# IEC61850 Client

## straton user guide Rev. 11

sales@straton-plc.com

CP_R31_006_H

straton

# Content

# 1. Overview

The aim of this document is to guide beginners in the configuration of an IEC61850 Client. Further information are available in the online help for advances users.

# 2. Requirement and setup

The required software is the IDE, for the configuration of the IEC61850 protocol, and some add-ons driver are required in order to support the IEC61850 Client protocol.

The application can be downloaded to any runtime that support the IEC61850 Client, please refer to the OEMs information for special device.

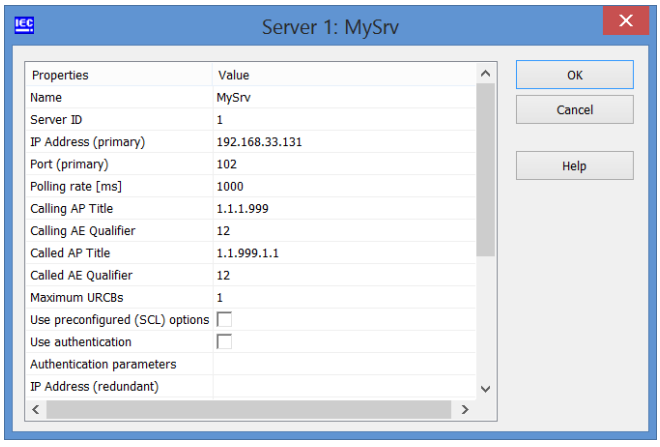
# 3. Create and configure an application

## 3.1. Create a new project

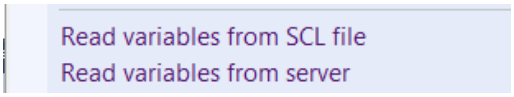
Open the IO Drivers window using the tool bar ( ) or right click on the project > Insert Shortcut > Fieldbus configurations

Insert a new Fieldbus using tool bar ( ) or menu Insert > Insert configuration and select the IEC61850 Client driver.

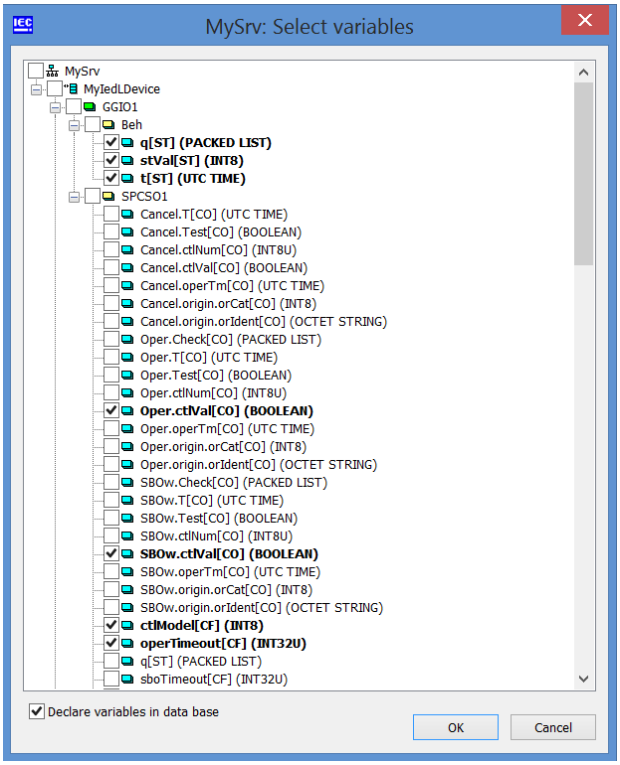
Insert a master using tool bar ( ) or menu Insert > Insert Master/Port.




Using the right mouse button select *Read variable From SCL file* or *Read variable from server*, in this last case the Server must be running.



Select the variables to declare.




## 3.2. Download the application

Download the application to the runtime:

- Select the communication parameters in menu Tools/Communication Parameters
- Establish the connection through menu Project/Online

**RESULT IS:**


The download is successful and application starts correctly.


The runtime is not started or communication parameters are wrong.


The application is not yet downloaded or an error occurs during startup. More detail can be found in the output view.

The application is not yet downloaded or an error occurs during startup. More detail can be found in the output view.

