MT: Python Scripting

Basic Operations

The following basic operations are available for the MQTT Transmission Module and can be executed within any of the available Ignition scopes using the format system.cirruslink.transmision.*operation*

- createConfig(type, name, config, *enabled*, *description*)
  - enabled (boolean) and description (string) are optional calls
  - returns a dictionary that represents the config.json of the specified resource
- deleteConfig(type, name)
  - returns a boolean
- getConfigNames(type, *enabled*)
  - enabled is an optional call
  - returns a list strings
- isConfigEnabled(type, name)
  - returns a boolean
- readConfig(type, name)
  - returns a dictionary that represents the config.json of the specified resource
- setConfigEnabled(type, name, enabled)
  - returns a boolean
- updateConfig(type, name, collisionPolicy, config)
  - returns a dictionary that represents the config.json of the specified resource

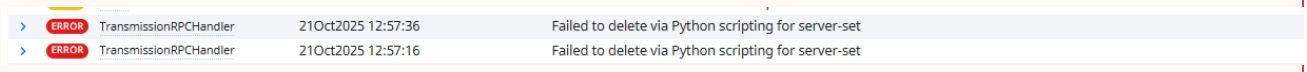
where:

- type
  - string denoting the type of configuration to operate on
- name
  - string denoting the unique name for the configuration resource to operate on
- config
  - dictionary of properties making up the configuration resource to operate on
    - general
    - file
    - history-store
    - record
    - cert-file
    - server
    - server-set
    - transmitter
    - uns-transmitter
- enabled
  - boolean
- collisionPolicy
  - string with options "MergeOverwrite" or "Overwrite"
    - "MergeOverwrite" will replace the existing properties in the resource name with only those included in the call
    - "Overwrite" will replace all properties in the resource name with those in the call

Configuration changes made through scripting do have the same validation as when made through the UI.

For example, attempting to delete a Sets configuring that is being referenced by a Servers configuration will result in no action being performed.

The logs will show an error



Extended Operations

The following extended operations are available for the MQTT Transmission Module and can be executed within any of the available Ignition scopes using the format system.cirruslink.transmision.*operation*

- publish(mqttServerName, topic, payload, qos, retained)

where:

- mqttServerName
  - string
  - name of the server or server set to use from the MQTT Transmission Server configuration.
  - this must match an existing server or server set name in the configuration
- topic

    - string
    - valid MQTT topic to publish on
- payload
    - array
    - bytes representing the MQTT payload
- qos
    - integer
    - the MQTT 'Quality of Service' to use
    - must be either 0,1 or 2
- retained
    - boolean
    - whether or not to publish this message with the retain flag set

Transmission 'type' keys

- general
- file
- history-store
- record
- cert-file
- server
- server-set
- transmitter
- uns-transmitter

Transmission 'general' keys

keyName	dataType	Description
enabled	boolean	Whether or not the configuration resource is enabled
name	string	The unique name for the configuration resource
enableTransmission	boolean	Whether or not Transmission is enabled
auditProfile	string	The optional Ignition Audit Profile to use

Transmission 'file' keys

keyName	dataType	Description
enabled	boolean	Whether or not the configuration resource is enabled
name	string	The unique name for the configuration resource
tagProvider	string	The name of the tag provider where the file control and information tags will be created
tagFolderPath	string	The path to the Tag folder under the specified Tag Provider where the file control and information tags will be created
enableAutoPublishing	boolean	Enable auto-publishing
fileScanRate	integer	The rate to scan the 'Files' directory specified in the 'Publish Files Path' tag for files to publish
fileScanRateTimeUnit	string	Time Unit for the 'File Scan Rate' configuration parameter. Default is SECONDS
groupId	string	To publish files MQTT Transmission uses a configured Transmitter. The Group ID must match an existing Sparkplug Edge Node and cannot be null
edgeNodeId	string	To publish files MQTT Transmission uses a configured Transmitter. The Edge Node ID must match an existing Sparkplug Edge Node and cannot be null
deviceId	string	To publish files MQTT Transmission uses a configured Transmitter. The Device ID that matches an existing Sparkplug Edge Node (Optional)
messageSize	integer	Number of bytes to transfer in one message
messagePacingPeriod	integer	Message Pacing Period in milliseconds
messageAckTimeout	integer	Message acknowledgement timeout in seconds

numberRetries	integer	Number of retries to publish a file or a chunk of a file
submitBasicFileAttributes	boolean	If checked, the metrics will contain basic file attributes (i.e. CreationTime, LastModifiedTime, and LastAccessTime)

