

IEC-61850

IEC-61850 Client Communication Protocol

Version 2016.2

Reference Manual

00058.02
December 2018

IEC-61850
Client Communication Protocol

Version 2016.2

Reference Manual

00058.02
December 2018

Table of Contents

1. GENERAL INFORMATION	4
1.1 Summary	4
1.2 Migration	4
1.3 The IEC 61850 Standard	5
1.4 Communication Module	8
1.4.1 Communication Library	8
1.4.2 Integration Library	11
1.5 Operation	14
2. CHANNEL SETTINGS	14
2.1 Protocol Options	14
2.2 Interfaces	14
2.3 Initial State	14
3. NODES SETTINGS	14
3.1 Primary Station	15
3.2 Secondary Station (Backup Station)	17
4. POINTS SETTINGS	17
4.1 Point Types	18
4.1.1 Binary output (BO)	19
4.1.2 DataSet (DS) (DS1) (DS2) (DS3) (DS4)	19
4.1.3 Report (BR) (RP)	20
4.1.4 Report Flags (BRF) (RPF)	20
4.2 Point Address	20
4.3 Access Type	22
4.3.1 AccessType Read	23
4.3.2 AccessType Write	23
5. BROWSER	24
5.1 General	24
5.2 Data Sources	25
5.3 Remote IED Connection	25
5.4 Reading SCL files	25
5.5 Data Objects	25
5.6 DataSets	26
5.7 Reports	27
5.8 Report Attribute Window	27
6. COMTRADE FILES	28
6.1 General information	28
6.2 Reading and Recording	29

7. DIAGNOSIS OF COMMUNICATION PROBLEMS	29
7.1 Logs of Operations (LOG)	29
7.1.1 SCADA Trace Window	29
7.1.2 Windows Event	30
7.1.3 Text File	30
7.2 Codes of Errors and Qualities	30
7.2.1 Command Errors (CO)	30
7.2.2 Quality	32

1. GENERAL INFORMATION

1.1 Summary

Communication Driver Name: IEC 61850 Client Communication Protocol

Current Version: 2016.2 – 1.3

Implementation DLL: T.ProtocolDriver. IEC61850.dll

Protocol: IEC-61850 Client communication protocol - Edition 2

Interface: TCP/IP

Description: IEC861850 Scada Client implements the part of the standard IEC61850 responsible for ACSI core services, as defined in part 8.1 of IEC861850 standard. It uses encapsulation of the ISO/OSI transport layer into the TCP/IP transport layer. It communicates with IEDs (Intelligent Electronic Devices), RTUs (Remote Terminal Units) and IO devices that support this protocol, acting as the client workstation.

Protocol Options: Timers for protocol control messages

Max number of nodes: User defined

PC Hardware requirements: Standard PC Ethernet interface board

Note: This communication module was rewritten for version **an-2016.2**, and uses a library that is different from the one previously used. Changes were made to the configuration procedures of channels, nodes and points. The following section provides notes to aid in the migration of projects that were already using the module from the previous version.

1.2 Migration

Scada users of communication module - Issue 1 need to update the channel registration, nodes, and project points. Below are the steps for migrating existing projects to the new issue.

1. Copy all points referring to the nodes of Project Communication Module 61850 to an Excel worksheet.
2. Delete all points referring to the nodes of Project Communication Module 61850.
3. Copy the names of the project nodes and channels to the notebook.
4. Delete all nodes.

5. Delete all channels.
6. Create the channels with the names previously copied in step 3. Configuring each channel according to session 2 - CHANNEL CONFIG – Does not allow copy and paste.
7. Create the nodes with the names previously copied in step 3. Configuring each node according to session 3 - CONFIGURING US (NODE CONFIG) – Does not allow copy and paste.
8. Paste the point definitions previously copied in step 1.
9. When starting the Communication Module, all the Point entries will be validated. If point mistakes occur, they will be recorded in the Log (See section 7.1 - **Logs of operations (LOG)**) and check how to correct them in session 4 - **CONFIGURATION OF POINTS (POINTS CONFIG)**.

1.3 The IEC 61850 Standard

This section aims to briefly present some information on the standard IEC 61850, which directly interferes in the definition and implementation of this communication module.

For more details and a formal description, refer to the standard's texts or documentation of the server IEDs.

The standard is divided into ten parts, listed below.

Chapter Description

1. Introduction and Overview
2. Glossary
3. General Requirements
4. System and Project Management
5. Communication Requirements for Device Functions and Models
6. Configuration of the Description Language for Communication in Substations with IEDs
7. Basic Communication Structures for Substations and Feeders: Principles and Models
 - 7.2 Basic Communication Structures for Substations and Feeders:
 - 7.3 Basic Communication Structures for Substations and Feeders: Common Data Class
 - 7.4 Basic Communication Structures for Substations and Feeds: Classes of Logic Nodes and Compatible Data
8. Mapping for MMS (ISO/IEC 9506-1 and ISO/IEC 9506-2) and for ISO/IEC8802-3
 - 9.1 Sample Values on One-way Multidrop Point-to-point
 - 9.2 Sample Values for ISO/IEC8802-3
- 10 Conformity Tests

As regards the support for data communication between computers and IEDs, as established by the standard, the types of messages defined are listed below.

Message Description

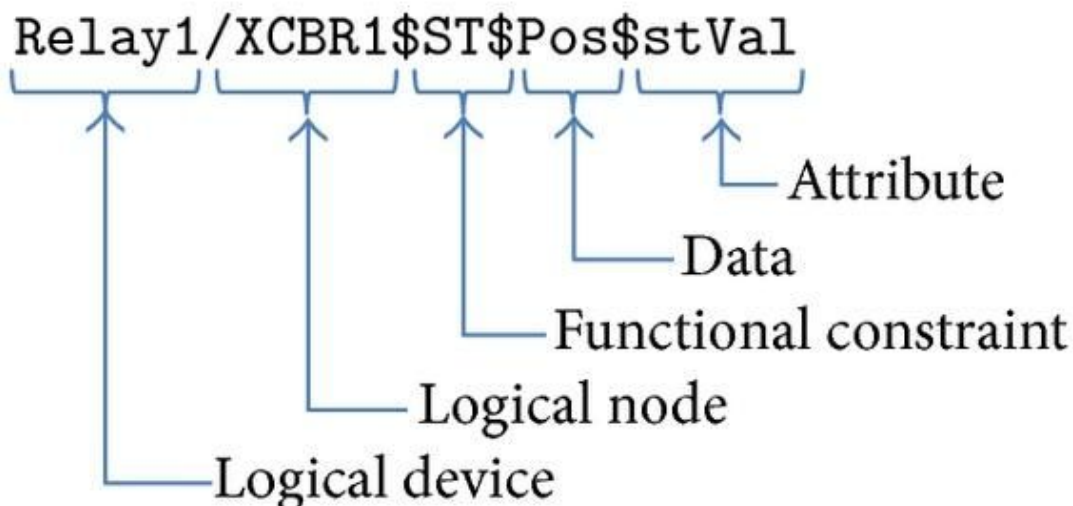
GOOSE Generic Object Oriented Substation - Multicast-type messages that load information between IEDs. They are responsible only for the traffic of messages about the performance of any protection or digital signal.

MMS Manufacturing Message Specification - Unicast-type messages that are used to exchange analog or digital (state) measurement data to indicate the state of the equipment and process.

One of the great advantages introduced by the standard in defining its data communication protocol is the fact that the internal points to an IED do not use **numbers for addresses**, as in other protocols, but **names** instead. Each point is considered an **object**, and its names and formats are standardized by the norm.

Internally to the IEC61850 IEDs there is a map of the **objects**, arranged in a hierarchical structure. In the communication between a client (for example, SCADA) and the server (for example, a digital-IED relay), these object names (at least when communication is initially established) travel in the protocol messages.

In order to have a general idea of these object's names, the next section briefly presents the structure of the address map hierarchy of the **IEC61850**.



