

15 IEC 61850

- 15.1 Introduction
- 15.2 IEC 61850 Server
- 15.3 IEC 61850 Client

15.1 Introduction

IEC 61850 is an international standard defining communication protocols for intelligent electronic devices at electrical substations. It is a part of the International Electrotechnical Commission's (IEC) Technical Committee 57 reference architecture for electric power systems. The abstract data models defined in IEC 61850 can be mapped to several protocols. Possible mappings in the standard can be MMS (Manufacturing Message Specification), GOOSE (Generic Object Oriented Substation Event), and SMV (Sampled Measured Values). These protocols can run over TCP/IP networks or substation LANs using high-speed switched Ethernet to obtain the necessary response times below four milliseconds for protective relaying.



As of version v1.5.0, WCC Lite supports MMS-type messaging. Logging and group setting services are not supported.

15.2 IEC 61850 Server

WCC Lite can act as an IEC 61850 server to serve data to remote SCADA systems. For example, WCC Lite can be used to acquire data from various protocols (Modbus, IEC 60870-5-103, etc.), this data can be redirected and propagated further to a single or multiple IEC 61850 clients. IEC 61850 Server supports TCP and TLS connection types. TCP connection can be secured with password authentication.

Commands


WCC Lite **IEC 61850 Server** implementation defines four command types which are described by their control model:

- **Case 1:** Direct control with normal security (direct-operate);
- **Case 2:** SBO control with normal security (operate-once or operate-many);
- **Case 3:** Direct control with enhanced security (direct-operate);
- **Case 4:** SBO control with enhanced security (operate-once or operate-many).

Normal security commands are considered for execution if the command signal is found in Excel configuration. There aren't any additional checks in command execution in any master protocol.

Enhanced security commands need feedback from the master protocol to either succeed or fail. If feedback is not received within the **command_ack_timeout_ms** timeframe, the command is considered failed.

Command value attributes (e.g. stVal) must be updated separately (if they need to be updated).

 When using SBO commands, select is not routed to the master protocol and select logic is performed only in IEC 61850 Server protocol.

Configuring data points

To use the IEC 61850 Server in WCC Lite, it has to be configured via an Excel configuration and the data model must be uploaded. This configuration contains two Excel sheets where parameters have to be filled in - Devices and Signals.

If a few devices were to connect to a server using the same virtual port, all of the IP addresses have to be specified on the host field separated by space. That way all of the clients will be able to connect from different IP addresses but using the same port as long as they all have the same subnet address.

IEC 61850 Server parameters for Devices tab

| Parameter | Type | Description | Required | Default value (when not specified) | Range | |
|--------------|----------------------------|--|----------|---------------------------------------|------------------|-----|
| | | | | | Min | Max |
| name | string | User-friendly name for a device | Yes | | | |
| description | string | Description of a device | No | | | |
| device_alias | string | Alphanumeric string to identify a device | Yes | | | |
| enable | boolean | Enabling/disabling of a device | No | 1 | 0 | 1 |
| protocol | string | Protocol to be used | Yes | | IEC 61850 Server | |
| bind_address | string (IP address format) | The IP address of an interface to use with the server | No | 0.0.0.0 | | |
| host | string (IP address format) | IP address list of allowed IPs (separated with spaces) | Yes | | | |
| port | integer | TCP communication port | No | 102 | | |

| | | | | | | |
|------------------------|---------|---|---------------------|-------|---------------------------|--|
| auth | string | Authorization type | Yes | | "NONE", "PASSWORD", "TLS" | |
| password | string | Authorization password for server device | Yes (for PASSWORD) | | | |
| tls_local_certificate | string | Local certificate for TLS connection | Yes (for TLS) | | | |
| tls_peer_certificate | string | Certificate authority file for TLS connection | Yes (for TLS) | | | |
| tls_private_key | string | A file consisting of the private key for TLS connection | Yes (for TLS) | | | |
| ied_name | string | Name of an Intelligent Electronic Device | Yes | | | |
| originator | string | Origin identification for the device | No | | | |
| model_filename | string | The filename of the server model, without the .server extension | Yes | | | |
| command_ack_timeout_ms | integer | Timeframe (ms) in which enhanced security commands must be acknowledged (Default: 3000) | No | 3000 | | |
| report_buffered_size | integer | Report control blocks buffer size in bytes (Default: 65536) | No | 65536 | | |
| report_unbuffered_size | integer | Unbuffered report control blocks buffer size in bytes (Default: 65513) | No | 65513 | | |

IEC 61850 Server parameters for Signals tab

| Parameter | Type | Description | Required | Default value (when not specified) | Range | |
|--------------|---------|---|----------|---------------------------------------|-------|-----|
| | | | | | Min | Max |
| signal_name | string | User-friendly signal name | Yes | | | |
| device_alias | string | Device alias from a Devices tab | Yes | | | |
| signal_alias | string | Unique alphanumeric name of the signal to be used | Yes | | | |
| enable | boolean | Enabling/disabling of an individual signal | No | 1 | 0 | 1 |
| log | boolean | Allow signal to be logged. If the log is 0 signal will not be logged. If the log is more than 0 signal will be logged | No | 0 | | |