Transmission 'history-store' keys

keyName	dataType	Description
enabled	boolean	Whether or not the configuration resource is enabled
name	string	The unique name for the configuration resource
transmissionHistoryStoreType	string	History Store type with options "In-Memory" or "Disk-Backed"
historyMaxSize	integer	Maximum number of megabytes history can use before dropping the data
historyMaxAge	integer	Maximum number of minutes to store history before dropping the data
flushQuantity	integer	Upon reestablishing communication, the maximum number of tags tags to publish at a time
flushPeriod	integer	Upon reestablishing communication, the period to wait in milliseconds between publishes in flushing messages
h2dbDirectory	string	Directory to store the H2 Database in. Applicable for "Disk-Backed" history store only
h2dbPort	integer	TCP Port to connect to H2 Database. Applicable for "Disk-Backed" history store only
dbMaxMetricSize	integer	The Max Metric size allowed in the Store and Forward DB. Applicable for "Disk-Backed" history store only
rollingHistoryBufferEnabled	boolean	Store data in a rolling buffer to cover data loss in keep alive timeout scenario
rollingHistoryPruneInterval	long	Interval at which to attempt to prune the rolling history buffer in seconds
rollingHistoryMaxAge	integer	The maximum age allowed of data in the rolling history buffer in seconds
rollingHistoryPruneQuantity	integer	The number of metrics to prune in a single block

Transmission 'record' keys

keyName	dataType	Description
enabled	boolean	Whether or not the configuration resource is enabled
name	string	The unique name for the configuration resource
tagProvider	string	The name of the tag provider
tagFolderPath	string	The path to the tag folder
recordType	string	Type of record
groupId	string	To publish records MQTT Transmission uses a configured Transmitter. The Group ID must match an existing Sparkplug Edge Node and cannot be null
edgeNodeId	string	To publish records MQTT Transmission uses a configured Transmitter. The Edge Node ID must match an existing Sparkplug Edge Node and cannot be null
deviceId	string	To publish records MQTT Transmission uses a configured Transmitter. The Device ID that matches an existing Sparkplug Edge Node
overridePublishTag	boolean	Overrides the default 'Publish Tag' in the Record defined folder path
publishTagPath	string	The full tag path to the tag to use as the trigger for the Record publish
enableSignature	boolean	Enables a digital signature field on all Records
hashingAlgorithm	string	The hashing algorithm to use when generating the digital signature. Options are SHA_1, SHA_224, SHA_256, SHA_384, SHA_512
signaturePassword	string	The password used to generate the PBE secret key for encrypting the digital signature

Transmission 'cert-file' keys

keyName	dataType	Description
enabled	boolean	Whether or not the configuration resource is enabled
fileName	string	The name of this certificate file or private key
description	string	The description of this certificate file or private key
fileContents	string	The contents of this certificate file or private key

Transmission 'server' keys

keyName	dataType	Description
enabled	boolean	Whether or not the configuration resource is enabled
name	string	The unique name for the configuration resource
url	string	The URL of the MQTT Server to connect to. Should be of the form tcp://mydomain.com:1883 or ssl://mydomain.com:8883
mqttServerConnectionEnabled	boolean	Enable this MQTT Server connection
serverSet	string	The Server Set this MQTT Server is associated with
clientId	string	Client ID for this MQTT connection
username	string	The username for this MQTT connection if required by the MQTT Server
password	string	The password for this MQTT connection if required by the MQTT Server
caCertFile	string	FileName of the CA Certificate file to use
clientCertFile	string	FileName of client certificate file to use
clientPrivateKeyFile	string	FileName of client private key file to use
clientKeyPassword	string	The password associated with the certificate's private key
hostNameVerification	boolean	Enable TLS Hostname Verification
tlsAlpnExtensions	string	The TLS ALPN Extensions to use with this connection (optional)
keepAlive	integer	The MQTT Client keep alive time (in seconds)
randomStartupDelay	string	The Random Startup Delay in milliseconds of the form 'min-max' where min is the low end and max is the high end of the random range
reconnectDelay	long	Transmission Client's reconnect delay in milliseconds
subscribeToLegacyStateTopic	boolean	Transmission Client will subscribe to a legacy state topic (e.g. STATE/IamHost) for primary host ID notifications
dataFormatTypeWrapper	string	The format of the data to send. Options are "Sparkplug_B_v1_0_Protobuf" or "Sparkplug_B_v1_0_JSON"
rpcClientId	string	Client ID for this RPC connection
rpcUsername	string	The username for RPC connection if required by the MQTT Server
rpcPassword	string	The password for RPC connection if required by the MQTT Server
rpcCaCertFile	string	FileName of the CA Certificate file to use
rpcClientCertFile	string	FileName of client certificate file to use
rpcClientPrivateKeyFile	string	FileName of client private key file to use
rpcClientKeyPassword	string	The password associated with the certificate's private key
rpcHostnameVerification	boolean	Enable TLS Hostname Verification
rpcTlsAplnExtensions	string	The TLS ALPN Extensions to use with the RPC connection

