

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](#)
- [readAlarms](#) (added 4.0.25)
- [clearAlarms](#) (added 4.0.25)
- [clearAllAlarms](#) (added 4.0.25)

There are [sample API scripts](#) available for each API call.



Configuration changes made through scripting do not have the same validation as when made through the UI.

For example, deleting a Sets configuring that is being referenced by a Servers configuration will result in the UI showing an Internal Error. Follow the instructions [here](#) to recover the configuration.

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
 - Sets
 - Def Namespace Filters
 - Def Namespace
 - Cust Namespace
 - Cust Namespace Convert (added 4.0.26)
 - Namespace Sets (added 4.0.26)
 - Map props
 - Dictionary of properties making up the configuration with the following keys and datatypes:
 - Cert Files
 - Id (long)(cannot be null)
 - Name (str)
 - FileContents (str)
 - Description (str)
 - Servers
 - Id (long)(cannot be null)
 - Name (str)
 - Enabled (bool)
 - Url (str)
 - ServerSetId (long)
 - 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)
- Sets
 - Id (long)(cannot be null)(added 4.0.22)
 - Name (str)(added 4.0.22)
 - Description (str)(added 4.0.22)
 - PrimaryHostEnabled (bool)(added 4.0.22)
 - PrimaryHostId (str)(added 4.0.22)
- Def Namespace Filters
 - Id (long)(cannot be null)
 - NamespaceId (long)
 - GroupId (str)
 - EdgeNodeId (str)
- Def Namespace
 - Id (long)(cannot be null)
 - Name (str) "Elecsys", "Sparkplug A", "Sparkplug B", or "Xirgo"
 - Enabled (bool)
- Cust Namespace
 - Id (long)(cannot be null)
 - Name (str)
 - Subscription (str)
 - QoS1 (bool)
 - RootFolder (str)
 - TagName (str)
 - JsonPayload (bool)
 - Charset (str)
 - WritableTags (bool)
 - NumbersAsFloats(bool)(added 4.0.28)
- Cust Namespace Convert (added 4.0.26)
 - Id (long)(cannot be null)
 - CustomNamespaceId (long)
 - Regex (bool)
 - OrderIndex (integer)
 - Source (str)
 - Replacement (str)
- Namespace Sets (added 4.0.26)
 - Id(long)(cannot be null)
 - ServerSetId (long)
 - NamespaceId (long)
- 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
 - Sets
 - Def Namespace Files
 - Def Namespace Filters
 - Def Namespace
 - Cust Namespace
 - Cust Namespace Convert (added 4.0.26)
 - Namespace Sets (added 4.0.26)
 - 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)(deprecated 4.0.22)
 - PrimaryHostId (str)(deprecated 4.0.22)
 - BlockNodeCommands (bool)
 - BlockDeviceCommands (bool)
 - BlockTagPropertyChanges (bool)
 - StoreHistoricalEvents (bool)

- EnableLatching (bool)
- LatchTimeout (long)
- LatchTagList (str)
- FilteredProperties (str)
- EnableBdSeqChecking (bool)
- CustomProperties (str)
- MetricTimestampValidation (bool)
- IncludeSecurityContext (bool)
- SecurityContextHashingAlgorithm (str) "SHA_1", "SHA_224", "SHA_256", "SHA_384" or "SHA_512"
- SecurityContextHashingPassword (str)
- PrimaryHostControlTagList (str)
- ReorderingTimeout (long)
- AlarmEventEnable (bool)(added 4.0.23)
- AlarmDisplayPath (str) (added 4.0.30) "IDS_AND_EDGE", "EDGE" or "ENGINE"
- H2dbDirectory (str)(added 4.0.25)
- H2dbPort (int)(added 4.0.23)
- UseCirrusEncoder (bool)(added 4.0.30)
- Cert Files
 - Id (long)
 - Name (str)
 - FileContents (str)
 - Description (str)
- Servers
 - Id (long)
 - Name (str)
 - Enabled (bool)
 - Url (str)
 - ServerSetId (long)
 - 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)
- Sets
 - Id (long)(added 4.0.22)
 - Name (str)(added 4.0.22)
 - Description (str)(added 4.0.22)
 - PrimaryHostEnabled (bool)(added 4.0.22)
 - PrimaryHostId (str)(added 4.0.22)
- 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)
 - NumbersAsFloats(bool)(added 4.0.28)
- Cust Namespace Convert (added 4.0.26)
 - Id (long)(cannot be null)
 - CustomNamespaceId (long)
 - Regex (bool)
 - OrderIndex (integer)
 - Source (str)
 - Replacement (str)
- Namespace Sets (added 4.0.26)
 - Id(long)(cannot be null)
 - ServerSetId (long)
 - NamespaceId (long)
- **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
 - Sets
 - Def Namespace Files
 - Def Namespace Filters
 - Def Namespace
 - Cust Namespace
 - Cust Namespace Convert (added 4.0.26)
 - Namespace Sets (added 4.0.26)
 - 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)(deprecated 4.0.22)
 - PrimaryHostId (str)(deprecated 4.0.22)
 - 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)
 - AlarmEventEnable (bool)(added 4.0.23)
 - AlarmDisplayPath (str) "IDS_AND_EDGE", "EDGE" or "ENGINE"
 - H2dbDirectory (str)(added 4.0.25)
 - H2dbPort (int)(added 4.0.23)
 - UseCirrusEncoder (bool)(added 4.0.30)
 - Cert Files
 - Id (long)
 - Name (str)
 - FileContents (str)
 - Description (str)
 - Servers
 - Id (long)
 - Name (str)
 - Enabled (bool)
 - Url (str)
 - ServerSetId (long)

