IEC 61850 south plugin

IEC 61850 Protocol stack configuration

The IEC 61850 protocol stack configuration specifies communication parameters and is a collection of entries containing information about OSI Transport and OSI Application layers objects.

Each entry is comprised of attributes that describe the object. All the configuration data are structured using JSON.

Each entry shall be mapped with the corresponding configuration function in the chosen implementation protocol library.

Attributes definition

Transport layer

Attributes	Definition	Example	Required
version	version number of the configuration file	2 digits x.y => x = major change, y = minor change	Yes
name	this identifies the protocol stack	iec104client, iec104server, tase2client, tase2server, 61850client, 61850server, etc...	Yes
ied_name	IED name	any non empty string	Yes
connections.srv_ip	IP address to remote IED server	IP address	Yes
connections.port	port number to remote IED server	default = 102	Yes
connections.osi.remote_tsel	remote transport layer address	valid TSelector	No
connections.osi.remote_ssel	remote session layer address	valid SSelector	No
connections.osi.remote_psel	remote presentation layer address	valid PSelector	No
connections.osi.remote_ap_title	remote OSI AP Title	any non empty string following the format "int0,int1,int2,int3", ex: 1,3,9999,23	No
connections.osi. remote_ae_qualifier	remote OSI AE Qualifier	integer	No
connections.osi.local_tsel	local transport layer address	valid TSelector	No
connections.osi.local_ssel	local session layer address	valid SSelector	No
connections.osi.local_psel	local presentation layer address	valid PSelector	No
connections.osi.local_ap_title	local OSI AP Title	any non empty string following the format "int0,int1,int2,int3", ex: 1,3,9999,23	No
connections.osi.local_ae_qualifier	local OSI AE Qualifier	integer	No
connections	array of connections to IED		Yes
connections.tls	Use TLS	false	No

Application Layer

Attributes	Definition	Example	Required
polling_interval	defines the polling interval in ms (0 = no polling)	100	Yes
datasets	Array of datasets		No
datasets. dataset_ref	Dataset reference		Yes

datasets.entries	Array of dataset entries,
	
	each entry is build using this pattern:
	
	iedName+ldInst/prefix+lnClass+lnInst+doName
	
	NB: prefix is optional
		`"entries": [`
		`"simpleIOGenericIO/GGIO1.AnIn1[MX]",`
		`"simpleIOGenericIO/GGIO1.AnIn2[MX]",`
		`"simpleIOGenericIO/GGIO1.AnIn3[MX]"`
		`]`
		
		This entry in the CID file:
		
		`<FCDA ldInst="GenericIO" lnClass="GGIO" fc="MX" lnInst="1" doName="AnIn1" daName="mag.f" />`
		
		should be translated to:
		
		`simpleIOGenericIO/GGIO1.AnIn1[MX]`
			Yes

datasets. dynamic	If dataset is set to dynamic = true then the client would try to create the dataset on the server, default is false (Static)		No

report_subscriptions	Array of report subscriptions		No
report_subscriptions.rcb_ref	Report Control Block Reference		Yes
report_subscriptions.dataset_ref	RCB dataset reference		Yes
report_subscription.trgops	RCB trigger options	"dchg": data_changed "qchg": quality_changed "dupd": data updated "period": data sent periodically "gi": general interrogation	: No
report_subscription.buftm	RCB buffer time in ms, time to wait before sending value after event occured		No
report_subscription.intgpd	RCB integrity period in ms, report is periodically send for the period defined (this requires the trgops to be set to "integrity")		No
report_subscription.gi	Force GI (General Interrogation) when the RCB is enabled		No