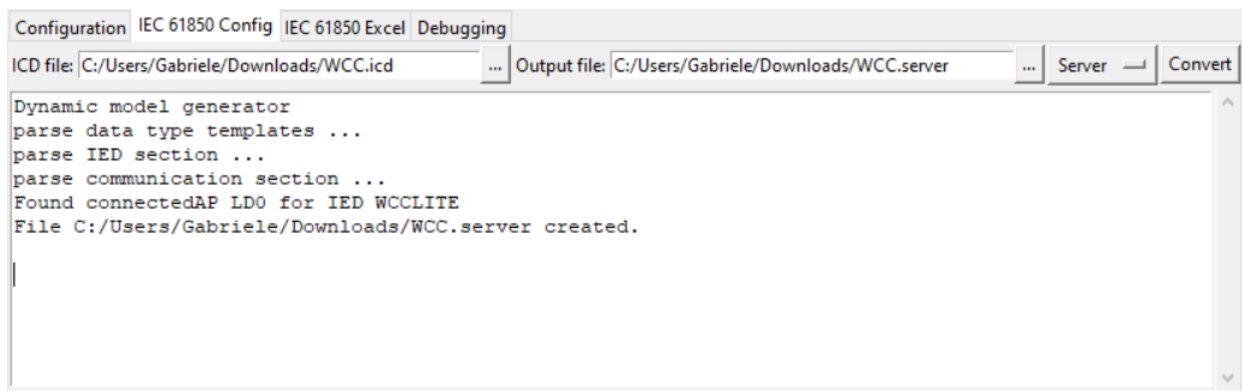
| | | | | | |
|---------------|---------|--|--------------------|-----------|--|
| number_type | string | Number format type (BOOLEAN, FLOAT, INT16, etc.) | Yes | | BOOLEAN, INT8, INT16, INT32, INT64, INT128, INT8U, INT24U, INT32U, FLOAT32, FLOAT64, ENUMERATED, OCTET STRING 64, OCTET STRING 6, OCTET STRING 8, VISIBLE STRING 32, VISIBLE STRING 64, VISIBLE STRING 65, VISIBLE STRING 129, VISIBLE STRING 255, UNICODE STRING 255, TIMESTAMP, QUALITY, CHECK, CODEDENUM, GENERIC BITSTRING, CONSTRUCTED, ENTRY TIME, PHYCOMADDR, CURRENCY, OPTFLDS, TRGOPS |
| ld_instance | string | An instance of a logical device | Yes | | |
| ln_class | string | Logical node class type | Yes | | |
| ln_instance | integer | An instance of a logical node | No | | |
| ln_prefix | string | Prefix of logical node string | No | | |
| cdc | string | Common Data Class (CDC) name | Yes | | SPS, DPS, INS, ACT, ACD, SEC, MV, CMV, WYE, DEL, SEQ, SPG, ING, ASG, ENS, ENG |
| data_object | string | Name of a data object in the dataset | Yes | | |
| da_value | string | Name of a data attribute value node | Yes | | |
| da_fc | string | Functional constrain for data object | Yes | | ST, MX, CO, SP |
| control_model | string | Model of output control | Yes (for commands) | read-only | read-only, direct-with-normal-security, sbo-with-normal-security, direct-with-enhanced-security, sbo-with-enhanced-security |

Device status signals

IEC 61850 has an additional signal which can be configured to show communication status. It is used to indicate if the client device has disconnected from the server (WCC Lite). To configure such a signal for the IEC 61850 protocol, job_todo and tag_job_todo fields with string values are required. For the IEC 61850 server required parameters for the status signal will be **signal_name device_alias, signal_alias, number_type, job_todo** and **tag_job_todo**. Job_todo value must be *device_status* and for tag_job_todo there are 4 variations: communication_status, device_running, device_error, unknown_error. Each signal has 4 possible values and is based on the same logic. If the signal returns the value of 0, it means an unknown error has appeared, 1 – device or protocol connection is on and working properly, 2 – device is off or protocol is disconnected, 3 – error or service is down.

Converting and uploading data model

To use the IEC61850 Server protocol in WCC Lite, the user must upload a data model in a specific format (file extension .server). These data models can be converted from SCL files (.icd, .cid or .scd files). To convert a data model, the user must use WCC Excel Utility. There's a separate tab for this operation as shown in the picture below.



The converted file can be uploaded in the WCC Lite web interface, Protocol Hub section. The current model can be also downloaded on the same page as shown in the picture below.

| PROTOCOL HUB | STATUS | SYSTEM | SERVICES | NETWORK | USERS | LOGOUT (ROOT) |
|---------------|------------------|-----------|----------------------|---------------|-------|---------------|
| CONFIGURATION | IMPORTED SIGNALS | EVENT LOG | PROTOCOL CONNECTIONS | SCRIPT-RUNNER | | |

Protocol configuration

IMPORT PROTOCOL CONFIGURATION

Here you can import Excel configuration file. Up to 1000 signals are allowed. All previous signals will be replaced.

