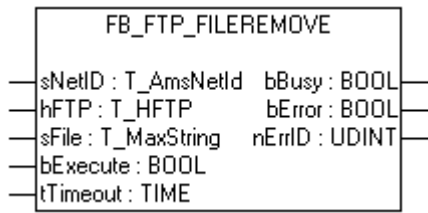
Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

Also see about this

📖 T_HFTP [► 43]

5.2.16 FB_FTP_FileRemove



You can remove files from a FTP-Server with this function block FB_FTP_FileRemove. This function block uses an existing connection to a FTP-Server.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId := '';
  hFTP       : T_HFTP;
  sFile      : T_MaxString := '';
  bExecute   : BOOL;
  tTimeout   : TIME       := T#15s;
END_VAR
```

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

hFTP : Handle to a FTP Server.

sFile : The name of the file, which will be removed.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

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

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [▶ 54] when the bError output is set.

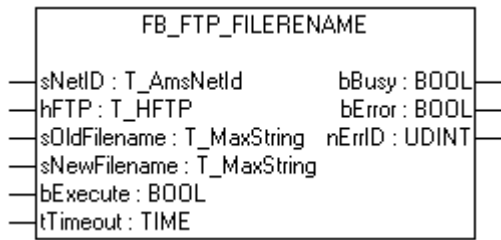
Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

Also see about this

📖 T_HFTP [▶ 43]

5.2.17 FB FTP_FileRename



You can rename files from a FTP-Server with this function block FB FTP_FileRename. This function block uses an existing connection to a FTP-Server.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId := '';
  hFTP        : T_HFTP;
  sOldFilename : T_MaxString := '';
  sNewFilename : T_MaxString := '';
  bExecute     : BOOL;
  tTimeout    : TIME      := T#15s;
END_VAR
```

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

hFTP : Handle to a FTP Server.

sOldFilename : The old filename.

sNewFilename : The new filename.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

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

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [► 54] when the bError output is set.

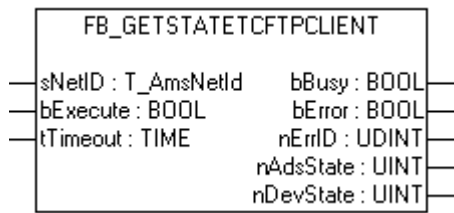
Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

Also see about this

T_HFTP [▶ 43]

5.2.18 FB_GetStateTcFTPClient



The function block allows to get the current state of the FTP Client.

VAR_INPUT

```
VAR_INPUT
    sNetID      : T_AmsNetID;
    bExecute    : BOOL;
    tTimeout    : TIME;
END_VAR
```

sNetID : Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

bExecute : The command is executed with the rising edge.

tTimeout : States the time before the function is cancelled.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrID     : UDINT;
    nAdsState  : UINT;
    nDevState  : UINT;
END_VAR
```

bBusy : The command is in the process of being transmitted by ADS. No new command will be accepted as long as "bBusy" remains TRUE.

bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [▶ 54] when the bError output is set.

nAdsState : Contains the state identification code of the ADS target device. The codes returned here are specified for all ADS servers:

- ADSSTATE_INVALID =0 ;
- ADSSTATE_IDLE =1 ;
- ADSSTATE_RESET =2 ;
- ADSSTATE_INIT =3 ;
- ADSSTATE_START =4 ;
- ADSSTATE_RUN =5 ;
- ADSSTATE_STOP =6 ;
- ADSSTATE_SAVECFG =7 ;
- ADSSTATE_LOADCFG =8 ;
- ADSSTATE_POWERFAILURE =9 ;
- ADSSTATE_POWERGOOD =10 ;

- ADSSTATE_ERROR =11;

nDevState : Contains the specific state identification code of the ADS target device. The codes returned here are supplementary information specific to the ADS device.

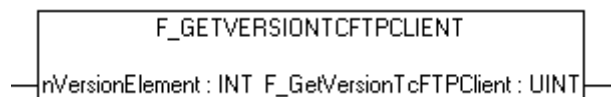
- 1 = TwinCAT FTP Client is started

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.3 Functions [obsolete]

5.3.1 F_GetVersionTcFTPClient



This function can be used to read PLC library version information.

FUNCTION F_GetVersionTcFTPClient: UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Version element to be read. Possible parameters:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.4 Data Types

5.4.1 T_HFTP

VAR_INPUT