Configuration JSON structure

```
{
    "protocol_stack":
    {
        "name": "iec61850client",
        "version": "0.0.1",
        "transport_layer": {
            "ied_name" : "IED1",
            "connections":[
                {
                    "ip_addr":"127.0.0.1",
                    "port": 102,
                    "osi": {
                        "local_ap_title":"1,3,9999.13",
                        "local_ae_qualifier":12,
                        "remote_ap_title":"1,2,1200,15,3",
                        "remote_ae_qualifier":1,
                        "local_psel":"0x12,0x34,0x56,0x78",
                        "local_ssel":"0,1,2,3,4",
                        "local_tsel":"0x00,0x01,0x02",
                        "remote_psel":"0x87,0x65,0x43,0x21",
                        "remote_ssel":"0,1",
                        "remote_tsel":"0x00,0x01"
                    },
                    "tls" : false
                }
            ]
        },
        "application_layer": {
            "polling_interval": 0,
            "datasets": [
                {
                  "dataset_ref": "simpleIOGenericIO/LLN0.AnalogueValues",
                  "entries": [
                    "simpleIOGenericIO/GGIO1.AnIn1[MX]",
                    "simpleIOGenericIO/GGIO1.AnIn2[MX]",
                    "simpleIOGenericIO/GGIO1.AnIn3[MX]"
                  ],
                  "dynamic": true
                },
                {
                  "dataset_ref": "simpleIOGenericIO/LLN0.CustomDataset",
                  "entries": [
                    "simpleIOGenericIO/GGIO1.AnIn1.mag.f[MX]",
                    "simpleIOGenericIO/GGIO1.AnIn2.q[MX]",
                    "simpleIOGenericIO/GGIO1.AnIn3.t[MX]"
                  ],
                  "dynamic": false
                }
            ],
            "report_subscriptions": [
                {
                    "rcb_ref" : "simpleIOGenericIO/LLN0.RP.EventsRCB",
                    "dataset_ref" : "simpleIOGenericIO/LLN0.AnalogueValues",
                    "trgops" : ["data_changed","quality_changed"],
                    "buftm" : 10,
                    "intgpd" : 23,
                    "gi" : true
                },
                {
                    "rcb_ref" : "simpleIOGenericIO/LLN0.RP.EventsRCB",
                    "dataset_ref" : "simpleIOGenericIO/LLN0.AnalogueValues",
                    "trgops" : ["data_changed","quality_changed"],
                    "buftm" : 10,
                    "intgpd" : 23,
                    "gi" : true
                }
            ]
        }
    }
}
```

Exchanged Data:

Attributes	Definition	Example	Required
datapoints	Exchange data object array		Yes
datapoints.pivot_id	Pivot_id used in filters		Yes
datapoints.label	Datapoint label		Yes
datapoints.protocols			Yes
datapoints.protocols.name	Protocol name (iec61850)		Yes
datapoints.protocols.objref	Data object's object reference		Yes
datapoints.protocols.cdc	Pivot CDC class	"SpcTyp"	Yes

Datapoints in exchanged data must be data objects and not attributes, otherwise they won't work, specific data attributes are chosen in dataset configuration

Example:

```
{
   "exchanged_data":{
      "datapoints":[
         {
            "pivot_id":"TS1",
            "label":"TS1",
            "protocols":[
               {
                  "name":"iec61850",
                  "objref":"simpleIOGenericIO/GGIO1.SPCSO1",
                  "cdc":"SpcTyp"
               }
            ]
         },
         {
            "pivot_id":"TS2",
            "label":"TS2",
            "protocols":[
               {
                  "name":"iec61850",
                  "objref":"simpleIOGenericIO/GGIO1.SPCSO2",
                  "cdc":"SpcTyp"
               }
            ]
         },
         {
            "pivot_id":"TS3",
            "label":"TS3",
            "protocols":[
               {
                  "name":"iec61850",
                  "objref":"simpleIOGenericIO/GGIO1.SPCSO3",
                  "cdc":"SpcTyp"
               }
            ]
         },
         {
            "pivot_id":"TS4",
            "label":"TS4",
            "protocols":[
               {
                  "name":"iec61850",
                  "objref":"simpleIOGenericIO/GGIO1.SPCSO4",
                  "cdc":"SpcTyp"
               }
            ]
         },
         {
```

```
            "pivot_id":"TM1",
            "label":"TM1",
            "protocols":[
               {
                  "name":"iec61850",
                  "objref":"simpleIOGenericIO/GGIO1.AnIn1",
                  "cdc":"MvTyp"
               }
            ]
         },
         {
            "pivot_id":"TM2",
            "label":"TM2",
            "protocols":[
               {
                  "name":"iec61850",
                  "objref":"simpleIOGenericIO/GGIO1.AnIn2",
                  "cdc":"MvTyp"
               }
            ]
         },
         {
            "pivot_id":"TM3",
            "label":"TM3",
            "protocols":[
               {
                  "name":"iec61850",
                  "objref":"simpleIOGenericIO/GGIO1.AnIn3",
                  "cdc":"MvTyp"
               }
            ]
         },
         {
            "pivot_id":"TM4",
            "label":"TM4",
            "protocols":[
               {
                  "name":"iec61850",
                  "objref":"simpleIOGenericIO/GGIO1.AnIn4",
                  "cdc":"MvTyp"
               }
            ]
         }
      ]
   }
}
```

PIVOT datapoint representation

This plugin uses the PIVOT datapoint representation, check it out here:

Here are the supported CDC Classes:

