BECKHOFF New Automation Technology

Manual | EN

TS6300

TwinCAT 2 FTP Client

Supplement | Communication

2023-08-10 | Version: 1.2

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 applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

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

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

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

Copyright

1.2 For your safety

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Signal words

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

Personal injury warnings

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

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

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

Warning of damage to property or environment

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

Information on handling the product

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

1.3 Notes on information security

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

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

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

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

2 Overview

The TwinCAT FTP Client is a supplement and provides the possibility to communicate from the TwinCAT System to a FTP Server on an easy way.

The TwinCAT FTP Client provides the following functions:

Connection establishment:

- with authentication

Transmission functions:

- Uploading files

- Downloading files

File functions:

- Renaming files

- Removing files

- Searching files

- Creating folders

- Removing folders

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 used with handles. Because of the handles you don't have to add the authentication values for every function.

3 Requirements

TwinCAT FTP Client is based on a implementation which requires .NET2.0 Framework or for CE the Compact Framework.

The products requires these components

- minimum TwinCAT PLC
- Microsoft .NET Framework 2.0 (or higher) or Microsoft .NET Compact Framework 2.0 (or higher) Nearly all Beckhoff CE devices contain .NET Compact Framework by default

Sample:

- Beckhoff CX9001 device contains .NET CF 2.0
- Beckhoff CX9000 devices does not contain .NET - as a result the TwinCAT FTP-Client will not run on this platform

4 Installation

This part of the documentation gives a step-by-step explaination of the TwinCAT FTP-Client setup process for Windows XP based operating systems. The following topics are part of this document:

- Downloading the setup file
- Starting the installation

Downloading the setup file

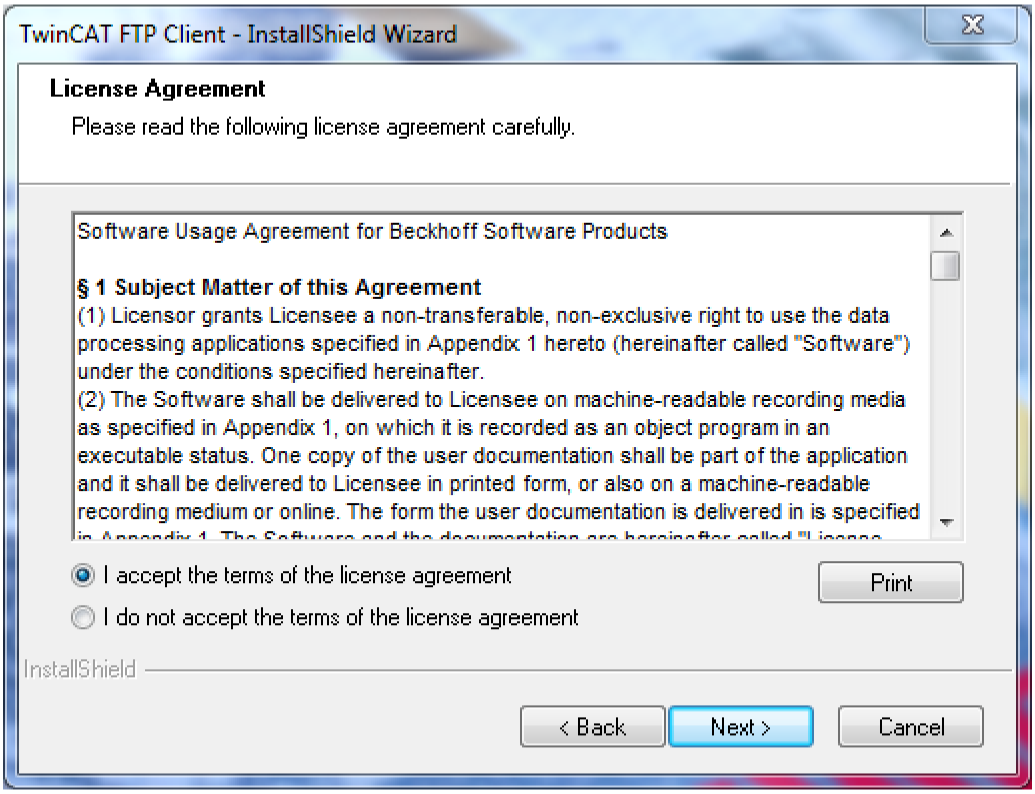
Like many other TwinCAT Supplement products,TwinCAT FTP-Client is available for download via the Beckhoff FTP-Server. The download represents the most current version. To download the setup file, perform the following steps:

1. Start an FTP-Client software of your choice, for example FileZilla or Total Commander.
2. Open a connection to TS6300 | TwinCAT FTP Client.
3. Select TS6250 TwinCAT Modus Server and start the download via the download-cart.
4. (Optional) Transfer the downloaded file to the TwinCAT runtime system, where you would like to install the Supplement.

Starting the installation

To install the Supplement, please perform the following steps:

1. Double-click the downloaded setup file**.**
   Under Windows 7 32-bit/64-bit, start the installation with Run as Administrator by right-clicking the setup file and selecting the corresponding option in the context menu.
2. Select an installation language.
3. Click on **Next** and accept the license agreement.



TwinCAT FTP Client - InstallShield Wizard

License Agreement

Please read the following license agreement carefully.

