# 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 ............................................................................................................................. 15  
4.1 Introduction to File Transfer Protocol (FTP) ........................................................................ 15  
4.2 Configure Data-Ports for active FTP .................................................................................... 16  
4.3 Activating an error logfile .................................................................................................... 17

5 PLC libraries ............................................................................................................................ 19  
5.1 Tc2_FTP ................................................................................................................................ 19  
5.2 Function Blocks .................................................................................................................... 20  
5.2.1 FB_FTP_HostResolve ......................................................................................................... 20  
5.2.2 FB_FTP_Open .................................................................................................................... 21  
5.2.3 FB_FTP_OpenEX ............................................................................................................... 22  
5.2.4 FB_FTP_Close ................................................................................................................... 23  
5.2.5 FB_FTP_CloseAll .............................................................................................................. 24  
5.2.6 FB_FTP_Info .................................................................................................................... 25  
5.2.7 FB_FTP_FileUpload .......................................................................................................... 26  
5.2.8 FB_FTP_FileUploadEx ...................................................................................................... 27  
5.2.9 FB_FTP_FileDownload .................................................................................................... 28  
5.2.10 FB_FTP_FileDownloadEx ............................................................................................... 29  
5.2.11 FB_FTP_DirCreate ......................................................................................................... 30  
5.2.12 FB_FTP_DirRemove ....................................................................................................... 31  
5.2.13 FB_FTP_FileList ............................................................................................................. 32  
5.2.14 FB_FTP_FileListEx ......................................................................................................... 34  
5.2.15 FB_FTP_FileExist .......................................................................................................... 35  
5.2.16 FB_FTP_FileRemove ..................................................................................................... 36  
5.2.17 FB_FTP_FileRename ...................................................................................................... 37  
5.2.18 FB_GetStateTcFTPClient .............................................................................................. 38  
5.3 Functions [obsolete] .............................................................................................................. 39  
5.3.1 F_GetVersionTcFTPClient .............................................................................................. 39  
5.4 Data Types ........................................................................................................................... 40  
5.4.1 T_HFTP .......................................................................................................................... 40  
5.4.2 ST_FTP_ConnInfo ........................................................................................................... 40  
5.4.3 ST_FTP_FileDetails ......................................................................................................... 41  
5.4.4 E_FTP_ConnMode ........................................................................................................... 41  
5.5 Constants ............................................................................................................................. 42  
5.5.1 Konstanten ..................................................................................................................... 42

6 Samples ..................................................................................................................................... 43
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Samples</td>
<td>43</td>
</tr>
<tr>
<td>6.2 TwinCAT FTP Client: Upload of a file to</td>
<td>43</td>
</tr>
<tr>
<td>a FTP Server</td>
<td></td>
</tr>
<tr>
<td>6.3 TwinCAT FTP Client: Downloading a file</td>
<td>44</td>
</tr>
<tr>
<td>from a FTP Server to an ADS device</td>
<td></td>
</tr>
<tr>
<td>6.4 TwinCAT FTP Client: Getting connection</td>
<td>45</td>
</tr>
<tr>
<td>information with the FB_FTP_Info</td>
<td></td>
</tr>
<tr>
<td>6.5 TwinCAT FTP Client: Removing a file from</td>
<td>47</td>
</tr>
<tr>
<td>the FTP Server</td>
<td></td>
</tr>
<tr>
<td>6.6 TwinCAT FTP Client: Reading of a filelist</td>
<td>48</td>
</tr>
<tr>
<td>from the FTP Server</td>
<td></td>
</tr>
<tr>
<td>7 Appendix</td>
<td>51</td>
</tr>
<tr>
<td>7.1 Return Codes</td>
<td>51</td>
</tr>
<tr>
<td>7.1.1 Overview of the TwinCAT FTP Client</td>
<td>51</td>
</tr>
<tr>
<td>Error Codes</td>
<td></td>
</tr>
<tr>
<td>7.1.2 ADS Return Codes</td>
<td>51</td>
</tr>
<tr>
<td>7.1.3 FTP Client Return Codes</td>
<td>56</td>
</tr>
<tr>
<td>7.2 Troubleshooting</td>
<td>57</td>
</tr>
<tr>
<td>7.2.1 Troubleshooting</td>
<td>57</td>
</tr>
<tr>
<td>7.2.2 Contact Beckhoff Support</td>
<td>57</td>
</tr>
</tbody>
</table>
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®, 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:


with corresponding applications or registrations in various other countries.

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

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

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

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

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

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

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

<table>
<thead>
<tr>
<th><strong>DANGER</strong></th>
<th>Serious risk of injury!</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>WARNING</strong></th>
<th>Risk of injury!</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>CAUTION</strong></th>
<th>Personal injuries!</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>NOTE</strong></th>
<th>Damage to the environment or devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.</td>
<td></td>
</tr>
</tbody>
</table>

**Tip or pointer**
This symbol indicates information that contributes to better understanding.
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) [15] | Important article about FTP basics. Describes the design of FTP and its specification.
Configure Data-Ports for active FTP [16] | Describes a feature to configure the used Data-Ports of TwinCAT FTP Client when using active FTP.
Activate an error logfile [17] | Describes how to activate an error logfile for diagnostics.
Overview about function blocks [19] | Provides an overview about all PLC Function Blocks of TwinCAT FTP Client.
Samples [43] | 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

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>
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 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

The following section describes how to install the TwinCAT 3 Function for Windows-based operating systems.