Configuration file: No file chosen

PLC (IEC-61499) Boot file: No file chosen

IEC61850 Client model file: No file chosen

IEC61850 Server model file: No file chosen

DOWNLOAD CONFIGURATION


Current configuration (WCC.xlsx):

Template configurations:

Current IEC 61850 Server model file (WCC.server):

Debugging an IEC 61850 server application

If the configuration for the IEC 61850 Server is set up, a handler for the protocol will start automatically. If the configuration is missing or contains errors, the protocol will not start. It is done intentionally to decrease unnecessary memory usage.

 If the IEC 61850 Server does not work properly (e.g. no communication between devices, data is corrupted, etc.), a user can launch a debug session from the command line interface and find out why the link is not functioning properly.

 To launch a debugging session, a user should stop `iec61850-server` process and run `iec61850-server` command with respective flags as you can see below:

Step 1: Service must be stopped by entering the following command into the wcc-lite:
`/etc/init.d/iec61850-server stop`

Step 2: After the service is stopped it must be started with the preferred configuration file (JSON files found in the `/etc/` folder) and a debug level 7:

`iec61850-server -c /etc/iec61850-server/iec61850-server.json -d7`

Additional output forming options described in the table below.

Step 3: Once the problem is diagnosed normal operations can be resumed with the following command:

`/etc/init.d/iec61850-server start`

`-h [--help]` Show help message

```
-c [--config] arg Configuration file location
-V [--version] Show version
-d [--debug] arg Set Debug level
-r [--redis] Show Redis messages
```

15.3 IEC 61850 Client

WCC Lite can be used as a master station to collect data from IEC 61850 compatible server devices such as protection relays. As relays require fast, secure and responsive interfaces, WCC Lite can be considered as a valid option. For additional security, a user can use encrypted transmission (TLS) or set up a password.

As TCP (TLS) connection can encounter issues and break, automatic reconnection is implemented. After every failed reconnection attempt the fallback delay is doubled starting from 1 second up until 32 seconds. After that connection reestablishment will be attempted every 32 seconds until a successful connection.

Acquiring data via report control blocks

As per the IEC 61850 standard, the report control block controls the procedures that are required for reporting values of data objects from one or more logical nodes to one client. Automatic reporting enables data servers (slave devices) to only send data on its (or its quality) change, thus saving network bandwidth. Instances of report control blocks are configured in the server at configuration time.

Report control blocks send information that is defined in their respective datasets. The dataset is a set of data elements grouped to represent some data group. For example, it is a common practice to group measurements and events into different groups.

A server restricts access to an instance of a report control block to one client at a time. That client exclusively shall own that instance and shall receive reports from that instance of report control blocks. There are two classes of report control blocks defined, each with a slightly different behaviour:

- buffered-report-control-block (BRCB) - internal events (caused by trigger options data-change, quality-change, and data-update) issue immediate sending of reports or buffer the events (to some practical limit) for transmission, such that values of the data object are not lost due to transport flow control constraints or loss of connection. BRCB provides the sequence-of-events (SOE) functionality;
- unbuffered-report-control-block (URCB) - internal events (caused by trigger options data-change, quality-change, and data-update) issue immediate sending of reports on a best efforts basis. If no association exists, or if the transport data flow is not fast enough to support it, events may be lost.

Buffered report control blocks are therefore useful to keep event data, for example, keeping the last known state of a relay switch where a loss of information might lead to confusion and even financial losses. Unbuffered report control blocks are particularly useful for data which is useful only momentarily, e.g. measurements of voltages, current or power. This information can change frequently and old measurements might not reflect the real state of a substation.

To allow multiple clients to receive the same values of data objects, multiple instances of the report control classes shall be made available.

Buffered report control blocks are usually configured to be used by a specific client implementing a well-defined functionality, for example, a SCADA master. The client may know the ObjectReference of the BRCB by configuration or by the use of a naming convention.

Parsing of report control blocks is based on types of Common Data Classes (CDC). Some of these types can have more than one data point of interest. The table below shows what data attributes are supported by various Common Data Classes. To select which data attribute should be used a `da_value` column should be filled with a data attribute name. Common Data Classes consist of data attributes with different Functional Constraints therefore to get the status points of interest correctly the user must fill in the correct value in `da_fc` the column.

IEC 61850 Client-supported data attributes:

| Common Data Class | Function Constraint | Data attributes |
|--------------------------|---------------------|---|
| SPS DPS INS ENS | ST | stVal |
| ACT | ST | general phsA phsB phsC neut |
| ACD | ST | general dirGeneral phsA dirPhsA phsB dirPhsB phsC dirPhsC neut dirNeut |

| | | |
|--------------------------|----|------------------|
| MV | MX | instMag mag |
| CMV | MX | instCVal cVal |
| SAV | MX | instMag |
| SPC DPC INC ENC | ST | stVal |
| BSC ISC | ST | valWTr |
| APC BAC | MX | mxVal |

Some of the data attributes are structures themselves, for example, `mag` attribute is a struct that can hold integer or float values. To select a fitting attribute the user should extend `da_value` parameter with additional attributes, for example, if float magnitude value is to be selected from MV Common Data Class, `da_value` column should be filled with `mag.f` value; if the user intends `cVal` magnitude value in float format from CMV Common Data Class, `da_value` should be filled with `cVal.mag.f` value. See IEC 61850-7-3 for more information about Common Data Classes.

