Quick start guide

FTP Client over OpenNETCF FTP Library

SER0011 – X2 panel application connects to a FTP server

Beijer ELECTRONICS

1 Function and area of use

This document provides guidelines when working with Sample_FTPClient.

This document explains how to use the third-party library OpenNETCF.Net.Ftp.dll to connect to a FTP server. It implements connect, disconnect, create a directory, delete a directory, change the working directory, upload a file, download a file, rename a file on the server, get file list, etc. You can get the Connected, Disconnected, CommandSent, and ResponseReceived events to handle your business logic.

2 About this document

This quick start document should not be considered as a complete manual. It is an aid to be able to startup a normal application quickly and easily.

Use the following hardware, software, drivers and utilities in order to obtain a stable application:

In this document we have used following software and hardware

- iX Developer 2.4 SP5/SP6
- All X2 panels and PC Runtime

For further information refer to

- Beijer Electronics knowledge database, HelpOnline

This document and other quick start documents can be obtained from our homepage.
Please use the address *support.europe@beijerelectronics.com* for feedback.

3 Table of Contents

4 Source code

The source code of the DLL used is distributed together with the sample project.

The OpenNETCF.Net.Ftp.dll Library is an open source project compiled for

.NET Compact Framework 3.5.

As we extended the BeginConnect Method of the DLL it is mandatory to use this version of the DLL – at least if you want to use the additional "Command Delay" property (for adjusting e.g. low-speed connections or connections between slower devices).

5 Adjusting the project

5.1 Command Delay

This is an additional parameter specified utilizing the 3rd overload of the BeginConnect method. Below one can see the BeginConnect method of the script module "FTPAdapter".
For X2 devices running either WinCE6 or WinCE8 (WEC2013) the project is pre-configured to run with a Command Delay of 250ms, for non-CE devices no 3rd parameter is specified → the default 100ms delay is used.

```
/// <summary>
/// Connect to the FTP server using the supplied username and password
/// </summary>
public void BeginConnect()
{
    try
    {
        // check OS and configure the command delay depending to your setup (server/client) to get a reliable system
        if(System.Environment.OSVersion.Platform == PlatformID.WinCE)
        {
            FTPInstance.BeginConnect(Globals.Tags.UserName.Value, Globals.Tags.Password.Value,250); //command delay 250ms
        }
        else
        {
            FTPInstance.BeginConnect(Globals.Tags.UserName.Value, Globals.Tags.Password.Value); //default delay = 100ms
        }
    }
```

5.2 FTP server specific adjustments

In multiple methods you have the possibility to adjust the script code to react properly to what the FTP server responses. One example is the "_ftp_ResponseReceived" method where one can adjust e.g. what the server responses after a directory change.

5.3 Active FTP vs. Passive FTP

The FTP protocol has 2 modes, PORT (also called "regular" or "normal" mode) and PASV

("passive" mode). The FTP client determines the mode that will be used. If the client issues a PORT command, it is attempting "PORT" mode. If the client issues a PASV command, it is attempting "PASV" mode.

PASV mode forces the client-server data connection to be established by the client, rather than the server (which is the default). PASV mode is required for users who are behind a router-based firewall.

Details:

In PORT and PASV mode the FTP session uses two port numbers.

The first port number is allocated on the server (default: port 21). This is referred to as the "Control or Command Channel".

If the client sends a PORT command, which contains the client IP address and (the second) allocated port number.

Then the FTP server connects to this port to send the data (referred to as the "Data Channel").

Client: PORT 127,0,0,1,12,82

Server: 200 Port command successful.

If the client sends a PASV command, then the FTP server responds back to the client with a port number that it has allocated.

In this case the "Data Channel" connection is created by the client to the IP Address and port number provided by the FTP server.