- The TwinCAT 3 Function setup file was downloaded from the Beckhoff website.
- Run the setup file as administrator. To do this, select the command **Run as administrator** in the context menu of the file.
  - The installation dialog opens.
- Accept the end user licensing agreement and click **Next**.
Installation

3. Enter your user data.

4. If you want to install the full version of the TwinCAT 3 Function, select **Complete** as installation type. If you want to install the TwinCAT 3 Function components separately, select **Custom**.
5. Select **Next**, then **Install** to start the installation.

![Ready to Install the Program dialog box](image)

A dialog box informs you that the TwinCAT system must be stopped to proceed with the installation.

6. Confirm the dialog with **Yes**.
7. Select **Finish** to exit the setup.

The TwinCAT 3 Function has been successfully installed and can be licensed (see Licensing [12]).

### 3.3 Licensing

The TwinCAT 3 function can be activated as a full version or as a 7-day test version. Both license types can be activated via the TwinCAT 3 development environment (XAE).

**Licensing the full version of a TwinCAT 3 Function**

A description of the procedure to license a full version can be found in the Beckhoff Information System in the documentation "TwinCAT 3 Licensing".

**Licensing the 7-day test version of a TwinCAT 3 Function**

A 7-day test version cannot be enabled for a TwinCAT 3 license dongle.

1. Start the TwinCAT 3 development environment (XAE).
2. Open an existing TwinCAT 3 project or create a new project.
3. If you want to activate the license for a remote device, set the desired target system. To do this, select the target system from the **Choose Target System** drop-down list in the toolbar.
   
   The licensing settings always refer to the selected target system. When the project is activated on the target system, the corresponding TwinCAT 3 licenses are automatically copied to this system.
4. In the Solution Explorer, double-click License in the SYSTEM subtree.

The TwinCAT 3 license manager opens.

5. Open the Manage Licenses tab. In the Add License column, check the check box for the license you want to add to your project (e.g. "TF6420: TC3 Database Server").

6. Open the Order Information (Runtime) tab.
   In the tabular overview of licenses, the previously selected license is displayed with the status "missing".
7. Click **7-Day Trial License...** to activate the 7-day trial license.

8. A dialog box opens, prompting you to enter the security code displayed in the dialog.

8. Enter the code exactly as it is displayed and confirm the entry.

9. Confirm the subsequent dialog, which indicates the successful activation.

9. In the tabular overview of licenses, the license status now indicates the expiry date of the license.

10. Restart the TwinCAT system.

10. The 7-day trial version is enabled.
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 [41] in the function block FB_FTP_OpenEx [22].

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 [16].
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


Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

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 registry keys "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.

### Requirements

**Development Environment**

- TwinCAT v3.0 Build 3102

**Target Platform**

- PC or CX (x86)

**PLC Libraries to include**

- 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:

32-bit: "HKEY_LOCAL_MACHINE\Software\Beckhoff\TwinCAT3 Functions\TF6300 FTP\Configuration\ErrorLog"
64-bit: "HKEY_LOCAL_MACHINE\Software\Wow6432Node\Beckhoff\TwinCAT3 Functions\TF6300 FTP 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.

**NOTE**

**Damage of storage medium**

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

**NOTE**

**Damage of storage medium**

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

**Requirements**

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB_GetStateTcFTPClient</td>
<td>Call state information.</td>
</tr>
<tr>
<td>FB_FTP_HostResolve</td>
<td>Resolves the given host name to an IPv4 address</td>
</tr>
<tr>
<td>FB_FTP_Open</td>
<td>Open a connection to a FTP Server. (Passive)</td>
</tr>
<tr>
<td>FB_FTP_OpenEx</td>
<td>Open a connection to a FTP Server. (Passive / Active)</td>
</tr>
<tr>
<td>FB_FTP_Close</td>
<td>Close a connection to a FTP Server.</td>
</tr>
<tr>
<td>FB_FTP_CloseAll</td>
<td>Close all connections to a FTP Server</td>
</tr>
<tr>
<td>FB_FTP_Info</td>
<td>Get information from all opened connections</td>
</tr>
<tr>
<td>FB_FTP_FileUpload</td>
<td>Copy a file to a FTP Server. Use an existing connection.</td>
</tr>
<tr>
<td>FB_FTP_FileUploadEx</td>
<td>Copy a file to a FTP Server.</td>
</tr>
<tr>
<td>FB_FTP_FileDownload</td>
<td>Copy a file from a FTP Server to an ADS device. Use an existing connection.</td>
</tr>
<tr>
<td>FB_FTP_FileDownloadEx</td>
<td>Copy a file from a FTP Server to an ADS device.</td>
</tr>
<tr>
<td>FB_FTP_DirCreate</td>
<td>Create a directory on the FTP Server</td>
</tr>
<tr>
<td>FB_FTP_DirRemove</td>
<td>Remove a directory on the FTP Server</td>
</tr>
<tr>
<td>FB_FTP_FileList</td>
<td>Select all files and folder which pass the searchmask.</td>
</tr>
<tr>
<td>FB_FTP_FileListEx</td>
<td>Select all files and folder with additional information which pass the searchmask.</td>
</tr>
<tr>
<td>FB_FTP_FileExist</td>
<td>Find out if the searched file exist on the FTP Server.</td>
</tr>
<tr>
<td>FB_FTP_FileRemove</td>
<td>Remove a file from the FTP Server.</td>
</tr>
<tr>
<td>FB_FTP_FileRename</td>
<td>Rename a file on the FTP Server.</td>
</tr>
</tbody>
</table>

Functions

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F_GetVersionTcFTPClient</td>
<td>Call version information.</td>
</tr>
</tbody>
</table>
Data Types

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_HFTP[40]</td>
<td>Handle of the FTP Server.</td>
</tr>
<tr>
<td>ST_FTP_ConnInfo[40]</td>
<td>Information of existing FTP connections.</td>
</tr>
<tr>
<td>ST_FTP_FileDetails[41]</td>
<td>Returning file information of the FTP &quot;LIST&quot; command</td>
</tr>
<tr>
<td>E_FTP_ConnMode[41]</td>
<td>FTP connection modes (Passiv / Aktiv).</td>
</tr>
</tbody>
</table>

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

5.2 Function Blocks

5.2.1 FB_FTP_HostResolve

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 \[51\] when the bError output is set.

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

### Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

#### 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**

```plaintext
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**

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

**bBusy** : Indicates if the function is active.

**bError** : Indicates if an error occurred during the operation.

**nErrID** : ErrorCode when the bError output is set.
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 [51] when the bError output is set.

hFTP : Handle of a FTP Server connection.

### Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

Also see about this

[T_HFTP [40]]

#### 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**

```plaintext
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 [41]**

- **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

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

### T_HFTP

**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 [51] when the bError output is set.

**hFTP**: Handle of a FTP Server connection.

### Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

### 5.2.4 FB_FTP_Close

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

#### VAR_INPUT

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

### T_HFTP

**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 [51] when the bError output is set.

### Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

#### 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 [51] when the bError output is set.

### Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

#### 5.2.6 FB_FTP_Info

![FB_FTP_Info Block Diagram]

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

**VAR_INPUT**

```plaintext
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 [40]**

- **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**

```plaintext
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 [51] when the bError output is set.

nEntries: Number of existing connections.

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

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

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

T_HFTP [40]

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

```plaintext
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 [51] when the bError output is set.

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

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

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

```plaintext
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 [51] when the bError output is set.

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

### Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

### 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

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

**T_HFTP [40]**

**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

VAR_INPUT

VAR_INPUT

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

5.2.10 FB_FTP_FileDownloadEx

FB_FTP_FILEDOWNLOADEX:

VAR_INPUT

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
nErrID : UDINT;
nProgress : UDINT;

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;

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

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 [51] 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.
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 [51] 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

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

5.2.11 FB_FTP_DirCreate

You can create folder on a 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 [40]

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 [51] when the bError output is set.

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

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

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.

dsDirectory : 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

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 [51] when the bError output is set.

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

Also see about this

T_HFTP [40]

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.

<table>
<thead>
<tr>
<th>Beispiel</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Read all filenames</td>
</tr>
<tr>
<td>*</td>
<td>Read all filenames and all folder names</td>
</tr>
<tr>
<td>*.txt</td>
<td>Read all filenames with the extension &quot;TXT&quot;</td>
</tr>
<tr>
<td>\Test*.*</td>
<td>Read all filenames of the subfolder Test</td>
</tr>
</tbody>
</table>

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 [51] when the bError output is set.

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

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

Also see about this

- T_HFTP [40]
5.2.14 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

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 : 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 [41] 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 [51] when the bError output is set.

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

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

Also see about this

T_HFTP [40]

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 := '/repos/';
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 [51] when the bError output is set.

bExist : Becomes TRUE, if the searched file exist.

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

Also see about this

T_HFTP [40]

5.2.16 FB_FTP_FileRemove

FB_FTP_FILEREMOVE

sNetID : T_AmsNetId := '';
bBusy : BOOL;
hFTP : T_HFTP;
sFile : T_MaxString := '';
bExecute : BOOL;
ErrID : UDINT;
tTimeout : TIME := T#15s;

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 |
| bBusy : BOOL |
| hFTP : T_HFTP |
| sFile : T_MaxString |
| 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.

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 [51] when the bError output is set.

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

Also see about this

[ T_HFTP [40] ]

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 [51] when the bError output is set.

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

Also see about this

T_HFTP [40]

### 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;
```
**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 [51] 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

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

### 5.3 Functions [obsolete]

#### 5.3.1 F_GetVersionTcFTPClient

```
F_GETVERSIONTCFTPCLIENT

nVersionElement : INT F_GetVersionTcFTPClient : UINT
```

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

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

#### 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

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

#### 5.4.4 E_FTP_ConnMode

```
TYPE E_FTP_ConnMode :
  eConnMode_PASSIVE := 0,
  eConnMode_ACTIVE := 1
```
Requirements

### Development Environment

<table>
<thead>
<tr>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

### Constants

#### Konstanten

<table>
<thead>
<tr>
<th>AMSPORT_FTPADSSRV</th>
<th>UINT</th>
<th>:= 10900;</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTPADS_IGR_CONNOPEN</td>
<td>UDINT</td>
<td>:= 16#100;</td>
</tr>
<tr>
<td>FTPADS_IGR_RESOLVEDNS</td>
<td>UDINT</td>
<td>:= 16#101;</td>
</tr>
<tr>
<td>FTPADS_IGR_CONNCLOSE</td>
<td>UDINT</td>
<td>:= 16#200;</td>
</tr>
<tr>
<td>FTPADS_IGR_CONNCLOSEALL</td>
<td>UDINT</td>
<td>:= 16#201;</td>
</tr>
<tr>
<td>FTPADS_IGR_CONNINFO</td>
<td>UDINT</td>
<td>:= 16#300;</td>
</tr>
<tr>
<td>FTPADS_IGR_FILEUPLOAD</td>
<td>UDINT</td>
<td>:= 16#10000;</td>
</tr>
<tr>
<td>FTPADS_IGR_FILEDOWNLOAD</td>
<td>UDINT</td>
<td>:= 16#20000;</td>
</tr>
<tr>
<td>FTPADS_IGR_DIRRENAME</td>
<td>UDINT</td>
<td>:= 16#30000;</td>
</tr>
<tr>
<td>FTPADS_IGR_DIRCREATE</td>
<td>UDINT</td>
<td>:= 16#30001;</td>
</tr>
<tr>
<td>FTPADS_IGR_FILEEXIST</td>
<td>UDINT</td>
<td>:= 16#40000;</td>
</tr>
<tr>
<td>FTPADS_IGR_FILEEREMOVE</td>
<td>UDINT</td>
<td>:= 16#40002;</td>
</tr>
<tr>
<td>FTPADS_IGR_FILELIST</td>
<td>UDINT</td>
<td>:= 16#40003;</td>
</tr>
<tr>
<td>FTPADS_IGR_FILELISTEX</td>
<td>UDINT</td>
<td>:= 16#40004;</td>
</tr>
<tr>
<td>MAX_FTP_CONNECTIONS</td>
<td>UDINT</td>
<td>:= 25;</td>
</tr>
<tr>
<td>MAX_FILELIST_ITEMS</td>
<td>UDINT</td>
<td>:= 255;</td>
</tr>
<tr>
<td>DEFAULT_FTP_PORT</td>
<td>UDINT</td>
<td>:= 21;</td>
</tr>
</tbody>
</table>
6 Samples

6.1 Samples

The following table shows all available samples.

<table>
<thead>
<tr>
<th>No.</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Uploading a file to a FTP-Server [43]</td>
</tr>
<tr>
<td>2</td>
<td>Downloading a file from a FTP-Server [44]</td>
</tr>
<tr>
<td>3</td>
<td>Reading connection information [45]</td>
</tr>
<tr>
<td>4</td>
<td>Removing a file from a FTP-Server [47]</td>
</tr>
<tr>
<td>5</td>
<td>Reading a file list from a FTP-Server [48]</td>
</tr>
</tbody>
</table>

Please note:

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

<table>
<thead>
<tr>
<th>TwinCAT Version</th>
<th>Sample download</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0</td>
<td>Download</td>
</tr>
<tr>
<td>3.1</td>
<td>Download</td>
</tr>
</tbody>
</table>

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

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

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

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(  
  aNetID:= ,
  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

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

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

```plaintext
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;
```
PLC Program

RisingEdge(CLK:= startstop);
IF RisingEdge.Q THEN
  state := 1;
END_IF
CASE state OF
  0:
    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
  1:
    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
  2:
    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

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>
7 Appendix

7.1 Return Codes

7.1.1 Overview of the TwinCAT FTP Client Error Codes

<table>
<thead>
<tr>
<th>Offset + Error Code</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000 + TwinCAT System Errors</td>
<td>0x00000000-0x00007800</td>
<td>TwinCAT System Errors (ADS-Error codes inclusive)</td>
</tr>
<tr>
<td>0x00008000 + Internal TwinCAT FTP Client Errors [5]</td>
<td>0x00008000-0x000081C4</td>
<td>Internal Errors of TwinCAT FTP Clients</td>
</tr>
</tbody>
</table>

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

7.1.2 ADS Return Codes

Grouping of error codes: 0x000[51], 0x500[52], 0x700[53], 0x1000[55],...

Global error codes
## Appendix

### TF6300

<table>
<thead>
<tr>
<th>Hex</th>
<th>Dec</th>
<th>HRESULT</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>0</td>
<td>0x9811 0000</td>
<td>ERR_NOERROR</td>
<td>No error.</td>
</tr>
<tr>
<td>0x1</td>
<td>1</td>
<td>0x9811 0001</td>
<td>ERR_INTERNAL</td>
<td>Internal error.</td>
</tr>
<tr>
<td>0x2</td>
<td>2</td>
<td>0x9811 0002</td>
<td>ERR_NORTIME</td>
<td>No real-time.</td>
</tr>
<tr>
<td>0x3</td>
<td>3</td>
<td>0x9811 0003</td>
<td>ERR_ALLOCCLOCKEDMEM</td>
<td>Allocation locked – memory error.</td>
</tr>
<tr>
<td>0x4</td>
<td>4</td>
<td>0x9811 0004</td>
<td>ERR_INSERTMAILBOX</td>
<td>Mailbox full – the ADS message could not be sent. Reducing the number of ADS messages per cycle will help.</td>
</tr>
<tr>
<td>0x5</td>
<td>5</td>
<td>0x9811 0005</td>
<td>ERR_WRONGRECEIVEHMSG</td>
<td>Wrong HMSG.</td>
</tr>
<tr>
<td>0x6</td>
<td>6</td>
<td>0x9811 0006</td>
<td>ERR_TARGETPORTNOTFOUND</td>
<td>Target port not found – ADS server is not started or is not reachable.</td>
</tr>
<tr>
<td>0x7</td>
<td>7</td>
<td>0x9811 0007</td>
<td>ERR_TARGETMACHINENOTFOUND</td>
<td>Target computer not found – AMS route was not found.</td>
</tr>
<tr>
<td>0x8</td>
<td>8</td>
<td>0x9811 0008</td>
<td>ERR_UNKNOWNCMDID</td>
<td>Unknown command ID.</td>
</tr>
<tr>
<td>0x9</td>
<td>9</td>
<td>0x9811 0009</td>
<td>ERR_BADTASKID</td>
<td>Invalid task ID.</td>
</tr>
<tr>
<td>0xA</td>
<td>10</td>
<td>0x9811 000A</td>
<td>ERR_NOIO</td>
<td>No IO.</td>
</tr>
<tr>
<td>0xB</td>
<td>11</td>
<td>0x9811 000B</td>
<td>ERR_UNKNOWNAMS_CMD</td>
<td>Unknown AMS command.</td>
</tr>
<tr>
<td>0xC</td>
<td>12</td>
<td>0x9811 000C</td>
<td>ERR_WIN32ERROR</td>
<td>Win32 error.</td>
</tr>
<tr>
<td>0xD</td>
<td>13</td>
<td>0x9811 000D</td>
<td>ERR_PORTNOTCONNECTED</td>
<td>Port not connected.</td>
</tr>
<tr>
<td>0xE</td>
<td>14</td>
<td>0x9811 000E</td>
<td>ERR_INVALIDAMSLENGTH</td>
<td>Invalid AMS length.</td>
</tr>
<tr>
<td>0xF</td>
<td>15</td>
<td>0x9811 000F</td>
<td>ERR_INVALIDAMSNETID</td>
<td>Invalid AMS Net ID.</td>
</tr>
<tr>
<td>0x10</td>
<td>16</td>
<td>0x9811 0010</td>
<td>ERR_LOWINSTLEVEL</td>
<td>Installation level is too low – TwinCAT 2 license error.</td>
</tr>
<tr>
<td>0x11</td>
<td>17</td>
<td>0x9811 0011</td>
<td>ERR_NODEBUGINTAVAILABLE</td>
<td>No debugging available.</td>
</tr>
<tr>
<td>0x12</td>
<td>18</td>
<td>0x9811 0012</td>
<td>ERR_PORTDISABLED</td>
<td>Port disabled – TwinCAT system service not started.</td>
</tr>
<tr>
<td>0x13</td>
<td>19</td>
<td>0x9811 0013</td>
<td>ERR_PORTALREADYCONNECTED</td>
<td>Port already connected.</td>
</tr>
<tr>
<td>0x14</td>
<td>20</td>
<td>0x9811 0014</td>
<td>ERR_AMSSYNC_W32ERROR</td>
<td>AMS Sync Win32 error.</td>
</tr>
<tr>
<td>0x15</td>
<td>21</td>
<td>0x9811 0015</td>
<td>ERR_AMSSYNC_TIMEOUT</td>
<td>AMS Sync Timeout.</td>
</tr>
<tr>
<td>0x16</td>
<td>22</td>
<td>0x9811 0016</td>
<td>ERR_AMSSYNC_AMSERROR</td>
<td>AMS Sync error.</td>
</tr>
<tr>
<td>0x17</td>
<td>23</td>
<td>0x9811 0017</td>
<td>ERR_AMSSYNC_NINDEXINMAP</td>
<td>No index map for AMS Sync available.</td>
</tr>
<tr>
<td>0x18</td>
<td>24</td>