LD - Logical Devices - These are logical devices within this physical server, which work as function containers (Logical Nodes) or even as gateways between different LD LNs. Generally, IED internal names are used in the initial part of the name (prefixes) and suffixes that indicate the main function of the LD. Examples of suffixes are CTRL, MEAS, etc.

LN - Logical Nodes - Objects that implement the well-defined basic functions within the Logical Device. The names of Logical Nodes are standardized. Each LN contains objects, data sets, Report definitions, Logs, parameter groups, and services that implement objects and functions. For example, a disconnect switch, its state, its control, has the name "Q0CSWI1." The LN in this case is the CSWI - "Switch controller." Other examples of LNs would be those that perform measurement functions (MMXU), voltage transformer (TVTR), overcurrent protection (PTOC), thermal protection (PTTR). A complete list of LNs for use in substation IEDs and the rules for forming their names are given in IEC61850-Part 5.

FC - Functional Constraints - The function defined by the LN will contain several objects that are distributed in categories defined as FC-Functional Constraints. These are "CO" that

will contain command objects, "ST" that will contain digital signaling objects, "RP" for report-type objects, and "MX" for measurement objects. The list of the following FCs was obtained from IEC61850 Part 7-2-2003 - Table 18. More details about each item can be found in the original table.

ST Status information

MX Measurements (analog values)

CO Control DataAttribute will be operated (control model) and read

SP Setpoint DataAttribute will represent a set-point; value may be controlled and read

SV Substitution DataAttribute will represent a substitution

CF Configuration DataAttribute will represent a configuration

DC Description DataAttribute will represent a description

SG Setting group logical devices that implement the SGCB class

SE Setting group

EX Extended definition

BR Buffered report

RP Unbuffered report

LG Logging Attribute will represent a log control

GO Goose control

GS Gsse control Attribute will represent a goose control

MS Multicast sampled value control

US Unicast sampled value control

DO - Data objects - Data Objects are objects with standard names that contain the information proper. These objects can be simple, such as the "Pos" object, which provides information on the circuit breaker position. There are more complex "A" objects, which represent a three-phase current. It is made up of four Data Attributes, each of which containing Child Data Attributes with the current information about each phase and a neutral. It has in the latter case multiple levels of Data Attributes in the hierarchy below a Data Object.

Examples of DataObjects are:

Beh - Behavior of LN

Health - State of the logical node related to HW and SW.

Loc - Switchover between local and remote operation

Mod - Mode and behavior: On, Blocked, Test, Test/Blocked, Off

Pos - Switch position

A - Current

neut - Phase neutral

phsA, phsB, PhsC - Phases A, B and C

Hz - Frequency

PPV - Phase to phase voltages

PhV - Phase to ground voltages for Phases 1, 2, and 3, including Angle

DA - Data Attributes - Each object hierarchically below a Data Object or another Data Attribute, in the case of "ctlNum" or a phase of the stream as "phsC." Examples:

In the case of "Pos" you can see four Data Attributes:

stVal - Containing the state of the circuit breaker

q - The quality of the data

t - Sampling timestamp

ctlNum - Data object status change sequence number

In the case of the current "pshC," besides being a Data Attribute itself, it also contains four Data Attributes below it:

cVal - The current measurement

instCval - The instantaneous measure of the current at the last acquisition

q - The quality of the data

t - Sampling timestamp.

For the end user, the way services are used is transparent. They are only required to know how to set up the user-friendly configuration interfaces for running the IEC61850 SCADA Client.

1.4 Communication Module

The communication module consists of two libraries that work together: Communication and Integration.

1.4.1 Communication Library

The communication library is developed in the C ++ programming language and is responsible for the communication between the Communication Module and the equipment. It is designed according to edition 2 of the IEC 61850 standard series.

1.4.1.1 Supported objects

The following table provides an overview of supported IEC 61850 objects.

IEC 61850 Objects	MMS Object
SERVER class	Virtual Manufacturing Device (VMD)
LOGICAL DEVICE class	Domain
LOGICAL NODE class	Named Variable
DATA class	Named Variable
DATA-SET class	Named Variable List
SETTING-GROUP-CONTROL-BLOCK class	Named Variable
REPORT-CONTROL-BLOCK class	Named Variable
LOG class	Journal
LOG-CONTROL-BLOCK class	Named Variable

GOOSE-CONTROL-BLOCK class	Named Variable
GSSE-CONTROL-BLOCK class	Named Variable
CONTROL class	Named Variable
Files	Files

1.4.1.2 Supported services

The following table provides an overview of supported IEC 61850 Abstract Communication Service Interface (ACSI) services.

IEC 61850 Model	IEC 61850 Services		AA: TP/M C	Client	Server
Server	S1	ServerDirectory	TP	✓	✓
Application association	S2	Associate	TP	✓	✓
	S3	Abort	TP	✓	✓
	S4	Release	TP	✓	✓
Logical device	S5	LogicalDeviceDirectory	TP	✓	✓
Logical node	S6	LogicalNodeDirectory	TP	✓	✓
	S7	GetDataValues	TP	✓	✓
Data	S8	GetDataValues	TP	✓	✓
	S9	SetDataValues	TP	✓	✓
	S10	GetDataDirectory	TP	✓	✓
	S11	GetDataDefinition	TP	✓	✓
Data set	S12	GetDataSetValues	TP	✓	✓
	S13	DataSetValues	TP	✓	✓
	S14	CreateDataSet	TP	✓	✓
	S15	DeleteDataSet	TP	✓	✓
	S16	GetDataSetDirectory	TP	✓	✓
Substitution	S17	SetDataValues	TP	✓	✓
Setting group control block	S18	SelectActiveSG	TP	✓	✓
	S19	SelectEditSG	TP	✓	✓
	S20	SetSGValues	TP	✓	✓
	S21	ConfirmEditSGValues	TP	✓	✓
	S22	GetSGValues	TP	✓	✓
	S23	GetSGCBValues	TP	✓	✓
Buffered report control block (BRCB)	S24	Report	TP	✓	✓
	S24-1	data-change (dchg)	TP	✓	✓
	S24-2	qchg-change (qchg)	TP	✓	✓
	S24-3	data-update (dupd)	TP	✓	✓
	S25	GetBRCBValues	TP	✓	✓
	S26	SetBRCBValues	TP	✓	✓
Unbuffered report control block (URCB)	S27	Report	TP	✓	✓
	S27-1	data-change (dchg)	TP	✓	✓

	S27-2	qchg-change (qchg)	TP	✓	✓
	S27-3	data-update (dupd)	TP	✓	✓
	S28	GetURCBValues	TP	✓	✓
	S29	SetURCBValues	TP	✓	✓
Log control block	S30	GetLCBValues	TP	✓	✓
	S31	SetLCBValues	TP	✓	✓
	S32	QueryLogByTime	TP	✓	✓
	S33	QueryLogAfter	TP	✓	✓
	S34	GetLogStatusValues	TP	✓	✓
GOOSE control block	S35	SendGOOSEMessage	MC	✓	✓
	S36	GetGoReference	TP		
	S37	GetGOOSEElementNumber	TP		
	S38	GetGoCBValues	TP	✓	✓
	S39	SetGoCBValues	TP	✓	✓
GSSE control block	S40	SendGSSEMessage	MC		
	S41	GetGsReference	TP		
	S42	GetGSSEElementNumber	TP		
	S43	GetGsCBValues	TP		
	S44	SetGsCBValues	TP		
Multicast SVC	S45	SendMSVMessage	MC		✓
	S46	GetMSVCBValues	TP	✓	✓
	S47	SetMSVCBValues	TP	✓	✓
Unicast SVC	S48	SendUSVMessage	TP		✓
	S49	GetUSVCBValues	TP	✓	✓
	S50	SetUSVCBValues	TP	✓	✓
Control	S51	Select	TP	✓	✓
	S52	SelectWithValue	TP	✓	✓
	S53	Cancel	TP	✓	✓
	S54	Operate	TP	✓	✓
	S55	Command-Termination	TP	✓	✓
	S56	TimeActivated-Operate	TP	✓	✓
File transfer	S57	GetFile	TP	✓	✓
	S58	SetFile	TP	✓	✓
	S59	DeleteFile	TP	✓	✓
	S60	GetFileAttributeValues	TP	✓	✓
Time	T1	Time resolution of internal clock		1ms	1ms
	T2	Time accuracy of internal clock			
	T3	Supported Timestamp resolution		1ms	1ms
AA - Application association					

TP – Two-party application association (MMS over TCP/IP or Unicast Sampled Value on top of Ethernet)
MC – Multicast application association (GOOSE or SV using multicast messages on top of Ethernet)
✓ – Supported

1.4.2 Integration Library

The integration library is developed in the C # programming language and is responsible for the integration between the Communication Module and SCADA. It aims to transact the read and write information between the device and SCADA.

