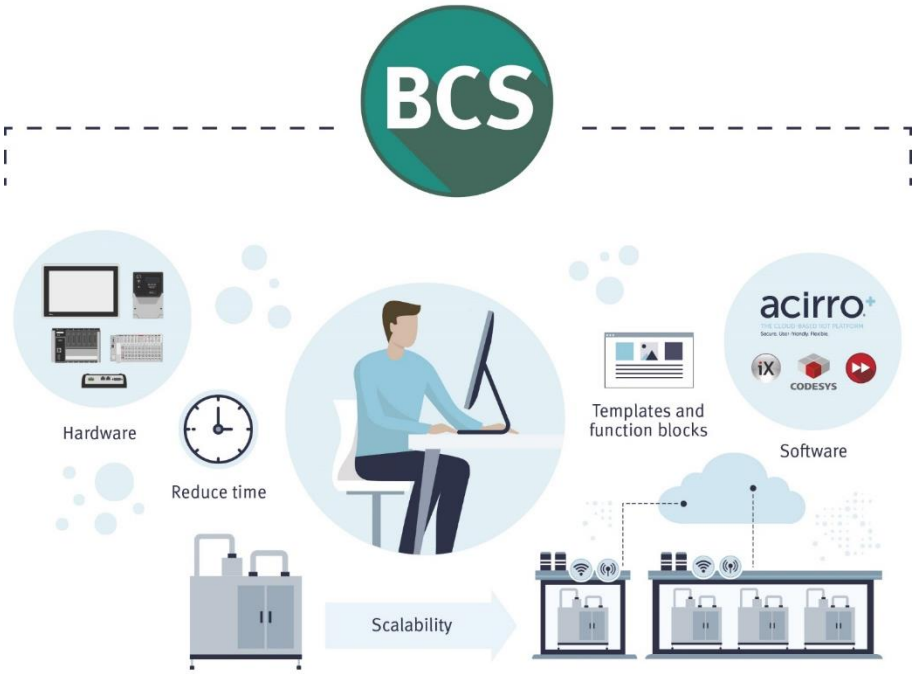
Quick start guide

Realise a TCP Client

SER0019 - TCP Client



Beijer ELECTRONICS

1 Function and area of use

This document provides guidelines when working with the CODESYS library known as SER0019, and it's a simple interface to create a TCP client or server.

Target device: X2 / BoX2 control series, with embedded CODESYS runtime.

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

- BCS Tools V3.31 or
- CODESYS 3.5 SP13 Patch 3
- X2 control and BoX2 Control

For further information refer to

- CODESYS online help
- Installation manual X2 control (MAxx202)
- 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 TCP client and server

This library gives a clean, simple interface to create a TCP client or server.

It is useful when there is a requirement to communicate with non-procedural third-party devices. Examples may be Barcode readers, Vision systems and Laser markers.

Third-part devices such as Laser marker, weigh scales and inkjet printers often allow socket communications. This library manages opening a socket and sending/receiving data. Helper methods are included to process the data prior to sending and assist parsing the incoming data.

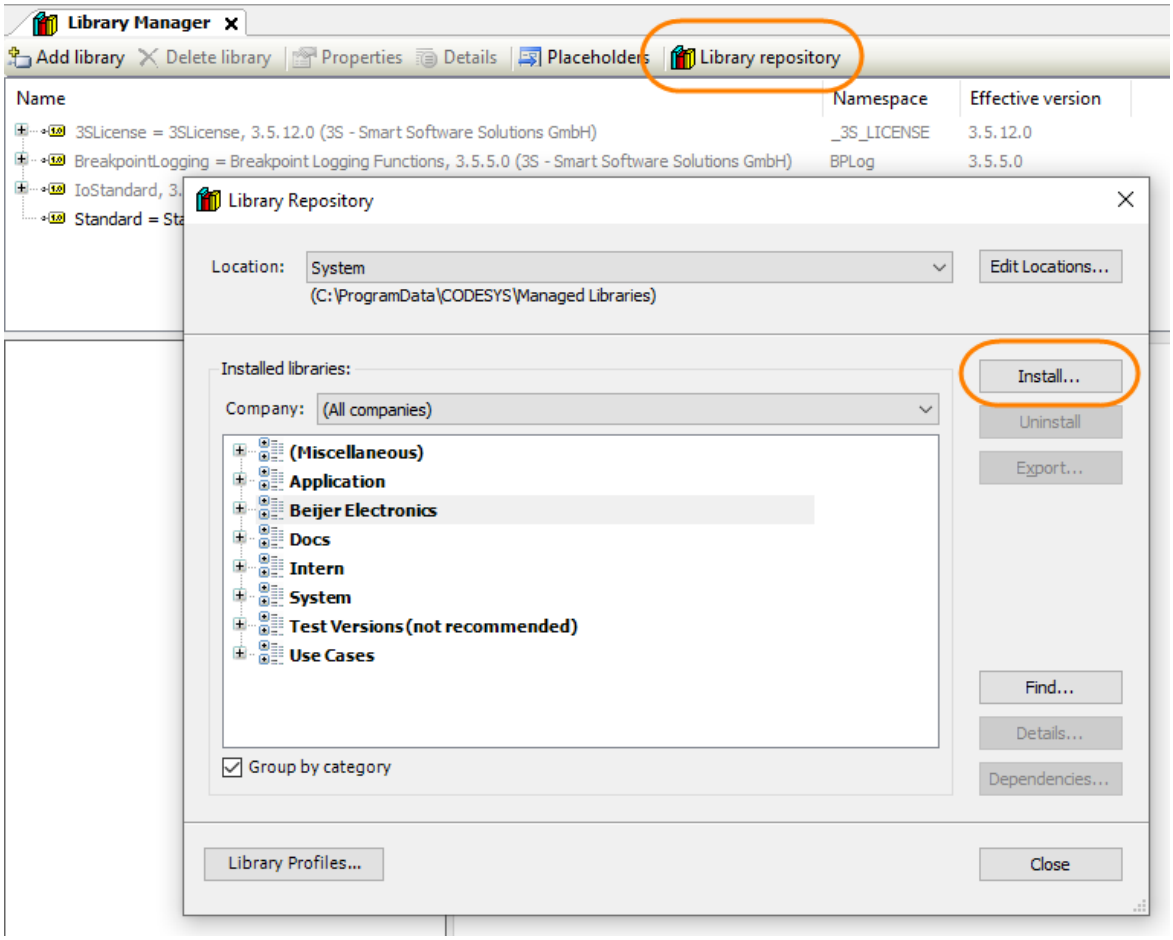
5 Preparing your editor

The following chapter describes important procedures and settings needed for a well-functioning system.

5.1 Installation of the library to your editor

The *.compiled-library needs to be made available in your system so it can be included in projects. This is done by accessing the 'Library Manager'→'Library Repository' then 'Install'

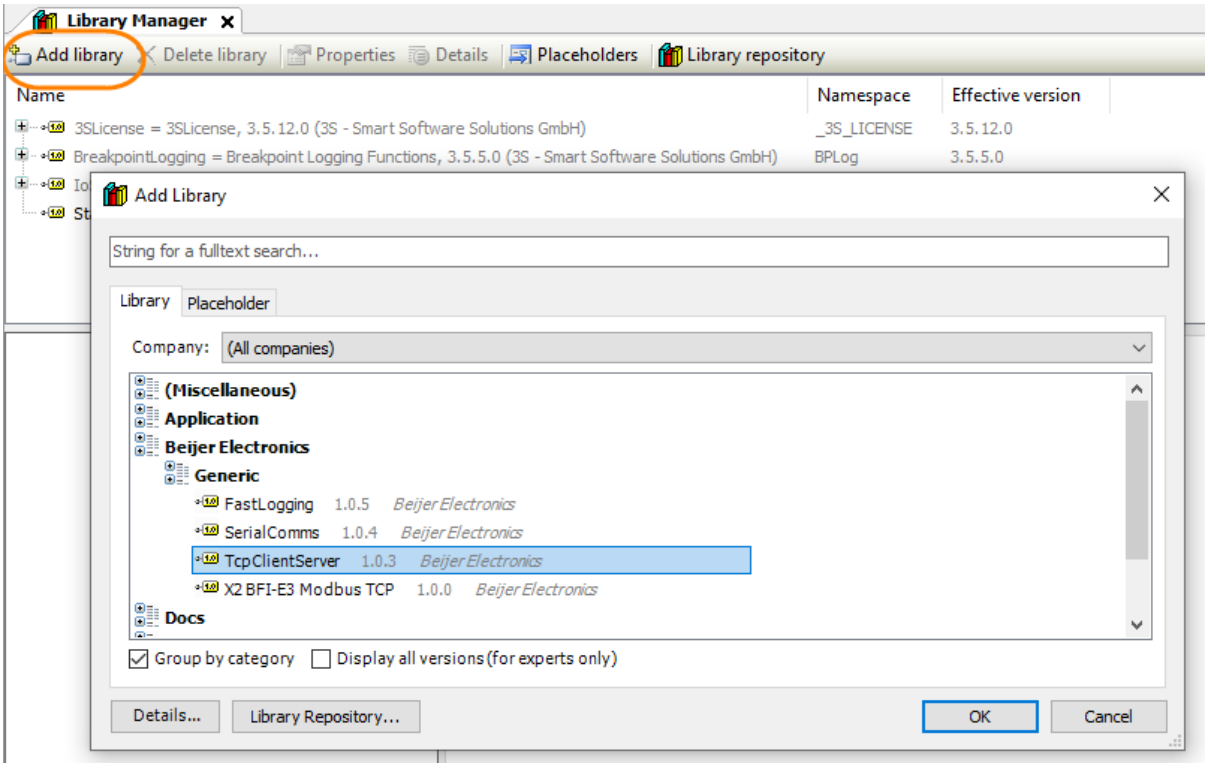
Navigate to the folder where you have put the *.compiled-library.
This procedure will need to be repeated if you use a new PC.