## 3.3. File transfer

The IEC61850 server from COPADATA needs to be configured in order to support file transfer. In the IEC61850 server Fieldbus configuration the *File transfer directory* needs to be set, if used. Only file under the folder's name will be available for file transfer.

| Name | Value |
|---|---|
| File transfer directory | User |
| Adapter identification | |
| Max. number of saved report... | 1000 |
| Max. number of simultaneous... | 0 |

In the IEC61850 Client Fieldbus configuration, the destination folder needs to be set too.

| Name | Value |
|---|---|
| Directory for file transfer | C:\Temp |
| Run communication from with... | ☐ |
| Queue all value changes | ☐ |
| Originator category (orCat) | Station-control |

Write an IEC61131-3 program to upload the desired file, for example:

```
IF bUploadFile THEN
    bErrorReport := IEC61850_FTGET(ServerId,'File.txt');
    bUploadFile := FALSE;
END_IF;
```

*ServerId* is a DINT variable, defined in the Master/Port in the IEC61850 Client Fieldbus configuration (cf: paragraph 3.1 in this document).

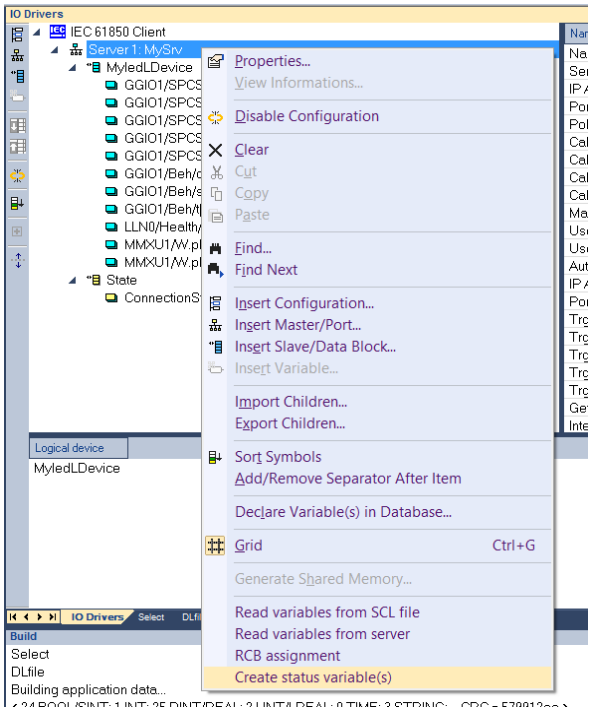
**NOTES:**

The file transfer directory's name must be set according to the used OS, for example if the application is downloaded on a Windows runtime it can be "C:\Temp", if it uses VXWORKS the file transfer directory can be "/ROM/user

## 3.4. Connection status bit

It is possible to look after a variable containing status bits about the state connection with the server. To create this variable, open "Fieldbus configurations", right click on the 61850 server to watch, and choose "Create status variable".

A new variable appears in global variables with a name like: "servername_ConnectionState".




The different values it can take are:

For the first connection:

| | |
|---|---|
| TCP_CONNECTING | 0x04 |
| TCP_CONNECTED | 0x02 |
| TCP_CONNECTFAILED | 0x08 |
| MMS_INITIALIZED | 0x10000 |
| MMS_RCBENABLEFAILED | 0x20000 |

For the second connection:

| | |
|---|---|
| TCP_CONNECTING | 0x40 |
| TCP_CONNECTED | 0x20 |
| TCP_CONNECTFAILED | 0x80 |
| MMS_INITIALIZED | 0x1000000 |
| MMS_RCBENABLEFAILED | 0x2000000 |

The standard evolution for this value is:

TCP_CONNECTFAILED => TCP_CONNECTING => TCP_CONNECTFAILED

Or:

TCP_CONNECTFAILED => TCP_CONNECTING => TCP_CONNECTED => TCP_CONNECTED | MMS_INITIALIZED => (or if error while enabling RCB: TCP_CONNECTED | MMS_INITIALIZED | MMS_RCBENABLEFAILED)

## 3.5. t-attribute

The Quality Time can be set by the Server (cf: *IEC61850 Server tutorial*)

It can be retrieved by the Client using the status bits #21, #22 and #49 of any variable. Commands in the Client looks like:

```
__BitOk21 := VSIGETBIT( MyClient_MyIedLDevice_XCBR1_Beh_stVal_ST, 21 );__
```

| Quality | Bit #21 | Bit #22 | Bit #49 |
|---|---|---|---|
| OK | TRUE | FALSE | FALSE |
| ClockNotSynchronized | TRUE | FALSE | TRUE |
| ClockFailure | FALSE | TRUE | TRUE |

**NOTES:**

The variable in the client must have a profile ( - profile for example)

Each bit's value is refreshed when the data changes.