1.4.2.1 Objects Implemented

The following table provides an overview of implemented IEC 61850 objects. The fields marked with a (✓) indicate the implementation of the service.

IEC 61850 Objects	
SERVER class	
LOGICAL DEVICE class	✓
LOGICAL NODE class	✓
DATA class	✓
DATA-SET class	✓
SETTING-GROUP-CONTROL-BLOCK class	
REPORT-CONTROL-BLOCK class	✓
LOG class	
LOG-CONTROL-BLOCK class	
GOOSE-CONTROL-BLOCK class	
GSSE-CONTROL-BLOCK class	
CONTROL class	✓
Files	

1.4.2.2 Services Implemented

The following table provides an overview of the implemented IEC 61850 Abstract Communication Service Interface (ACSI) services. The fields marked with a (✓) indicate the implementation of the service.

IEC 61850 Model	IEC 61850 Services		
Server	S1	ServerDirectory	✓
Application association	S2	Associate	✓
	S3	Abort	✓
	S4	Release	✓
Logical device	S5	LogicalDeviceDirectory	✓
Logical node	S6	LogicalNodeDirectory	✓
	S7	GetAllDataValues	✓
Data	S8	GetDataValues	✓
	S9	SetDataValues	✓

	S10	GetDataDirectory	✓
	S11	GetDataDefinition	✓
Data set	S12	GetDataSetValues	✓
	S13	DataSetValues	
	S14	CreateDataSet	
	S15	DeleteDataSet	
	S16	GetDataSetDirectory	✓
Substitution	S17	SetDataValues	
Setting group control block	S18	SelectActiveSG	
	S19	SelectEditSG	
	S20	SetSGValues	
	S21	ConfirmEditSGValues	
	S22	GetSGValues	
	S23	GetSGCBValues	
Buffered report control block (BRCB)	S24	Report	✓
	S24-1	data-change (dchg)	✓
	S24-2	qchg-change (qchg)	✓
	S24-3	data-update (dupd)	✓
	S25	GetBRCBValues	✓
	S26	SetBRCBValues	✓
Unbuffered report control block (URCB)	S27	Report	✓
	S27-1	data-change (dchg)	✓
	S27-2	qchg-change (qchg)	✓
	S27-3	data-update (dupd)	✓
	S28	GetURCBValues	✓
	S29	SetURCBValues	✓
Log control block	S30	GetLCBValues	
	S31	SetLCBValues	
	S32	QueryLogByTime	
	S33	QueryLogAfter	
	S34	GetLogStatusValues	
GOOSE control block	S35	SendGOOSEMessage	
	S36	GetGoReference	
	S37	GetGOOSEElementNumber	
	S38	GetGoCBValues	
	S39	SetGoCBValues	
GSSE control block	S40	SendGSSEMessage	
	S41	GetGsReference	
	S42	GetGSSEElementNumber	
	S43	GetGsCBValues	
	S44	SetGsCBValues	

Multicast SVC	S45	SendMSVMessage	
	S46	GetMSVCBValues	
	S47	SetMSVCBValues	
Unicast SVC	S48	SendUSVMessage	
	S49	GetUSVCBValues	
	S50	SetUSVCBValues	
Control	S51	Select	✓
	S52	SelectWithValue	✓
	S53	Cancel	
	S54	Operate	✓
	S55	Command-Termination	
	S56	TimeActivated-Operate	
File transfer	S57	GetFile	
	S58	SetFile	
	S59	DeleteFile	
	S60	GetFileAttributeValues	
Time	T1	Time resolution of internal clock	
	T2	Time accuracy of internal clock	

1.4.2.3 Functional Constraints Implemented

The following table provides an overview of the Functional Constraint property found in each attribute of a logical node. Fields marked with a (✓) indicate the implementation of the reading or writing.

Functional Constraint		Read	Write
BL	Blocking		
BR	Buffered report	✓	✓
CF	Configuration	✓	✓
CO	Control		✓
DC	Description	✓	✓
EX	Extended definition	✓	✓
LG	Log control blocks		
MS	Multicast SV		
MX	Measurements (analog values)	✓	
OR	Operate received		
RP	Unbuffered report	✓	✓
SE	Setting group editable		
SG	Setting group		
SP	Setpoint	✓	✓
SR	Service response / Service tracking		

ST	Status information	✓	
SV	Substitution		
U S	Unicast SV		

1.5 Operation

The communication module follows the sequence of steps below.

1. Channel initialization

Reading is performed. Read the settings in the IEC61850.xml file. Validation of information is performed.

2. Initialization of the node(s)

Communication is established and the data structure of the equipment read. This reading is essential for the operation of the Communication Module. This step is recursive, that is, only when it is successfully completed will the Communication Module proceed to the next step.

3. Initialization of the point(s)

The reading of the registered points is performed. The validation of the points registry is performed. It also evaluates if the point exists and if the Communication Module has the implementation to read or write on the point.

4. Starts read and write tasks

The task of reading COMTRADE and dot files is started according to the node settings. The writing task is started and triggered when a point of the writing type changes value. The task of reading received reports is started and the report points are signed.

5. Cycles through tasks

Read and write tasks are performed as configured. If connection is lost, return to step 2.

2. CHANNEL SETTINGS

2.1 Protocol Options

Name	Protocol	ProtocolOptions	Interface
IEC61850	IEC61850	ProtocolOptions	CustomTCPIP
IEC8705		Logging level	TCPIP
IEC8705		Log files	TCPIP
SinapWis		Log file size	TCPIP
SinapWis		Command password	TCPIP
		Command timeout (s)	

Figure 1 – Protocol Options

Logging Level – You can choose from this list the logging mode created by the communication module.

Logging level	Debug	All messages are registered in the LOG.
	Info	Only Info, Warning and Error messages are registered in the LOG.
	Warning	Only Warning and Error messages are registered in the LOG.
	Error	Only Error messages are registered in the LOG.

Log files: Maximum number of log files kept before the oldest one is deleted.

Log file size: Maximum size of each log file. Specify the maximum size with KB, MB, or GB suffixes.

Command password: To increase the security of sending commands, normally initiated only by a change in the state of a tag, it is possible here to specify a password of up to nine digits. The communication module will verify, at the time it receives the command, this password against the current value of the *EstimatedValue* attribute tag involved in the command. Therefore, in the operation of sending a command through a window, script, etc. this number must be loaded in this attribute. The communication module, after executing the command, changes the value of *EstimatedValue* to ZERO. This verification does not occur if leaving this option as zero.

Command timeout (s) - Time in seconds to determine that a command has reached the maximum time to execute.

2.2 Interfaces

Interface:

- **Custom TCP/IP:** This option is unique for this protocol.

2.3 Initial State

Enabled

3. NODES SETTINGS

Each node corresponds to a remote IED. Below is a figure showing a list of settings.

Dataset T3 read polling rate (ms) - Time period in milliseconds for read requests from DataSet (DS3) points. A zero value (0) disables this reading.

Dataset T4 read polling rate (ms) - Time period in milliseconds for read requests from DataSet (DS4) points. A zero value (0) disables this reading.

Report enable polling rate (ms) - Time in milliseconds to check if report points are enabled. The check is done by reading the value of the **RptEna** attribute of each report point. Zero value (0) disables this cyclic check, and the system will only try to enable the subscribed Reports once during startup.

Communication status tag - Enter the Name of a previously created integer TAG. Module will control the value: 0 = Offline communication; 1 = Online Communication.

