

MQTT Client Driver

© 2017 PTC Inc. All Rights Reserved.

Table of Contents

MQTT Client Driver	1
Table of Contents	2
MQTT Client Driver	4
Overview	5
Setup	6
Channel Properties — General	6
Channel Properties — Advanced	7
Channel Properties — Connection	7
Channel Properties — Timing	8
Channel Properties — Authentication	8
Device Properties — General	9
Device Properties — Scan Mode	11
Data Types Description	12
Address Descriptions	13
Event Log Messages	14
Datatype conversion failed. Device = '<device>', Address = '<address>', Expected = '<datatype>', Actual = '<datatype>'	14
Unexpected character found while parsing MQTT payload. Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>	14
Invalid Unicode character found while parsing MQTT payload. Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>	14
Missing matching quote while parsing MQTT payload. Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>	14
Invalid value found while parsing MQTT payload. Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>	15
Invalid number found while parsing MQTT payload. Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>	15
Unexpected token found within an array while parsing MQTT payload. Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>	15
Unexpected token found within an object while parsing MQTT payload. Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>	15
Unexpected token found while looking for the end of the MQTT payload. Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>	16
Failed to connect to MQTT server. Channel = '<channel>', Server = '<hostname:port>'	16
Failed to connect to MQTT server. Channel = '<channel>', Server = '<hostname:port>', Reason = '<non-localized reason>'	16
Failed to connect to MQTT server. Channel = '<channel>', Server = '<hostname:port>', Error Code = '<code>'	17

Failed to connect to MQTT server. Channel = '<channel>', Server = '<hostname:port>', Reason = '<reason>', Error Code = '<code>'.	17
Failed to connect to MQTT server. Channel = '<channel>', Server = '<hostname:port>', Reason = 'Connection refused; MQTT Version not supported'.	18
Failed to connect to MQTT server. Channel = '<channel>', Server = '<hostname:port>', Reason = 'Connection refused; Client ID rejected'.	18
Failed to connect to MQTT server. Channel = '<channel>', Server = '<hostname:port>', Reason = 'Connection refused; MQTT Server unavailable'.	18
Failed to connect to MQTT server. Channel = '<channel>', Server = '<hostname:port>', Reason = 'Connection refused; Bad Username or Password'.	19
Failed to connect to MQTT server. Channel = '<channel>', Server = '<hostname:port>', Reason = 'Connection refused; Not authorized to connect'.	19
Failed to connect to MQTT server. Channel = '<channel>', Server = '<hostname:port>', Reason = 'Connection refused', Error Code = '<code>'.	19
Connection with MQTT server lost. Channel = '<channel>', Server = '<hostname:port>'.	20
Unable to subscribe to topic. Channel = '<channel>', