IEC 61850

IEC 61850 protocol

Protocol description

The IEC 61850 protocol is an IEC standard designed for communication with Intelligent Electronic Devices (IED) at electrical substations (breakers, protections, transformers, switches, etc).

The IEC 61850 protocol is - like IEC 60870-6 ICCP/TASE.2 - based on the Manufacturing Message Specification (MMS) and therefore the implementation in the D2000 KOM process uses a common MMS framework.

Implementation of the IEC 61850 protocol in the D2000 system supports this functionality:

1. Periodic reading of values (polling)
2. Reading of values from predefined Datasets made available in predefined Information Reports - both buffered and unbuffered
3. Reading of array elements of which are simple values
4. Writing to simple types of values (writing to structures is not implemented yet)
5. Writing to array elements of which are simple values
6. Browsing - when communication is established, the following items can be browsed:
    - logical devices within a physical device (in the station address configuration)
    - objects located within the logical device (in the I/O tag configuration)
    - reports located within the logical device (in the I/O tag configuration)

There is no support for the processing of GOOSE and GSSE reports.

The ISO over TCP/IP transmission layer is implemented according to the RFC 1006 specification.

The protocol was tested with software simulators and with ABB Substation management unit COM600.

Communication line configuration

- Communication line category: TCP/IP-TCP.
- TCP Parameters:
    - Host: string max. 80 characters – server name in INET format (a name or numerical address a.b.c.d)
    - Port: TCP port number (0 to 65535), port 102 is used by default.
    - Line number: not used, set to 1

A valid hostname or IP address of the device must be entered according to the rules above.

The port number where the device is listening must also be entered. The *ISO over TCP/IP* standard uses port 102.

In the case of redundant systems, it is also possible to enter multiple comma-separated names/addresses. When the connection is broken, the communication process will again attempt to establish a connection to the device at the specified address. If the attempt is unsuccessful, then the process tries to establish a connection to the next address. This is repeated cyclically until the connection with one server is established.

Communication line protocol parameters

Communication lines - configuration dialog box - tab **Protocol parameters**.
These parameters influence some optional parameters of the protocol. You can define the following parameters:

Table 1

Parameter	Meaning	Unit / Type	Default value
Address parameters of local side (D2000 KOM process)			

Local TSEL (hex)	Octet string that represents the local Transport Selector. It identifies Transport SAP. The maximum size is 32 octets (64 ASCII encoded hexadecimal digits).	octet string	00 01
Local SSEL (hex)	Octet string that represents the local Session Selector. It identifies Session SAP. The maximum size is 16 octets (32 ASCII encoded hexadecimal digits).	octet string	00 01
Local PSEL (hex)	Octet string that represents the local Presentation Selector. It identifies Presentation SAP. The maximum size is 16 octets (32 ASCII encoded hexadecimal digits).	octet string	00 00 00 01
Local AP Title	Local Application Process Title is an identifier that is assigned by an address manager. It represents a specific application process.	string	1.3.9999.1
Local AE Qualifier	Integer value used to identify the local Application Entity.	string	1
Address parameters of remote side (the IED device)			
Remote TSEL (hex)	Octet string that represents the remote Transport Selector. It identifies Transport SAP. The maximum size is 32 octets (64 ASCII encoded hexadecimal digits).	octet string	00 01
Remote SSEL (hex)	Octet string that represents the remote Session Selector. It identifies Session SAP. The maximum size is 16 octets (32 ASCII encoded hexadecimal digits).	octet string	00 01
Remote PSEL (hex)	Octet string that represents the remote Presentation Selector. It identifies Presentation SAP. The maximum size is 16 octets (32 ASCII encoded hexadecimal digits).	octet string	00 00 00 01
Remote AP Title	Remote Application Process Title is an identifier that is assigned by an address manager. It represents a specific application process.	string	1.3.9999.1
Remote AE Qualifier	Integer value used to identify the remote Application Entity.	string	1
Common address parameters			
Bilateral Table ID	Name of the bilateral table. In the case of IEC 61850, this can be an empty string.	-	
Max MMS Message Size	Maximum size of MMS messages.	bytes	32000
Max Requests Pending	Maximum proposed transactions that could be sent and unacknowledged. The value can be reduced if the other party suggests a lower value at negotiation.	1 .. 32	5
Max Indications Pending	Maximum proposed transactions that could be received without sending an acknowledgment. The value is sent to the other party.	1 .. 32	5
Max Nesting Level	The maximum level of nesting for MMS data structures.	1 .. 10	5
ISO TPDU Size	The maximum packet size for "ISO over TCP" protocol level (according to RFC 1006)	8192/4096 /2048 /1024/512 /256/128 bytes	1024 bytes
Heartbeat	Timer, which permits the sending ICCP message I*dentify request* to the communicating party to find out the validity of TCP/IP connection. To enable the timer a nonzero value must be specified. If the connection was aborted on the TCP/IP level, fast detection ensures its faster restoring. This parameter is recommended in situations when transmitting data via large networks or when a small amount of data is transmitted via the active connection. The value 0 (implicit) turns off the HeartBeat timer. A positive value means the seconds to send a Heartbeat message. If any message is sent, the Heartbeat timer is reset and the Heartbeat message is sent only after a timeout elapses without any communication between partners.	sec	0
Retry Delay	Delay inserted before repeated attempt to establish the connection after it has been broken. If the connection should be restored as fast as possible, set a small value or 0 seconds.	sec	10 sec
Inter Read Timeout	Maximum waiting time to receive TCP data. After this timeout elapses, the possible requirements (data) for sending to a communicating party are checked. As communication is executed via one thread, the high value can reduce the speed of interaction with the party. The recommended value is 50 to 150 milliseconds.	msec	100 msec
Parallel Stations	The parameter activates creating TCP connections for each station. If the device supports multiple clients, communication can be parallelized. If the parameter is set to NO, a single TCP connection will be created for all communication stations.	YES/NO	NO