Last command error tag - Name of Integer-type TAG to receive the last command error code. The errors come with y codes and they can be viewed in section 7.2.1.

SCL file name to config points - SCL file name, which will be used for point reading by the browser in point configuration. The file search directory is: C:\Action.NET\Projects\

Quality changes timestamp - Enables the date/time update when only the point quality changes. **Disabled** - Disables the date/time update when only the point quality changes.

Ignore polling on points updated by reports - Enabling report points will ignore the polling reading request. There is some performance gain in this case, but loss of the integrity of the data read as well. By disabling this option, the Points of Enabled Report will make the polling request also. There is loss of performance in this case, but a gain in the integrity of the data read.

COMTRADE files reading mode - Two modes:

- **Download:** In this mode of operation, the driver will transfer the COMTRADE files to the computer.
- **Download/Delete:** In this mode of operation, the driver will transfer the COMTRADE files to the computer and delete them from the IED after the transfer completes successfully.

COMTRADE files download folder - Directory where the transferred COMTRADE files will be saved to.

COMTRADE files download polling rate (ms) - Time period in milliseconds for read requests from COMTRADE files. Zero value (0) disables this reading.

FTP username - User with permission to read COMTRADE files by FTP protocol. If this field is complete, the reading of the files will be performed by FTP, otherwise it will be performed by MMS.

FTP password - User password with permission to read COMTRADE files by FTP protocol.

3.2 Secondary Station (Backup Station)

The same settings adjusted for the primary station can be adjusted to a secondary station, if it exists in the installation.

4. POINTS SETTINGS

The Points table in SCADA must contain the match between SCADA tag names and IEC61850 data object names in remote IEDs, which are the source of the information or targets of commands.

The address format of points in SCADA is separated by a colon and must be composed of the Type and Address fields. However, some point types have additional fields after the address.

4.1 Point Types

The types of points were created for registration purposes in order to organize, group and categorize the addresses under the IEC 61850.

Points with write access are updated when the tag changes the value. Points with read access are read according to the read request period defined in the node configuration. Some more complex types are explained in subsections (4.1.x).

In the SCADA points table, the following types of points can be used:

Type Name		Tag Type	Access	Description
AI	Analog input	Double	Read	Analog input
BI	Binary input	Integer	Read	Digital input
AO	Analog output	Double	Write	Writing "setpoints" for remote IEDs. Write the current value of point in float format (32 bits). Uses IEC61850 APC Common Data Class Model.
BO	Binary output	Digital	Write	Outputs - Send to remote corresponding point the output parameter set to this point (16-bit integer). Uses IEC61850 SPC and DPC Common Data Class Models.
RP	Unbuffered report	Digital	Read	Points of this type serve to indicate to SCADA the name of an "Unbuffered Report" that should be activated. Value of this tag: (0) Report unsubscribed (1) Report subscribed
BR	Buffered report	Digital	Read	Points of this type serve to indicate to SCADA the name of a "Buffered Report" that should be activated. Value of this tag: (0) Report unsubscribed (1) Report subscribed
RPF	Unbuffered report flag	Digital Integer Text	Read	This type is used to represent attributes of an "Unbuffered Reports" control block.
BRF	Buffered report flag	Digital Integer Text	Read	This type is used to represent attributes of a "Buffered Reports" control block.
DS	DataSet	Digital	Read	Points of this type serve to indicate to the communication module the name of a DataSet whose reading will be periodically required.
DS1	DataSet T1	Digital	Read	Reading points of this data set with this sample time
DS2	DataSet T2	Digital	Read	Reading points of this data set with this sample time
DS3	DataSet T3	Digital	Read	Reading points of this data set with this sample time

DS4	DataSet T4	Digital	Read	Reading points of this data set with this sample time
INC	Integer output control	Integer	Write	Writing of 32-bit integer set-point, following the IEC61850 Data Class Model INC – Controllable Integer Status
BSC	Binary step position control	Integer	Write	Writing of Raise and Lower control, following the model BSC – Binary Step Controlled position. Possible states: (0) Stop (1) Lower (2) Higher (3) Reserved
ISC	Integer control step position	Integer	Write	Writing of a position with values between -64 and 63. Uses IEC61850 Common Data Class Model Integer controlled step position information (ISC)
CS	Command signaling	Digital	Read	Command Signaling

4.1.1 Binary output (BO)

The Binary output (BO) type has the additional **Command Parameter** field. This field is used to set a fixed value to be sent by command, when the digital tag associated with the value changes points. The following is an example of usage.

Step		Parameter	
1	Create tag	Name	BREAKER_OFF
		Type	Digital
2	Create tag	Name	BREAKER_ON
		Type	Digital
3	Create point	Tag	BREAKER_OFF
		Address	Binary output (BO) IED7SJ63CTRL/QOCSWI1\$CO\$Pos : 0
4	Create point	Tag	BREAKER_ON
		Address	Binary output (BO) IED7SJ63CTRL/QOCSWI1\$CO\$Pos : 11

Associating the Tag (BREAKER_OFF) with the button “OPEN” and the Tag (BREAKER_ON) “CLOSE” with the action ToggleValue. By switching the tag value from 0 to 1 or from 1 to 0, a command will be generated by passing the fixed value set in the address.

4.1.2 DataSet (DS) (DS1) (DS2) (DS3) (DS4)

A DataSet is an object that contains a list of Data Objects’ names. DataSets are usually preconfigured on server IEDs. The contents (state, measure, timestamp, quality) of the Data Object in a DataSet can be obtained in two ways:

- By requesting a reading of the DataSet, by reference to its name. You must have a DataSet type point configured in the points table, in addition to having the points relative to each Data Object of interest;
- By activating a Report that points to this DataSet, causing the unsolicited sending of this content (described in the “Reports” section below).

DataSet points are set to read by polling and are set to DS type. Types DS1, DS2, DS3, and DS4 exist if you want to set different read times for each group of DataSet points.

TagName	Node	Address	DataType	Modifiers	AccessType
ANA61[1]	NO61850	DS:IED7SJ63CTRL/LLN0\$DATASET1:0	Native		Read

4.1.3 Report (BR) (RP)

For each Report configured on the server you want to enable SCADA to receive, you must include an entry in the points table. This real-time point will contain the current status of Report (0 - Disabled, 1 - Enabled). The information about the enabled state is extracted from the current state, reading the RptEna attribute of the Report. The time interval of this reading is 1 second.

The OptFlds and TrgOps attributes have the following default values:

- **OptFlds** = SEQ_NUM, TIME_STAMP, REASON_FOR_INCLUSION, DATA_SET, DATA_REFERENCE, ENTRY_ID, and CONF_REV
- **TrgOps** = DATA_CHANGED and GI

These fields can be changed to receive other values for each Report. This is possible by using a browser window function as described in section 5.7.

4.1.4 Report Flags (BRF) (RPF)

Some Report attribute points must be registered for monitoring and control.

EntryID – Text. Identifier of the received Report. It is required for proper control of the Buffered Report. If there is a disconnection, the Communication Module re-signs the Report from the last EntryID entered in the Tag. As it is a value which must be stored, it is imperative that the Tag is registered to use the Retentive functionality of SCADA.

SqNum – Integer. Sequential number of Reports received.

TagName	Node	Address	DataType	Mo...	AccessType
COMM1.ENTRYID	LINHA	BRF:IED7SJ63CTRL/LLN0\$BR\$brcbA01\$EntryID	Native		Read
COMM1.BRSEQ	LINHA	BRF:IED7SJ63CTRL/LLN0\$BR\$brcbA01\$SqNum:0	Native		Read
COMM1.BRSTAT	LINHA	BR:IED7SJ63CTRL/LLN0\$BR\$brcbA01	Native		Read

4.2 Point Address

Addresses generally follow the IEC 61850 format:

LogicalDevice / LogicalNode \$ FunctionalConstraint \$ DataObject \$ DataAttribute

For each desired Data Object, you must create a point in the points table. However, there are cases of more complex Data Objects, such as Current, Three-phase Voltage, and so on. These have more than one hierarchical level of Data Attributes below them.