## 3.6. Quality bits

The quality bits of IEC61850 common data attribute "q" are mapped according to the following table:

| Attribute name | Attribute type Value/Value range | bit |
|---|---|---|
| Validity | CODED ENUM good \| invalid \| reserved \| questionable<br>Bits 6 and 7 respectively 0 0 \| 1 0 \| 0 1 \| 1 1 | 6,7 |
| detailQual | PACKED LIST | |
| Overflow | BOOLEAN DEFAULT FALSE | 5 |
| Out Of Range | BOOLEAN DEFAULT FALSE | 4 |
| Bad Reference | BOOLEAN DEFAULT FALSE | 3 |
| Oscillatory | BOOLEAN DEFAULT FALSE | 2 |
| Failure | BOOLEAN DEFAULT FALSE | 1 |
| Old Data | BOOLEAN DEFAULT FALSE | 0 |
| Inconsistent | BOOLEAN DEFAULT FALSE | 15 |
| Inaccurate | BOOLEAN DEFAULT FALSE | 14 |
| Source | CODED ENUM process \| substituted | 13 |
| Test | BOOLEAN DEFAULT FALSE | 11 |
| Operator Blocked | BOOLEAN DEFAULT FALSE | 12 |

**NOTES:**

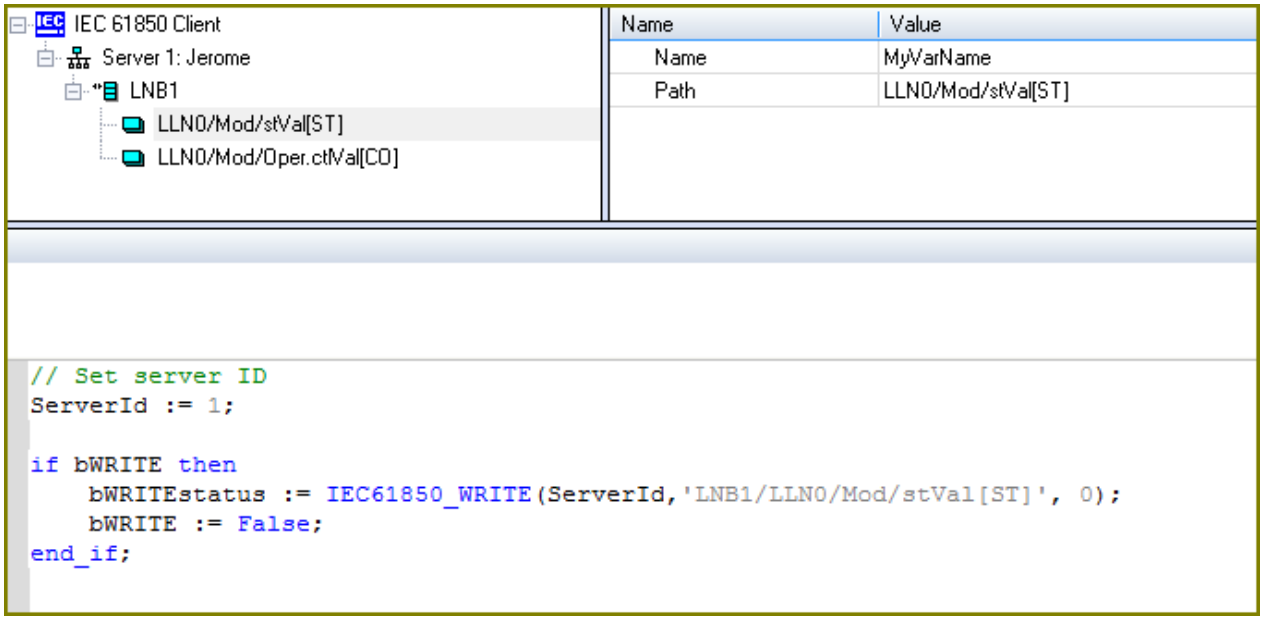
The "detailQual" bits are linked with the "Validity" bits. For example, if the "detailQual" is *overflow* then the "Validity" shall be set to *invalid*.

More details are available in the standard, see IEC61850-7-3 and IEC61850-8-1.

## 3.7. Write an IEC 61850 attribute

A function block is available to write a variable through IEC61850/MMS protocol. The function required the ServerID and the path name.

Path Name is composed of the Logical Device Name plus the Path you can find in the client driver.



## 3.8. Commands (Control model)

The driver supports the command 'Direct Operate' and 'Select Before Operate' (SBO) in normal and in enhanced security. In IEC61850_OPER the driver automatically checks the actual Control Model and executes the command in the corresponding way.