Map IEC 61850 flags	The way IEC 61850 flags are mapped into D2000 flags A .. M. IEC 61850 protocol has quality flags mapped into 13 bits of the Quality attribute, details are given in the description of the parameter Quality Index: Mapping can be: • *None* - flags A .. M are not set • *Simple* - Quality attribute directly maps to flags A .. M **Note**: after the change of this parameter we recommend a restart of the KOM process or communication partner so that all values come into the system with properly set flags.	None / Simple	None
Report Trigger Options	The setting of Trigger parameters used to trigger reports (TrgOps). The following bits are used according to the standard (the numbering is from the highest bit, which is bit 0): • bit 0 - *reserved* • bit 1 - *data-change* • bit 2 - *quality-change* • bit 3 - *data-update* • bit 4 - *integrity* • bit 5 - *general-interrogation* The default value is 0x74 (all bits are active except *integrity* - this is activated if the parameter Integrity Period is specified in the report configuration. Note: ABB Substation Management Unit COM600 required 0x64, it did not support bit 3 - *data-update*.	1-byte octet string	74
Additional Browse Items	The parameter enables specifying additional names of objects to query during browsing for I/O tags. Names must be separated by a space, e.g. "Obj1 Obj2". This makes it possible to handle the situation when IED deliberately does not list some objects in the GetNameList-Response message (response to a requested list of objects within a logical device). Note: The ABB Substation Management Unit COM600 required parameter setting to *LLN0* - this object was not propagated but it was important (it contained e.g. buffered and unbuffered reports).		
Authentication Type	Type of authentication. These types of authentication are supported: • None - no authentication • Password - password-based authentication. The password is transmitted unencrypted within *ISO 8650-1 OSI Association Control Service* layer.	None Password	None
Password	Password used if Authentication Type = Password	string	
Debug settings			
Debug I/O binary packets info	Enables debug information on the level of binary packets. See Note 1.	YES/NO	NO
Debug ISO packet level info	Enables debug information on the ISO OSI layer. See Note 1.	YES/NO	NO
Debug MMS level info	Enables debug information on the MMS data level. See Note 1.	YES/NO	NO
Full IEC61850 level info	Enables debug information on the top level of IEC 61850 data. See Note 1.	YES/NO	NO
Incoming values info	Enables detailed debug information about incoming values (data values). See Note 1.	YES/NO	NO
Outgoing values info	Enables detailed debug information about outgoing values (data values). See Note 1.	YES/NO	NO

Note 1

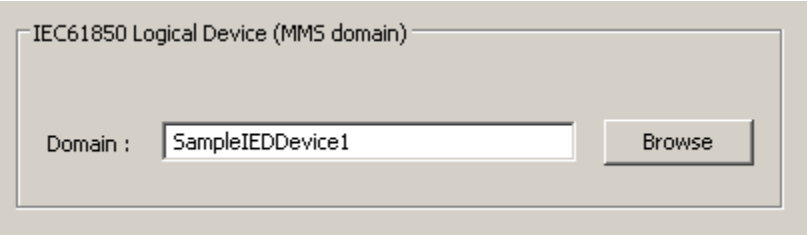
If all debug info is enabled, it could cause an overload of communication workstation and decrease of data transfer rate from IED to D2000. After the communication is tuned and debugged, we recommend minimizing the amount of debugging information.

Communication station configuration

- Communication protocol "**IEC 61850**".
- Polling parameters are used for reading I/O tags of the "Periodically polled value" type.
- The protocol does not support time synchronization.
- in the Time parameters tab, monotonic UTC time (with offset 0) should be set, as IEC 61850 uses UTC-based timestamps (unless a specific implementation is non-compliant with this part of the standard).

Station address

Img. No. 1, Station address



The station corresponds to one logical device. In one physical device, one or more logical devices can be defined. The domain is the name of the logical device values of which we want to read. Multiple stations can be on one line. For each station, a dedicated TCP connection to a defined physical device will be created.

The **Browse** button enables getting a list of logical devices (if the KOM process is running and communication is established). For Browsing functionality, the device must implement support for getNameList request with parameter ObjectClass = DOMAIN.