<td>0x9811 0018</td>
<td>ERR_INVALIDAMSPORT</td>
<td>Invalid AMS port.</td>
</tr>
<tr>
<td>0x19</td>
<td>25</td>
<td>0x9811 0019</td>
<td>ERR_NOMEMORY</td>
<td>No memory.</td>
</tr>
<tr>
<td>0x1A</td>
<td>26</td>
<td>0x9811 001A</td>
<td>ERR_TCPSEND</td>
<td>TCP send error.</td>
</tr>
<tr>
<td>0x1B</td>
<td>27</td>
<td>0x9811 001B</td>
<td>ERR_HOUSTUNREACHABLE</td>
<td>Host unreachable.</td>
</tr>
<tr>
<td>0x1C</td>
<td>28</td>
<td>0x9811 001C</td>
<td>ERR_INVALIDAMSFRAGMENT</td>
<td>Invalid AMS fragment.</td>
</tr>
<tr>
<td>0x1D</td>
<td>29</td>
<td>0x9811 001D</td>
<td>ERR_TLSSEND</td>
<td>TLS send error – secure ADS connection failed.</td>
</tr>
<tr>
<td>0x1E</td>
<td>30</td>
<td>0x9811 001E</td>
<td>ERR_ACCESSDENIED</td>
<td>Access denied – secure ADS access denied.</td>
</tr>
</tbody>
</table>

### Router error codes

<table>
<thead>
<tr>
<th>Hex</th>
<th>Dec</th>
<th>HRESULT</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x500</td>
<td>1280</td>
<td>0x9811 0500</td>
<td>ROUTERERR_NOLOCKEDMEMORY</td>
<td>Locked memory cannot be allocated.</td>
</tr>
<tr>
<td>0x501</td>
<td>1281</td>
<td>0x9811 0501</td>
<td>ROUTERERR_RESIZEMEMORY</td>
<td>The router memory size could not be changed.</td>
</tr>
<tr>
<td>0x502</td>
<td>1282</td>
<td>0x9811 0502</td>
<td>ROUTERERR_MAILBOXFULL</td>
<td>The mailbox has reached the maximum number of possible messages.</td>
</tr>
<tr>
<td>0x503</td>
<td>1283</td>
<td>0x9811 0503</td>
<td>ROUTERERR_DEBUGBOXFULL</td>
<td>The Debug mailbox has reached the maximum number of possible messages.</td>
</tr>
<tr>
<td>0x504</td>
<td>1284</td>
<td>0x9811 0504</td>
<td>ROUTERERR_UNKNOWNPORTTYPE</td>
<td>The port type is unknown.</td>
</tr>
<tr>
<td>0x505</td>
<td>1285</td>
<td>0x9811 0505</td>
<td>ROUTERERR_NOTINITIALIZED</td>
<td>The router is not initialized.</td>
</tr>
<tr>
<td>0x506</td>
<td>1286</td>
<td>0x9811 0506</td>
<td>ROUTERERR_PORTALREADYINUSE</td>
<td>The port number is already assigned.</td>
</tr>
<tr>
<td>0x507</td>
<td>1287</td>
<td>0x9811 0507</td>
<td>ROUTERERR_NOTREGISTERED</td>
<td>The port is not registered.</td>
</tr>
<tr>
<td>0x508</td>
<td>1288</td>
<td>0x9811 0508</td>
<td>ROUTERERR_NOMOREQUEUES</td>
<td>The maximum number of ports has been reached.</td>
</tr>
<tr>
<td>0x509</td>
<td>1289</td>
<td>0x9811 0509</td>
<td>ROUTERERR_INVALIDPORT</td>
<td>The port is invalid.</td>
</tr>
<tr>
<td>0x50A</td>
<td>1290</td>
<td>0x9811 050A</td>
<td>ROUTERERR_NOTACTIVATED</td>
<td>The router is not active.</td>
</tr>
<tr>
<td>0x50B</td>
<td>1291</td>
<td>0x9811 050B</td>
<td>ROUTERERR_FRAGMENTBOXFULL</td>
<td>The mailbox has reached the maximum number for fragmented messages.</td>
</tr>
<tr>
<td>0x50C</td>
<td>1292</td>
<td>0x9811 050C</td>
<td>ROUTERERR_FRAGMENTTIMEOUT</td>
<td>A fragment timeout has occurred.</td>
</tr>
<tr>
<td>0x50D</td>
<td>1293</td>
<td>0x9811 050D</td>
<td>ROUTERERR_TOBEREMOVED</td>
<td>The port is removed.</td>
</tr>
</tbody>
</table>
General ADS error codes
### Appendix

<table>
<thead>
<tr>
<th>Hex</th>
<th>Dec</th>
<th>HRESULT</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x700</td>
<td>1792</td>
<td>0x9811 0700</td>
<td>ADSERR_DEVICE_ERROR</td>
<td>General device error.</td>
</tr>
<tr>
<td>0x701</td>
<td>1793</td>
<td>0x9811 0701</td>
<td>ADSERR_DEVICE_SRVNOTSUPP</td>
<td>Service is not supported by the server.</td>
</tr>
<tr>
<td>0x702</td>
<td>1794</td>
<td>0x9811 0702</td>
<td>ADSERR_DEVICE_INVALIDGRP</td>
<td>Invalid index group.</td>
</tr>
<tr>
<td>0x703</td>
<td>1795</td>
<td>0x9811 0703</td>
<td>ADSERR_DEVICE_INVALIDOFFSET</td>
<td>Invalid index offset.</td>
</tr>
<tr>
<td>0x704</td>
<td>1796</td>
<td>0x9811 0704</td>
<td>ADSERR_DEVICE_INVALIDACCESS</td>
<td>Reading or writing not permitted.</td>
</tr>
<tr>
<td>0x705</td>
<td>1797</td>
<td>0x9811 0705</td>
<td>ADSERR_DEVICE_INVALIDSIZE</td>
<td>Device is not ready to operate.</td>
</tr>
<tr>
<td>0x706</td>
<td>1798</td>
<td>0x9811 0706</td>
<td>ADSERR_DEVICE_INVALIDDATA</td>
<td>Device is busy.</td>
</tr>
<tr>
<td>0x707</td>
<td>1799</td>
<td>0x9811 0707</td>
<td>ADSERR_DEVICE_NOTREADY</td>
<td>Invalid operating system context. This can result</td>
</tr>
<tr>