Client: PASV

Server: 227 Entering Passive Mode (127,0,0,1,12,251)

Also checkout the below website for more information about PASV and PORT mode.

http://slacksite.com/other/ftp.html

6 Add additional controls

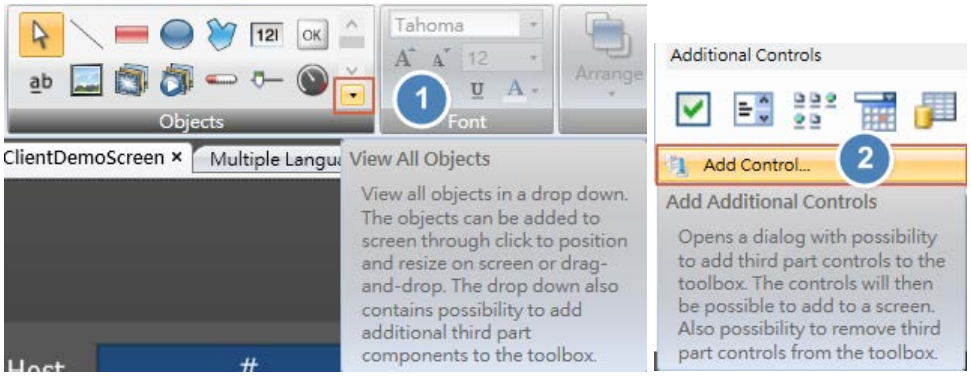
Before you could open this sample successfully, you need to add WinForms controls that are used by the “FTPClientDemoScreen” demonstration. The controls should be added as follows:

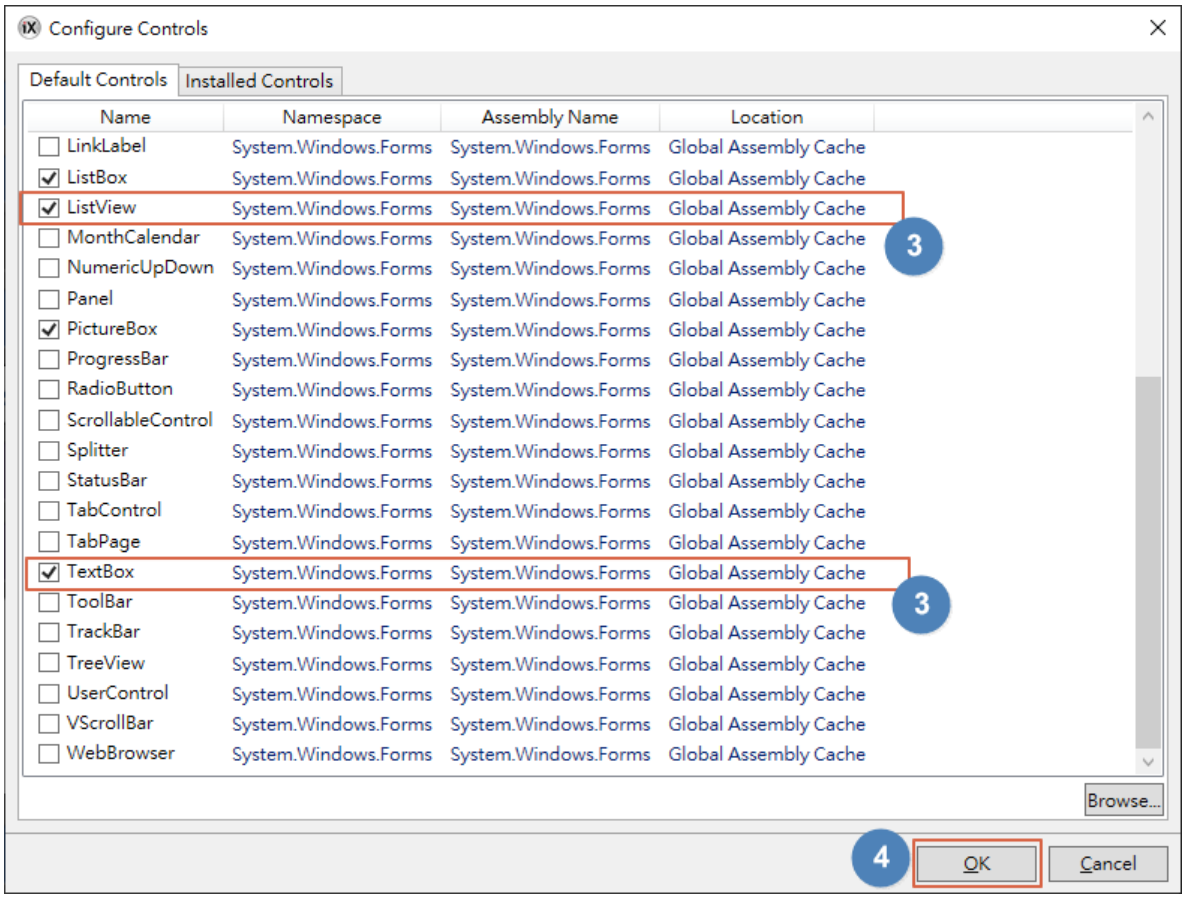
- ListView (System.Windows.Forms)
- TextBox (System.Windows.Forms)