Software Usage Agreement for Beckhoff Software Products

§ 1 Subject Matter of this Agreement

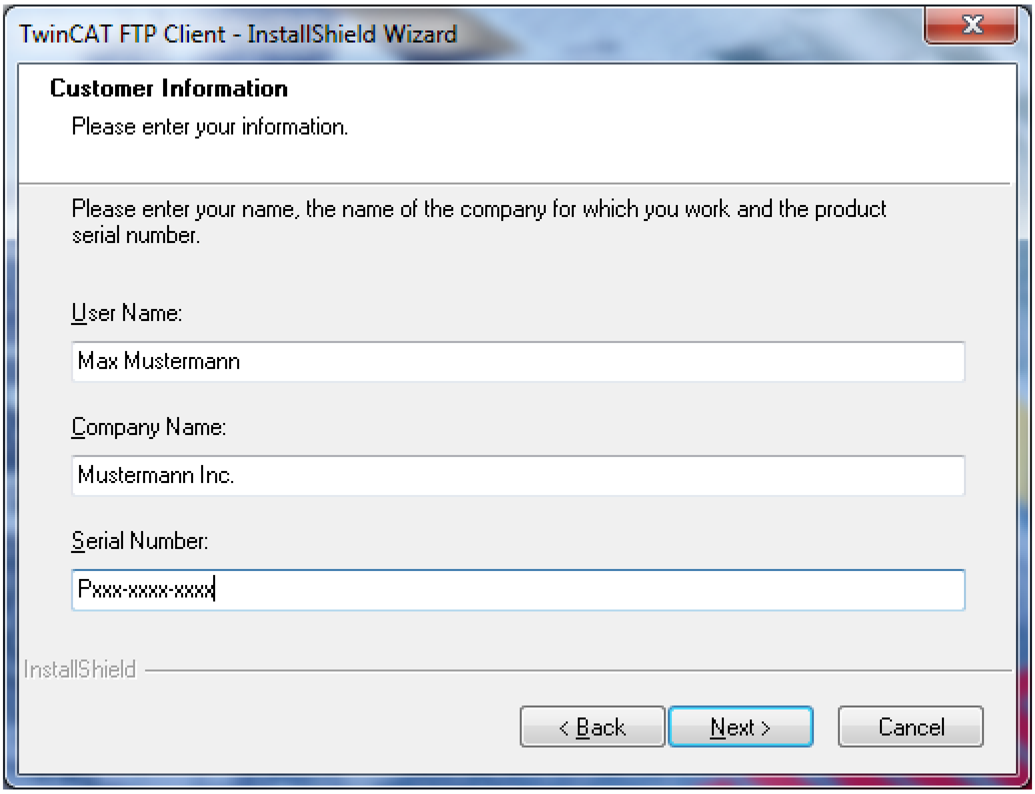
(1) Licensor grants Licensee a non-transferable, non-exclusive right to use the data processing applications specified in Appendix 1 hereto (hereinafter called "Software") under the conditions specified hereinafter.

(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

● I accept the terms of the license agreement

○ I do not accept the terms of the license agreement

4. Enter your user information. All fields are mandatory.



TwinCAT FTP Client - InstallShield Wizard

Customer Information

Please enter your information.

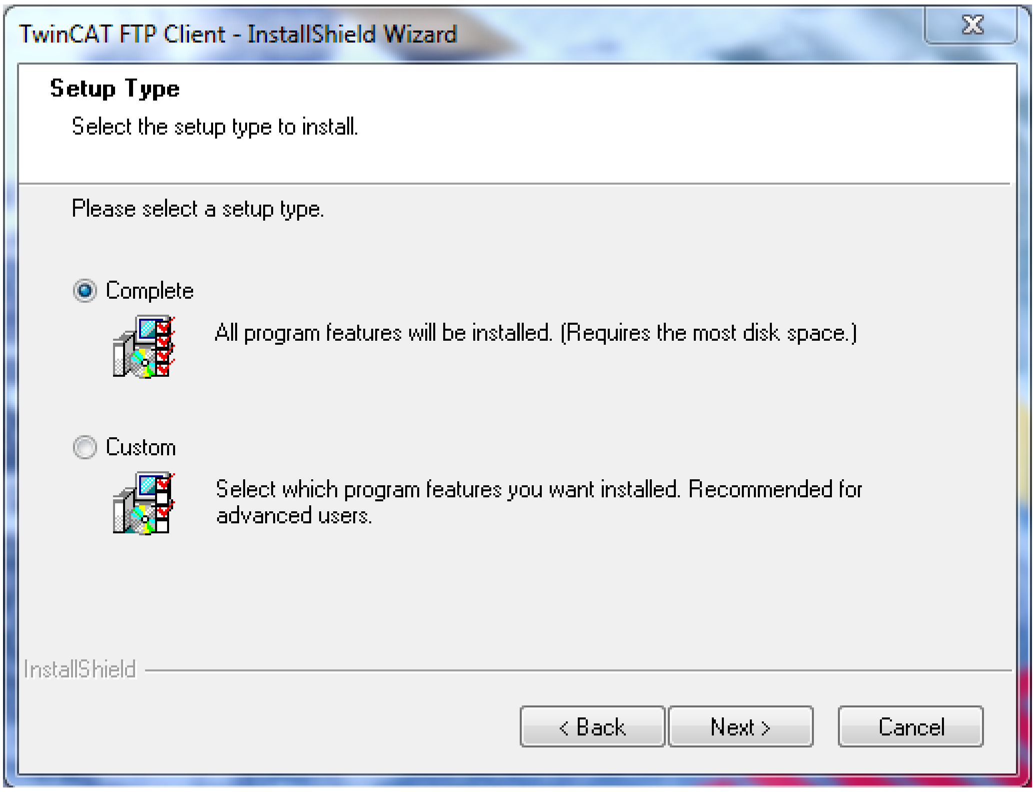
Please enter your name, the name of the company for which you work and the product serial number.

User Name: Max Mustermann

Company Name: Mustermann Inc.

Serial Number: Pxxx-xxxx-xxxx

5. Select **Complete** and click on **Next**.



TwinCAT FTP Client - InstallShield Wizard

Setup Type

Select the setup type to install.

Please select a setup type.

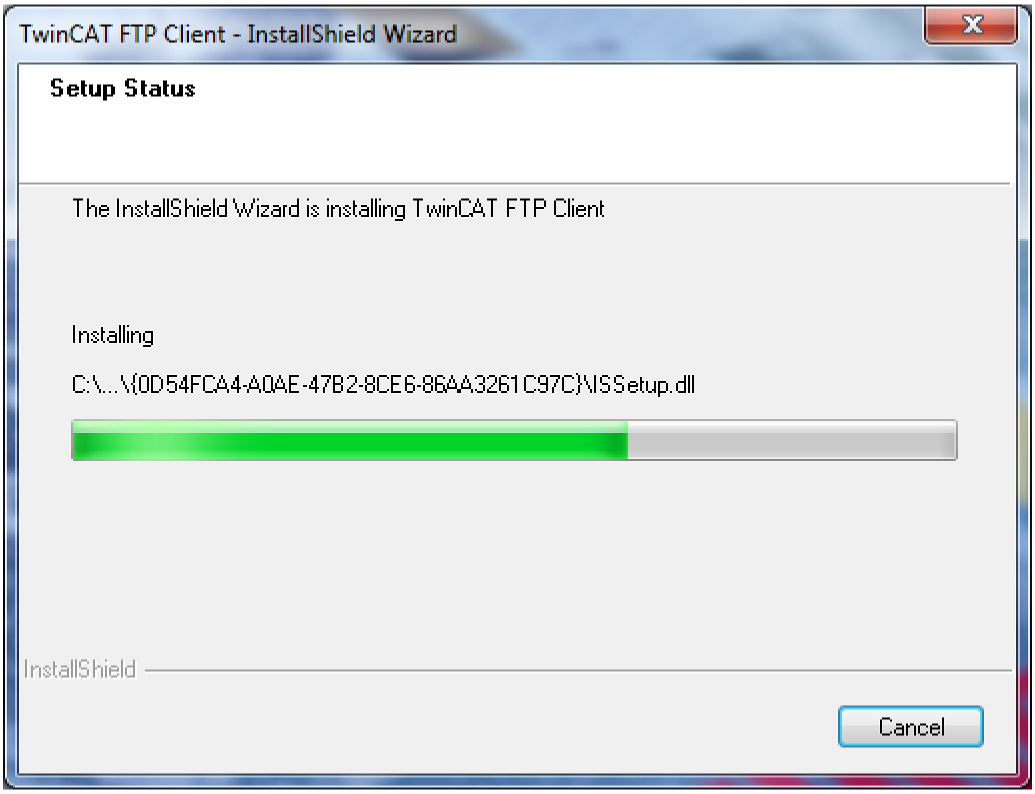
● Complete

All program features will be installed. (Requires the most disk space.)