Station protocol parameters

Communication station - configuration dialog box - tab **Protocol parameters**.
These parameters influence some optional parameters of the protocol. You can set the following station parameters:

Table 2

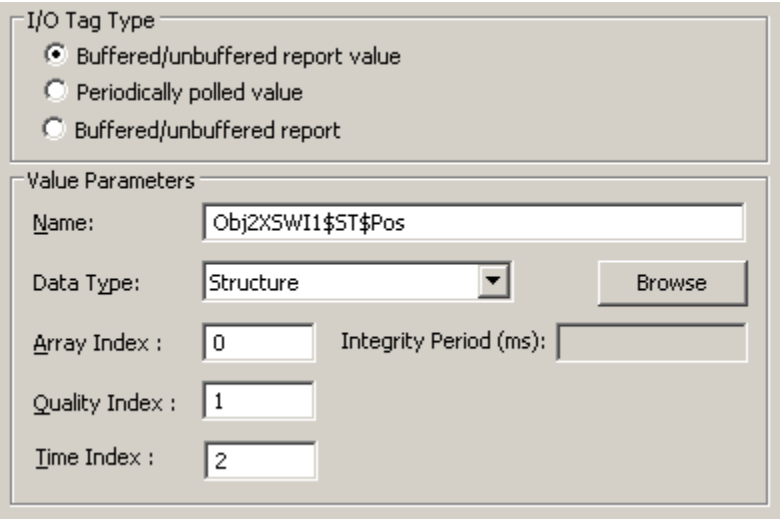
Object group	Parameter	Meaning	Unit / size	
Interpretation of quaternary values	QERR Value	Interpretation of Quaternary value Error from the received integer value or from 2-bit State value.	0, 1, 2, 3	3 (11 binary)
	QOFF Value	Interpretation of Quaternary value Off from the received integer value or from 2-bit State value.	0, 1, 2, 3	2 (10 binary)
	QON Value	Interpretation of Quaternary value On from the received integer value or from 2-bit State value.	0, 1, 2, 3	1 (01 binary)
	QTRANS Value	Interpretation of Quaternary value Transient/Moving from the received integer value or from 2-bit State value.	0, 1, 2, 3	0 (00 binary)

I/O tag configuration

Permitted I/O tag types: **Ai**, **Ao**, **Ci**, **Co**, **Di**, **Dout**, **TiA**, **ToA**, **TiR**, **ToR, Qi**, **TxtI**, **TxtO**

I/O tag address

Img. No. 2, I/O tag address (Data Value)



The configuration of the I/O tag address requires the following data:

I/O Tag Type

Type of I/O tag:

- **Buffered/unbuffered report value** - the I/O tag represents the value obtained from a buffered or unbuffered report. Such values are sent by the device spontaneously after they are changed.
- **Periodically polled value** - the I/O tag represents the periodically read value (polling). The reading period is configured in the station parameters. **Note:** Periodic reading may result in loss of values as a result of rapid changes, so it should only be used for slowly changing values or if the value can not be retrieved from the report.
- **Buffered/unbuffered report** - the I/O tag represents a buffered or unbuffered report. It is important that all reports that are used (i.e. which contain the requested data) are configured as I/O tags. Based on these I/O tags, a report is activated when a communication is established (write is performed to its attributes *$TrgOps*, $*RptEna*, *$GI* and if the parameter *Integrity period* is specified, also to the attribute *$IntgPd*).

If the I/O tag representing the report has a text value, the value will be set to the name of the Dataset (the *$DatSet* attribute) that the report publishes.
If the I/O tag representing the report has an integer or a real value, the value will be

  - 0 - when a communication has been established - after receiving the initiate-Response message and requesting the reading of the Dataset name (*$DatSet* attribute) that the report publishes
  - 1 - after reading the Dataset name (receiving the device response to the Dataset name read request)
  - 2 - after reading report parameters
  - 3 - after reading a list of objects that are contained in Dataset
  - 4 - after successful report activation
  - 5 and more - with the arrival of each additional Information Report, the value of the object that represents it will be increased
  - invalid - if any of the communication steps fails

This mechanism can be used to define "watchdogs" guarding, for example, a communication failure, connection failure, or non-working sending of Informati on Reports.

Name

A unique text string that identifies the IEC 61850 object or its attribute (a reference). The maximum size is 64 characters.
Note: if the I/O tag's address is specified as **%IGNORE**, such an I/O tag will be ignored.

According to the standard ISO/IEC IEC 61850-8-1 the reference is in form
*<LNVariableName>$<FC>$<LNDataName>$<AttributeName>* (napr. XCBR1STPos$stVal)
respectively - if subattributes are used - in form
<LNVariableName>$<FC>$<LNDataName>$<AttributeName>$<subDataAttributeName> (napr. XBCR1STPos$origin$orCat)
where:

- *LNVariableName* is a name of a Logical Node. The name can be arbitrary, e.g. LLN0, Obj1XCBR1, Obj2XSWI1
- *FC* is a Functional Constraint. Table of defined functional constraints can be seen below.
- *LNDataName* is the name of a DataObject. The name can be arbitrary, the standard contains recommended rules concerning naming conventions.
- *AttributeName* is the name of Attribute of a DataObject. A table of frequently used attributes can be seen below.
- *subDataAttributeName* is the name of the Attribute of a SubDataObject. A table of frequently used attributes can be seen below.