6.1 Add Control

The following description shows how to add additional controls.

“Home” → “View All Objects” of “Object” → “Add Control” of “Additional Controls” → check controls as desired → “OK”



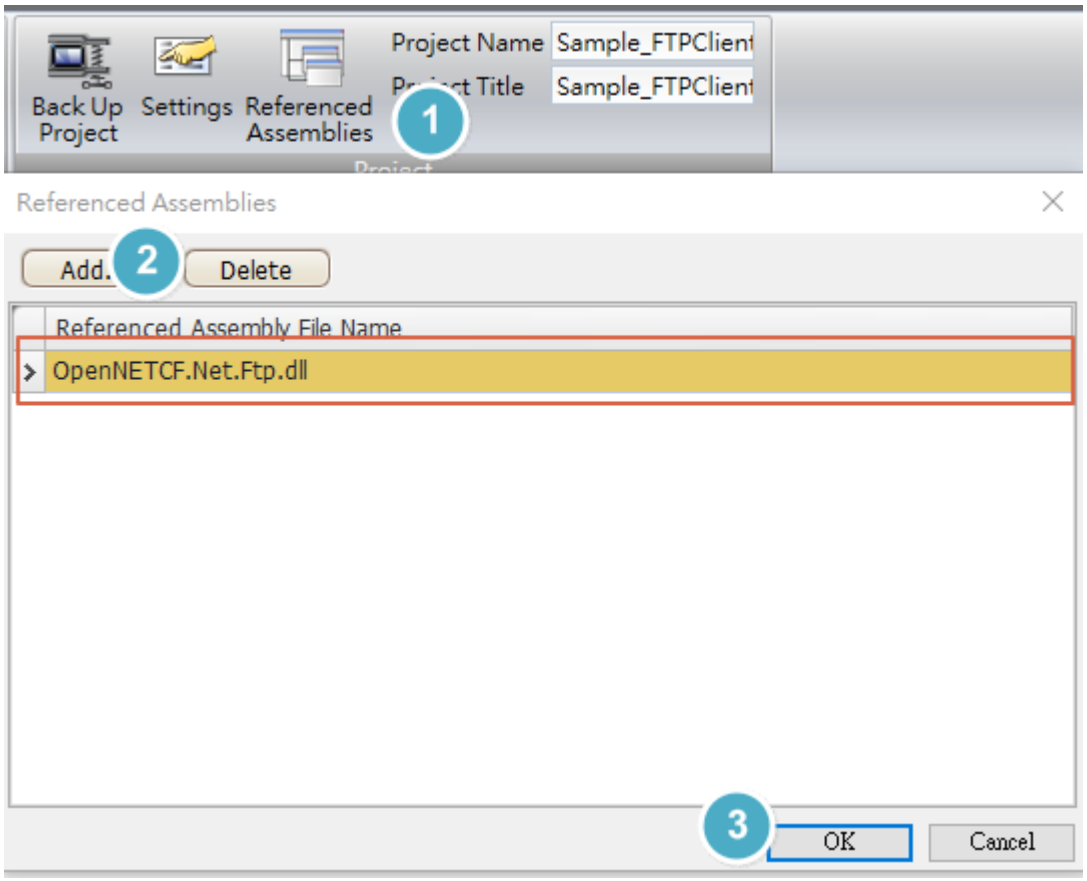


7 Migration

7.1 Add reference

"Project" → "Referenced Assemblies" → "Add" → browse to and select the assembly "OpenNETCF.Net.Ftp.dll" → "OK"

Open your iX project and click the "Referenced Assemblies" on the "Project" tab to add the reference.



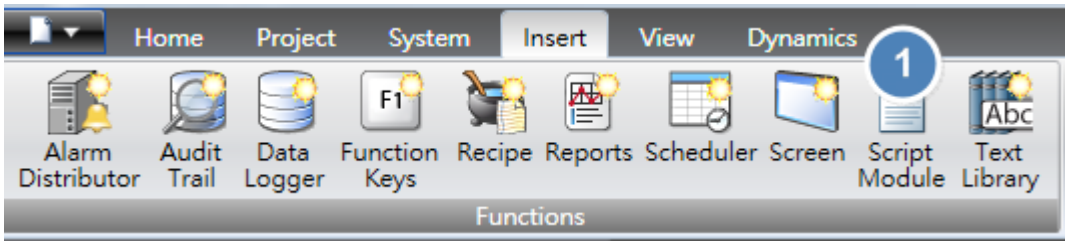