○ Custom

Select which program features you want installed. Recommended for advanced users.

6. Click on **Install** to start installation.



TwinCAT FTP Client - InstallShield Wizard

Setup Status

The InstallShield Wizard is installing TwinCAT FTP Client

Installing

C:\...\{0D54FCA4-A0AE-47B2-8CE6-86AA3261C97C}\ISSetup.dll

7. At the end of the setup process, restart your computer.

5 Installation Windows CE

The installation on Windows CE is a two-step procedure:

Step 1: Installation of TwinCAT FTP-Client Supplement Software on a host computer

This topic will be covered in a separate installation manual and should not be discussed on this page.

By installing the Supplement-Software, the necessary files for Windows CE will be automatically installed to the TwinCAT sub directory **..\TwinCAT\CE\TwinCAT FTP Client CE** on the host computer.

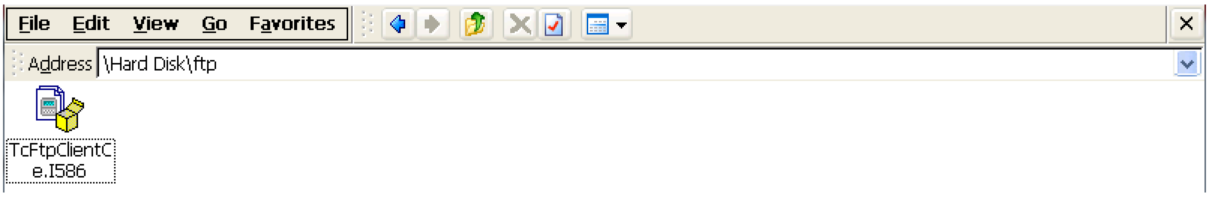
Step 2: Transfer of CAB files for Windows CE to the CE device

Transfer the corresponding CAB-File from the TwinCAT sub directory **..\TwinCAT\CE\TwinCAT FTP Client CE** to the CE device (e.g. via MemoryStick/FTP/Public-Folder/CF-Adapter...). Which CAB file you actually need depends on the CPU architecture of your CE device:

- **TcFtpClientCe.I586.cab:** TwinCAT FTP-Client for x86 based CPUs (e.g. CX10xx, CP62xx, C69xx, ...)
- **TcFtpClientCe.ARM.cab:** TwinCAT FTP-Client for ARM based CPUs (e.g. CX9001, CX9010, CP6608, ...)

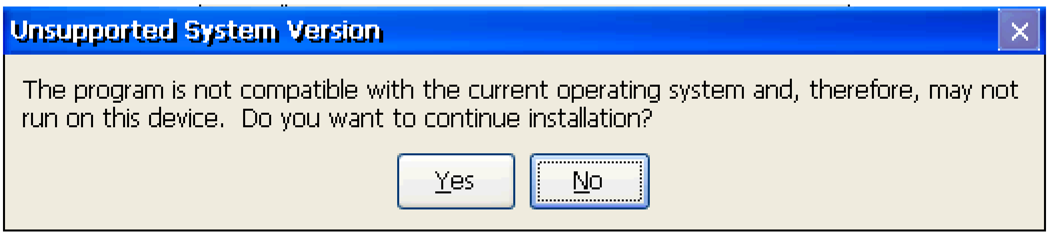
Please perform the following steps after you have transferred the CAB file to the CE device:

- Navigate to the directory where you transferred the setup file to



Address: \Hard Disk\ftp

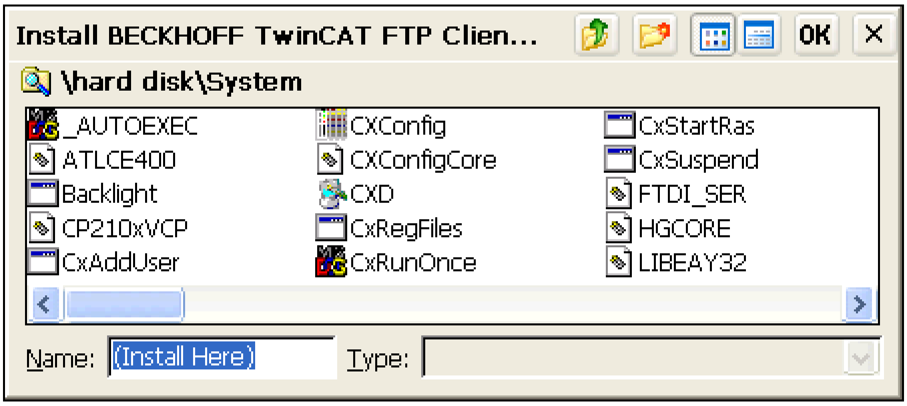
- Double-click the CAB file. Should you get a message box stating that this program is not compatible with the current operating system, please re-check if you used the correct CAB-file (ARM, I586) for your IPC/Embedded-PC.
- If you are sure that the CAB-file corresponds to the Embedded-PC/IPC, please acknowledge this message box with **"Yes"**.



Unsupported System Version

The program is not compatible with the current operating system and, therefore, may not run on this device. Do you want to continue installation?

- Confirm that **"\Hard Disk\System"** has been selected as the destination directory

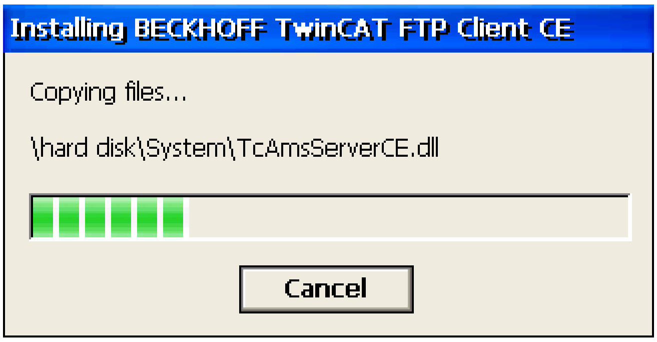


\hard disk\System

Name: (Install Here)

Type:

- To start installation, click on **"Ok"**



Installing BECKHOFF TwinCAT FTP Client CE

Copying files...