<td>0x708</td>
<td>1800</td>
<td>0x9811 0708</td>
<td>ADSERR_DEVICE_BUSY</td>
<td>from use of ADS function blocks in different tasks. It may be possible</td>
</tr>
<tr>
<td>0x709</td>
<td>1801</td>
<td>0x9811 0709</td>
<td>ADSERR_DEVICE_INVALIDCONTEXT</td>
<td>to resolve this through multitasking</td>
</tr>
<tr>
<td>0x70A</td>
<td>1802</td>
<td>0x9811 070A</td>
<td>ADSERR_DEVICE_NOMEMORY</td>
<td>synchronization in the PLC.</td>
</tr>
<tr>
<td>0x70B</td>
<td>1803</td>
<td>0x9811 070B</td>
<td>ADSERR_DEVICE_INVALIDPARAM</td>
<td>Invalid memory.</td>
</tr>
<tr>
<td>0x70C</td>
<td>1804</td>
<td>0x9811 070C</td>
<td>ADSERR_DEVICE_NOTFOUND</td>
<td>Invalid parameter values.</td>
</tr>
<tr>
<td>0x70D</td>
<td>1805</td>
<td>0x9811 070D</td>
<td>ADSERR_DEVICE_SYNTAX</td>
<td>Not found (files, ...).</td>
</tr>
<tr>
<td>0x70E</td>
<td>1806</td>
<td>0x9811 070E</td>
<td>ADSERR_DEVICE_INCOMPATIBLE</td>
<td>Syntax error in file or command.</td>
</tr>
<tr>
<td>0x70F</td>
<td>1807</td>
<td>0x9811 070F</td>
<td>ADSERR_DEVICE_EXISTS</td>
<td>Objects do not match.</td>
</tr>
<tr>
<td>0x710</td>
<td>1808</td>
<td>0x9811 0710</td>
<td>ADSERR_DEVICE_SYMBOLNOTFOUND</td>
<td>Object already exists.</td>
</tr>
<tr>
<td>0x711</td>
<td>1809</td>
<td>0x9811 0711</td>
<td>ADSERR_DEVICE_SYMBOLVERSIONINVALID</td>
<td>Symbol not found.</td>
</tr>
<tr>
<td>0x712</td>
<td>1810</td>
<td>0x9811 0712</td>
<td>ADSERR_DEVICE_INVALIDSTATE</td>
<td>Invalid symbol version. This can occur due to an online change.</td>
</tr>
<tr>
<td>0x713</td>
<td>1811</td>
<td>0x9811 0713</td>
<td>ADSERR_DEVICE_TRANSMODENOTSUPP</td>
<td>Create a new handle.</td>
</tr>
<tr>
<td>0x714</td>
<td>1812</td>
<td>0x9811 0714</td>
<td>ADSERR_DEVICE_NOTIFYHINDINVALID</td>
<td>AdsTransMode not supported.</td>
</tr>
<tr>
<td>0x715</td>
<td>1813</td>
<td>0x9811 0715</td>
<td>ADSERR_DEVICE_CLIENTUNKNOWN</td>
<td>Notification handle is invalid.</td>
</tr>
<tr>
<td>0x716</td>
<td>1814</td>
<td>0x9811 0716</td>
<td>ADSERR_DEVICE_NOMOREHDDS</td>
<td>Notification client not registered.</td>
</tr>
<tr>
<td>0x717</td>
<td>1815</td>
<td>0x9811 0717</td>
<td>ADSERR_DEVICE_INVALIDWATCHSIZE</td>
<td>No further notification handle available.</td>
</tr>
<tr>
<td>0x718</td>
<td>1816</td>
<td>0x9811 0718</td>
<td>ADSERR_DEVICE_NOTINIT</td>
<td>Notification size too large.</td>
</tr>
<tr>
<td>0x719</td>
<td>1817</td>
<td>0x9811 0719</td>
<td>ADSERR_DEVICE_TIMEOUT</td>
<td>Device not initialized.</td>
</tr>
<tr>
<td>0x71A</td>
<td>1818</td>
<td>0x9811 071A</td>
<td>ADSERR_DEVICE_NODATA</td>
<td>Interface query failed.</td>
</tr>
<tr>
<td>0x71B</td>
<td>1819</td>
<td>0x9811 071B</td>
<td>ADSERR_DEVICE_INVALIDINTERFACE</td>
<td>Wrong interface requested.</td>
</tr>
<tr>
<td>0x71C</td>
<td>1820</td>
<td>0x9811 071C</td>
<td>ADSERR_DEVICE_INVALIDCLSID</td>
<td>Invalid ID.</td>
</tr>
<tr>
<td>0x71D</td>
<td>1821</td>
<td>0x9811 071D</td>
<td>ADSERR_DEVICE_INVALIDOBJID</td>
<td>Class ID is invalid.</td>
</tr>
<tr>
<td>0x71E</td>
<td>1822</td>
<td>0x9811 071E</td>
<td>ADSERR_DEVICE_PENDING</td>
<td>Object ID is invalid.</td>
</tr>
<tr>
<td>0x71F</td>
<td>1823</td>
<td>0x9811 071F</td>
<td>ADSERR_DEVICE_ABORTED</td>
<td>Request pending.</td>
</tr>
<tr>
<td>0x720</td>
<td>1824</td>
<td>0x9811 0720</td>
<td>ADSERR_DEVICE_WARNING</td>
<td>Request is aborted.</td>
</tr>
<tr>
<td>0x721</td>
<td>1825</td>
<td>0x9811 0721</td>
<td>ADSERR_DEVICE_INVALIDARRAYIDX</td>
<td>Signal warning.</td>
</tr>
<tr>
<td>0x722</td>
<td>1826</td>
<td>0x9811 0722</td>
<td>ADSERR_DEVICE_SYMBOLNOTACTIVE</td>
<td>Invalid array index.</td>
</tr>
<tr>
<td>0x723</td>
<td>1827</td>
<td>0x9811 0723</td>
<td>ADSERR_DEVICE_ACCESSDENIED</td>
<td>Invalid symbol not active.</td>
</tr>
<tr>
<td>0x724</td>
<td>1828</td>
<td>0x9811 0724</td>
<td>ADSERR_DEVICE_LICENSENOTFOUND</td>
<td>Symbol not active.</td>
</tr>
<tr>
<td>0x725</td>
<td>1829</td>
<td>0x9811 0725</td>
<td>ADSERR_DEVICE_LICENSEEXPIRED</td>
<td>Access denied.</td>
</tr>
<tr>
<td>0x726</td>
<td>1830</td>
<td>0x9811 0726</td>
<td>ADSERR_DEVICE_LICENSEEXCEEDED</td>
<td>Missing license.</td>
</tr>
<tr>
<td>0x727</td>
<td>1831</td>
<td>0x9811 0727</td>
<td>ADSERR_DEVICE_LICENSEINVALID</td>