Notice: If you want to replace an assembly with a new one that has the same file name, you need to "Delete" the current assembly and "Save" and "Close" the project (iX Developer). And then, re-open the project and add a new reference because iX Developer otherwise will not replace an existing DLL file.

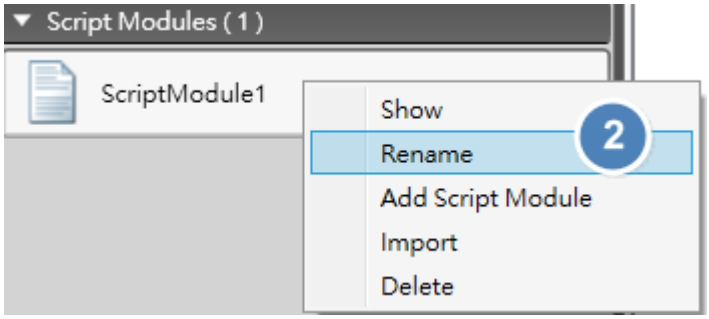
7.2 Add a ScriptModule

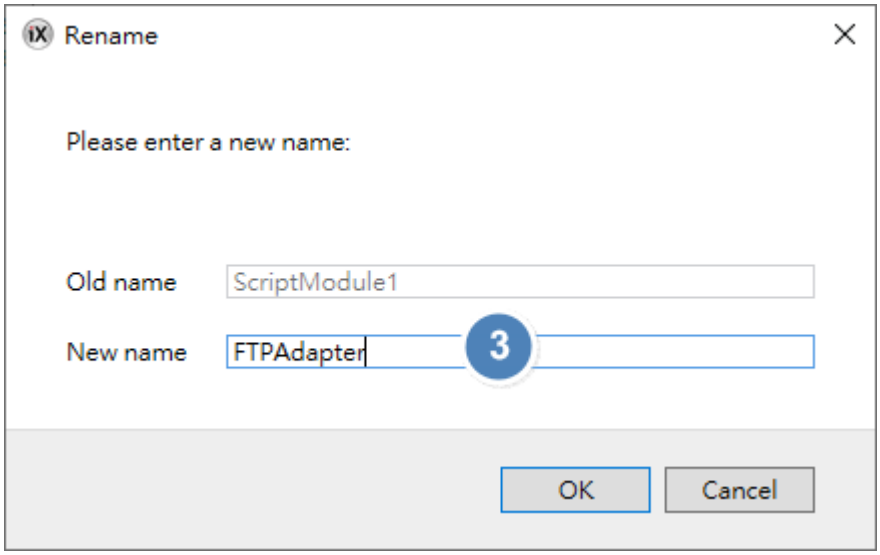
"Insert" → "Script Module" → "Rename" → input new name "FTPAdapter" → copy all codes from the "FTPAdapter" of this sample & paste these codes on "FTPAdapter" of your project → Hookup "FTPAdapter_Created" event handler with "Created" event of the "FTPAdapter"script module