The Communication Module knows some structures and automatically obtains each of the phases, but it is necessary to specify each of them in the points table (in this case, each of the Data Attributes in the first hierarchical level below the current Data Object must be a point in the points table).

Points that the Communication Module does not know how to identify which value to fetch in a given structure will display startup errors. This deficiency can be corrected by improving the knowledge of the Communication Module in relation to the structures or the address can be registered specifying the attribute that will be the value. The quality and timing of these attributes will be read from the structure to which it belongs.

Example of an address registered to the structure:

IED7SJ63MEAS / MMXU1 \$ MX \$ PhV \$ phsA

- Ignore => IED7SJ63MEAS / MMXU1 \$ MX \$ PhV \$ phsA \$ instCVal \$ mag \$ f
- Value => IED7SJ63MEAS / MMXU1 \$ MX \$ PhV \$ phsA \$ cVal \$ mag \$ f
- Quality => IED7SJ63MEAS / MMXU1 \$ MX \$ PhV \$ phsA \$ q
- Time => IED7SJ63MEAS / MMXU1 \$ MX \$ PhV \$ phsA \$ t

Example of address registered to the attribute:

IED7SJ63MEAS / MMXU1 \$ MX \$ PhV \$ phsA \$ cVal \$ mag \$ f

- Value => IED7SJ63MEAS / MMXU1 \$ MX \$ PhV \$ phsA \$ cVal \$ mag \$ f
- Quality => IED7SJ63MEAS / MMXU1 \$ MX \$ PhV \$ phsA \$ q
- Time => IED7SJ63MEAS / MMXU1 \$ MX \$ PhV \$ phsA \$ t

Status Specification

Status objects follow the general rule mentioned above – one point for each Data Object desired. The reading will retrieve all the related Data Attributes (example: stVal, q (quality) and timestamp t together. The SCADA will use stVal to update the state of the point. Attributes q (quality) and t (timestamp) will be used for the status and the time stamp of the point.

Circuit Breakers and Switchgears

In the case of signaling circuit breakers and switchgears, a digital point is defined in SCADA for the desired Data Object, which can be single or multiple digital (depending on the type of signaling for the object defined in the IED server). Attribute Data stVal will be the source of information for the state signaling. Attributes q (quality) and t (timestamp) will be used for point status and the time stamp.

For remote control, two points are defined in the SCADA points table, one for ON and one for OFF, with output parameters 1 and 0 respectively. The same Data Object is referenced on both, as shown below.

TagName	Node	Address	DataType	Modifiers	AccessType
* AN61_TH[4]	NO61850_TH	AI:IEDRelay1\$T1MMXU0\$MX\$TotVAr:0	Native		Read
AN61_TH[3]	NO61850_TH	AI:IEDRelay1\$T1MMXU0\$MX\$A\$neut:0	Native		Read
AN61_TH[2]	NO61850_TH	AI:IEDRelay1\$T1MMXU0\$MX\$TotVA:0	Native		Read
AN61_TH[1]	NO61850_TH	AI:IEDRelay1\$T1MMXU0\$MX\$AR\$neut:0	Native		Read
AN61_TH[0]	NO61850_TH	AI:IEDRelay1\$T1MMXU0\$MX\$A\$phsB:0	Native		Read
CMDDIG61DB_L	NO61850	AI:IED7SJ63CTRL\$Q0CSWI1\$CO\$POS\$SBOW:1	Native		ReadWrite
CMDDIG61DB_D	NO61850	BO:IED7SJ63CTRL\$Q0CSWI1\$CO\$POS\$SBOW:0	Native		ReadWrite
CMDDIG61_D	NO61850	BO:IED7SJ63CTRL\$Q0GAPC2\$CO\$SPCSO\$SBOW:0	Native		ReadWrite
CMDDIG61_L	NO61850	BO:IED7SJ63CTRL\$Q0GAPC2\$CO\$SPCSO\$SBOW:1	Native		ReadWrite
ANA61[7]	NO61850	BR:IED7SJ63CTRL\$LLN0\$BR\$BRCBA01	Native		Read
dig61[6]	NO61850	BI:IED7SJ63CTRL\$Q0GAPC2\$ST\$SPCSO	Native		Read

In the case of direct commands, the OPER object must be chosen.

In the case of multiple stage commands such as Select Before Operate (SBO) or Select Before Operate With Value (SBOw), it's only necessary to place the corresponding SBO or SBOw point in the points table. No additional points are required to perform the command operation.

Set Points

For points of the set-point kind (AO, INC, BCS and ISC), the implementation is as follows: for the table points, **AccessType Write** must be used. The **Address** column must be edited to match the correct type (AO, INC, BCS and ISC). On run time SCADA mode, a value change into one of these points will trigger a command sending the newly entered value. The remaining fields in the control object, operT, origin, ctlNum, etc., are sent with default states and

TagName	Node	Address	DataType	Access...	Scaling
* ATCC.STEP_CH	XTORM	BSC:Xtorm_ServerLogicalDevice/ATCC1\$COSTapChg\$Oper:0	Native	Write	None
ATCC.STEP_POS	XTORM	BI:Xtorm_ServerLogicalDevice/ATCC1\$ST\$TapChg\$valWTr\$posVal	Native	Read	None
ATCC.TAP_POS	XTORM	BI:Xtorm_ServerLogicalDevice/ATCC1\$ST\$TapPos\$valWTr\$posVal	Native	Read	None
ATCC.TAP_SP	XTORM	ISC:Xtorm_ServerLogicalDevice/ATCC1\$COSTapPos\$Oper:0	Native	Write	None
CTLV	XTORM	AI:Xtorm_ServerLogicalDevice/ATCC1\$MX\$CtlV	Native	Read	None
DJ_D	XTORM	BO:Xtorm_ServerLogicalDevice/GGIO1\$CO\$SPCSO\$Oper:0	Native	Write	None
DJ_L	XTORM	BO:Xtorm_ServerLogicalDevice/GGIO1\$CO\$SPCSO\$Oper:1	Native	Write	None
▶ FREQ	XTORM	AI:Xtorm_ServerLogicalDevice/MMXU1\$MX\$Hz	Native	Read	None
T_INC	XTORM	INC:Xtorm_ServerLogicalDevice/GGIO1\$CO\$ISCSO\$Oper:0	Native	Write	None

values.

4.3 Access Type

Two Access Types are used by this communication module: Read and Write.

Below is the information about settings of these access types.

4.3.1 AccessType Read

Read	
Name	Read
ReadOnStartup	True
ReadEnable	Always
ReadPollingRate	00:00:01.00 (one second)
ReadTrigger	
ReadStatus	
ReadCompleted	
WriteEventEnabled	False
WriteEvent	Changed
WriteTrigger	
WriteStatus	
WriteCompleted	
AcceptUnsolicited	False
UseStaticBlocks	False
BlockCommand	
Description	

4.3.2 AccessType Write

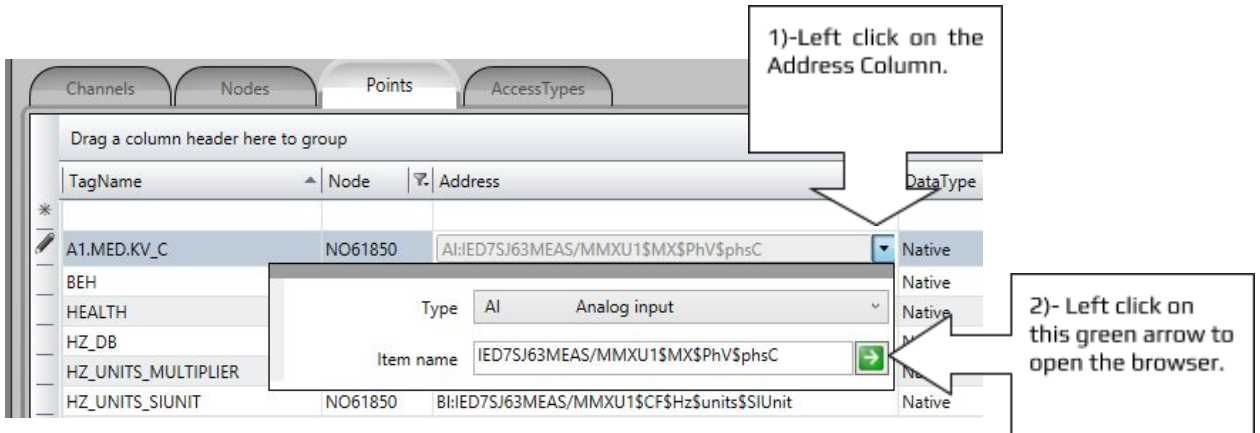
Write	
Name	Write
ReadOnStartup	False
ReadEnable	Never
ReadPollingRate	
ReadTrigger	
ReadStatus	
ReadCompleted	
WriteEventEnabled	True
WriteEvent	Changed
WriteTrigger	
WriteStatus	
WriteCompleted	
AcceptUnsolicited	False
UseStaticBlocks	False
BlockCommand	
Description	



5. BROWSER

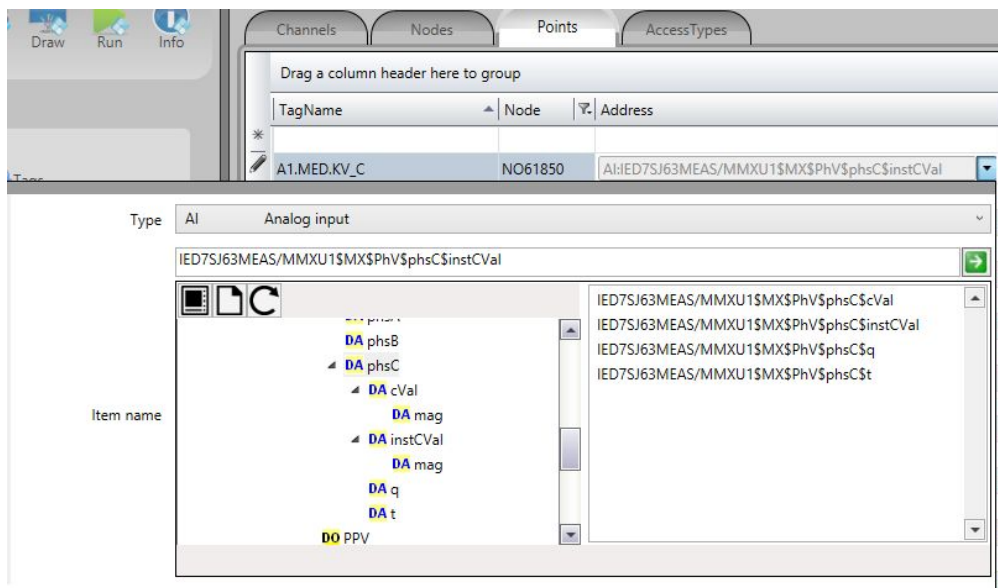
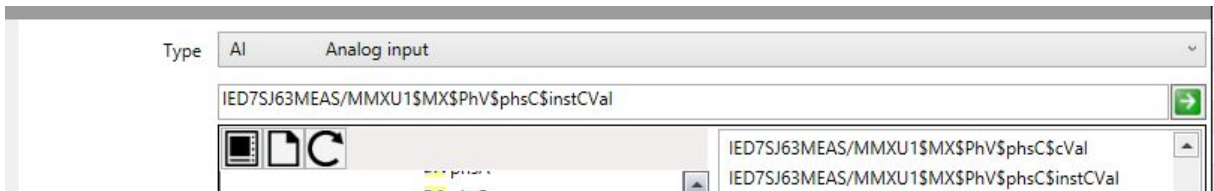
5.1 General

To facilitate the registration of remote IEC61850 points into the SCADA points table, an IEC61850 Browser feature, shown below, is available.

To start the feature follow the steps 1 and 2 below.



The browser will work over an IED () or over a SCL file (). This way, if the File SCL option is enabled and an IED IP was provided, the browser will begin accessing the IED and the double arrows will alternate between the selected choices.



The window consists of two frames presented side by side. The left frame displays a hierarchical list in the form of a tree, with the IED objects. Clicking on the branches of the tree will list on the right frame the objects available for that hierarchical level.

The left frame shows the server designated by the Node field of that particular point record. Left clicking on the server name on the tree will show the IEC61850 objects that belong to that particular server.

The IEC61850 tree structure, briefly:

LD – Logical Devices – Logical devices inside a physical device.

LN – Logical Nodes – Implements functions inside an LD

FC – Functional Constraints

DO – Data objects

DA – Data attributes

5.2 Data Sources

To load the objects on this window from Data Objects, Reports and DataSets configured in the IED server, the user can make a direct connection with the actual IED or reading an SCL (Substation Configuration Language) supported file. The supported files are ICD (IED Capability Description), SCD (Substation Configuration Description), CID (Configured IED Description) and SSD (Substation).

By right clicking on the name of the current server, the user can choose the browsing option: via remote connection or via description file.

5.3 Remote IED Connection

If the user chooses connection with the remote IED, clicking on the name of a remote server for the first time will cause the browser to attempt a connection to this server. If the connection is successful, the tree will expand to show the Logical Devices on the server. Clicking on each Logical Device will read all objects included in it.

In case the data takes too long to load, there may have been a failure to communicate with the IED. In this case, messages like “Not connected” or “Time Out” will appear at the bottom of the Browser window. Therefore, the server icon in the tree will change to show a red “X” indicating connection failure. To make a new connection attempt click on the LD items (Logical Devices) again or close the screen and try again, for a full reset of the browser.

5.4 Reading SCL files

To load the browser window from the reading of an SCL file, its name must have been specified in the Node configuration and the file must have been previously placed in the directory pointed to in the Node configuration options section of the project. In this case,

clicking on the server name starts the reading of the file and display the Logical Devices. When clicking the LDs, all existing data objects are read hierarchically.

As already mentioned in the SCADA channel description for handling IEC61850 protocol servers, it is not necessary for the server name specified to be the same as the one configured within the IED. In case of reading SCL files, to assist the user if these names are different and because there could be several IEDS described in SCD-type files, a window appears listing the IEDS names found in the file. The user must choose the desired one to be used in that Node.

5.5 Data Objects

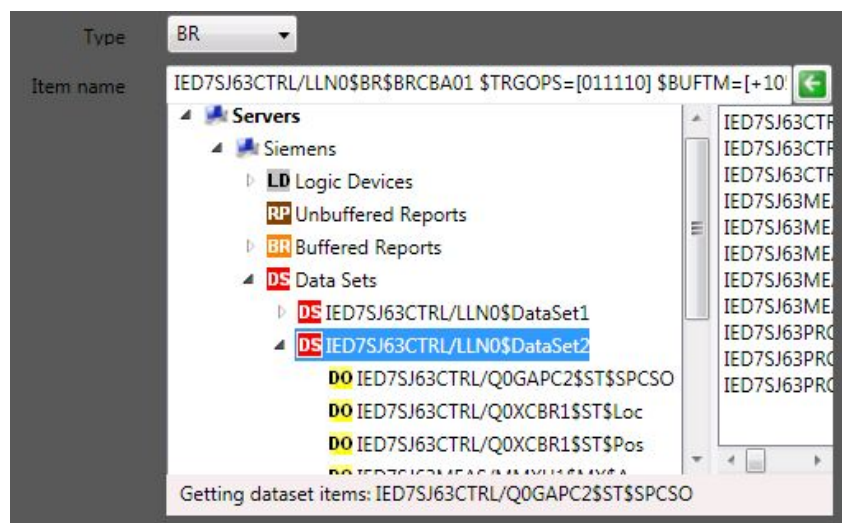
When it comes to selecting the desired level of Data Object, just click on it in the right frame of the browser and press OK to trigger the transfer to the points table Address field, in the current field register. The Node field should be set with the name of the desired target Node. The list on the right side of the window allows the user to select multiple objects. If the user selected more than one item, after clicking on OK, new records will be created in the points table to hold all the selected items.