\hard disk\System\TcAmsServerCE.dll

After installation, the setup file will be automatically deleted.

Please note that the TCP/IP Server will be available after a restart of your CE device.

If the TcFTPClient don't start automatically after a reboot and you get the ADS-Error 6, please check the version of the "StartUp.exe". Version 1.35 or later is necessary for the automatic start of TcFTPClient.

Installation contents on the CE device:

- \Hard Disk\System"
  - TcFTPClientCe.exe
  - TcAmsServerCe.dll
  - TwinCAT.Ads.dll
  - TwinCAT.Ads.Server.dll

6 Connection mode

For sending and receiving files or for transmitting directory lists (the default-port is port 20) a separate TCP-connection will be used per process. FTP knows two different modes to connect to these kinds of connections: "**active FTP**" and **"passive FTP"**.

The TwinCAT FTP Client Version 1.0.8 can use both modes. Earlier Version only support **"passive FTP"** connection mode.

active FTP

At active FTP ("active mode") the client open a random port and send a PORT-command to the FTP Server which contains the port and the IP-Address. This is typically a port of the client which is greater than 1023, can also be another server that has been switched to passive mode, i.e. is waiting for a connection (so-called FXP). The data transmitting at server site use the port 20. The communication with commands uses exclusive the control port. We also call this control "Out of Band". Therefore, it is possible to keep on communicate with the partner while data transmitting.

passive FTP

At passive FTP ("passive mode") the client sends a PASV-command, the server open a port and transmit this port together with the IP-Address to the client. The client use therefore a port higher than 1023 and the server use the port which he has transmit. These technique will be used, if the client isn't available for the server. This could be, if the client is located behind a router, which circumscribe the address of the client with NAT, or if a firewall refuse the access to the network of the client.

7 PLC library

Overview

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

Function Blocks

Name	Description
FB_GetStateTcFTPClient [▶ 34]	Call state information.
FB_FTP_HostResolve [▶ 17]	Resolves the given host name to an IPv4 address
FB_FTP_Open [▶ 18]	Open a connection to a FTP Server. (Passive)
FB_FTP_OpenEx [▶ 19]	Open a connection to a FTP Server. (Passive / Active)
FB_FTP_Close [▶ 20]	Close a connection to a FTP Server.
FB_FTP_CloseAll [▶ 21]	Close all connections to a FTP Server
FB_FTP_Info [▶ 21]	Get information from all opened connections
FB_FTP_FileUpload [▶ 22]	Copy a file to a FTP Server. Use an existing connection.
FB_FTP_FileUploadEx [▶ 23]	Copy a file to a FTP Server.
FB_FTP_FileDownload [▶ 24]	Copy a file from a FTP Server to an ADS device. Use an existing connection.
FB_FTP_FileDownloadEx [▶ 25]	Copy a file from a FTP Server to an ADS device.
FB_FTP_DirCreate [▶ 26]	Create a directory on the FTP Server
FB_FTP_DirRemove [▶ 27]	Remove a directory on the FTP Server
FB_FTP_FileList [▶ 28]	Select all files and folder which pass the searchmask.
FB_FTP_FileListEx [▶ 29]	Select all files and folder with additional information which pass the searchmask.
FB_FTP_FileExist [▶ 31]	Find out if the searched file exist on the FTP Server.
FB_FTP_FileRemove [▶ 32]	Remove a file from the FTP Server.
FB_FTP_FileRename [▶ 33]	Rename a file on the FTP Server.

Functions

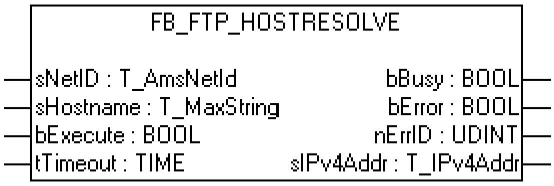
Name	Description
F_GetVersionTcFTPClient [▶ 35]	Call version information.

Data Types

Name	Description
T_HFTP [▶ 35]	Handle of the FTP Server.
ST_FTP_ConnInfo [▶ 36]	Information of existing FTP connections.
ST_FTP_FileDetails [▶ 36]	Returning file information of the FTP "LIST" command
E_FTP_ConnMode [▶ 37]	FTP connection modes (Passiv / Aktiv).

7.1 Function Blocks

7.1.1 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.

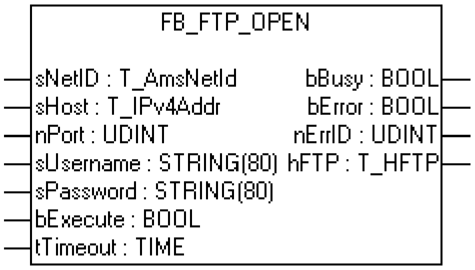
nErrID : Supplies the ErrorCode [▸ 45] when the bError output is set.

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

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.2 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.

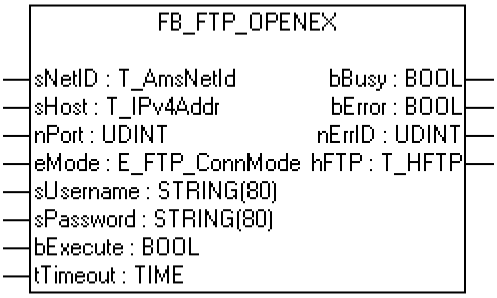
nErrID : Supplies the ErrorCode [▸ 45] when the bError output is set.

hFTP : Handle [▸ 35] of a FTP Server connection.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.3 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
```

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 [▶ 37] (active / passive).

sUsername : Username for the FTP Server authentification

sPassword : Password for the FTP Server authentification

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.

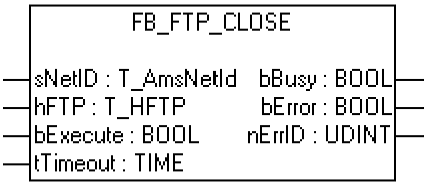
nErrID : Supplies the ErrorCode [▶ 45] when the bError output is set.

hFTP : Handle [▶ 35] of a FTP Server connection.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib (from TcFTPClient Version 1.0.8)
TwinCAT v2.10.0	CX (ARM)	

7.1.4 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
```

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

hFTP : Handle [▶ 35] 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.