The standard defines the following naming conventions for individual components of a reference:

Defined Functional Constraints according to ISO/IEC IEC 61850-8-1:

FC Name	Description
MX	Measurands (analog values)
ST	Status information
CO	Control service parameters
CF	Configuration
DC	Description
SP	Setting (outside setting group)
SG	Setting group
RP	Unbuffered report control blocks
LG	Log control blocks
BR	Buffered report control blocks
GO	GOOSE control blocks
GS	GSSE control blocks

SV	Substituted values
SE	Setting group editing
MS	Multicast Sampled Values control block
US	Unicast Sampled Values control block
EX	Name space for model extension
SR	Service tracking
OR	Operate received
BL	Blocking

Names of DataObject Attributes and SubDataObject Attributes are defined in ISO/IEC IEC 61850-7-3 and the following table contains only the most common ones:

Attribute Name	Description
ctlModel	Specifies the control model of IEC 61850-7-2 that corresponds to the behavior of the data.
d	Textual description of the data.
evalTm	Time window applied to interharmonic calculations. The value shall be represented in ms.
frequency	Nominal frequency of the power system or some other fundamental frequency in Hz.
instMag	Magnitude of the instantaneous value of a measured value.
mag	Deadbanded value. Shall be based on a dead band calculation from instMag.
numCyc	The number of cycles of power frequency, which are used for harmonic, subharmonic and interharmonic calculations.
numHar	The number of harmonic and subharmonics or interharmonic values that are to be returned as the value attribute.
phsAHar, phsBHar, phsCHar	This array shall contain the harmonic and subharmonics or interharmonic values related to phase A (resp. B, C).
phsABHar, phsBCHar, phsCAHar	This array shall contain the harmonic and subharmonics or interharmonic values related to phase A to phase B (resp. B to C, resp. C to A).
q	Quality of the attribute(s) representing the value of the data
setCharact	This attribute shall describe the curve characteristic.
setMag	The value of an analog setting or set point.
stVal	The status value of the data
swRev	SW-revision
t	The timestamp of the last change in one of the attribute(s) representing the value of the data or in the q attribute.
units	Units of the attribute(s) representing the value of the data (ISO/IEC IEC 61850-7-3 Annex A)
vendor	Name of the vendor.

Data Type

The list of permitted data types:

Data Type	Popis

---Autodetect---	The Data Type will be queried by a GetVariableAccessAttributes-Request message after the connection establishment. **Note 1:** information is currently stored only in KOM process memory, therefore after its restart and the first connection establishment the data type detection is performed for all Autodetect I/O tags. **Note 2:** data types State and StateQ are undistinguishable by this mechanism, therefore objects of these types are detected as StateQ. **Note 3:** the same mechanism for data type detection is used in browsing.
Discrete *	ICCP: Integer 32-bit value signed
DiscreteQ *	ICCP: Integer 32-bit signed value + ICCP Validity
DiscreteQTimeTag *	ICCP: Integer 32-bit signed value + ICCP Validity + Time stamp
DiscreteExtended *	ICCP: Integer 32-bit signed value + ICCP Validity + Current Source + Extended time stamp
Real *	ICCP: Float 32
RealQ *	ICCP: Float 32 + ICCP Validity
RealQTimeTag *	ICCP: Float 32 + ICCP Validity + Time stamp
RealExtended *	ICCP: Float 32 + ICCP Validity + Current Source + Extended time stamp
State *	ICCP: Discrete 2-bit value
StateQ *	ICCP: Discrete 2-bit value + ICCP Validity
StateQTimeTag *	ICCP: Discrete 2-bit value + ICCP Validity + Time stamp
StateExtended *	ICCP: Discrete 2-bit value + ICCP Validity + Current Source + Extended time stamp
Boolean	Boolean value
Float32	32-bit real value
Float64	64-bit real value
Integer8	8-bit signed integer value
Integer16	16-bit signed integer value
Integer32	32-bit signed integer value
Integer64	64-bit signed integer value
Unsigned8	8-bit unsigned integer value
Unsigned16	16-bit unsigned integer value
Unsigned24	24-bit unsigned integer value
Unsigned32	32-bit unsigned integer value
OctetString	Variable-length binarny string
VisibleString	Variable-length text string
UnicodeString	Variable-length text string in UTF8 encoding
BitString	Variable-length bit string
UtcTime	Absolute time (format seconds since 1.1.1970 + milliseconds)
TimeOfDay	Absolute/relative time (format seconds and milliseconds of a day + optionally number of days since 1.1.1984)
Array of Boolean	An array of Boolean values