```

TYPE T_HFTP :
STRUCT
    hClient      : UDINT;
END_STRUCT
END_TYPE
    
```

hClient : Handle of the FTP connection.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.4.2 ST_FTP_ConnInfo

VAR_INPUT

```

TYPE ST_FTP_ConnInfo :
STRUCT
    sHost        : T_IPv4Addr;
    nPort        : UDINT;
    hFTP         : T_HFTP;
    sUsername    : STRING;
END_STRUCT
END_TYPE
    
```

sHost : IPv4-address of the FTP Server.

nPort : FTP port .

hFTP : Handle of the FTP connection.

sUsername : Username of the connected User.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.4.3 ST_FTP_FileDetails

VAR_INPUT

```

TYPE ST_FTP_FileDetails:
STRUCT
  bDir      : BOOL;
  sPermission : STRING(10);
  nSize     : UDINT;
  nFilecode  : UDINT;
  tTimestamp : DT;
  sOwner    : STRING(79);
  sGroup    : STRING(79);
  sFilename  : STRING(79);
END_STRUCT
END_TYPE

```

bDir : Shows if it is a file or a folder. (won't be supported by every FTP Server)

sPermission : Shows the permissions of the file/folder. (won't be supported by every FTP Server)

nSize : Supply the size of the file.

nFilecode : Supply the file code. (won't be supported by every FTP Server)

tTimestamp : Returns the timestamp of the file/folder.

sOwner : Returns the owner of the file/folder. (won't be supported by every FTP Server)

sGroup : Shows the group of the file/folder. (won't be supported by every FTP Server)

sFilename : Shows the name of the file/folder.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.4.4 E_FTP_ConnMode

```

TYPE E_FTP_ConnMode : (
  eConnMode_PASSIVE := 0,
  eConnMode_ACTIVE  := 1
);
END_TYPE

```

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

5.5 Constants

5.5.1 Konstanten

```

AMSPORT_FTPADSSRV      : UINT      := 10900;

FTPADS_IGR_CONNOPEN    : UDINT      := 16#100;
FTPADS_IGR_RESOLVEDNS  : UDINT      := 16#101;
FTPADS_IGR_CONNCLOSE   : UDINT      := 16#200;
FTPADS_IGR_CONNCLOSEALL : UDINT     := 16#201;

FTPADS_IGR_CONNINFO    : UDINT      := 16#300;

FTPADS_IGR_FILEUPLOAD  : UDINT      := 16#10000;
FTPADS_IGR_FILEDOWNLOAD : UDINT     := 16#20000;

FTPADS_IGR_DIRREMOVE   : UDINT      := 16#30000;
FTPADS_IGR_DIRCREATE   : UDINT      := 16#30001;

FTPADS_IGR_FILEEXIST   : UDINT      := 16#40000;
FTPADS_IGR_FILERENAME  : UDINT      := 16#40001;
FTPADS_IGR_FILEREMOVE  : UDINT      := 16#40002;
FTPADS_IGR_FILELIST    : UDINT      := 16#40003;
FTPADS_IGR_FILELISTEX  : UDINT      := 16#40004;

MAX_FTP_CONNECTIONS   : UDINT      := 25;
MAX_FILELIST_ITEMS    : UDINT      := 255;

DEFAULT_FTP_PORT       : UDINT      := 21;
    
```

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

6 Samples

6.1 Samples

The following table shows all available samples.

No.	Sample
1	Uploading a file to a FTP-Server [▶ 46]
2	Downloading a file from a FTP-Server [▶ 47]
3	Reading connection information [▶ 49]
4	Removing a file from a FTP-Server [▶ 50]
5	Reading a file list from a FTP-Server [▶ 51]

Please note:

All samples can also be downloaded as a single TwinCAT 3 Solution. The following table provides download links for each TwinCAT version.

TwinCAT Version	Sample download
3.0	Download
3.1	Download

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

6.2 TwinCAT FTP Client: Upload of a file to a FTP Server

This sample will illustrate uploading a file from an ADS device to a FTP Server.

To execute this sample you have to create a user with the username "TestUser" and the password "TestPwd123" on the FTP Server.

If you set a rising edge at the input variable "bExecute" with the help of the variable "startstop", the file "\Program Files\TestFile1.txt" from the ADS device will be copied to the FTP Server. On the FTP Server the name of the file will be set to "\TestFolder\File1.txt". The folder "TestFolder" will be created if it isn't available.

Notice that the user "TestUser" has rights for reading and writing.

Variable declaration