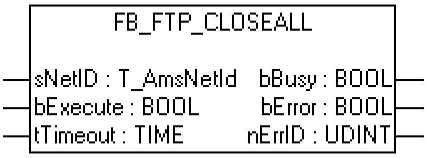
bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [▶ 45] when the bError output is set.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.5 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.

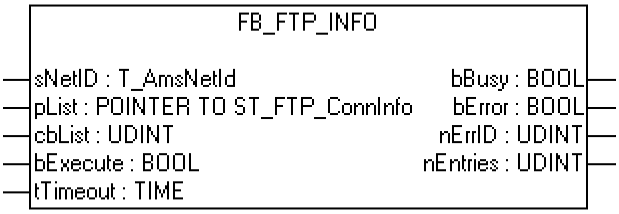
bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [▶ 45] when the bError output is set.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.6 FB_FTP_Info



You can get information about the exiting 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
```

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 [▶ 36]. 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 if "bBusy" remains TRUE.

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

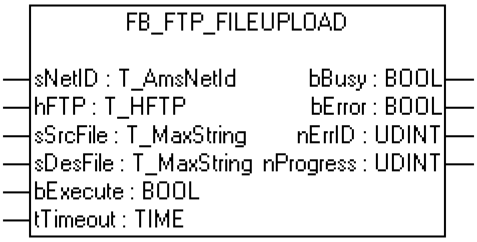
nErrID : Supplies the ErrorCode [▶ 45] when the bError output is set.

nEntries : Number of existing connections.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.7 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
```

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

hFTP : Handle [▶ 35] 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 [▸ 45] when the bError output is set.

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

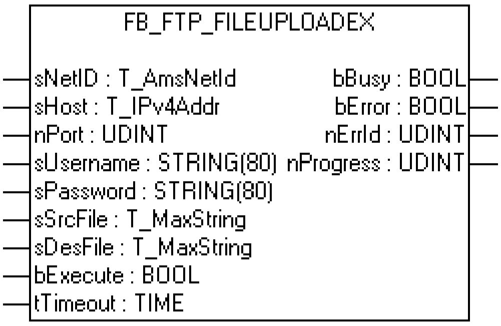
Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

Also see about this

- ADS Return Codes [▸ 46]

7.1.8 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.

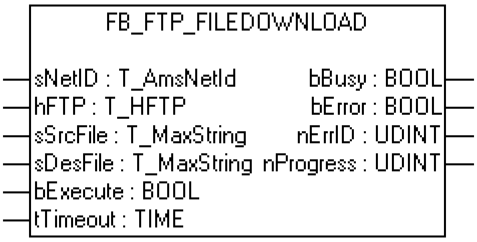
nErrID : Supplies the ErrorCode [▶ 45] when the bError output is set.

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

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.9 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
```

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

hFTP : Handle [▸ 35] 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.

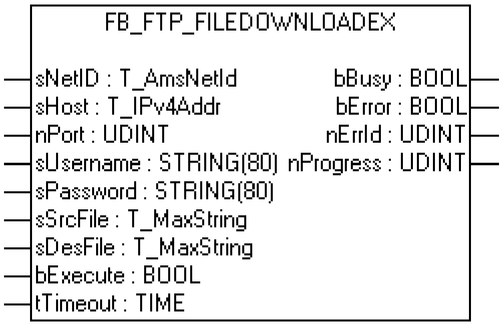
nErrID : Supplies the ErrorCode [▸ 45] when the bError output is set.

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 system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.10 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.

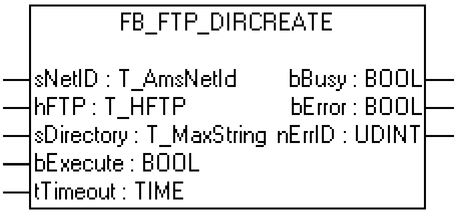
nErrID : Supplies the ErrorCode [▶ 45] when the bError output is set.

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 system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.11 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
```

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

hFTP : Handle [▶ 35] 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.

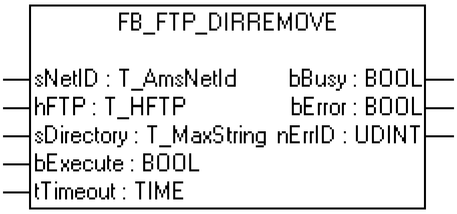
bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [▶ 45] when the bError output is set.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.12 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 [▶ 35] 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.

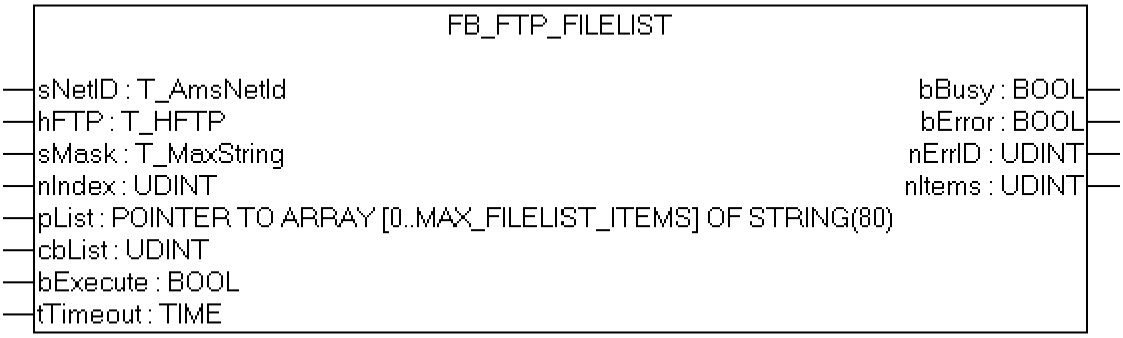
bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [▶ 45] when the bError output is set.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.13 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 [▶ 35] 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 [▸ 45] when the bError output is set.

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

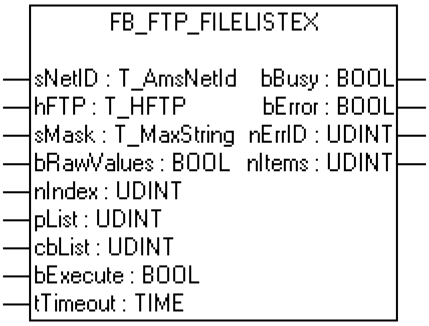
Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

Also see about this

ADS Return Codes [▸ 46]

7.1.14 FB_FTP_FileListEx



You can read a list of file- and foldernames 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 [▶ 35] 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 [▶ 36] 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 [▶ 45] when the bError output is set.