To ensure the integrity of configuration, WCC Lite has additional checks implemented at configuration time. If the report control block (or its dataset) with a predefined ObjectReference doesn't exist, it is considered that IEC 61850 Client has not been configured properly or configuration has been changed in either of IEC 61850 devices and cannot be matched, therefore should be considered invalid.

Controlling remote equipment via commands

The control model provides a specific way to change the state of internal and external processes by a client. The control model can only be applied to data object instances of a controllable Common Data Class (CDC) whose `ctlModel` DataAttribute is not set to status - only. Such data objects can be referred to as control objects. If controls are enabled in an IEC 61850 Server device the user can configure controls by filling the `control_model` column in Excel configuration with a control model (*direct-with-normal-security*, *sbo-with-normal-security*, *direct-with-enhanced-security*, *sbo-with-enhanced-security*) as well as setting functional constraint in `da_fc` column to CO.

Depending on the application, different behaviours of a control object shall be used. Therefore, different state machines are defined. Four cases are defined:

- **Case 1:** Direct control with normal security (direct-operate);
- **Case 2:** SBO control with normal security (operate-once or operate-many);
- **Case 3:** Direct control with enhanced security (direct-operate);
- **Case 4:** SBO control with enhanced security (operate-once or operate-many).

IEC 61850 standard enables the user to plan command transmission - set the timer when the command should be issued. However, as this possibility is rarely used in practice, it is not implemented as of version v1.8 All issued commands are executed immediately.

For more information on the control class model, please consult the IEC 61850-7-2 standard.

If `ctlModel` is read-only, messages from the internal database will be ignored for this point, otherwise, a subscribe callback will be launched to handle commands as soon as they are sent. If the CDC of a signal does not have a means of control, the `ctlModel` parameter is ignored.

Originator identification can be attached to a station so that replies to command requests can be forwarded to only one device. To use this functionality a user should select an origin identifier by filling value in the Excel configuration, originator column. The originator category is always enforced to tell that a remote control command is issued.

Configuring datapoints

To use the IEC 61850 Client in WCC Lite, it has to be configured via an Excel configuration. This configuration contains two Excel sheets where parameters have to be filled in - Devices and Signals tables.

Table IEC 61850 Client parameters for *Devices* tab

| Parameter | Type | Description | Required | Default value (when not specified) | Range | |
|-----------|--------|---------------------------------|----------|---------------------------------------|-------|-----|
| | | | | | Min | Max |
| name | string | User-friendly name for a device | Yes | | | |

| | | | | | | |
|-----------------------|----------------------------|---|--------------------|-----|---------------------|---|
| description | string | Description of a device | No | | | |
| device_alias | string | Alphanumeric string to identify a device | Yes | | | |
| enable | boolean | Enabling/disabling of a device | No | 1 | 0 | 1 |
| protocol | string | Protocol to be used | Yes | | IEC 61850 Client | |
| host | string (IP address format) | The IP address of server device | Yes | | | |
| port | integer | TCP communication port | Yes | 102 | | |
| auth | string | Authorization type | Yes | | none, password, tls | |
| password | string | Authorization password for server device | Yes (for PASSWORD) | | | |
| tls_local_certificate | string | Local certificate for TLS connection | Yes (for TLS) | | | |
| tls_peer_certificate | string | Certificate authority file for TLS connection | Yes (for TLS) | | | |
| tls_private_key | string | A file consisting of the private key for TLS connection | Yes (for TLS) | | | |
| ied_name | string | Name of an Intelligent Electronic Device | Yes | | | |
| originator | string | Origin identifier for the device | No | | | |
| model_filename | string | The filename of the client model uploaded to WCC (must contain .client extension) | Yes | | | |

Table IEC 61850 Client parameters for *Signals* tab

| Parameter | Type | Description | Required | Default value (when not specified) | Range | |
|--------------|---------|---|----------|---------------------------------------|-------|-----|
| | | | | | Min | Max |
| signal_name | string | User-friendly signal name | Yes | | | |
| device_alias | string | Device alias from a Devices tab | Yes | | | |
| signal_alias | string | Unique alphanumeric name of the signal to be used | Yes | | | |
| enable | boolean | Enabling/disabling of an individual signal | No | 1 | 0 | 1 |
| log | boolean | Allow signal to be logged. If the log is 0 signal will not be logged. If the log is more than 0 signal will be logged | No | 0 | | |