```
PROGRAM MAIN
VAR
  FB_FTPFileUpload1 : FB_FTP_FileUploadEx;
  startstop         : BOOL;
  busy              : BOOL;
  err               : BOOL;
  errid            : UDINT;
  progress         : UDINT;
END_VAR
```

SPS Programm

```
FB_FTPFileUpload1(
  sNetID:= ,
  sHost:= '172.16.235.100',
  nPort:= DEFAULT_FTP_PORT,
  sUsername:= 'TestUser',
  sPassword:= 'TestPwd123',
  sSrcFile:= '\Program Files\TestFile1.txt',
  sDesFile:= '\TestFolder\File1.txt',
  bExecute:= startstop,
  tTimeout:= T#15s,
  bBusy=> busy,
  bError=> err,
  nErrId=> errid,
  nProgress=> progress);
```

The following PLC libraries need to be referenced for this sample: **Tc2_FTP**, Tc2_System, Tc2_Standard, Tc3_Interfaces, Tc3_Modules.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

6.3 TwinCAT FTP Client: Downloading a file from a FTP Server to an ADS device

Download: [Beispiel1 Hochladen einer Datei auf einen FTP Server](#)

This sample will illustrate the downloading of files from a FTP Server.

To execute this sample you have to create a user with the username "TestUser" and the password "TestPwd123" on the FTP Server. If you set a rising edge at the variable "startstop", the downloading will be started. At first a connection to a FTP Server will be opened with the function block FB_FTP_Open. After that the given file "\TestFolder\File1.txt" from the FTP Server will be downloaded with the function block FB_FTP_FileDownload. The following path "\Program Files\TestFile1.txt" is the place where the file will be stored on the ADS device. At the end the connection to the FTP Server will be closed with the function block FB_FTP_Close.

Notice that the user "TestUser" has rights for reading and writing.

Variable declaration

```
PROGRAM MAIN
VAR
  RisingEdge      : R_TRIG;
  startstop       : BOOL;
  state           : BYTE;
  FB_FTP_Open1    : FB_FTP_Open;
  FB_FTP_FileDownload1 : FB_FTP_FileDownload;
  FB_FTP_Close1   : FB_FTP_Close;
  busy            : BOOL;
  err             : BOOL;
  errid           : UDINT;
  handle          : T_HFTP;
  progress        : UDINT;
END_VAR
```

PLC Program

```

RisingEdge(CLK:= startstop);
IF RisingEdge.Q THEN
    state := 1;
END_IF

CASE state OF
    0:
        ;

    1:
        FB_FTP_Open1(
            sNetID:= '5.0.252.142.1.1',
            sHost:= '172.16.9.223',
            nPort:= 21,
            sUsername:= 'TestUser',
            sPassword:= 'TestPwd123',
            bExecute:= TRUE,
            tTimeout:= T#15s,
            bBusy=> busy,
            bError=> err,
            nErrId=> errid,
            hFTP=> handle);

        IF NOT busy AND NOT err THEN
            FB_FTP_Open1(bExecute:= FALSE);
            state := 2;
        END_IF

    2:
        FB_FTP_FileDownload1(
            sNetID:= '5.0.252.142.1.1',
            hFTP:= handle,
            sSrcFile:= '\\TestFolder\File1.txt',
            sDesFile:= '\\Program Files\TestFile1.txt',
            bExecute:= TRUE,
            tTimeout:= T#15s,
            bBusy => busy,
            bError => err,
            nErrId => errid,
            nProgress => progress);

        IF NOT busy AND NOT err THEN
            FB_FTP_FileDownload1(bExecute:= FALSE);
            state := 3;
        END_IF

    3:
        FB_FTP_Close1(
            sNetID:= '5.0.252.142.1.1',
            hFTP:= handle,
            bExecute:= TRUE,
            tTimeout:= T#15s,
            bBusy => busy,
            bError => err,
            nErrId => errid);