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

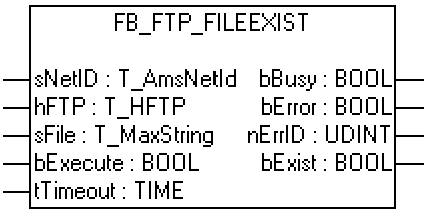
Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

Also see about this

- ADS Return Codes [▶ 46]

7.1.15 FB_FTP_FileExist



You can check, if a destined file exists 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 [▸ 35] 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.

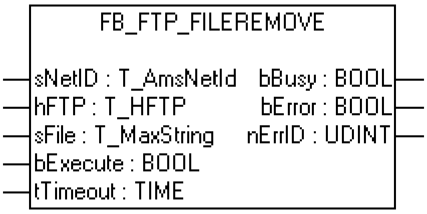
nErrID : Supplies the ErrorCode [▸ 45] when the bError output is set.

bExist : Becomes TRUE, if the searched file exist.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.16 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 [▶ 35] 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.

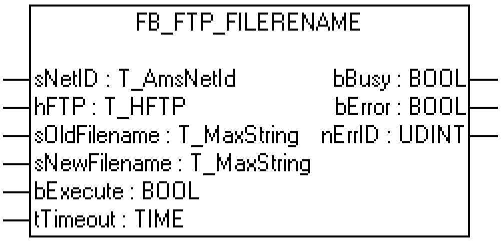
bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [▶ 45] when the bError output is set.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.17 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 [▶ 35] 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 if "bBusy" remains TRUE.

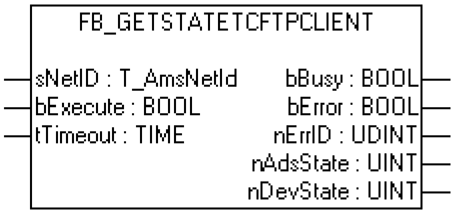
bError : Becomes TRUE, as soon as an error occurs.

nErrID : Supplies the ErrorCode [▶ 45] when the bError output is set.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.18 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 [▶ 45] 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.

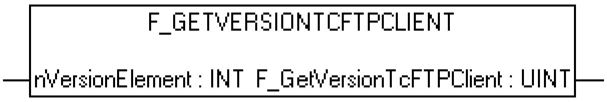
- 1 = TwinCAT FTP Client is started

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.2 Functions

7.2.1 FUNCTION 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 System	PLC Libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.3 Data Types

7.3.1 TYPE T_HFTP

VAR_INPUT

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

hClient : Handle of the FTP connection.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.3.2 TYPE 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 system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.3.3 TYPE 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)

tTimstamp : 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 system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.3.4 TYPE E_FTP_ConnMode

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

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib (from TcFTPClient Version 1.0.8)
TwinCAT v2.10.0	CX (ARM)	

7.4 Constants

7.4.1 Global Variables

```
VAR_GLOBAL CONSTANT

    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;

END_VAR
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

8 Samples

8.1 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
```

PLC Program

```
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 "**TcFTPClient.lib**", "TcSystem.lib", "TcBase.lib" and "STANDARD.lib" libraries are to be linked.

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

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 "**TcFTPClient.lib**", "TcSystem.lib", "TcBase.lib" and "STANDARD.lib" libraries are to be linked.

8.3 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

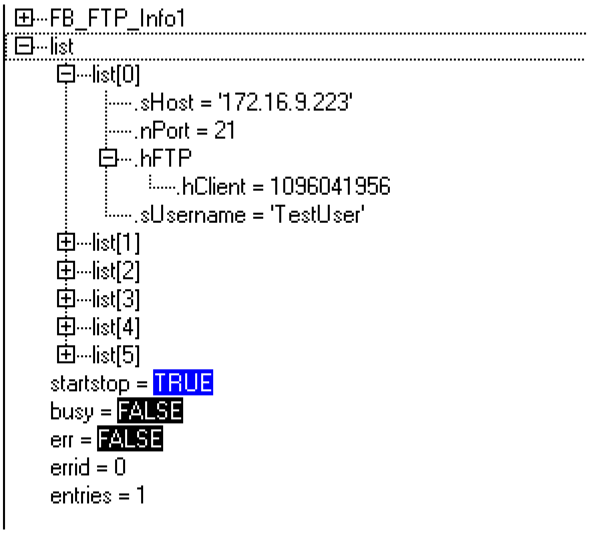
```
PROGRAM MAIN
VAR
    FB_FTP_Info1    : FB_FTP_Info;
    list        : ARRAY [0..5] OF ST_FTP_ConnInfo;
    startstop           : BOOL;
    busy        : BOOL;
    err         : BOOL;
    errid       : BOOL;
    entries     : UDINT;
END_VAR
```

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 "**TcFTPClient.lib**", "TcSystem.lib", "TcBase.lib" and "STANDARD.lib" libraries are to be linked.

At the following picture you can see a possible output:



```
⊞FB_FTP_Info1
⊟list
    ⊟list[0]
        .sHost = '172.16.9.223'
        .nPort = 21
        ⊟.hFTP
            .hClient = 1096041956
        .sUsername = 'TestUser'
    ⊞list[1]
    ⊞list[2]
    ⊞list[3]
    ⊞list[4]
    ⊞list[5]
startstop = TRUE
busy = FALSE
err = FALSE
errid = 0
entries = 1
```

8.4 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:= '5.0.252.142.1.1',
            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:= '5.0.252.142.1.1',
            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:= '5.0.252.142.1.1',
            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:= '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
            state := 0;
            FB_FTP_Close1(bExecute:=FALSE);
        END_IF
END_CASE
```

The "**TcFTPClient.lib**", "TcSystem.lib", "TcBase.lib" and "STANDARD.lib" libraries are to be linked.

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

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

To be able to run this example, you should have created a user with the name "TestUser" and the password "TestPwd123" at your FTP server.

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

At first a connection to the FTP Server will be created with the function block FB_FTP_Open. Then all files with the extension "*.txt" are loaded from the FTP server with the function block FB_FTP_FileList.

At the end the connection to the FTP Server will be closed with the function block FB_FTP_Close.

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:= '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_FileList1(
            sNetID:= '5.0.252.142.1.1',
            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:= '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 "**TcFTPClient.lib**", "TcSystem.lib", "TcBase.lib" and "STANDARD.lib" libraries are to be linked.