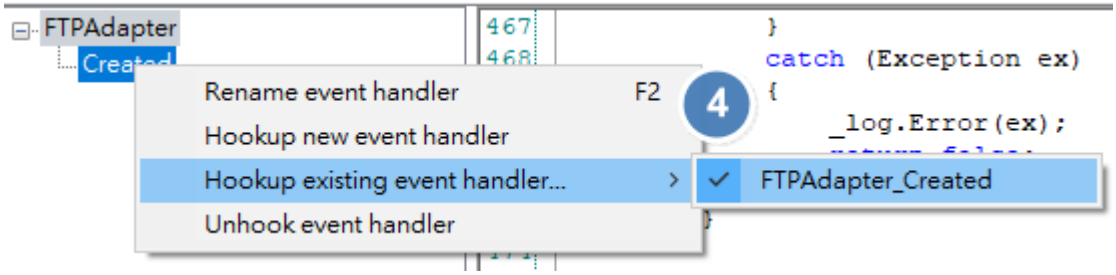
You can utilize it to interact with an FTP Server like the "FTPClientDemoScreen" demonstration.

Insert a script module by clicking the “Script Module” on the “Insert” tab. Rename the Script Module as desired.









A Script Module is a class in C# .NET. It would be used to create a singleon object while initialization processes of an iX application. All the objects created by you via iX Developer, i.e. not created by scripting, would be the properties of the Globals object. So you can use the “Globals” object name to get all these objects. For example, after inserting a Script Module with the name, ExampleScriptModule, and write a public method, ExampleMethod, you can use “Globals.ExampleScriptModule.ExampleMethod()” in scripting to call the method of it.

7.3 Tags

Mandatory Tag of FTPAdapter

Tag Name	Global Data Type	Description
FTPServerUri	STRING	Remote FTP server URI
FTPServerPort	INT32	Remote FTP server port number
FTPIsConnected	BOOL	Indicate the connection status
UserName	STRING	The account/user name is used to log in FTP server
Password	STRING	The password is used to log in FTP server
FTPServerType	STRING	Get the FTP server type (It would be updated after the connection established.)
FTPRemoteDirectory	STRING	Current working directory on FTP server (It would be available after the connection established.)

Other tags are not necessary, they just are used to build the demonstrations.
You can refer to the demos to find out how they work.

7.4 APIs

FTPAdapter class : Script Module

Properties:

Name	Data Type	Description
IsConnected	bool	Get the connection status
RemoteDirectory	string	The remote directory
TransferType	FTPTransferType	The transfer type: Binary or ASCII
BufferSize	int	Sets the size of the internal buffer used for FTP transfers. Default is 512 bytes
ConnectionTimeout	int	Timeout for FTP operations
ExceptionOnError	bool	When set, if an error is encountered while using the class instance, an exception will be thrown
ServerType	FTPServerType	The remote server type: Unix-compliant server, Windows/IIS-compliant server or Unknown server type