5.2 Add the library into your project

The library is now available for you to include in your specific project.

Note, the first time you need to go 'Advanced' menu below to find the library, and the next time you don't need to. Because previously used libraries are available immediately from the Add library dialog.



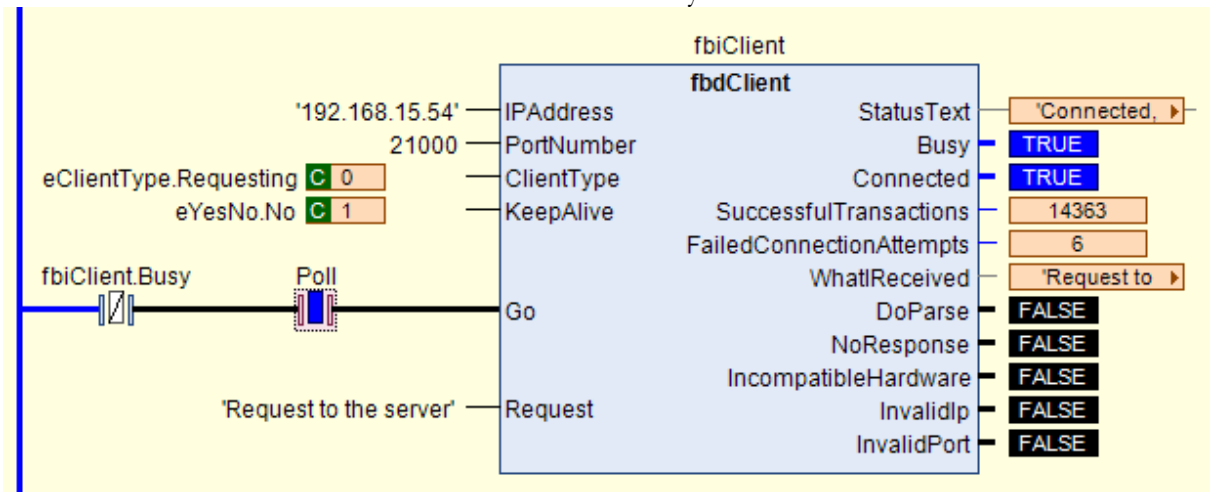
6 Using the Function Blocks

Two FBs are provided:

- **fbdClient** (the main block)
- **fbdLimiter** (contains helper methods to prepare data for sending, and parsing a response)

6.1 fbdClient

Invoke one instance of this block for each TCP Server you want to talk to.



Name	Scope	Data Type	Description
IPAddress	VAR_IN	STRING(16)	IP address of the server
PortNumber	VAR_IN	UINT	Port which the server is listening on
ClientType	VAR_IN	eClientType	Select client type
KeepAlive	VAR_IN	eYesNo	Select whether to maintain a socket, or open it for each transaction
Go	VAR_IN	BOOL	If client type is **Requesting**, this triggers a send/receive operation. If the client is **NonRequesting**, the client listens to the server as long as this is high.
Request	VAR_IN	STRING(255)	Request to send to the server
StatusText	VAR_OUT	STRING	
Busy	VAR_OUT	BOOL	
Connected	VAR_OUT	BOOL	Indicates a TCP connection is established to the IPEndPoint
SuccessfulTransactions	VAR_OUT	UDINT	Info Only
FailedConnectionAttempts	VAR_OUT	UDINT	Info Only
WhatIReceived	VAR_OUT	STRING(255)	Server's response
DoParse	VAR_OUT	BOOL	Response ready to be dealt with
NoResponse	VAR_OUT	BOOL	A server response was not received within an acceptable time. Valid only for **Requesting** type client
IncompatibleHardware	VAR_OUT	BOOL	Not X2Control

6.2 fbdLimiter

This FB and its associated Helper methods is provided to help format a string to send to the server, then parse the response into something usable. Method to delimit (convert to array of strings) and relimit (convert to <char> separated). For example, a laser marker may receive its variables in the following format:

“SetVar;12 December 2019;Batch number; 12345678<CR><LF>”