        IF NOT busy AND NOT err THEN
            FB_FTP_Close1(bExecute:= FALSE);
            state := 0;
        END_IF

END_CASE

```

The following PLC libraries need to be referenced for this sample: **Tc2_FTP**, Tc2_System, Tc2_Standard, Tc3_Interfaces, Tc3_Modules.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

6.4 TwinCAT FTP Client: Getting connection information with the FB_FTP_Info

This sample illustrates the using of the function block FB_FTP_Info.

Variable declaration

```
PROGRAM MAIN
VAR
  FB_FTP_Info1 : FB_FTP_Info;
  list         : ARRAY [0..5] OF ST_FTP_ConnInfo;
  startstop    : BOOL;
  busy        : BOOL;
  err         : BOOL;
  errid       : UDINT;
  entries     : UDINT;
END_VAR
```

PLC Program

```
FB_FTP_Info1(
  sNetID:= ,
  pList:= ADR(list),
  cbList:= SIZEOF(list),
  bExecute:= startstop,
  tTimeout:= T#15s,
  bBusy=> busy,
  bError=> err,
  nErrID=> errid,
  nEntries=> entries);
```

The following PLC libraries need to be referenced for this sample: Tc2_FTP, Tc2_System, Tc2_Standard, Tc3_Interfaces, Tc3_Modules.

At the following picture you can see a possible output:

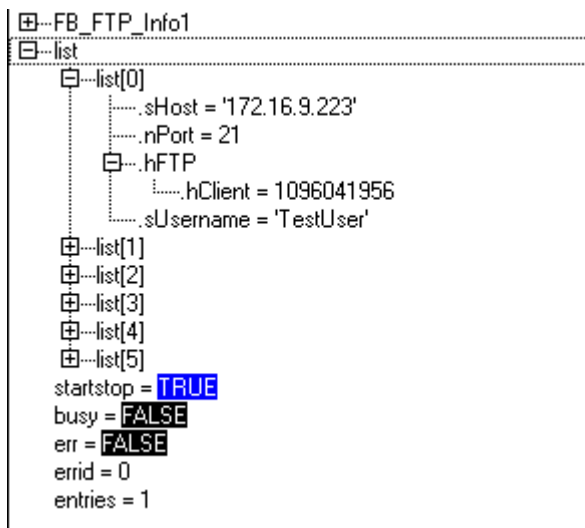


Fig. 2: Info_Sample

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

6.5 TwinCAT FTP Client: Removing a file from the FTP Server

This sample illustrates the removing of an existing file from the FTP Server.

At first a connection to the FTP Server will be created with the function block FB_FTP_Open. After that the function block FB_FTP_FileExist checks if the file "\\TestFolder\File1.txt" is available on the FTP Server. The function block FB_FTP_FileRemove deletes the specified file. At the end the connection to the FTP Server will be closed with the function block FB_FTP_Close.

You can start the sample with a rising edge at the variable "startstop".

Variable declaration

```
PROGRAM MAIN
VAR
  RisingEdge      : R_TRIG;
  startstop       : BOOL;
  state           : BYTE;
  FB_FTP_Open1    : FB_FTP_Open;
  FB_FTP_FileExist1 : FB_FTP_FileExist;
  FB_FTP_FileRemove1 : FB_FTP_FileRemove;
  FB_FTP_Close1   : FB_FTP_Close;
  busy            : BOOL;
  err             : BOOL;
  errid           : UDINT;
  handle          : T_HFTP;
  exist           : BOOL;
END_VAR
```

PLC Program

```
RisingEdge(CLK:=startstop);
IF RisingEdge.Q THEN
  state := 1;
END_IF

CASE state OF
  0:
    ;
  1:
    FB_FTP_Open1(
      sNetID:= ,
      sHost:= '172.16.9.223',
      nPort:= DEFAULT_FTP_PORT,
      sUsername:= 'TestUser',
      sPassword:= 'TestPwd123',
      bExecute:= TRUE,
      tTimeout:= T#15s,
      bBusy=> busy,
      bError=> err,
      nErrID=> errid,
      hFTP=> handle);