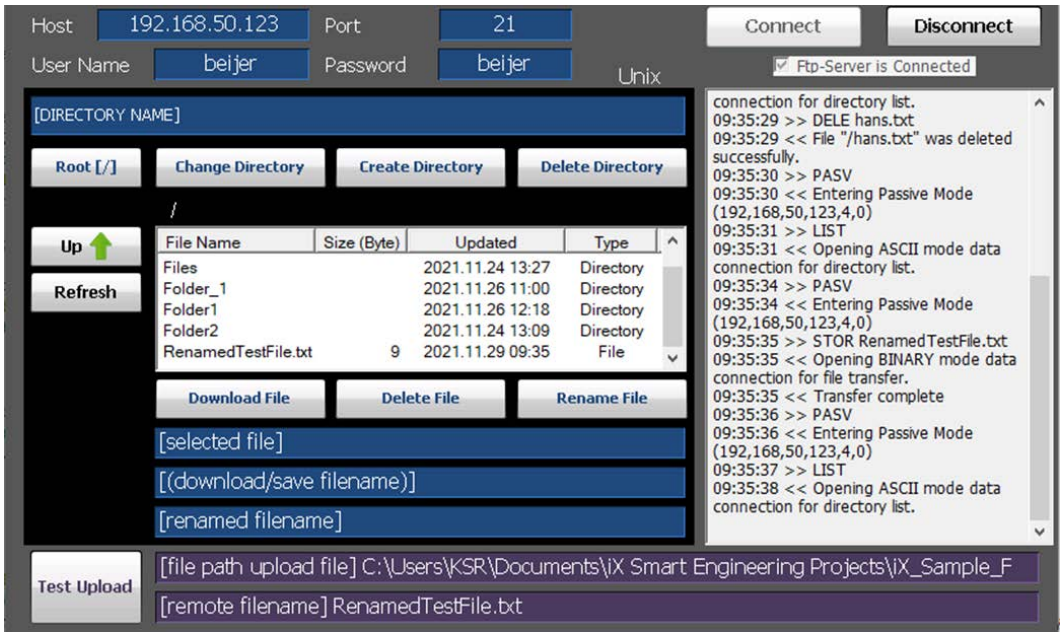
Events:

Name	Description
Connected	Triggered after the connection established
Disconnected	Triggered after the connection closed
CommandSent	Triggered after a command sent
ResponseReceived	Triggered after a response received

Methods:

Name	Return Data Type	Description
BeginConnect()	void	Connect to the FTP server using the supplied username and password
Disconnect()	void	Disconnect from the FTP server
ChangeDirectory(string **directory)**	void	Change the remote working directory
CreateDirectory(string **directory)**	void	Create a directory on the FTP server
DeleteDirectory(string **directory)**	void	Delete an empty directory from the FTP server
DeleteFile(string **fileName)**	void	Delete a file on the FTP server
EnumFiles()	FTPFiles	Get a list of the files in the current remote directory
GetFile(string **remoteFileName,** string **localFileName,** bool **overwrite)**	void	Retrieves a file from the FTP server
GetFileList(bool **detailed)**	string	Retrieves the filelist string as the FTP server sends it
RenameFile(string **currentFileName,** string **newFileName)**	void	Rename a file on the FTP server
SendCommand(string **command)**	bool	Send a generic or server-specific command to the FTP server
SendFile(string **localFilePath,** string **remoteFileName)**	bool	Send a file to the FTP server

8 Example screen



9 About Beijer Electronics

Beijer Electronics is a multinational, cross-industry innovator that connects people and technologies to optimize processes for business-critical applications. Our offer includes operator communication, automation solutions, digitalization, display solutions and support. As experts in user-friendly software, hardware and services for the Industrial Internet of Things, we empower you to meet your challenges through leading-edge solutions. Beijer Electronics is a Beijer Group company.

Since its start-up in 1981, BEIJER GROUP has evolved into a multinational group with sales of 1.4 billion SEK 2020. BEIJER GROUP is listed on the NASDAQ Stockholm Main Market under the ticker BELE. www.beijergroup.com


China


Denmark


France


Germany


India


Norway


South Korea


Sweden HQ


Taiwan


Turkey


United Kingdom


USA

9.1 Contact us

Global offices and distributors