Array of Float32	An array of 32-bit real values
Array of Float64	An array of 64-bit real values
Array of Integer8	An array of 8-bit signed integer values
Array of Integer16	An array of 16-bit signed integer values
Array of Integer32	An array of 32-bit signed integer values
Array of Integer64	An array of 64-bit signed integer values
Array of Unsigned8	An array of 8-bit unsigned integer values
Array of Unsigned16	An array of 16-bit unsigned integer values
Array of Unsigned24	An array of 24-bit unsigned integer values
Array of Unsigned32	An array of 32-bit unsigned integer values
Array of OctetString	An array of variable-length binary strings
Array of VisibleString	An array of variable-length text strings
Array of UnicodeString	An array of a variable-length text string in UTF8 encoding
Array of BitString	An array of variable-length bit strings
Array of UtcTime	An array of absolute times (format seconds since 1.1.1970 + milliseconds)
Array of TimeOfDay	An array of absolute/relative times (format seconds and milliseconds of a day + optionally number of days since 1.1.1984)
Structure	Structure or Array. The structure can contain simple types, arrays, and nested structures. Arrays consist of elements that can be simple types (here, however, it is more efficient to use some of the *Array of* types), structures, and arrays. If a data type of an I/O tag is configured as a *Structure*, the Array Index parameter specifies the complex address of the structure element to be read in the I/O tag. **Note:** multiple I/O tags can have the same *Name* parameter, the *Structure* data type, and differ only by the Array Index parameter to access the various elements of the structure. If these I/O tags are configured with the *I / O Tag Type = Periodically polled value*, a single reading request will be sent for all such points.

Note * - types marked with asterisk (*) are used in IEC 60870-6 ICCP/TASE.2 protocol (which uses a common MMS framework). For IEC 61850, we recommend not using these types.

Array Index

- For the Autodetect object type and for arrays of elementary types (*Array of*): it is possible to specify an **index of an element** within an array. The arrays in the IEC 61850 protocol are indexed from 0. Not specifying the index results in accessing the 0-th element.

Note: For arrays of elementary types (*Array of*), the IEC61850 supports writing the received values to a target column of a structured variable. If the Destination tab in the configuration of the I/O tag has a Destination column set to a column of a structured variable, the corresponding array items will be written into it. This only works if the Array Index is not entered or is equal to 0.

- For the Structure object type: it is necessary to specify a **complex address** of element (of simple type) within a structure.
  The complex address is in form x.y.z.. e.g. 0.2. Individual indices indicate order within a structure or field, a dot indicates a descend deeper.
  Example 1: Communication line log shows a structure consisting of VisibleString elements. Individual elements will be accessed by complex addresses 0.0, 0.1 and 0.2

Log line	Note	Simple element

08:08:14.322 29-06-2018|D|MMS> [1] STRUCTURE{	The object of *Structure* type	
08:08:14.323 29-06-2018|D|MMS> [1] Structure{	Level 0	
08:08:14.325 29-06-2018|D|MMS> [1] VisibleString=INFO TECH	Level 0.0	yes
08:08:14.327 29-06-2018|D|MMS> [1] VisibleString=1.0	Level 0.1	yes
08:08:14.329 29-06-2018|D|MMS> [1] VisibleString=Current harmonics measurement	Level 0.2	yes
08:08:14.332 29-06-2018|D|MMS> [1] }	End of structure	
08:08:14.334 29-06-2018|D|MMS> [1] }	End of the *Structure* object	

Example 2: Communication line log shows an array consisting of two structures that contain structures that contain elements of the *Float32* type. Individual elements will be accessed by complex addresses 0.0.0 and 1.0.0

Log line	Note	Simple element
08:08:14.169 29-06-2018|D|MMS> [1] ARRAY{	The object of *Array* type	
08:08:14.172 29-06-2018|D|MMS> [1] Structure{	Level 0	
08:08:14.173 29-06-2018|D|MMS> [1] Structure{	Level 0.0	
08:08:14.175 29-06-2018|D|MMS> [1] Float32= 0.00000E+00	Level 0.0.0	yes
08:08:14.177 29-06-2018|D|MMS> [1] }	End of structure	
08:08:14.179 29-06-2018|D|MMS> [1] }	End of structure	
08:08:14.181 29-06-2018|D|MMS> [1] Structure{	Level 1	
08:08:14.183 29-06-2018|D|MMS> [1] Structure{	Level 1.0	
08:08:14.185 29-06-2018|D|MMS> [1] Float32= 1.00000E+02	Level 1.0.0	yes
08:08:14.186 29-06-2018|D|MMS> [1] }	End of structure	
08:08:14.188 29-06-2018|D|MMS> [1] }	End of structure	
08:08:14.188 29-06-2018|D|MMS> [1] }	End of Array	

Quality Index

For the Structure object type: it is possible to specify the index where IEC 61850 Quality is located. The format of the *Quality Index* is the same as a complex address.
IEC 61850 Quality is of Bitstring type with a length of 2 bytes. 13 quality bits are defined as follows (the standard numbers the bits so that the highest bit is 0 and the lowest is 15):