Transmission 'server-set' keys

keyName	dataType	Description
enabled	boolean	Whether or not the configuration resource is enabled
name	string	The unique name for the configuration resource
primaryHostId	string	Primary Host ID of the backend application the MQTT clients in MQTT Transmission should remain connected to
rpcClientEnabled	boolean	Enable the RPC MQTT Client which is used for publishing from Ignition Python scripts
autoReconnectRpcClient	boolean	If set, the RPC client will automatically reconnect to the server otherwise will reconnect at publish time
randomizeServerConnections	boolean	Whether or not to randomly connect to each server in the set. If true, the server to connect to will be randomly selected. Otherwise, it connects to the next server in the list

Transmission 'transmitter' keys

keyName	dataType	Description
enabled	boolean	Whether or not the configuration resource is enabled
name	string	The unique name for the configuration resource
tagPath	string	A path to the root folder where the tag tree starts
tagPacingPeriod	long	The waiting period in milliseconds after an initial tag change event before publishing all changed tags
serverSet	string	The name of the Server Set to use with this Transmitter
discoveryDelay	long	The Discovery Delay in milliseconds
historyStore	string	The name of the History Store to use with this Transmitter
enableStoreForwardByDefault	boolean	Enable store and forward by default on all tags
inOrderHistory	boolean	Flush history in-order (synchronously) before live data resumes
alarmEventEnable	boolean	Whether or not to publish alarm events on tags via MQTT
alarmJournalName	string	The pre-configured alarm journal name to use for alarm reconciliation
aliasedTags	boolean	Use aliases for tag names to optimize payload sizes when publishing data. Not supported when publishing UDTs
compressionTypeWrapper	string	The algorithm to use for compressing payloads before publishing. Options are "NONE" "DEFLATE" or "GZIP"
convertUDTs	boolean	Converts UDT members to normal Tags before publishing
deviceLevelUdtsAsDevices	boolean	Treat 'device level UDTs as Sparkplug devices'. ConvertUDTs must be true
publishUdtDefinitions	boolean	Publish UDT Definitions in BIRTH
optimizeUdts	boolean	Optimizes UDT payload sizes in NDATA and DDATA payloads
blockCommands	boolean	Block incoming commands (writes) to Edge Node and Device Tags
validateSecurityContext	boolean	Whether or not to validate the Security Context in write commands
securityContextHashingAlgorithm	string	The hashing algorithm to use when decrypting the Security Context. Options are SHA_1, SHA_224, SHA_256, SHA_384, SHA_512
securityContextHashingPassword	string	The hashing password to use when decrypting the Security Context
publishBED	boolean	Publish data using BED format instead of Sparkplug format
hcpId	string	The HCP ID of the Transmitter for BED publishing
echoRequestTimeout	integer	The Echo Request Timeout of the Transmitter for BED publishing - set to zero to disable
groupId	string	An ID representing a logical grouping of Edge Nodes and Devices (optional)

edgeNodeId	string	An ID representing an Edge or Network (EoN) Node (optional)
deviceId	string	An ID representing a Device (optional)
filteredProperties	string	A semicolon delimited list of Tag properties to filter/block from being published
rebirthDebounceDelay	long	The amount of time to delay before processing Rebirth NCMD requests after one has been processed (in milliseconds)
includeSparkplugDataTypes	boolean	Whether or not to include Sparkplug DataTypes for Metrics in Sparkplug DATA payloads
useCirrusEncoder	boolean	Whether to use the Cirrus Link Encoder which will use special encoding for non-Sparkplug supported PropertyDataTypes
birthSettleDelay	long	The amount of time to wait after setting the BIRTH timestamp before grabbing the cached BIRTH payloads
calculatedSparkplugIds	string	Sparkplug Id created using Ignition tagpath
sendAllProps	boolean	Send all properties, including non-default properties, in Sparkplug BIRTH messages
reconciliationWindow	long	The reconciliation window in milliseconds