According to the FC (Functional Constraint) of the chosen point (the points in the figure have the "MX" FC, which means measures), SCADA automatically suggests the point type for the Types column in the points table (in this case, the EA-type Analog Input will be suggested). The type suggested may be altered if necessary.

5.6 DataSets

In the tree, after the LD section, branches with Reports (buffered and unbuffered) and DataSets appear.

A DataSet is an object in the form of a set of Data Objects (DO), pre-configured in the IED.



The figure above shows the list of defined datasets in an IED server. Clicking on the name of the DataSets shows the data objects (DO) that were set up in this DataSet. A list of these objects also appears in the right frame of the window.

To set DataSet readings by requested sampling, the user should choose the name of the DataSet and create a record in the points table, as described in the configuration of points section.

5.7 Reports

Reports are actions consisting in dispatching unsolicited data, initiated by IEDs servers for clients who activated these Reports.

Each Report has its own configuration attributes that define its behavior. One is the name of the DataSet to be sent by the Report. Each time a change in the objects on a DataSet specified by the Report is sensed, a new instance of this Report is created and sent to the registered client.

When the user clicks on the branches RP - Unbuffered Reports or BR - Buffered Reports, these branches will drop down to display the Reports configured in the IED server.

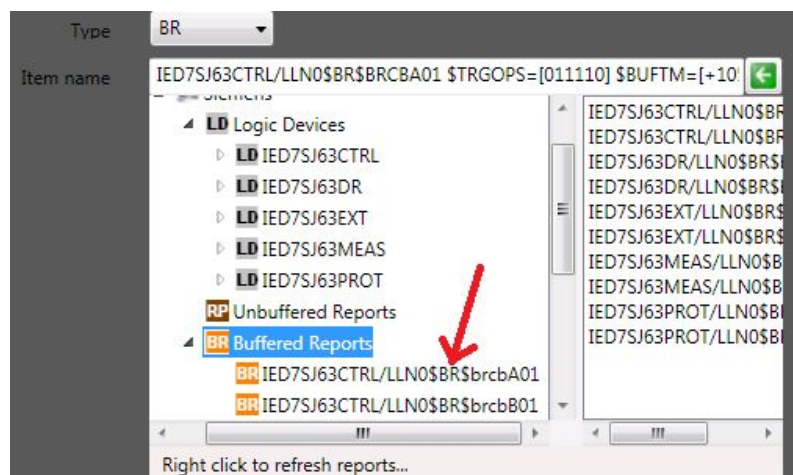
The way this reading is performed, in the case of a direct connection to the IED, requires the LDs to be previously loaded in order to show the respective Reports. After one of these loads, if some reports are not appearing, right click on the main RP or BR branch and choose the "refresh" item from the menu. For ICD files, the data load is always performed together.

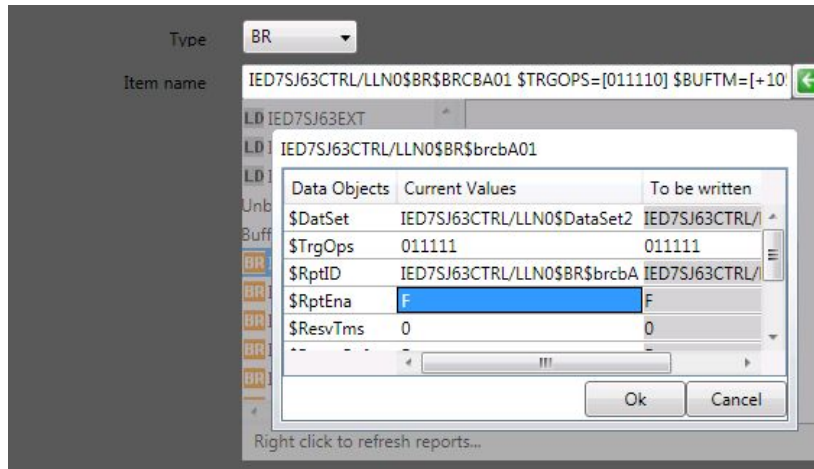
To choose a Report whose denomination must be transferred to the points table, the name of that same Report that appears in the tree below Logical Nodes in BR or RP can be used. This choice can also be made in the RP or BR branches, after the LDs, which list the Reports. By expanding these branches the names of the report control blocks appear in the right frame of the browser window.

5.8 Report Attribute Window

Clicking on a Report, on BR or RP tree's Section, as indicated by the red arrow in figure 13, the control block attributes window of the Report (BRCB) is shown (Figure 14).

This window (Figure 14) shows the contents of the Report attributes: The attribute names are on the left column (Data Objects), the central column (Current Values) shows the values currently read from the browser, and the right column opens a window, with a click on the desired attribute, where the user can set a new value for it. The value will be loaded on the remote server for the next SCADA real time execution.





In the example below (Figure 15), the attributes that have been modified are **\$BufTm** and **\$TrgOps**.

TagName	Node	Address	Data
* ANA61[6]	NO61850	BR:IED7SJ63CTRL/LLN0\$BR\$BRCBA01 \$TRGOPS=[011110] \$BUFTM=[+105]	Nati

6. COMTRADE FILES

6.1 General information

The COMTRADE standard (IEEE Standard Common Format for Transient Data Exchange) defines a standard format for the data files and other files needed to represent faults, tests, or simulations in the electrical sector. Up to 4 types of files can be generated with the *.HDR, *.CFG, *.DAT and *.INF extensions.

- **HDR** (Header) - This is an optional text file. It contains general information about the event. It is recorded in ASCII format and its content does not need to be stored following a specific order.
- **CFG** (Configuration) - This is a required text file. It is recorded in ASCII format and its content is organized in order to allow its automated reading through a computational algorithm. It consists of data such as the substation name, number of analog and digital channels, channel names, units, conversion factors, electrical system frequency, sampling rates, date and time, and the format in which the data file was recorded.
- **DAT** - This is a required text file. It is recorded in ASCII or binary format. It contains only numbers and corresponds to samples taken of the analog and digital quantities. Its correct interpretation depends on the information contained in the configuration file. The configuration file specifies means to translate these values appropriately to real quantities, such as volts and amperes. Digital data is represented by the digits "0" and "1."

- **INF** (Information) - This is an optional text file. It is recorded in ASCII or UTF-8 format. It contains extra information beyond the minimum required to apply the dataset. The format provides information that can be read and used by any user, and information that can be accessible only to users of a particular class. The information store is divided into sections, which can be of the public or private type. Each section consists of a header line followed by a certain number of input lines. Public sections contain information that can be used by equipment and software of various types of manufacturers; private sections contain information that can be used only by a specific manufacturer of software, hardware or equipment.

6.2 Reading and Recording

The driver will scan the IED file directories for files with the *.HDR, *.CFG, *.DAT, and *.INF extensions. This scan will be performed by MMS or FTP, every X period of time and saved in a directory of the computer according to the settings defined in the node.

If the file already exists in the directory, it will have its name and last modified date compared to that of the file in the IED and this will define if the file should be transferred or deleted, if this option is enabled in the configuration of the node.

The operations are logged at the DEBUG level in the LOG file and they will be logged if the logging level in the channel configuration is set to DEBUG. For cases where FTP is used, a new LOG file named FTP.log will be generated within the folder of each channel. It has been separated to have a better debugging organization when the operation uses FTP to read the COMTRADE files.

7. DIAGNOSIS OF COMMUNICATION PROBLEMS

7.1 Logs of Operations (LOG)

The protocol uses a feature that records events into a text file LOG. There are four types of messages: **Debug, Info, Warning, and Error**.

Debug	Messages at the lowest level. Used when it is necessary to identify in more detail each process. Usually used by developers, testers and integrators.
Info	Important information messages for monitoring the operation of the Communication Module.
Warning	Warning messages, which can be bypassed or need to be scanned.
Error	Error messages that cannot be ignored and need immediate attention and correction for the correct operation of the Communication Module.

7.1.1 SCADA Trace Window

SCADA has the trace window, which runs on the server and displays the real-time operations log of each channel and node. It has 4 important columns (Info 1, Info 2, Info 3 and Info 4), which show the Channel, Node, Point address and method executed by the driver.