<table>
<tr><th>Bit(s)</th><th>Note</th></tr>
<tr><td>0-1</td><td>Validity:
<table>
<tr><th>Value</th><th>Description</th><th>D2000 attribute</th></tr>
<tr><td>0 0</td><td>Good</td><td>Valid</td></tr>
<tr><td>0 1</td><td>Invalid</td><td>Invalid</td></tr>
<tr><td>1 0</td><td>Reserved</td><td>-</td></tr>
<tr><td>1 1</td><td>Questionable</td><td>Weak</td></tr>
</table></td></tr>
<tr><td>2</td><td>Overflow</td></tr>
<tr><td>3</td><td>OutofRange</td></tr>
<tr><td>4</td><td>BadReference</td></tr>
<tr><td>5</td><td>Oscillatory</td></tr>
<tr><td>6</td><td>Failure</td></tr>
<tr><td>7</td><td>OldData</td></tr>
</table>

8	Inconsistent
9	Inaccurate
10	Source (0-Process, 1-Substituted)
11	Test
12	OperatorBlocked
13-15	unused bits

If at least one of bits number 2-12 is set, the value in D2000 has a Weak attribute.
See the example in Time Index.

Time Index

For the Structure object type: it is possible to specify the index where the timestamp is located. The format of the *Time Index* is the same as a complex address.

If the *Time Index* is not specified, the timestamp of the information report (for I/O Tag Type=*Buffered/unbuffered report value*) or the current time (for I/O Tag Type=*Periodically polled value*) will be used.

Example: information report contains a structure that contains a Bitstring value followed by quality and timestamp. Therefore the object's address has *Array Index* = 0, *Quality Index* = 1 and *Time Index* = *2*.

Log line	Note
14:35:36.198 09-07-2018|D|MMS> [1] STRUCTURE{	Object of *Structure* type
14:35:36.200 09-07-2018|D|MMS> [1] Bitstring=<80>	Value (*Array Index* = 0)
14:35:36.202 09-07-2018|D|MMS> [1] Bitstring=<00><00>	Quality (*Quality Index* = 1)
14:35:36.204 09-07-2018|D|MMS> [1] seconds= 1531115260 fraction= 3223372800 (09-07-2018 07:47:40.751)	Timestamp (*Time Index* = 2)
14:35:36.206 09-07-2018|D|MMS> [1] }	End of *Structure* object

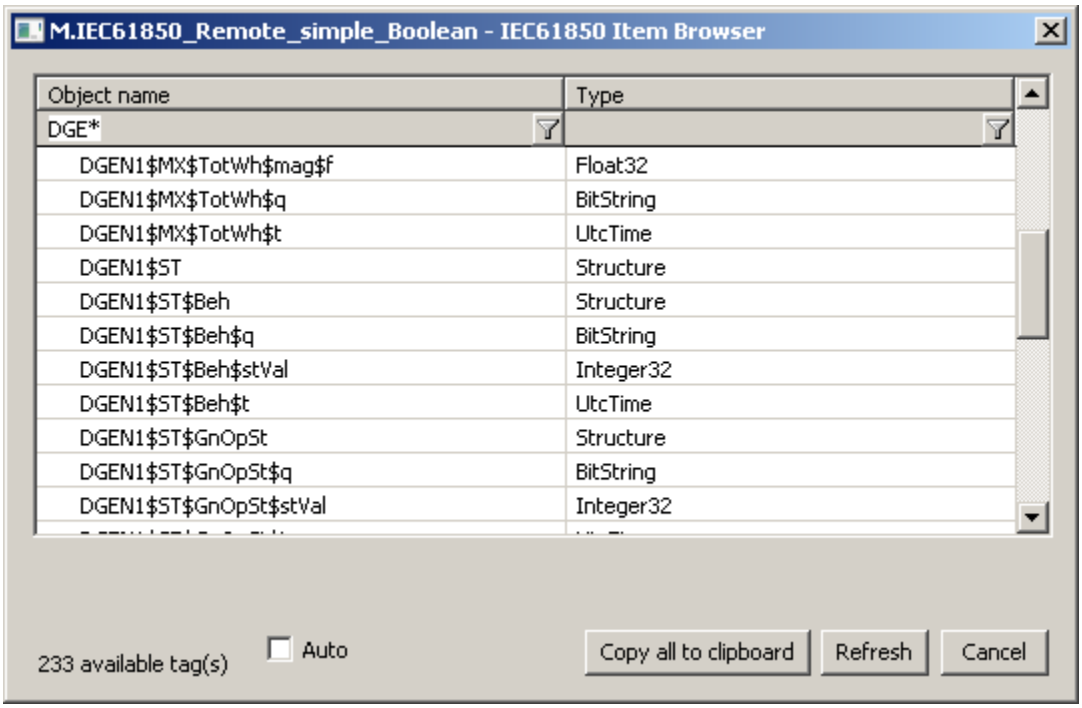
Integrity Period

This parameter is enabled for the I/O tags of the "**Buffered/unbuffered Report**" type and it specifies a period (in ms) after which a report containing the values of all objects in the given datasheet is generated. Therefore, if a loss of value has occurred in some way, this Integrity Report will periodically restore the integrity of the data. Value 0 turns off the generation of the Integrity Report. The non-zero value causes a write to the *$IntgPd* report attribute during the report activation.
Note: Not all IEC 61850 servers support this parameter.