For this, first the attribute */ctlModel[CF] of the Data Object is read out. Depending on the value of this attribute, the corresponding command is executed using the respective Control model:

| Value | Action |
|---|---|
| 0 | No action, execute command not possible (status only) |
| 1 | An 'Operate' service (direct control with normal security) is executed |
| 2 | 'Select' and 'Operate' services (select before operate with normal security) are executed |
| 3 | An 'Operate' service (direct control with enhanced security) is executed |
| 4 | 'SelectWithValue' and 'Operate' services (select before operate with enhanced security) are executed |

**OPERATE COMMAND:**

The value is set directly to */Oper[CO].

Select commands are processed sequentially:

- Select command normal security: read from */SBO[CO], write to */Oper[CO].
- Select command enhanced security: write to */SBOw[CO], write to */Oper[CO].

The required command sequences are automatically performed in the right sequence by the driver. No further configuration steps are necessary.

Additionally, the driver supports manual Select and Cancel. The driver - according to the ctlModel of the Data Object - executes the Select (only when ctlModel=2 or 4). The values of ctlModel=0, 1 or 3 are treated as negative answer of Select.

When the IEC 61850 server responds the Select positive, the driver stores information about selected the Data Object and in the following command (in the following IEC61850_OPER) the driver only executes the Operate.

### 3.8.1. IEC61850_OPER (Oper or Select and Oper)

**PARAMETERS:**

```
VAR_INPUT
   ENABLE : BOOL;                (* Operate on a rising edge *)
   SERVERID : UDINT;     (* Id of the server set in the IEC61850 configuration *)
   PATH : STRING;        (* IedNameLogicalDeviceName/LnodeName/Pos/Oper.ctlVal[CO] *)
   CHECK : USINT;
   VALUE : ANY;
END_VAR

VAR_OUTPUT
   STATUS : UDINT;
   RESULT : DINT;
END_VAR
```

### 3.8.2. IEC 61850_OPER (Oper or Select and Oper)

**PARAMETERS:**

```
VAR_INPUT
   ENABLE : BOOL;
   SERVERID : UDINT;
   PATH : STRING;
   CHECK : USINT;
   VALUE : ANY;
END_VAR


VAR_OUTPUT
   STATUS : UDINT;
   RESULT : DINT;
END_VAR
```

### 3.8.3. IEC 61850 CANCEL (Cancel)

**PARAMETERS:**

```
VAR_INPUT
   ENABLE : BOOL;
   SERVERID : UDINT;
   PATH : STRING;
END_VAR

VAR_OUTPUT
   STATUS : UDINT;
   RESULT : DINT;
END_VAR
```

### 3.8.4. Parameters

ENABLE: the trigger (edge) of the command. Note: the command will be started only when the last command is already done (compare STATUS).

SERVERID: Server ID

PATH: Path (without server name) e.g. 'GEDeviceF650/XCBR1/Pos/Oper.ctlVal[CO]'

CHECK: IEC61850 Check attribute:

- 0: No check
- 1: Interlock check
- 2: Synchro check
- 3: Interlock and synchro check

VALUE: value of the command (any Data type)

### 3.8.5. Status and Result

STATUS: Status of last command:

- 0: BUSY
- 1: DONE
- 2: ERROR

RESULT:

when > 0 responds the IEC61850 AddCause

- < 0 internal error code:
- -1, VALUE wrong parameter
- -2, driver not loaded
- -3, write command cannot be given (e.g. not valid SERVERID)

## 3.9. Report Control Block (RCB)

The IEC61850 Client gets the data from the server by polling by default. The definition of RCB control block optimize the communication by using an event based communication.

### RCB IS AN ABSTRACT FOR:

#### URCB (unbuffered report control blocks)

The unbuffered are sent to the client and destroyed immediately by the server after sending without any acknowledgment.

The server sends a block immediately after value or quality change.

#### BRCB (buffered report control blocks)

The buffered are sent to the client till the reception of the acknowledgment from the client.

They are resent as long as the client accepts them.

They are stored (up to the specified limit with the actual server) in case the communication between server and client fails.

The client asks for the buffered reports since a given number only if he detects a connection failure or (some) lost reports, otherwise the BRCBs act like URCBs and the server sends a block immediately after value or quality change.