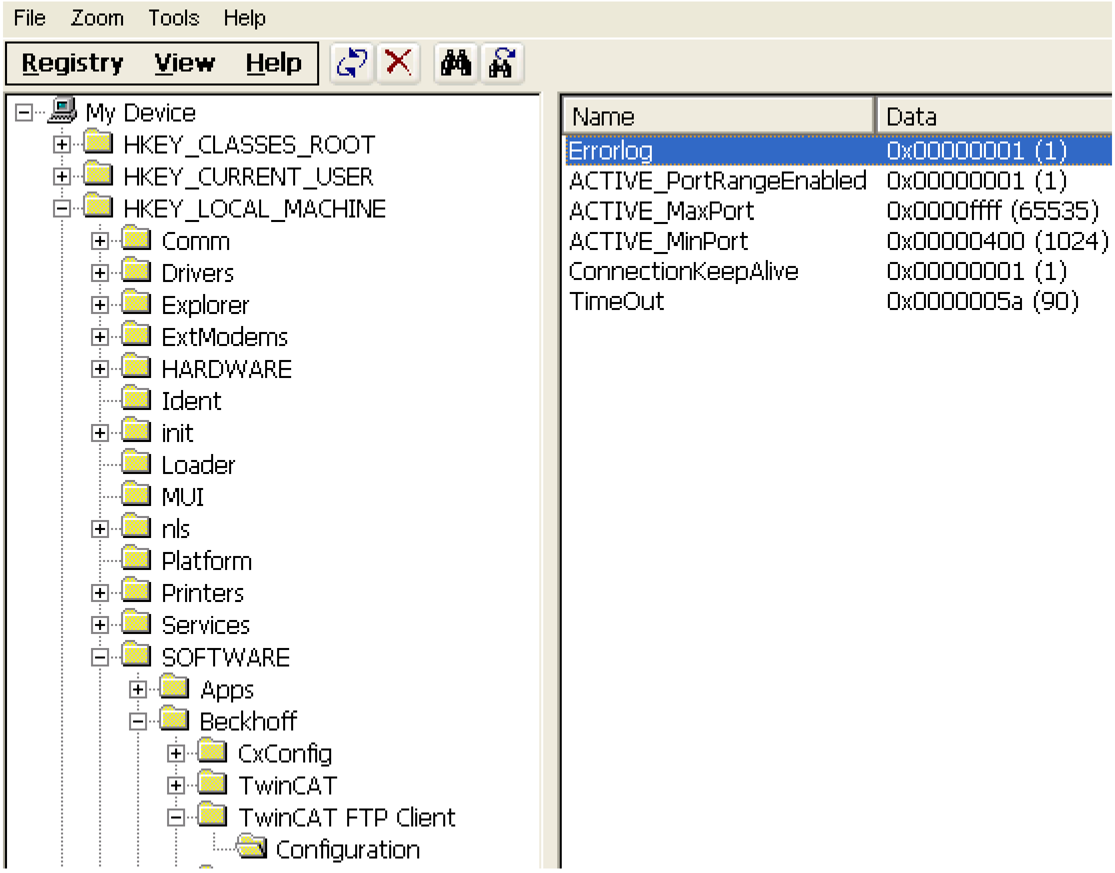
9 Appendix

9.1 Creating an Errorlogfile

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 (usually "\Hard Disk\System").
You are able to activate the functionality with the following Registry Key:

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



Name	Data
Errorlog	0x00000001 (1)
ACTIVE_PortRangeEnabled	0x00000001 (1)
ACTIVE_MaxPort	0x0000ffff (65535)
ACTIVE_MinPort	0x00000400 (1024)
ConnectionKeepAlive	0x00000001 (1)
TimeOut	0x0000005a (90)

You start the functionality with the Value 1.
You stop the functionality with the Value 0.
All changes takes effect after a restart of the TwinCAT System.

NOTICE

Damage of storage medium

To many write cycles to the Compact Flash Card can shorten its service life.

NOTICE

Damage of storage medium

Use the function of the Errorlogfile only if you do tests!

9.2 Port Range (Active FTP)

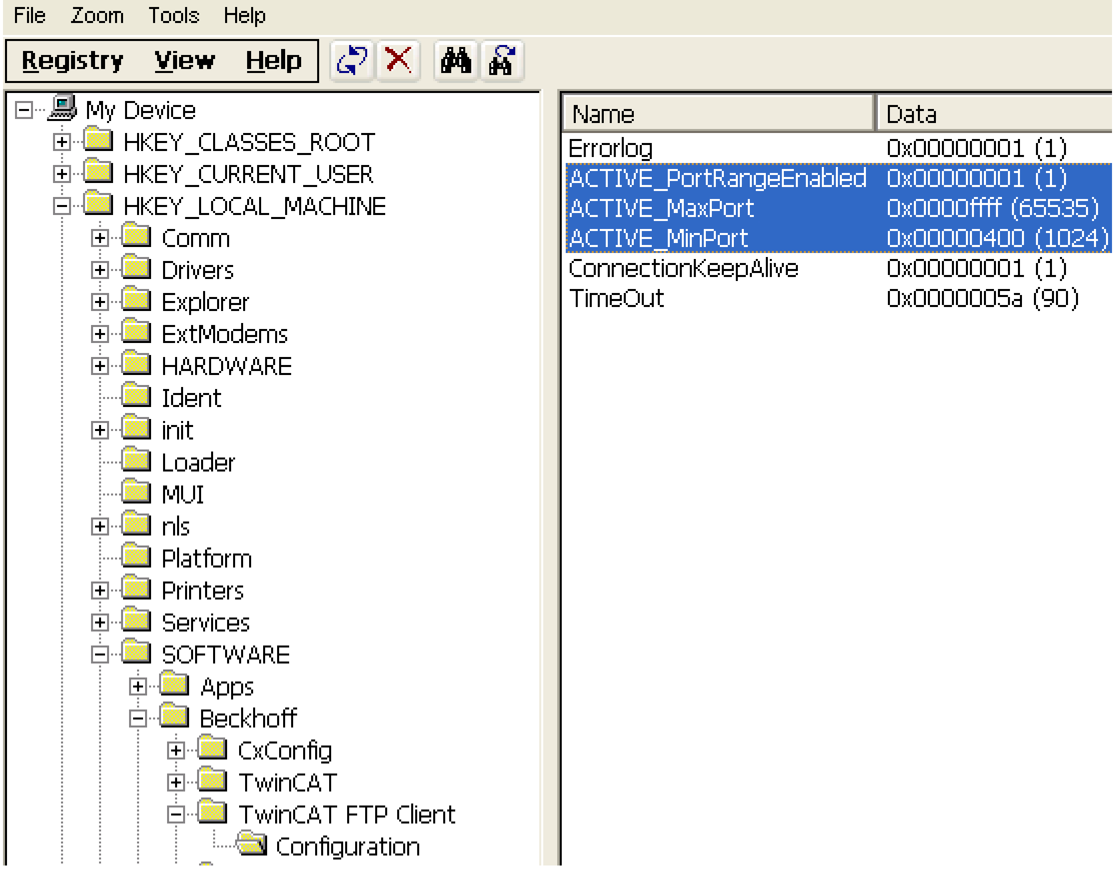
If you use the Active-FTP mode for connecting to the FTP Server, the Client has to choose the port for the connection establishment. For this reason it is possible to declare a port range. The TwinCAT FTP Client will choose a free port out of the range to create a connection to the FTP Server.