<td>License expired.</td>
</tr>
<tr>
<td>0x728</td>
<td>1832</td>
<td>0x9811 0728</td>
<td>ADSERR_DEVICE_LICENSESYSTEMID</td>
<td>License exceeded.</td>
</tr>
<tr>
<td>0x729</td>
<td>1833</td>
<td>0x9811 0729</td>
<td>ADSERR_DEVICE_LICENSEENOTIMELIMIT</td>
<td>Invalid license.</td>
</tr>
<tr>
<td>0x72A</td>
<td>1834</td>
<td>0x9811 072A</td>
<td>ADSERR_DEVICE_LICENSEFUTUREISSUE</td>
<td>Invalid license.</td>
</tr>
<tr>
<td>0x72B</td>
<td>1835</td>
<td>0x9811 072B</td>
<td>ADSERR_DEVICE_LICENSETIMELONG</td>
<td>Invalid license.</td>
</tr>
<tr>
<td>0x72C</td>
<td>1836</td>
<td>0x9811 072C</td>
<td>ADSERR_DEVICE_EXCEPTION</td>
<td>Invalid license.</td>
</tr>
<tr>
<td>0x72D</td>
<td>1837</td>
<td>0x9811 072D</td>
<td>ADSERR_DEVICE_LICENSEDEPRECATED</td>
<td>Invalid license.</td>
</tr>
<tr>
<td>0x72E</td>
<td>1838</td>
<td>0x9811 072E</td>
<td>ADSERR_DEVICE_SIGNATUREINVALID</td>
<td>Invalid license.</td>
</tr>
<tr>
<td>0x72F</td>
<td>1839</td>
<td>0x9811 072F</td>
<td>ADSERR_DEVICE_CERTIFICATEINVALID</td>
<td>Invalid certificate.</td>
</tr>
<tr>
<td>0x730</td>
<td>1840</td>
<td>0x9811 0730</td>
<td>ADSERR_DEVICE_LICENSEOEMNOTFOUND</td>
<td>Public key not known from OEM.</td>
</tr>
<tr>
<td>0x731</td>
<td>1841</td>
<td>0x9811 0731</td>
<td>ADSERR_DEVICE_LICENSERESTRICTED</td>
<td>License not valid for this system ID.</td>
</tr>
<tr>
<td>0x732</td>
<td>1842</td>
<td>0x9811 0732</td>
<td>ADSERR_DEVICE_LICENSEDEDEMOLISHED</td>
<td>Demo license prohibited.</td>
</tr>
<tr>
<td>0x733</td>
<td>1843</td>
<td>0x9811 0733</td>
<td>ADSERR_DEVICE_INVALIDFNID</td>
<td>Invalid function ID.</td>
</tr>
<tr>
<td>0x734</td>
<td>1844</td>
<td>0x9811 0734</td>
<td>ADSERR_DEVICE_OUTOFRANGE</td>
<td>Outside the valid range.</td>
</tr>
<tr>
<td>0x735</td>
<td>1845</td>
<td>0x9811 0735</td>
<td>ADSERR_DEVICE_INVALIDALIGNMENT</td>
<td>Invalid alignment.</td>
</tr>
</tbody>
</table>
## TCP Winsock error codes

<table>
<thead>
<tr>
<th>Hex</th>
<th>Dec</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x274C</td>
<td>10060</td>
<td>WSAETIMEOUT</td>
<td>A connection timeout has occurred - error while establishing the connection, because the remote terminal did not respond properly after a certain period of time, or the established connection could not be maintained because the connected host did not respond.</td>
</tr>
<tr>
<td>0x274D</td>
<td>10061</td>
<td>WSAECONNREFUSED</td>
<td>Connection refused - no connection could be established because the target computer has explicitly rejected it. This error usually results from an attempt to connect to a service that is inactive on the external host, that is, a service for which no server application is running.</td>
</tr>
<tr>
<td>0x2751</td>
<td>10065</td>
<td>WSAEHOSTUNREACH</td>
<td>No route to host - a socket operation referred to an unavailable host.</td>
</tr>
</tbody>
</table>

More Winsock error codes: Win32 error codes
# 7.1.3 FTP Client Return Codes

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

Requirements

<table>
<thead>
<tr>
<th>Development Environment</th>
<th>Target Platform</th>
<th>PLC Libraries to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT v3.0 Build 3102</td>
<td>PC or CX (x86)</td>
<td>Tc2_FTP</td>
</tr>
</tbody>
</table>

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 [51].

• 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 [15] for more information about FTP ports.

Should the steps mentioned above don't help, please contact our support department [57].

7.2.2 Contact Beckhoff Support

7.2.2.1 Contact Beckhoff Support

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General information [57]</td>
<td>Shows which general system information needs to be forwarded to our Support department</td>
</tr>
<tr>
<td>Product-related information [58]</td>
<td>Shows which product-related information needs to be forwarded to our Support department</td>
</tr>
</tbody>
</table>

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 [60] 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 [58] 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"

![CX Configuration Tool](image-url)
More Information:
www.beckhoff.com/tf6300