Browse

For the I/O tags, it is possible to find a list of objects and their data types, as long as the KOM process is running and communication is established. When a *Browse* button is clicked, the IEC61850 Browser window opens, and the KOM process begins to query the list of objects with the GetNameList-Request message and then their data types with the GetVariableAccessAttributes-Request message.

Img. no. 3, the IEC 61850 Item Browser window



Meaning of individual choices and buttons:

Auto

If this choice is active, the Data Type will be set to Autodetect, otherwise to value discovered during browsing, e.g. Boolean or Integer32.

Copy all to clipboard

Copies the displayed objects and their respective data types into the Windows Clipboard.

Refresh

By pressing the Refresh button it is possible to enforce re-querying of the list of objects from the device. By default, the KOM process reads the list of objects and their respective data types only during the first browse request and stores them in memory. This reading can take a longer time, depending on the number of objects and speed of the device. These cached lists are sent to the CNF process(es), so that consecutive filling of the Browse window is fast.

Filtering in the list of objects

The browse window enables filtering by the object name and data type. It is not necessary to enter the full text in the filter field. The notation "*FILTERED EXPRESSION*" is supported. The symbol * represents any text before and after the expression (e.g. **momen**).

Note 1

In addition to objects with supported data types (Boolean, Integer32 ...), the object list can contain objects with Structure type. These are structured objects that can not be directly read. The implementation only supports work with simple types and fields whose elements are simple types, therefore individual items of structured objects need to be communicated.

Note 2

In D2000 versions from 20th December 2018 and newer, the recycling of browser dialog has been implemented. If the dialog is closed by the *Cancel* button or after selecting an object, it is actually only hidden and it is available for browsing by another I/O tag within the same station so that the tree structure of the browsed objects is preserved. Clicking on the close icon at the top right corner will cause the dialog to be really closed.

Note 3

Writing to I/O tag with the Array data type (Array of *) is supported for text I/O tags (TxtO). When writing, individual values must be separated by commas, e.g. "1, 2, 3, 7".

Comments on ABB Substation management unit COM600

During the testing of the ABB Substation Management Unit COM600 the following facts were found:

- The device contained several configurable logical names, so it required configuration of several stations on the line (addresses e.g. RF_TS3CTRL, RF_TS3DR, RF_TS3LD0).

- The device has implemented buffered reports. For each report, it supported several instances differing by number (e.g., LLN0BRrcbStatUrg01 .. LLN0BRrcbStatUrg05, LLN0BRrcbMeasReg01 .. LLN0BRrcbMeasReg05). For multiple clients, each one had to subscribe to a specific instance of the report in order to avoid conflict.
- Value of Report Trigger Options parameter had to be 64 (hexadecimal). That means, the device did not support bit 3 - *data-update* during activation of buffered reports.
- The value of the Additional Browse Items parameter had to be *LLN0* so that multiple objects and information reports are found during browsing.
- The buffered reports supported the Integrity Period parameter.
- The individual items of the information reports were of different structure:
  Booleans with quality flags and a timestamp (*Array Index* = 0, *Quality Index* = 1, *Time Index* = 2):

```
15:59:54.444 07-12-2018|D|MMS>  [1]    STRUCTURE{
15:59:54.446 07-12-2018|D|MMS>  [1]      Boolean=FALSE
15:59:54.448 07-12-2018|D|MMS>  [1]      Bitstring=<00><03>
15:59:54.450 07-12-2018|D|MMS>  [1]      seconds= 1544108937 fraction= 3898870784 (06-12-2018 16:08:57.908) quality 1 (Accuracy of
fraction: 1 bits)
15:59:54.452 07-12-2018|D|MMS>  [1]    }
```

  Integers with quality flags and a timestamp (*Array Index* = 0, *Quality Index* = 1, *Time Index* = 2):

```
15:59:54.432 07-12-2018|D|MMS>  [1]    STRUCTURE{
15:59:54.434 07-12-2018|D|MMS>  [1]      Integer8= 1
15:59:54.436 07-12-2018|D|MMS>  [1]      Bitstring=<00><03>
15:59:54.438 07-12-2018|D|MMS>  [1]      seconds= 1544108937 fraction= 3834446336 (06-12-2018 16:08:57.893) quality 1 (Accuracy of
fraction: 1 bits)
15:59:54.440 07-12-2018|D|MMS>  [1]    }
```

  Floats (inside a structure) with quality flags and a timestamp (*Array Index* = 0.0, *Quality Index* = 1, *Time Index* = 2):

```
16:00:03.979 07-12-2018|D|MMS>  [1]    STRUCTURE{
16:00:03.980 07-12-2018|D|MMS>  [1]      Structure{
16:00:03.981 07-12-2018|D|MMS>  [1]       Float32= 3.11465E+01
16:00:03.982 07-12-2018|D|MMS>  [1]      }
16:00:03.982 07-12-2018|D|MMS>  [1]      Bitstring=<00><03>
16:00:03.984 07-12-2018|D|MMS>  [1]      seconds= 1544184591 fraction= 1623978240 (07-12-2018 13:09:51.378) quality 0 (Accuracy of
fraction: 0 bits)
16:00:03.984 07-12-2018|D|MMS>  [1]    }
```

