ME: Python Scripting

The following API calls are available for the MQTT Engine Module and can be executed within any of the available Ignition scopes.

- publish
- createConfig
- readConfig
- updateConfig
- deleteConfig

There are sample API scripts available for each API call.

APIs

- **system.cirruslink.engine.publish(mqttServerName, topic, payload, qos, retained) void**
    - Parameters
        - String mqttServerName
            - The name of the server connection to use in the MQTT Engine Server configuration. This must match an existing server name in the configuration.
        - String topic
            - The MQTT topic to publish on. This must be a valid MQTT topic
        - byte[] payload
            - An array of bytes representing the MQTT payload
        - int qos
            - The MQTT 'Quality of Service' to use. This must be either 0, 1, or 2
        - boolean retained
            - Whether or not to publish this message with the retained flag set
    - Return Value
        - None
- **system.cirruslink.engine.createConfig(type, props) void**
    - Parameters
        - String type
            - The type of configuration to create with valid options:
                - Cert Files
                - Servers
                - Def Namespace Filters
                - Def Namespace
                - Cust Namespace
        - Map props
            - Dictionary of properties making up the configuration with the following keys and datatypes:
                - Cert Files
                    - Id (long)
                    - Name (str)
                    - FileContents (str)
                    - Description (str)
                - Servers
                    - Id (long)
                    - Name (str)
                    - Enabled (bool)
                    - Url (str)
                    - Username (str)
                    - Password (str)
                    - CaCertFileId (long)
                    - ClientCertFileId (long)
                    - ClientPrivateKeyFileId (long)
                    - ClientKeyPassword (str)
                    - HostnameVerification (bool)
                    - TlsAlpnExtensions (str)
                    - ClientId (str)
                    - CleanSession (bool)
                    - KeepAlive (int)
                    - FilteredNamespaces (str)
                    - EnableLegacyStateMessages (bool)
                    - LegacyStateUsername (str)
                    - LegacyStatePassword (str)
                    - LegacyStateClientId (str)
                    - LegacyStateCaCertFileId (long)
                    - LegacyStateClientCertFileId (long)
                    - LegacyStateClientPrivateKeyFileId (long)
                    - LegacyStateClientKeyPassword (str)
                    - LegacyStateHostnameVerification (bool)
                    - LegacyStateTlsAlpnExtensions (str)
                - Def Namespace Filters
                    - Id (long)
                    - NamespaceId (long)

                        - GroupId (str)
                        - EdgeNodeId (str)
                    - Def Namespace
                        - Id (long)
                        - Name (str) "Elecsys", "Sparkplug A", Sparkplug B", or "Xirgo"
                        - Enabled (bool)
                    - Cust Namespace
                        - Id (long)
                        - Name (str)
                        - Subscription (str)
                        - QoS1 (bool)
                        - RootFolder (str)
                        - TagName (str)
                        - JsonPayload (bool)
                        - Charset (str)
                        - WritableTags (bool)
        - Return Value
            - A map representing the newly created configuration after it has been written to the database. (added 4.0.16)
- **system.cirruslink.engine.readConfig(type) List**
    - Parameters
        - String type
            - The type of configuration to read with valid options:
                - General
                - Cert Files
                - Servers
                - Def Namespace Files
                - Def Namespace Filters
                - Def Namespace
                - Cust Namespace
    - Return Value
        - A List of Dictionaries of properties making up the configurations with the following keys and datatypes:
            - General
                - Id (long)
                - Enabled (bool)
                - PrimaryHostEnabled (bool)
                - PrimaryHostId (str)
                - BlockNodeCommands (bool)
                - BlockDeviceCommands (bool)
                - BlockTagPropertyChanges (bool)
                - StoreHistoricalEvents (bool)
                - EnableLatching (bool)
                - LatchTimeout (long)
                - LatchTagList (str)
                - FilteredProperties (str)
                - EnableBdSeqChecking (bool)
                - CustomProperties (str)
                - EnablePrimaryHostSubscriptions (bool)
                - MetricTimestampValidation (bool)
                - IncludeSecurityContext (bool)
                - SecurityContextHashingAlgorithm (str) "SHA_1", "SHA_224", "SHA_256", "SHA_384" or "SHA_512"
                - SecurityContextHashingPassword (str)
                - PrimaryHostControlTagList (str)
                - ReorderingTimeout (long)
            - Cert Files
                - Id (long)
                - Name (str)
                - FileContents (str)
                - Description (str)
            - Servers
                - Id (long)
                - Name (str)
                - Enabled (bool)
                - Url (str)
                - Username (str)
                - Password (str)
                - CaCertFileId (long)
                - ClientCertFileId (long)
                - ClientPrivateKeyFileId (long)
                - ClientKeyPassword (str)
                - HostnameVerification (bool)
                - TlsAlpnExtensions (str)
                - ClientId (str)
                - CleanSession (bool)
                - KeepAlive (int)
                - FilteredNamespaces (str)
                - EnableLegacyStateMessages (bool)