    IF NOT busy AND NOT err THEN
      state := 2;
      FB_FTP_Open1(bExecute:=FALSE);
    END_IF
  2:
    FB_FTP_FileExist1(
      sNetID:= ,
      hFTP:= handle,
      sFile:= '\\TestFolder\File1.txt',
      bExecute:= TRUE,
      tTimeout:= T#15s,
      bBusy=> busy,
```

```

        bError=> err,
        nErrID=> errid,
        bExist=> exist);

    IF NOT busy AND NOT err THEN
        IF exist THEN
            state := 3;
        ELSE
            state := 4;
        END_IF
        FB_FTP_FileExist1(bExecute:= FALSE);
    END_IF
3:
    FB_FTP_FileRemove1(
        sNetID:= ,
        hFTP:= handle,
        sFile:= '\TestFolder\File1.txt',
        bExecute:= TRUE,
        tTimeout:= T#15s,
        bBusy=> busy,
        bError=> err,
        nErrID=> errid);

    IF NOT busy AND NOT err THEN
        state := 4;
        FB_FTP_FileRemove1(bExecute:=FALSE);
    END_IF
4:
    FB_FTP_Close1(
        sNetID:= ,
        hFTP:= handle,
        bExecute:= TRUE,
        tTimeout:= T#15s,
        bBusy=> busy,
        bError=> err,
        nErrID=> errid);

    IF NOT busy AND NOT err THEN
        state := 0;
        FB_FTP_Close1(bExecute:=FALSE);
    END_IF
END_CASE

```

The following PLC libraries need to be referenced for this sample: **Tc2_FTP**, Tc2_System, Tc2_Standard, Tc3_Interfaces, Tc3_Modules.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

6.6 TwinCAT FTP Client: Reading of a filelist from the FTP Server

This sample illustrates the reading of a filelist from the FTP Server.

At first a connection to the FTP Server will be created with the function block FB_FTP_Open. After that the function block FB_FTP_FileList will read all files and folders which pass the searchmask from the FTP Server. At the end the connection to the FTP Server will be closed with the function block FB_FTP_Close.

You can start the sample with a rising edge at the variable "startstop".

Variable declaration

```

PROGRAM MAIN
VAR
    RisingEdge          : R_TRIG;
    startstop           : BOOL;
    state               : BYTE;

```

```

FB_FTP_Open1      : FB_FTP_Open;
FB_FTP_FileList1  : FB_FTP_FileList;
FB_FTP_Close1    : FB_FTP_Close;
busy              : BOOL;
err               : BOOL;
errid             : UDINT;
handle            : T_HFTP;
FileList          : ARRAY [0..MAX_FILELIST_ITEMS] OF STRING;
Items             : UDINT;

```

```
END_VAR
```

PLC Program

```

RisingEdge(CLK:= startstop);
IF RisingEdge.Q THEN
    state := 1;
END_IF

CASE state OF
    0:
        ;

    1:
        FB_FTP_Open1(
            sNetID:= ,
            sHost:= '172.16.9.223',
            nPort:= 21,
            sUsername:= 'TestUser',
            sPassword:= 'TestPwd123',
            bExecute:= TRUE,
            tTimeout:= T#15s,
            bBusy=> busy,
            bError=> err,
            nErrId=> errid,
            hFTP=> handle);

        IF NOT busy AND NOT err THEN
            FB_FTP_Open1(bExecute:= FALSE);
            state := 2;
        END_IF

    2:
        FB_FTP_FileList1(
            sNetID:= ,
            hFTP:= handle,
            sMask:= '*.txt',
            nIndex:= 0,
            pList:= ADR(FileList),
            cbList:= SIZEOF(FileList),
            bExecute:= TRUE,
            tTimeout:= T#15s,
            bBusy=> busy,
            bError=> err,
            nErrID=> errid,
            nItems=> Items);

        IF NOT busy AND NOT err THEN
            FB_FTP_FileList1(bExecute:= FALSE);
            state := 3;
        END_IF

    3:
        FB_FTP_Close1(
            sNetID:= ,
            hFTP:= handle,
            bExecute:= TRUE,
            tTimeout:= T#15s,
            bBusy => busy,
            bError => err,
            nErrId => errid);