Transmission 'uns_transmitter' keys

keyName	dataType	Description
enabled	boolean	Whether or not the configuration resource is enabled
name	string	The unique name for the configuration resource
tagProvider	string	The Name of the tag provider
tagPath	string	A path to the root folder where the tag tree starts
serverSet	string	The name of the Server Set to use with this Transmitter
discoveryDelay	long	The Discovery Delay in milliseconds
propsQos	integer	The MQTT Quality of Service to use for 'properties' messages
propsRetain	boolean	Whether or not to set the MQTT retain flag to true for 'properties' messages
dataQos	integer	The MQTT Quality of Service to use for 'data' messages
dataRetain	boolean	Whether or not to set the MQTT retain flag to true for 'data' messages
historyStore	string	The name of the History Store to use with this Transmitter
enableStoreForwardByDefault	boolean	Enable store and forward by default on all tags
inOrderHistory	boolean	Flush history in-order (synchronously) before live data resumes
convertUDTs	boolean	Converts UDT members to normal Tags before publishing
deviceLevelUdtsAsDevices	boolean	Treat 'device level UDTs as Sparkplug devices'. ConvertUDTs must be true
publishUdtDefinitions	boolean	Publish UDT Definitions in BIRTH
optimizeUdts	boolean	Optimizes UDT payload sizes in NDATA and DDATA payloads
filteredProperties	string	A semicolon delimited list of Tag properties to filter/block from being published
rebirthDebounceDelay	long	The amount of time to delay before processing Rebirth NCMD requests after one has been processed (in milliseconds)
includeSparkplugDataTypes	boolean	Whether or not to include Sparkplug DataTypes for Metrics in Sparkplug DATA payloads
birthSettleDelay	long	The amount of time to wait after setting the BIRTH timestamp before grabbing the cached BIRTH payloads
sendAllProps	boolean	Send all properties, including non-default properties, in .props messages

Sample Scripts

Cirrus Link provides these scripts as examples only, they are not supported or guaranteed to meet any particular functionality. Cirrus Link cannot provide any assistance to modify these scripts to meet a particular need.

```
# Cirrus Link provides these scripts as examples only, they are not supported or guaranteed to meet any
particular functionality. Cirrus Link cannot provide any assistance to modify these scripts to meet a
particular need.

# Create Transmitter One
transmitterProps = {}
transmitterProps["tagProvider"] = "default"
transmitterProps["tagPath"] = ""
transmitterProps["tagPacingPeriod"] = 1000
transmitterProps["serverSet"] = "Default"
transmitterProps["discoveryDelay"] = 0
transmitterProps["aliasedTags"] = False
transmitterProps["compressionTypeWrapper"] = "NONE"
transmitterProps["blockCommands"] = False
transmitterProps["convertUdts"] = True
transmitterProps["publishUdtDefinitions"] = True
transmitterProps["optimizeUdts"] = True
transmitterProps["cacheBirthsEnabled"] = False
#transmitterProps["historyStore"] = None
transmitterProps["enableStoreForwardByDefault"] = True
transmitterProps["inOrderHistory"] = False
transmitterProps["groupId"] = "Test1G"
transmitterProps["edgeNodeId"] = "Test1E"
transmitterProps["deviceId"] = None
transmitterProps["filteredProperties"] = "accessRights;clampMode;deadband;deadbandMode;formatString;
historicalDeadband;historicalDeadbandMode;historicalDeadbandStyle;historyEnabled;historyMaxAge;
historyMaxAgeUnits;historyProvider;historySampleRate;historySampleRateUnits;historyTagGroup;historyTimeDeadband;
historyTimeDeadbandUnits;opcItemPath;opcServer;permissionModel;rawHigh;rawLow;sampleMode;scaleFactor;scaleMode;
scaledHigh;scaledLow;tagGroup;valueSource;expression;expressionType;ConfiguredTagPath;eventScripts;
readPermissions;writePermissions;eventScripts"
transmitterProps["rebirthDebounceDelay"] = 5000
system.cirruslink.transmission.createConfig("transmitter", "CustomOne", transmitterProps, True, "My First
Custom Transmitter")

# MQTT Transmission Update Existing Transmitter using MergeOverwrite
transmitterProps = {}
transmitterProps["groupId"] = "Test2G"
transmitterProps["edgeNodeId"] = "Test2E"
system.cirruslink.transmission.updateConfig(“transmitter”, "CustomOne", “MergeOverwrite”, transmitterProps)

# Delete Config
system.cirruslink.transmission.deleteConfig("transmitter", "CustomOne")
```