        - LegacyStateUsername (str)
        - LegacyStatePassword (str)
        - LegacyStateClientId (str)
        - LegacyStateCaCertFileId (long)
        - LegacyStateClientCertFileId (long)
        - LegacyStateClientPrivateKeyFileId (long)
        - LegacyStateClientKeyPassword (str)
        - LegacyStateHostnameVerification (bool)
        - LegacyStateTlsAlpnExtensions (str)
    - Def Namespace Files
        - Id (long)
        - IgnoreFiles (bool)
        - FileHostType (str) "ACTIVE" or "PASSIVE"
        - BaseFileDirectory (str)
        - AddTopicTokens (bool)
        - FileStoringPolicy (str)
    - Def Namespace Filters
        - Id (long)
        - NamespaceId (long)
        - GroupId (str)
        - EdgeNodeId (str)
    - Def Namespace
        - Id (long)
        - Name (str) "Elecsys", "Sparkplug A", Sparkplug B", or "Xirgo"
        - Enabled (bool)
    - Cust Namespace
        - Id (long)
        - Name (str)
        - Subscription (str)
        - QoS1 (bool)
        - RootFolder (str)
        - TagName (str)
        - JsonPayload (bool)
        - Charset (str)
        - WritableTags (bool)

- **system.cirruslink.engine.updateConfig(type, id, collisionPolicy, props) void**
    - Parameters
        - String type
            - The type of configuration to update with valid options:
                - General
                - Cert Files
                - Servers
                - Def Namespace Files
                - Def Namespace Filters
                - Def Namespace
                - Cust Namespace
        - Long id
            - The ID of the record to update
        - String collisionPolicy
            - "MergeOverwrite" or "Overwrite"
                - "MergeOverwrite" will replace the existing properties in the record ID with only those included in the call
                - "Overwrite" will replace all properties in the record ID with those in the call
        - Map props
            - Dictionary of properties making up the configuration with the following keys and datatypes:
                - General
                    - Id (long)
                    - Enabled (bool)
                    - PrimaryHostEnabled (bool)
                    - PrimaryHostId (str)
                    - BlockNodeCommands (bool)
                    - BlockDeviceCommands (bool)
                    - BlockTagPropertyChanges (bool)
                    - StoreHistoricalEvents (bool)
                    - EnableLatching (bool)
                    - LatchTimeout (long)
                    - LatchTagList (str)
                    - FilteredProperties (str)
                    - EnableBdSeqChecking (bool)
                    - CustomProperties (str)
                    - EnablePrimaryHostSubscriptions (bool)
                    - MetricTimestampValidation (bool)
                    - IncludeSecurityContext (bool)
                    - SecurityContextHashingAlgorithm (str) "SHA_1", "SHA_224", "SHA_256", "SHA_384" or "SHA_512"
                    - SecurityContextHashingPassword (str)
                    - PrimaryHostControlTagList (str)
                    - ReorderingTimeout (long)

        - Cert Files
            - Id (long)
            - Name (str)
            - FileContents (str)
            - Description (str)
        - Servers
            - Id (long)
            - Name (str)
            - Enabled (bool)
            - Url (str)
            - Username (str)
            - Password (str)
            - CaCertFileId (long)
            - ClientCertFileId (long)
            - ClientPrivateKeyFileId (long)
            - ClientKeyPassword (str)
            - HostnameVerification (bool)
            - TlsAlpnExtensions (str)
            - ClientId (str)
            - CleanSession (bool)
            - KeepAlive (int)
            - FilteredNamespaces (str)
            - EnableLegacyStateMessages (bool)
            - LegacyStateUsername (str)
            - LegacyStatePassword (str)
            - LegacyStateClientId (str)
            - LegacyStateCaCertFileId (long)
            - LegacyStateClientCertFileId (long)
            - LegacyStateClientPrivateKeyFileId (long)
            - LegacyStateClientKeyPassword (str)
            - LegacyStateHostnameVerification (bool)
            - LegacyStateTlsAlpnExtensions (str)
        - Def Namespace Files
            - Id (long)
            - IgnoreFiles (bool)
            - FileHostType (str) "ACTIVE" or "PASSIVE"
            - BaseFileDirectory (str)
            - AddTopicTokens (bool)
            - FileStoringPolicy (str)
        - Def Namespace Filters
            - Id (long)
            - NamespaceId (long)
            - GroupId (str)
            - EdgeNodeId (str)
        - Def Namespace
            - Id (long)
            - Name (str) "Elecsys", "Sparkplug A", Sparkplug B", or "Xirgo"
            - Enabled (bool)
        - Cust Namespace
            - Id (long)
            - Name (str)
            - Subscription (str)
            - QoS1 (bool)
            - RootFolder (str)
            - TagName (str)
            - JsonPayload (bool)
            - Charset (str)
            - WritableTags (bool)
    - Return Value
        - A map representing the updated configuration after it has been written to the database. (added 4.0.16)
- **system.cirruslink.engine.deleteConfig(type, id) void**
    - Parameters
        - String type
            - The type of configuration to delete with valid options:
                - Cert Files
                - Servers
                - Def Namespace Filters
                - Def Namespace
                - Cust Namespace
        - Long id
            - The ID of the record to delete
    - Return Value
        - True if the record was successfully deleted otherwise False is returned (added 4.0.16)

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.
# MQTT Transmission Create Transmitter

# Create new server
serverProps = {}
serverProps["Name"] = "My Server"
serverProps["Enabled"] = True
serverProps["Url"] = "ssl://192.0.2.1:8883"
serverProps["Username"] = "MyUserName"
serverProps["Password"] = "MyPassword"
serverProps["KeepAlive"] = "30"
system.cirruslink.engine.createConfig("Servers", serverProps)

# Read server configuration
savedServers = system.cirruslink.engine.readConfig("Servers")

# Find server Id
for config in savedServers:
        if config ["Name"] == "My Server":
        serverId = config["id"]

# Update server properties using MergeOverwrite
serverProps = {}
serverProps["KeepAlive"] = "60"
system.cirruslink.engine.createConfig("Servers", serverId, "MergeOverwrite", serverProps)

# Delete server
system.cirruslink.engine.deleteConfig("Servers", serverId)
```