        IF NOT busy AND NOT err THEN
            FB_FTP_Close1(bExecute:= FALSE);
            state := 0;
        END_IF
END_CASE

```

The following PLC libraries need to be referenced for this sample: **Tc2_FTP**, Tc2_System, Tc2_Standard, Tc3_Interfaces, Tc3_Modules.

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

7 Appendix

7.1 Return Codes

7.1.1 Overview of the TwinCAT FTP Client Error Codes

Offset + Error Code	Range	Description
0x00000000 + TwinCAT System Errors	0x00000000-0x00007800	TwinCAT System Errors (ADS-Error codes inclusive)
0x00008000 + Internal TwinCAT FTP Client Errors [▶ 59]	0x00008000-0x000081C4	Internal Errors of TwinCAT FTP Clients

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

7.1.2 ADS Return Codes

Error codes: 0x000 [▶ 55]..., 0x500 [▶ 55]..., 0x700 [▶ 56]..., 0x1000 [▶ 58]...

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 occurred 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_NOMORESVM	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 valid
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_IRQNOTLESSOREQUAL	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

7.1.3 FTP Client Return Codes

Hex	Dez	Description
0x00008001	32768 + 1 := 32769	Internal Error TwinCAT FTP Client.
0x00008002	32768 + 2 := 32770	File Error (e.g. File not found, Access denied)
0x00008003	32768 + 3 := 32771	Transmission Error (e.g. Connection broken)
0x00008004	32768 + 4 := 32772	Connecting Error. Couldn't create connection to a FTP Server.
0x00008005	32768 + 5 := 32773	Connection Error. No respond received from the FTP Server.
0x000081F4	32768 + 500 = 33268	Syntax error, command unrecognized. This may include errors such as command line too long.
0x000081F5	32768 + 501 = 33269	Syntax error in parameters or arguments.
0x000081F6	32768 + 502 = 33270	Command not implemented.
0x000081F7	32768 + 503 = 33271	Bad sequence of commands.
0x000081F8	32768 + 504 = 33272	Command not implemented for that parameter.
0x00008212	32768 + 530 = 33298	Not logged in.
0x00008214	32768 + 532 = 33300	Need account for storing files.
0x00008226	32768 + 550 = 33318	Requested action not taken. File unavailable (e.g., file not found, no access).
0x00008227	32768 + 551 = 33319	Requested action aborted. Page type unknown.
0x00008228	32768 + 552 = 33320	Requested file action aborted. Exceeded storage allocation (for current directory or dataset).
0x00008229	32768 + 553 = 33321	Requested action not taken. File name not allowed.
0x000081A5	32768 + 421 = 33189	Service not available, closing control connection. This may be a reply to any command if the service knows it must shut down.
0x000081A9	32768 + 425 = 33193	Can't open data connection.
0x000081AA	32768 + 426 = 33194	Connection closed; transfer aborted.
0x000081C2	32768 + 450 = 33218	Requested file action not taken.
0x000081C3	32768 + 451 = 33219	Requested action aborted. Local error in processing.
0x000081C4	32768 + 452 = 33220	Requested action not taken. Insufficient storage space in system. File unavailable (e.g., file busy).

Requirements

Development Environment	Target Platform	PLC Libraries to include
TwinCAT v3.0 Build 3102	PC or CX (x86)	Tc2_FTP

7.2 Troubleshooting

7.2.1 Troubleshooting

The following list provides some basic help in case any errors should occur and should be read before contacting our support department. Please also consult our list of error codes [▶ 54].

- **One of the PLC function blocks returns error 0x6: Target port not found**