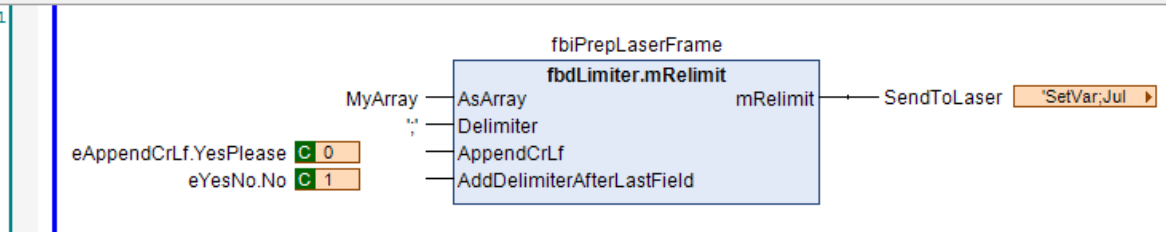
Or

“SetVar;12 December 2019;Batch number; 12345678;<CR><LF>” (extra delimiting character at end!)

The **mRelimit** method allows the user to build up the final string in an array of strings, and then insert the delimiting characters (; in this case) and append a <CR><LF>

In the following example, the string is built up simply in an array of strings. The method assembles 1 string and inserts the delimiting character. It optionally add a <CR><LF>

MyArray	ARRAY [0..20] OF S...	
MyArray[0]	STRING(40)	'SetVar'
MyArray[1]	STRING(40)	'July 2024'
MyArray[2]	STRING(40)	'Batch'
MyArray[3]	STRING(40)	'QZ19223'
MyArray[4]	STRING(40)	''
MyArray[5]	STRING(40)	''
MyArray[6]	STRING(40)	''
MyArray[7]	STRING(40)	''
MyArray[8]	STRING(40)	''
MyArray[9]	STRING(40)	''
MyArray[10]	STRING(40)	''
MyArray[11]	STRING(40)	''
MyArray[12]	STRING(40)	''
MyArray[13]	STRING(40)	''
MyArray[14]	STRING(40)	''
MyArray[15]	STRING(40)	''
MyArray[16]	STRING(40)	''
MyArray[17]	STRING(40)	''
MyArray[18]	STRING(40)	''
MyArray[19]	STRING(40)	''
MyArray[20]	STRING(40)	''
SendToLaser	STRING(255)	'SetVar;July 2024;Batch;QZ19223RN'
fbiPrepLaserFrame	FBDLIMITER	

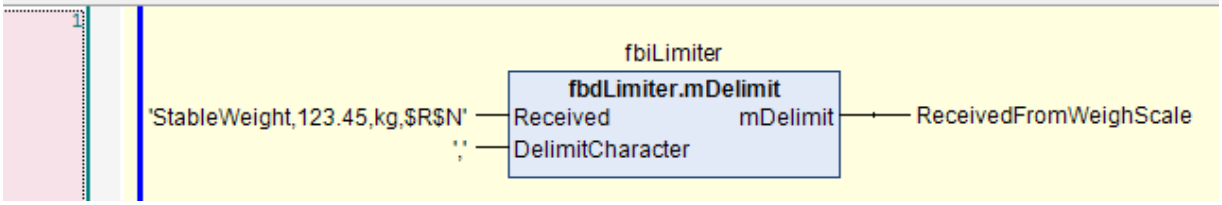


The **mDelimit** method does the opposite i.e. it simplifies further processing of the incoming string.

For example, a weighscale may send its weigh as:

“StableWeight,123.45,kg<CR><LF>”

ReceivedFromWeighScale	ARRAY [0..20] OF STRING(40)	
ReceivedFromWeighS...	STRING(40)	'StableWeight'
ReceivedFromWeighS...	STRING(40)	'123.45'
ReceivedFromWeighS...	STRING(40)	'kg'
ReceivedFromWeighS...	STRING(40)	'RN'
ReceivedFromWeighS...	STRING(40)	''
ReceivedFromWeighS...	STRING(40)	''
ReceivedFromWeighS...	STRING(40)	''



The component parts of the message are delimited and presented in an array.

The **mDelimit** and **mRelimit** methods are executed synchronously

7 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. Beijer Group has a sale over 1.6 billion SEK in 2019 and is listed on the NASDAQ OMX Nordic Stockholm Small Cap list under the ticker BELE. **www.beijergroup.com**

China
Shanghai

NORWAY
Drammen

TAIWAN
Taipei

DENMARK
Roskilde

SOUTH KOREA
Seoul

TURKEY
Istanbul

FRANCE
Paris

SWEDEN
Göteborg
Malmö
Stockholm

UNITED KINGDOM
Nottingham

GERMANY
Nürtingen

USA
Salt Lake City

7.1 Contact us

Global offices and distributors