7.1.2 Windows Event

The Communication Module is configured internally to always log all Warning and Error messages. This rule is independent of the log level setting that is set.

The format of the message consists of: Channel, Node (when available), Point address (when available), Date / Time, Message type, Thread, Method and message.

7.1.3 Text File

- The Communication Module will log messages based on the hierarchical level defined in the channel configuration (Session 2 of the Manual).
- The directory consists of folders for each channel and node that was initialized. The root is at: C:\Action.NET\Projects\Logs\Drivers\
- The text file is created in the format: Log-DAY-MONTH-YEAR.log
- For example, if the maximum storage of 30 files is configured, each with a maximum size of 35 Megabytes then each day cycle a file will be created if it has not exceeded the maximum size. If this size is exceeded once, a copy of that day's file with a final 1 will be created and another, new file will be created. If the burst repeats on the same day, the new file replaces the copy and a new file is started. If not, a new daily archive cycle is created. Therefore, each node can store approximately 1076 Megabytes to disk.
- The message format consists of: Point address (when available), Date/Time, Message type, Thread, Method and message.

7.2 Codes of Errors and Qualities

7.2.1 Command Errors (CO)

Command error codes are read from the lastAppError attribute. The attribute is valid only for command data classes and its value displays the status of the last command. It is updated when command responses are received from the devices. Used by Functional Point Constraint CO.

Code	Cause	Description
0	UNKNOWN	Unknown
1	NOT_SUPPORTED	Not supported
2	BLOCKED_BY_SWITCHING_HIERARCHY	Not successful, because one of the downstream Loc switches has the value TRUE.
3	SELECT_FAILED	Canceled due to an unsuccessful selection (select service).
4	INVALID_POSITION	Control action is aborted due to invalid switch position.
5	POSITION_REACHED	Switch is already in the intended position.
6	PARAMETER_CHANGE_IN_EXECUTION	Control action is blocked due to running parameter change.

7	STEP_LIMIT	Control action is blocked because tap changer has reached the limit EndPosR or EndPosL.
8	BLOCKED_BY_MODE	Control action is blocked because the LN is in a mode (Mod) which does not allow any switching.
9	BLOCKED_BY_PROCESS	Control action is blocked due to some external event at process level that prevents a successful operation.
10	BLOCKED_BY_INTERLOCKING	Control action is blocked due to interlocking of switching devices.
11	BLOCKED_BY_SYNCHROCHECK	Control action with synchrocheck is aborted due to exceed of time limit and missing synchronism condition.
12	COMMAND_ALREADY_IN_EXECUTION	Control select or cancel service is rejected because control action is already running.
13	BLOCKED_BY_HEALTH	Control action is blocked due to some internal event that prevents a successful operation (Health).
14	1_OF_N_CONTROL	Control action is blocked because another control action in a domain (for example substation) is already running.
15	ABORTION_BY_CANCEL	Control action is aborted due to cancel service.
16	TIME_LIMIT_OVER	Control action is terminated due to exceed of some time limit.
17	ABORTION_BY_TRIP	Control action is aborted due to a trip.
18	OBJECT_NOT_SELECTED	Control action is rejected because control object was not selected.
19	OBJECT_ALREADY_SELECTED	Select action is not executed because the addressed object is already selected.
20	NO_ACCESS_AUTHORITY	Control action is blocked due to lack of access authority.
21	ENDED_WITH_OVERSHOOT	Control action executed but the end position has overshoot.
22	ABORTION_DUE_TO_DEVIATION	Control action is aborted due to deviation between the command value and the measured value.
23	ABORTION_BY_COMMUNICATION_LOSS	Control action is aborted due to the loss of connection with the client that issued the control.
24	ABORTION_BY_COMMAND	Control action is blocked due to the data attribute CmdBlk.stVal is TRUE.
25	NONE	Command not successful due to Unknown causes.
26	INCONSISTENT_PARAMETERS	The parameters between successive control services are not consistent for example the ctrlNum of Select and Operate service are different.
27	LOCKED_BY_OTHER_CLIENT	Another client has already reserved the object.

99	CONTROL_MODEL_STATUS_ONLY	The object is not controllable, only the services that apply to a status object are supported. The attribute ctlVal does not exist.
----	---------------------------	---

7.2.2 Quality

The Quality Table standard adopted by SCADA is Open Platform Communications (OPC). The table below is used to convert a quality from IEC 61850 to an OPC quality code.

61850 Name	OPC Code	OPC Name	OPC Description
Validity Invalid Detail Overflow	0	Bad [Non-Specific]	Non-specific. The value is bad but no specific reason is known.
Config Error	4	Bad [Configuration Error]	Configuration Error. There is some server-specific problem with the configuration. For example the item in question has been deleted from the configuration.
	8	Bad [Not Connected]	Not Connected. The input is required to be logically connected to something but is not. This quality may reflect that no value is available at this time, for reasons like the value may have not been provided by the data source.
Detail Failure	12	Bad [Device Failure]	Device Failure. A device failure has been detected.
Time Quality (Clock Failure / Not Synchronized)	16	Bad [Sensor Failure]	Sensor Failure. A sensor failure had been detected (the "Limits" field can provide additional diagnostic information in some situations).
	20	Bad [Last Known Value]	Last Known Value. Communications have failed. However, the last known value is available. Remark: The "age" of the value may be determined from the TIMESTAMP in the OPCITEMSTATE.
	24	Bad [Communication Failure]	Comm Failure. Communications have failed. There is no last known value available.
	28	Bad [Out of Service]	Out of Service. The block is off scan or otherwise locked. This quality is also used when the active state of the item or

			the group containing the item is Inactive.
	32	Bad [Waiting For Initial Data]	Waiting for Initial Data. After Items are added to a group, it may take some time for the server to actually obtain values for these items. In such cases the client might perform a read (from cache), or establish a ConnectionPoint-based subscription and/or execute a Refresh on such a subscription before the values are available. This substatus is only available from OPC DA 3.0 or newer servers.
Validity Questionable	64	Uncertain [Non-Specific]	Non-specific. There is no specific reason why the value is uncertain.
Detail Old Data Operator Blocked	68	Uncertain [Last Usable]	Last Usable Value. Whatever was writing this value has stopped doing so. The returned value should be regarded as "stale." Remark: This differs from a BAD value with Substatus 5 (Last Known Value). That status is associated specifically with a detectable communications error on a "fetched" value. This error is associated with the failure of some external source to "put" something into the value within an acceptable period of time. Remark: The "age" of the value can be determined from the <code>TIMESTAMP</code> in <code>OPCITEMSTATE</code> .
Detail Out of Range Detail Bad Reference Detail Oscillatory Detail Inconsistent Detail Inaccurate	80	Uncertain [Sensor Not Accurate]	Sensor Not Accurate. Either the value has "pegged" at one of the sensor limits (in which case the limit field should be set to 1 or 2) or the sensor is otherwise known to be out of calibration via some form of internal diagnostics (in which case the limit field should be 0).

	84	Uncertain [EU Exceeded]	Engineering Units Exceeded. The returned value is outside the limits defined for this parameter. Remark: In this case (per the Fieldbus Specification) the "Limits" field indicates which limit has been exceeded but does NOT necessarily imply that the value cannot move farther out of range.
	88	Uncertain [Sub-Normal]	Sub-Normal. The value is derived from multiple sources and has less than the required number of Good sources.
Validity Good Validity Test	192	Good [Non-Specific]	Non-specific. The value is good. There are no special conditions.
Source Substituted	216	Good [Local Override]	Local Override. The value has been Overridden. Typically this means the input has been disconnected and a manually entered value has been "forced."

7.2.2.1 Communication Failure

When the communication module establishes a connection with the equipment and there is a communication failure between them, the quality of the points will be updated to the value 24, which corresponds to the Bad [Communication Failure] code of the OPC. The value and date/time of the points will not change. If the communication module has never established communication with the equipment, it will not be considered as a communication failure that changes the quality value of the points.