| | | | | | | |
|----------------------|-----------------|---|-----|-----------|--|--|
| number_type | string | Number format type | Yes | | BOOLEAN, INT8, INT16, INT32, INT64, INT128, INT8U, INT24U, INT32U, FLOAT32, FLOAT64, ENUMERATED, OCTETSTRING6, OCTETSTRING8, OCTETSTRING64, VISIBLESTRING32, VISIBLESTRING64, VISIBLESTRING65, VISIBLESTRING129, VISIBLESTRING255, UNICODESTRING255, TIMESTAMP, QUALITY, CHECK, CODEDENUM, GENERICBITSTRING, CONSTRUCTED, ENTRYTIME, PHYCOMADDR, CURRENCY, OPTFLDS, TRGOPS | |
| Id_instance | string | An instance of a logical device | Yes | | | |
| In_class | string | Logical node class type | Yes | | | |
| In_instance | integer | An instance of a logical node | No | | | |
| In_prefix | string, integer | Prefix of logical node string | No | | | |
| cdc | string | Common Data Class (CDC) name | Yes | | SPS, DPS, INS, ACT, ACD, SEC, MV, CMV, WYE, DEL, SEQ, SPG, ING, ASG, ENS, ENG | |
| data_object | string | Name of a data object in the dataset | Yes | | | |
| da_value | string | Name of a data attribute value node | Yes | | | |
| da_fc | string | Functional constrain for data object | Yes | | ST, MX, CO, SP, SE | |
| control_model | string | Model of output control | No | read-only | read-only, direct-with-normal-security, sbo-with-normal-security, direct-with-enhanced-security, sbo-with-enhanced-security | |
| dataset | string | Full object reference of a dataset | Yes | | | |
| report_control_block | string | Full object reference of a report control block | Yes | | | |
| intgPd | integer | Integrity period in milliseconds | No | 0 | | |



It should be noted that ACT and ACD messages can only be parsed from the report if either only the 'general' attribute or all attributes attached to all three phases and neutral can be found in the report

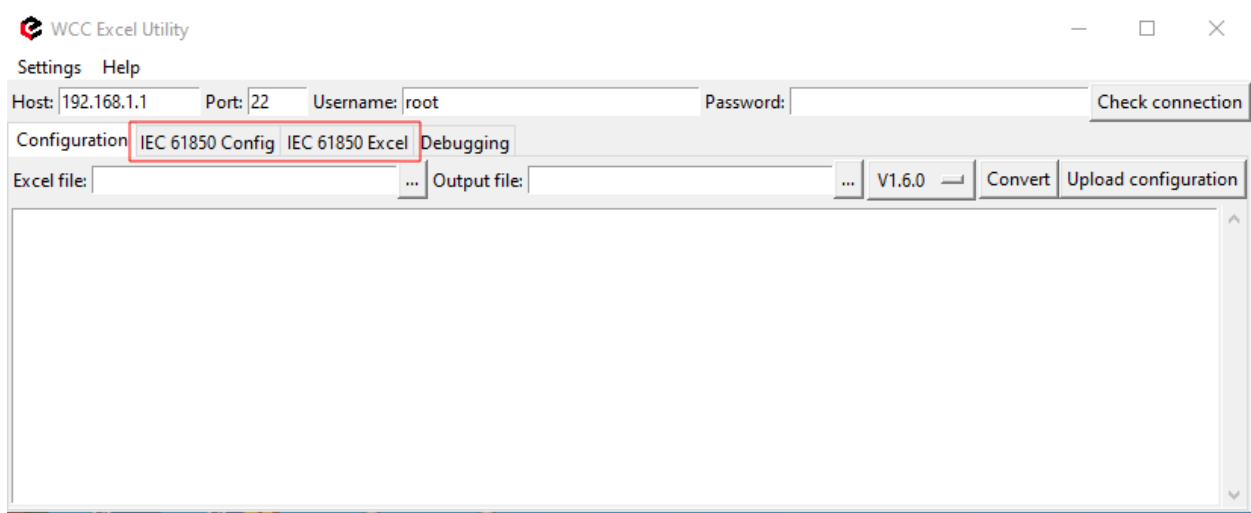
Device status signals

IEC 61850 has an additional signal which can be configured to show communication status. It is used to indicate if the server device has disconnected from the client (WCC Lite). To configure such a signal for the IEC 61850 protocol, job_todo and tag_job_todo fields with string values are required. For IEC 61850 client required parameters for the status signal will be: **signal_name**, **device_alias**, **signal_alias**, **number_type**, **job_todo** and **tag_job_todo**. Job_todo value must be *device_status* and for tag_job_todo there are 4 variations: communication_status, device_running, device_error, unknown_error. Each signal has 4 possible values and is based on the same logic. If the signal returns the value of 0, it means an unknown error has appeared, 1 - device or protocol connection is on and working properly, 2 - device is off or protocol is disconnected, 3 - error or service is down.

Configuration

Configuration of IEC61850 Client for WCC Lite is done via WCC Utility. Elseta WCC Utility has two IEC 61850 selections - IEC 61850 Config and IEC 61850 Excel:

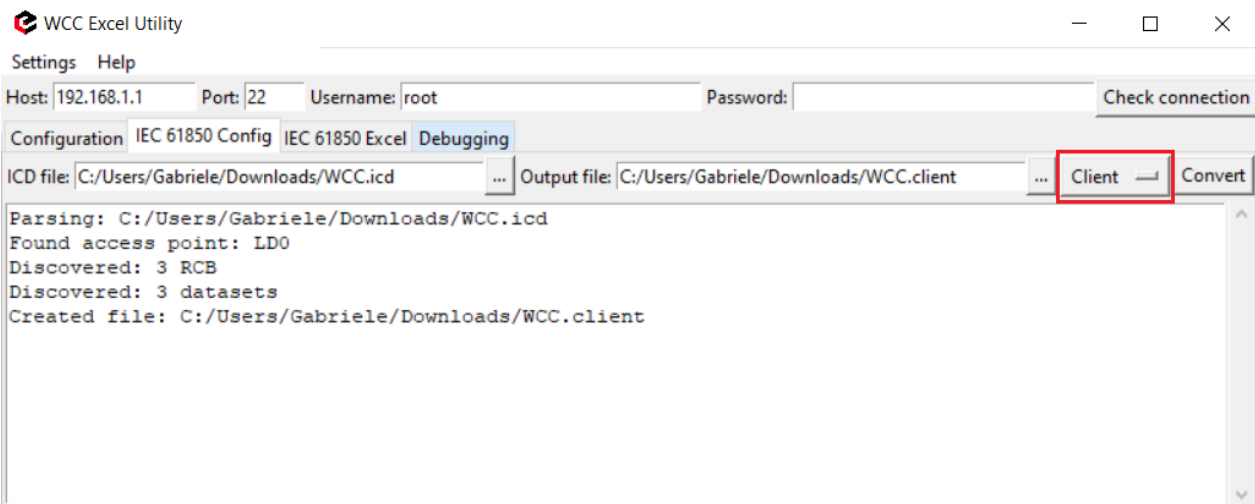
- **IEC 61850 Config** is used to create a configuration model file, which IEC 61850 Client service will use to parse reports from the server.
- **IEC 61850 Excel** is used to generate Excel configuration files which in turn will be used to generate configuration.json via excel-utility.



WCC Utility with IEC61850 selections

Generate model file

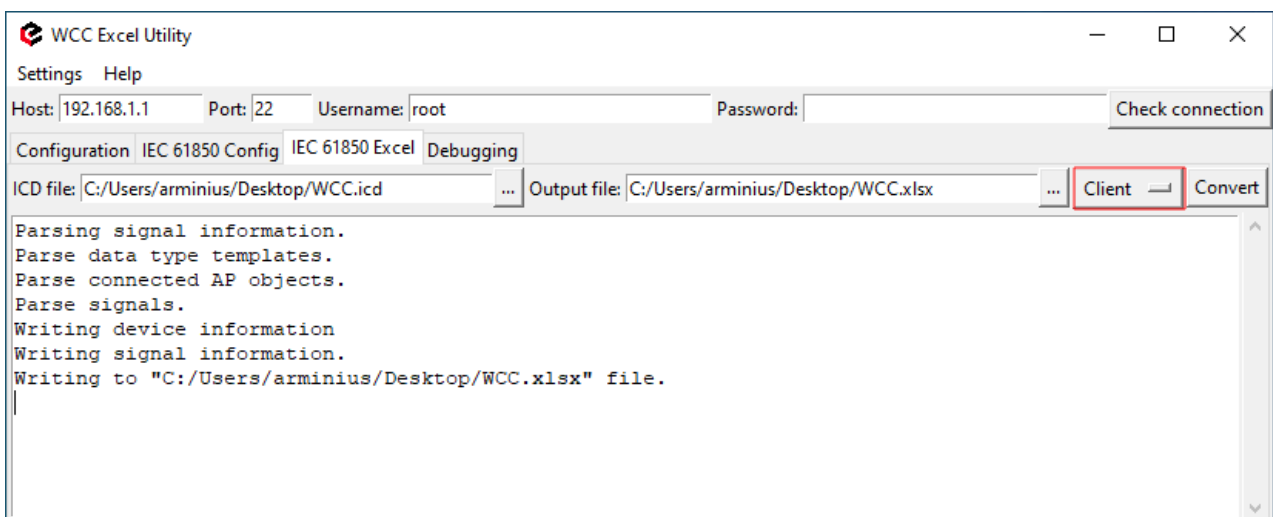
To generate the IEC 61850 Client model file select "Client" in the drop-down selection tab. Then select where to output the generated model and upload a file with extensions .icd, .scd or .cid.



Generate IEC 61850 Client model file

Generate excel file

To generate the IEC 61850 Client Excel file select "Client" in the drop-down selection tab. Then select where to output the generated model upload file with extensions .icd, .scd or .cid.



Generate IEC 61850 Client Excel file

After generating the Excel file additional configuration information must be written in the devices sheet:

- A valid host ip address must be provided.
- An authorization method must be provided (if it is a complex authorization method, additional parameters might be required).
- Model filename must be provided. The model filename must be the same as that was generated one step earlier (The model filename can include an extension, but it is not mandatory).

| I | J | K |
|-----------------|---------------|----------------|
| host | authorization | model_filename |
| 192.168.122.146 | none | WCC |

Excel configuration (Devices sheet)

In the signals sheet, signals which are not used or needed can be removed. Their information might be modified as well.