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 have to set the registrykey "ACTIVE_PortRangeEnabled" to 1. Else the TwinCAT FTP Client will be 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 range.



Name	Data
Errorlog	0x00000001 (1)
ACTIVE_PortRangeEnabled	0x00000001 (1)
ACTIVE_MaxPort	0x0000ffff (65535)
ACTIVE_MinPort	0x00000400 (1024)
ConnectionKeepAlive	0x00000001 (1)
TimeOut	0x0000005a (90)

9.3 Return Codes

9.3.1 Overview of the TwinCAT FTP Client Error Codes

Requirements

Offset + Error Code	Range	Description
0x00000000 + TwinCAT System Errors [▶ 46]	0x00000000-0x00007800	TwinCAT System Errors (ADS-Error codes inclusive)
0x00008000 + Internal TwinCAT FTP Client Errors [▶ 49]	0x00008000-0x000081C4	Internal Errors of TwinCAT FTP Clients

9.3.2 ADS Return Codes

Error codes: 0x000... [▸ 46], 0x500... [▸ 46], 0x700... [▸ 46], 0x1000... [▸ 46], 0x274C... [▸ 46]

Global Error Codes

Hex	Dec	Description	Possible Causes	Solution
0x0	0	no error		
0x1	1	Internal error		
0x2	2	No Rtime		
0x3	3	Allocation locked memory error		
0x4	4	Insert mailbox error	No ADS mailbox was available to process this message.	Reduce the number of ADS calls (e.g ADS-Sum commands or Max Delay Parameter)
0x5	5	Wrong receive HMSG		
0x6	6	target port not found	ADS Server not started	
0x7	7	target machine not found	Missing ADS routes	
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 AMS 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	Description	Possible Causes	Solution
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	Check the connection between the communication partners

Hex	Dec	Description	Possible Causes	Solution
			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	Check the connection to the debug monitor
0x504	1284	ROUTERERR_UNKNOWNPORTTYPE	The port type is unknown	
0x505	1285	ROUTERERR_NOTINITIALIZED	Router is not initialised	
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	The port is invalid.	
0x50A	1290	ROUTERERR_NOTACTIVATED	TwinCAT Router not active	
0x50B	1291	ROUTERERR_FRAGMENTBOXFULL		
0x50C	1292	ROUTERERR_FRAGMENTTIMEOUT		
0x50D	1293	ROUTERERR_TOBEREMOVED		

General ADS Error Codes

Hex	Dec	Description	Possible Causes	Solution
0x700	1792	error class <device error>		
0x701	1793	Service is not supported by server		
0x702	1794	invalid index group		
0x703	1795	invalid index offset		
0x704	1796	reading/writing not permitted		
0x705	1797	parameter size not correct		
0x706	1798	invalid parameter value(s)		
0x707	1799	device is not in a ready state		
0x708	1800	device is busy		
0x709	1801	invalid context (must be in Windows)		
0x70A	1802	out of memory		
0x70B	1803	invalid parameter value(s)		
0x70C	1804	not found (files, ...)		
0x70D	1805	syntax error in command or file		
0x70E	1806	objects do not match		
0x70F	1807	object already exists		
0x710	1808	symbol not found		
0x711	1809	symbol version invalid	Onlinechange	Release handle and get a new one
0x712	1810	server is in invalid state		
0x713	1811	AdsTransMode not supported		

Hex	Dec	Description	Possible Causes	Solution
0x714	1812	Notification handle is invalid	Onlinechange	Release handle and get a new one
0x715	1813	Notification client not registered		
0x716	1814	no more notification handles		
0x717	1815	size for watch too big		
0x718	1816	device not initialized		
0x719	1817	device has a timeout		
0x71A	1818	query interface failed		
0x71B	1819	wrong interface required		
0x71C	1820	class ID is invalid		
0x71D	1821	object ID is invalid		
0x71E	1822	request is pending		
0x71F	1823	request is aborted		
0x720	1824	signal warning		
0x721	1825	invalid array index		
0x722	1826	symbol not active	Onlinechange	Release handle and get a new one
0x723	1827	access denied		
0x724	1828	missing license		Activate license for TwinCAT 3 function
0x72c	1836	exception occured during system start		Check each device transistions
0x740	1856	Error class <client error>		
0x741	1857	invalid parameter at service		
0x742	1858	polling list is empty		
0x743	1859	var connection already in use		
0x744	1860	invoke ID in use		
0x745	1861	timeout elapsed		Check ADS routes of sender and receiver and your firewall setting
0x746	1862	error in win32 subsystem		
0x747	1863	Invalid client timeout value		
0x748	1864	ads-port not opened		
0x750	1872	internal error in ads sync		
0x751	1873	hash table overflow		
0x752	1874	key not found in hash		
0x753	1875	no more symbols in cache		
0x754	1876	invalid response received		
0x755	1877	sync port is locked		

RTime Error Codes

Hex	Dec	Description	Possible Causes
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

Hex	Dec	Description	Possible Causes
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
0x1008	4104	TwinCAT reserved.	
0x1009	4105	TwinCAT reserved.	
0x100A	4106	TwinCAT reserved.	
0x100B	4107	TwinCAT reserved.	
0x100C	4108	TwinCAT reserved.	
0x100D	4109	RTERR_EXTIRQALREADYDEF	An external synchronisation interrupt is already applied
0x100E	4110	RTERR_EXTIRQNOTDEF	No external synchronsiation interrupt applied
0x100F	4111	RTERR_EXTIRQINSTALLFAILED	The apply of the external synchronisation 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 BIOS.
0x1019	4121	RTERR_VMXCONTROLSMISSING	Missing feature in Intel VT-x extension.
0x101A	4122	RTERR_VMXENABLEFAILS	Enabling Intel VT-x fails.

TCP Winsock Error Codes

Hex	Dec	Description	Possible Causes	Solution
0x274c	10060	A socket operation was attempted to an unreachable host	Host unreachable	Check network connection via ping
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.	Host unreachable	Check network connection via ping
0x2751	10065	No connection could be made because the target machine actively refused it		
		Further Winsock error codes: Win32 Error Codes		

9.3.3 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.

Hex	Dez	Description
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).

More Information:
www.beckhoff.com/ts6300

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