- 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)
- Sets
 - Id (long)(added 4.0.22)
 - Name (str)(added 4.0.22)
 - Description (str)(added 4.0.22)
 - PrimaryHostEnabled (bool)(added 4.0.22)
 - PrimaryHostId (str)(added 4.0.22)
- 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)
 - NumbersAsFloats(bool)(added 4.0.28)
- Cust Namespace Convert (added 4.0.26)
 - Id (long)(cannot be null)
 - CustomNamespaceId (long)
 - Regex (bool)
 - OrderIndex (integer)
 - Source (str)
 - Replacement (str)
- Namespace Sets (added 4.0.26)
 - Id(long)(cannot be null)
 - ServerSetId (long)
 - NamespaceId (long)
- 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
 - Sets
 - Def Namespace Filters
 - Def Namespace
 - Cust Namespace

- Cust Namespace Convert (added 4.0.26)
 - Namespace Sets (added 4.0.26)
- 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)
- **system.cirruslink.engine.readAlarms() void (added 4.0.25)**
 - Parameters
 - None
 - Return Value
 - A list of alarm maps
- **system.cirruslink.engine.clearAlarm(id) void (added 4.0.25)**
 - Parameters
 - String id
 - The ID of the alarm to delete
 - Return Value
 - True if the alarm was successfully deleted otherwise False is returned
- **system.cirruslink.engine.clearAllAlarms() void (added 4.0.25)**
 - Parameters
 - None
 - Return Value
 - True if the alarms were successfully deleted otherwise False is returned

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 new server set
serverProps = {}
serverProps["Name"] = "Server Set 1"
serverProps["Description"] = ""
serverProps["PrimaryHostEnabled"] = True
serverProps["PrimaryHostId"] = "MyPrimaryHost"
system.cirruslink.engine.createConfig("Sets", serverProps)

# Find server set Id
savedSets = system.cirruslink.engine.readConfig("Sets")
for config in savedSets:
    if config ["Name"] == "Server Set 1":
        serverSetId = config["id"]

# Create new server
serverProps = {}
serverProps["Name"] = "My Server"
serverProps["Enabled"] = True
serverProps["Url"] = "tcp://192.0.2.1:1883"
serverProps["ServerSetId"] = serverSetId
serverProps["Username"] = "MyUserName"
serverProps["Password"] = "MyPassword"
system.cirruslink.engine.createConfig("Servers", serverProps)

# Find server Id
savedServers = system.cirruslink.engine.readConfig("Servers")
for config in savedServers:
    if config ["Name"] == "My Server":
        serverId = config["id"] # Read server configuration

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

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

# Read alarms
system.cirruslink.engine.readAlarms()
# Example list returned for 4 alarms
[{ "id": "59ff123d-62af-4272-83c8-cbc444891982", "name": "Alarm4", "alarmSource": "prot:MQTT:/src:Ignition-pi5:/prov:edge:/tag:Edge Nodes/G1/E1/D1/tag1:/alm:Alarm4", "priority": "Critical", "activeTime": 1731373561034, "groupId": "G1", "edgeNodeId": "E1", "deviceId": "D1", "state": "ActiveUnacked", "ackedBy": null }, { "id": "fa076e0f-bc37-4ce0-a7a3-709d2e2389b2", "name": "Alarm1", "alarmSource": "prot:MQTT:/src:Ignition-pi5:/prov:edge:/tag:Edge Nodes/G1/E1/D1/tag1:/alm:Alarm1", "priority": "Low", "activeTime": 1731373531034, "groupId": "G1", "edgeNodeId": "E1", "deviceId": "D1", "state": "ActiveUnacked", "ackedBy": null }, { "id": "2ac2c263-b552-49fc-a969-98313ca107b6", "name": "Alarm2", "alarmSource": "prot:MQTT:/src:Ignition-pi5:/prov:edge:/tag:Edge Nodes/G1/E1/D1/tag1:/alm:Alarm2", "priority": "Medium", "activeTime": 1731373541033, "groupId": "G1", "edgeNodeId": "E1", "deviceId": "D1", "state": "ActiveUnacked", "ackedBy": null }, { "id": "74a19264-8672-4e5b-ab9b-420c6aae7877", "name": "Alarm3", "alarmSource": "prot:MQTT:/src:Ignition-pi5:/prov:edge:/tag:Edge Nodes/G1/E1/D1/tag1:/alm:Alarm3", "priority": "High", "activeTime": 1731373551034, "groupId": "G1", "edgeNodeId": "E1", "deviceId": "D1", "state": "ActiveUnacked", "ackedBy": null } ]

# Clear alarm
system.cirruslink.engine.clearAlarm("59ff123d-62af-4272-83c8-cbc444891982")
```