| A | B | C | D | E | F | G | H | I | J | K |
|------------------|------------------------------|------------------------------|-------------|----------|-------------|-----------|-----|-------------|-------|----------------|
| device alias | signal name | signal alias | ld_instance | ln_class | ln_instance | ln_prefix | cdc | data_object | da_fc | number_type |
| iec 61850 client | LD0_GGIO_1_SPS1_stVal | LD0_GGIO_1_SPS1_stVal | LD0 | GGIO | 1 | | SPS | SPS1 | ST | BOOLEAN |
| iec 61850 client | LD0_GGIO_1_SPS2_stVal | LD0_GGIO_1_SPS2_stVal | LD0 | GGIO | 1 | | SPS | SPS2 | ST | BOOLEAN |
| iec 61850 client | LD0_GGIO_2_SPC1_origin.orCat | LD0_GGIO_2_SPC1_origin.orCat | LD0 | GGIO | 2 | | SPC | SPC1 | ST | INT8 |
| iec 61850 client | LD0_GGIO_2_SPC1_origin.ordnt | LD0_GGIO_2_SPC1_origin.ordnt | LD0 | GGIO | 2 | | SPC | SPC1 | ST | OCTET STRING 6 |
| iec 61850 client | LD0_GGIO_2_SPC1_ctiNum | LD0_GGIO_2_SPC1_ctiNum | LD0 | GGIO | 2 | | SPC | SPC1 | ST | INT8U |
| iec 61850 client | LD0_GGIO_2_SPC1_stSeld | LD0_GGIO_2_SPC1_stSeld | LD0 | GGIO | 2 | | SPC | SPC1 | ST | BOOLEAN |
| iec 61850 client | LD0_GGIO_2_SPC1_stVal | LD0_GGIO_2_SPC1_stVal | LD0 | GGIO | 2 | | SPC | SPC1 | ST | BOOLEAN |
| iec 61850 client | LD0_GGIO_2_SPC1_Oper.ctiVal | LD0_GGIO_2_SPC1_Oper.ctiVal | LD0 | GGIO | 2 | | SPC | SPC1 | CO | BOOLEAN |
| iec 61850 client | LD0_GGIO_2_SPC2_origin.orCat | LD0_GGIO_2_SPC2_origin.orCat | LD0 | GGIO | 2 | | SPC | SPC2 | ST | INT8 |
| iec 61850 client | LD0_GGIO_2_SPC2_origin.ordnt | LD0_GGIO_2_SPC2_origin.ordnt | LD0 | GGIO | 2 | | SPC | SPC2 | ST | OCTET STRING 6 |
| iec 61850 client | LD0_GGIO_2_SPC2_ctiNum | LD0_GGIO_2_SPC2_ctiNum | LD0 | GGIO | 2 | | SPC | SPC2 | ST | INT8U |
| iec 61850 client | LD0_GGIO_2_SPC2_stSeld | LD0_GGIO_2_SPC2_stSeld | LD0 | GGIO | 2 | | SPC | SPC2 | ST | BOOLEAN |
| iec 61850 client | LD0_GGIO_2_SPC2_stVal | LD0_GGIO_2_SPC2_stVal | LD0 | GGIO | 2 | | SPC | SPC2 | ST | BOOLEAN |
| iec 61850 client | LD0_GGIO_2_SPC2_Oper.ctiVal | LD0_GGIO_2_SPC2_Oper.ctiVal | LD0 | GGIO | 2 | | SPC | SPC2 | CO | BOOLEAN |
| iec 61850 client | LD0_GGIO_3_DPS1_stVal | LD0_GGIO_3_DPS1_stVal | LD0 | GGIO | 3 | | DPS | DPS1 | ST | CODEENUM |
| iec 61850 client | LD0_GGIO_3_DPS2_stVal | LD0_GGIO_3_DPS2_stVal | LD0 | GGIO | 3 | | DPS | DPS2 | ST | CODEENUM |
| iec 61850 client | LD0_GGIO_4_DPC1_origin.orCat | LD0_GGIO_4_DPC1_origin.orCat | LD0 | GGIO | 4 | | DPC | DPC1 | ST | INT8 |
| iec 61850 client | LD0_GGIO_4_DPC1_origin.ordnt | LD0_GGIO_4_DPC1_origin.ordnt | LD0 | GGIO | 4 | | DPC | DPC1 | ST | OCTET STRING 6 |
| iec 61850 client | LD0_GGIO_4_DPC1_ctiNum | LD0_GGIO_4_DPC1_ctiNum | LD0 | GGIO | 4 | | DPC | DPC1 | ST | INT8U |
| iec 61850 client | LD0_GGIO_4_DPC1_stSeld | LD0_GGIO_4_DPC1_stSeld | LD0 | GGIO | 4 | | DPC | DPC1 | ST | BOOLEAN |
| iec 61850 client | LD0_GGIO_4_DPC1_stVal | LD0_GGIO_4_DPC1_stVal | LD0 | GGIO | 4 | | DPC | DPC1 | ST | CODEENUM |
| iec 61850 client | LD0_GGIO_4_DPC1_Oper.ctiVal | LD0_GGIO_4_DPC1_Oper.ctiVal | LD0 | GGIO | 4 | | DPC | DPC1 | CO | BOOLEAN |
| iec 61850 client | LD0_GGIO_4_DPC2_origin.orCat | LD0_GGIO_4_DPC2_origin.orCat | LD0 | GGIO | 4 | | DPC | DPC2 | ST | INT8 |
| iec 61850 client | LD0_GGIO_4_DPC2_origin.ordnt | LD0_GGIO_4_DPC2_origin.ordnt | LD0 | GGIO | 4 | | DPC | DPC2 | ST | OCTET STRING 6 |
| iec 61850 client | LD0_GGIO_4_DPC2_ctiNum | LD0_GGIO_4_DPC2_ctiNum | LD0 | GGIO | 4 | | DPC | DPC2 | ST | INT8U |
| iec 61850 client | LD0_GGIO_4_DPC2_stSeld | LD0_GGIO_4_DPC2_stSeld | LD0 | GGIO | 4 | | DPC | DPC2 | ST | BOOLEAN |
| iec 61850 client | LD0_GGIO_4_DPC2_stVal | LD0_GGIO_4_DPC2_stVal | LD0 | GGIO | 4 | | DPC | DPC2 | ST | CODEENUM |
| iec 61850 client | LD0_GGIO_4_DPC2_Oper.ctiVal | LD0_GGIO_4_DPC2_Oper.ctiVal | LD0 | GGIO | 4 | | DPC | DPC2 | CO | BOOLEAN |
| iec 61850 client | LD0_GGIO_5_AnIn1_mag.f | LD0_GGIO_5_AnIn1_mag.f | LD0 | GGIO | 5 | | MV | AnIn1 | MX | FLOAT32 |
| iec 61850 client | LD0_GGIO_5_AnIn2_mag.f | LD0_GGIO_5_AnIn2_mag.f | LD0 | GGIO | 5 | | MV | AnIn2 | MX | FLOAT32 |