### 3.9.1. Unbuffered Report Control Block (URBC)

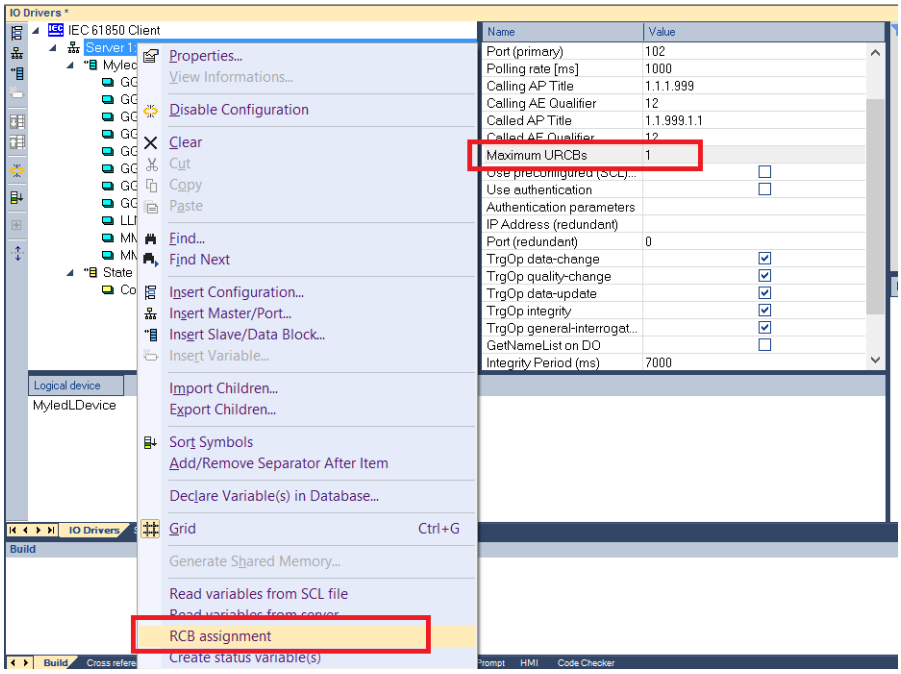
The IEC61850 Client gets the data from the server by polling by default. The definition of RCB control block optimize the communication by using an event based communication.

**RCB IS AN ABSTRACT FOR:**

- URCB (unbuffered report control blocks)
  - The unbuffered are sent to the client and destroyed immediately by the server after sending without any acknowledgment.
  - The server sends a block immediately after value or quality change.
- BRCB (buffered report control blocks)
  - The buffered are sent to the client till the reception of the acknowledgment from the client.
  - They are resent as long as the client accepts them.
  - They are stored (up to the specified limit with the actual server) in case the communication between server and client fails.
  - The client asks for the buffered reports since a given number only if he detects a connection failure or (some) lost reports, otherwise the BRCBs act like URCBs and the server sends a block immediately after value or quality change.

## 3.9.2. Buffered Report Control Block (BRCB)

For Buffered RCB and statically assigned RCB, this is done by right click on the server and RCB Assignment.

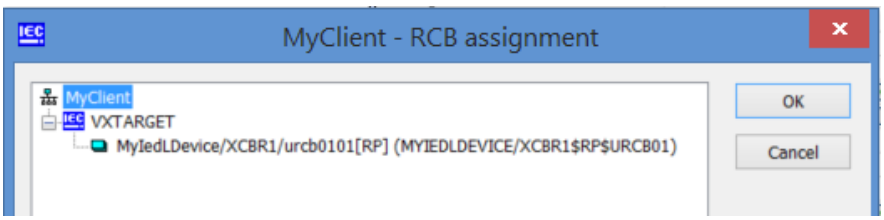



In that case you can import it from a server or enter the name of the RCB: exampleLDevice/XCBR1/urcb[RP]

Where:

- "example" is the IED Name
- "LDevice" is the name of the logical device
- "XCBRxx" the class of the logical node and xx the instance number
- "urcb" the name of the report control block.

To receive the RCB correctly, one must enter the name of the client's machine, in capital letters. For example on a Mitsubishi Q24 (VXWORKS) the configuration can be:



### 3.9.3. Trigger options

In a RCB, the data are pushed by the server on a trigger. The trigger supports several options:

TrgOps data-change: Data value change

TrgOps quality-change: Quality change

TrgOps data-update: Timestamp change

TrgOps Integrity: Minimum every bufTm (bufTm is define in the RCB)

TrgOps general-interrogation: On request by the client (GI)

These options are set by the server if the “Use preconfigured TrgOps” is set.

Otherwise the defined value in the client configuration are used.

# 4. Frequently Asked Questions

### CONNECTION BETWEEN CLIENT AND SERVER FAILS?

At least one variable needs to be linked in the client so that straton/zenon logic can establish the connection. For instance LLN0.Mod.stVal is always present in any server.

### FILE TRANSFER FAILED?

When IEC61850 client upload a file it keep the folder/sub-folder architecture locally. These folders need to be present in the IEC61850 working directory defined in the configuration tool.