CDC Class	Pivot CDC
SPS	SpsTyp
DPS	DpsTyp
INS	InsTyp
ENS	EnsTyp
MV	MvTyp
BSC	BscTyp
SPC	SpcTyp
DPC	DpcTyp

APC	ApcTyp
INC	IncTyp

Monitoring direction:

<Root>.<type>	{CDC}		
<Root>.<type>.q.operatorBlocked	Boolean	false	
<Root>.<type>.q.Source = "substituted"	String	"process"	"process" | "substituted"
<Root>.<type>.q.test	Boolean	false	
<Root>.<type>.q.Validity = "invalid"	String	"good"	"good" | "invalid" | "reserved" | "questionable"
<Root>.<type>.q.Validity = "questionable" <Root>.<type>.q.DetailQuality.oldData = true	String	"good"	"good" | "invalid" | "reserved" | "questionable"
<Root>.<type>.q.Validity = "questionable" <Root>.<type>.q.DetailQuality.overflow = true	Boolean	false	
<Root>.<type>.t.SecondSinceEpoch	Integer		
<Root>.BscTyp.valWTr.posVal	Integer		Int 8
<Root>.BscTyp.valWTr.transInd	Boolean		Boolean
<Root>.Cause.stVal	Integer		See Cause of Transmission
<Root>.ComingFrom	String		
<Root>.Confirmation.stVal	Boolean	false	
<Root>.DpsTyp.stVal	String		intermediate-state | off | on | bad-state
<Root>.Identifier	String		exchanged_data.datapoints.pivot_id
<Root>.MvTyp.mag.f	Float		Float 32
<Root>.MvTyp.mag.i	Integer		Int 32
<Root>.SpsTyp.stVal	Boolean		
<Root>.TmOrg.stVal	String		"genuine" | "substituted"
<Root>.TmValidity.stVal = "invalid"	String	"good"	"good" | "invalid" | "reserved" | "questionable"

Control Direction:

Key	Type	Default Value	Notes
<Root>.Cause.stVal	Integer		See Cause of Transmission
<Root>.ComingFrom	String		This plugin should always use the value "iec104" when converting to pivot
<Root>.Identifier	String		exchanged_data.datapoints.pivot_id
<Root>.Select.stVal	Boolean	false	• 0 is mapped with false, for Execute • 1 is mapped with true, for Select before Execute
<Root>.<type>	{CDC}		exchanged_data.datapoints.pivot_type (see <type> conversion table above)
<Root>.<type>.q.test	Boolean	false	
<Root>.<type>.t.SecondSinceEpoch	Integer		
<Root>.SpcTyp.ctlVal	Boolean		0 or 1
<Root>.DpcTyp.ctlVal	String		intermediate-state | off | on | bad-state
<Root>.IncTyp.ctlVal	Integer		Int 32
<Root>.ApcTyp.ctlVal	Float		Float 32
<Root>.BscTyp.ctlVal	String		stop | lower | higher | reserved

Example of Datapoint:

```
{
   "GTIC":{
      "ComingFrom":"iec61850",
      "SpcTyp":{
         "q":{
            "test":0
         },
         "t":{
            "SecondSinceEpoch":1700566837,
            "FractionOfSecond":15921577
         },
         "ctlVal":1
      },
      "Identifier":"TS1",
      "Select":{
         "stVal":0
      }
   }
}
```

TLS configuration

The IEC61850 standard can also be used with TLS to realize secure and authenticated connections.

Parameters are needed to set up the TLS secured connection:

private_key	client private key	valid private key	YES
own_cert	client certificate	valid certificate	YES
ca_certs	allows to specify the ca certificates if not included in the owner certificate	list of valid certificates	NO
remote_certs	allows to specify the server certificates, so if specified, only these certificates are accepted	list of valid certificates	NO

Fledge's certificate store allows certificates to be stored and used by the south plugins.

```
{
   "private_key":"iec104_client.key",
   "own_cert":"iec104_client.cer",
   "ca_certs":[
      {
         "cert_file":"iec104_ca.cer"
      },
      {
         "cert_file":"iec104_ca2.cer"
      }
   ],
   "remote_certs":[
      {
         "cert_file":"iec104_server.cer"
      }
   ]
}
```

Certificates and keys:

The certificates should be a .cert, .cer, .crt, .pem, .p12, .der file.

The keys should be a .key or .pem file.

Certificate store

Certificates and keys can be added by clicking the “Import” button in the Certificate Store tab in the Fledge GUI.

TLS Parameters

Set the key and certificate values in the TLS parameters section of the plugin configuration.