Signals sheet

Important! Information such as ld_instance and other data taken directly from SCD configuration files should not be modified, otherwise, access to info of these reports can be broken.

Uploading configuration

First, upload the model configuration file.

CONFIGURATION

IMPORTED SIGNALS

EVENT LOG

PROTOCOL CONNECTIONS

SCRIPT-RUNNER

Protocol configuration

IMPORT PROTOCOL CONFIGURATION

Here you can import Excel configuration file. Up to 1000 signals are allowed. All previous signals will be replaced.

Configuration file:

Choose File No file chosen

Import configuration

PLC (IEC-61499) Boot file:

Choose File No file chosen

Import FBOOT file

IEC61850 Client model file:

Choose File WCC.client

Import client model file

IEC61850 Server model file:

Choose File No file chosen

Import server model file

DOWNLOAD CONFIGURATION

Template configurations:

Download

Uploading model configuration file

After uploading the model configuration file it should appear under the *DOWNLOAD CONFIGURATION* tab.

DOWNLOAD CONFIGURATION

Template configurations:

Download

Current IEC 61850 client model file (WCC.client):

Download

Delete

Uploaded IEC 6180 Client configuration file

Then upload the Excel configuration (same as with every other protocol).

PROTOCOL HUB

STATUS

SYSTEM

SERVICES

NETWORK

USERS

LOGOUT (ROOT)

CONFIGURATION

IMPORTED SIGNALS

EVENT LOG

PROTOCOL CONNECTIONS

Protocol configuration

IMPORT PROTOCOL CONFIGURATION

Here you can import Excel configuration file. Up to 1000 signals are allowed. All previous signals will be replaced.

Configuration file:

Choose File
 WCC.xlsx

Import configuration

PLC (IEC-61499) Boot file:

Choose File
 No file chosen

Import FBOOT file

IEC61850 Client model file:

Choose File
 No file chosen

Import client model file

IEC61850 Server model file:

Choose File
 No file chosen

Import server model file

Uploading excel configuration

After successful configuration upload, both configurations should appear under the*DOWNLOAD CONFIGURATION* tab. If any errors occur during Excel upload, fix them along Excel utility guidelines.

DOWNLOAD CONFIGURATION

Current configuration (WCC.xlsx):

Download

Template configurations:

Download

Current IEC 61850 client model file (WCC.json):

Download

Delete

Uploaded configurations

IEC 61850 Client command line debugging options

- Step 1:** Service must be stopped by entering the following command into the wcclite:
/etc/init.d/iec61850-client stop
- Step 2:** After the service is stopped it must be started with the preferred configuration file (JSON files found in the /etc/ folder) and a debug level 7: **iec61850-client -c /etc/iec61850-client/iec61850-client.json -d7**
 Additional output forming options described in the table below.
- Step 3:** Once the problem is diagnosed normal operations can be resumed with the following command: **/etc/init.d/iec61850-client start**

```

-h [ -help ] Show help message
-c [ -config] arg Configuration file location
-V [ -version] Show version
-d [ -debug] arg Set debugging level
-r [ -redis] Show Redis messages

```

⚠ If the IEC 61850 Client does not work properly (e.g. no communication between devices, data is corrupted, etc.), a user can launch a debug session from the command line interface and find out why the link is not functioning properly.

ℹ To launch a debugging session, a user should stop `iec61850-client` process by running `/etc/init.d/iec61850-client stop` and run `iec61850-client` command with respective flags as was shown above.