  More complicated structures - e.g. multiple Booleans with common quality flags and a timestamp:

```
16:00:04.118 07-12-2018|D|MMS>  [1]    STRUCTURE{
16:00:04.119 07-12-2018|D|MMS>  [1]      Boolean=FALSE
16:00:04.120 07-12-2018|D|MMS>  [1]      Boolean=FALSE
16:00:04.120 07-12-2018|D|MMS>  [1]      Boolean=FALSE
16:00:04.121 07-12-2018|D|MMS>  [1]      Boolean=FALSE
16:00:04.122 07-12-2018|D|MMS>  [1]      Bitstring=<00><03>
16:00:04.123 07-12-2018|D|MMS>  [1]      seconds= 1544108877 fraction= 3564805632 (06-12-2018 16:07:57.830) quality 0 (Accuracy of
fraction: 0 bits)
16:00:04.124 07-12-2018|D|MMS>  [1]    }
```

  or multiple Booleans and Integers with common quality flags and a timestamp:

```
16:00:04.107 07-12-2018|D|MMS>  [1]    STRUCTURE{
16:00:04.108 07-12-2018|D|MMS>  [1]      Boolean=FALSE
16:00:04.109 07-12-2018|D|MMS>  [1]      Integer8= 0
16:00:04.110 07-12-2018|D|MMS>  [1]      Boolean=FALSE
16:00:04.111 07-12-2018|D|MMS>  [1]      Integer8= 0
16:00:04.112 07-12-2018|D|MMS>  [1]      Boolean=FALSE
16:00:04.112 07-12-2018|D|MMS>  [1]      Integer8= 0
16:00:04.113 07-12-2018|D|MMS>  [1]      Boolean=FALSE
16:00:04.114 07-12-2018|D|MMS>  [1]      Integer8= 0
16:00:04.115 07-12-2018|D|MMS>  [1]      Bitstring=<00><03>
16:00:04.116 07-12-2018|D|MMS>  [1]      seconds= 1544184591 fraction= 2075243776 (07-12-2018 13:09:51.483) quality 0 (Accuracy of
fraction: 0 bits)
16:00:04.117 07-12-2018|D|MMS>  [1]    }
```

Tell commands

Command	Syntax	Description
STWATCH	STWATCH StationName	Tell command sends requests for reading values of all I/O tags.

Literature

- RFC 1006 (ISO Transport Service on top of the TCP, Version: 3)
- International Standard ISO/IEC 8073 (Open Systems Interconnection — Protocol for providing the connection-mode transport service)

- International Standard ISO/IEC 8327-1 (Open Systems Interconnection — Connection-oriented Session protocol: Protocol Specification)
- International Standard ISO/IEC 8823-1 (Open Systems Interconnection — Connection-oriented Presentation protocol: Protocol Specification)
- International Standard ISO/IEC 8650-1 (Open Systems Interconnection — Connection-oriented protocol for the Association Control Service Element: Protocol Specification)
- International Standard ISO/IEC IEC 61850-1 (Communication networks and systems in substations – Part 1: Introduction and overview)
- International Standard ISO/IEC IEC 61850-2 (Communication networks and systems in substations – Part 2: Glossary)
- International Standard ISO/IEC IEC 61850-3 (Communication networks and systems in substations – Part 3: General requirements)
- International Standard ISO/IEC IEC 61850-4 (Communication networks and systems in substations – Part 4: System and project management)
- International Standard ISO/IEC IEC 61850-5 (Communication networks and systems in substations – Part 5: Communication requirements for functions and device models)
- International Standard ISO/IEC IEC 61850-6 (Communication networks and systems in substations – Part 6: Configuration description language for communication in electrical substations related to IEDs)
- International Standard ISO/IEC IEC 61850-7-1 (Communication networks and systems in substations – Part 7-1: Basic communication structure for substation and feeder equipment – Principles and models)
- International Standard ISO/IEC IEC 61850-7-2 (Communication networks and systems in substations – Part 7-2: Basic communication structure for substation and feeder equipment – Abstract communication service interface (ACSI))
- International Standard ISO/IEC IEC 61850-8-1 (Communication networks and systems in substations – Part 8-1: Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3)
- International Standard ISO/IEC IEC 61850-9-1 (Communication networks and systems in substations – Part 9-1: Specific Communication Service Mapping (SCSM) – Sampled values over serial unidirectional multidrop point to point link)
- International Standard ISO/IEC IEC 61850-9-2 (Communication networks and systems in substations – Part 9-2: Specific Communication Service Mapping (SCSM) – Sampled values over ISO/IEC 8802-3)

Changes and modifications

-

Document revisions

- Ver. 1.0 - June 20, 2018 - Creation of document.

Related pages:

Communication protocols