Please make sure that the process TcFtpClient.exe is running by checking the Windows Task Manager. If it isn't running, please reinstall the product TF6300 FTP.

- **One of the PLC function blocks returns error 0x00008005: No response from FTP-Server**

- Please make sure that the FTP-Server is available and that no firewall blocks access to the FTP ports. Please also consult our introduction to FTP [▶ 17] for more information about FTP ports.

Should the steps mentioned above don't help, please contact our support department [▶ 60].

7.2.2 Contact Beckhoff Support

7.2.2.1 Contact Beckhoff Support

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

Type	Description
General information [▶ 60]	Shows which general system information needs to be forwarded to our Support department
Product-related information [▶ 61]	Shows which product-related information needs to be forwarded to our Support department

7.2.2.2 General system information

- What kind of hardware is being used on the computer running TF6300 FTP?
 - Beckhoff Industrial PC: Which product number does the IPC have?
 - Beckhoff Embedded PC: Which product number does the Embedded PC have?
 - Which Operating System image version [▶ 63] is currently installed on that computer?
- What kind of Operating System has been installed on the computer running TF6300 FTP?

- Windows CE5, Windows CE6, Windows CE7?
- Windows XP, Windows 7, Windows Embedded?
- Which version of TwinCAT [▶ 61] is being used in conjunction with TF6300 FTP?
 - Are there multiple TwinCAT versions installed on that computer? (For example: TwinCAT 2 and TwinCAT 3 installed in parallel)
- Please provide an exact description of the issue

7.2.2.3 Product-related system information

- Which version of TF6300 FTP is being used?
- Which function blocks of the Tc2_Ftp library are being used in the PLC program?
- Which FTP-Server software is being used?
 - Microsoft Internet Information Service (which version?)
 - FileZilla FTP-Server
 - Linux/Unix NcFtpd
 -
- Please provide an exact description of the environment in which the product TF6300 FTP is being used
 - Where is the computer running TF6300 FTP located?
 - Where is the FTP-Server located? (Local network, Internet)
 - What are the IP settings of the FTP-Server and the computer running TF6300 FTP? (IP address, subnet mask, gateway, DNS)
 - In case that there is a firewall between both computers:

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

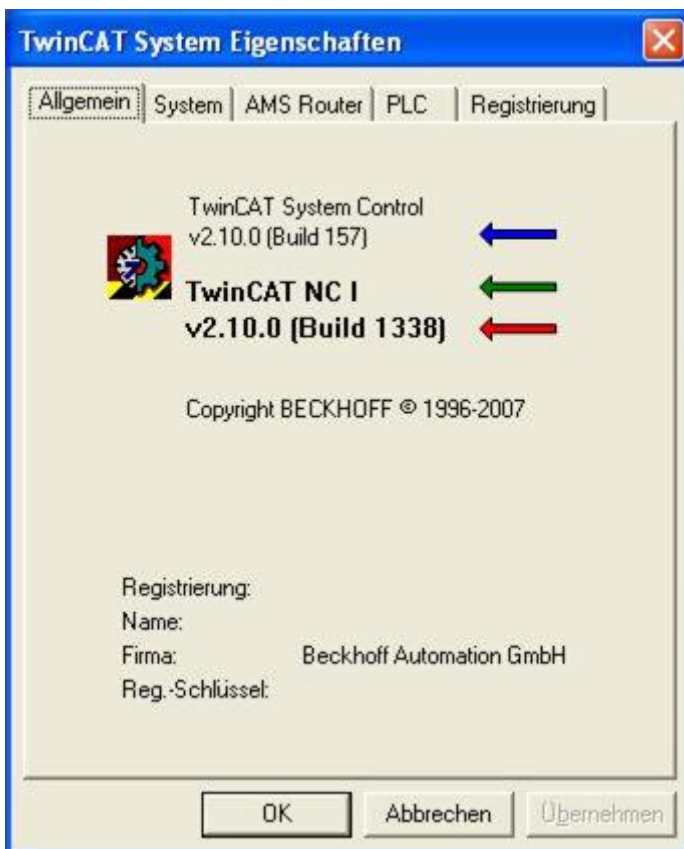
Has the Firewall been configured to allow FTP connections?

7.2.2.4 How to determine TwinCAT version

This article describes how you can determine the currently used version of TwinCAT.

Windows XP, Windows 7

You can find the TwinCAT version in the **TwinCAT System Tray** icon.



Windows CE

You can find the TwinCAT version in the **TwinCAT System Tray** icon.



7.2.2.5 How to determine Operating System image version

This article describes - in case a Beckhoff IPC/EPC is being used - how you can determine the currently used version of the Operating System image.

Windows XP, Windows 7

You can find the image version in the **System Properties** window. Please perform the following steps on the Beckhoff IPC/EPC:

- Open the Windows Control Panel
- Double-click on "System"



Windows CE

You can find the image version in the **CX Configuration Tool**. Please perform the following steps on the Beckhoff IPC/EPC:

- Open the Windows Control Panel
- Double-click on "CX Configuration Tool"

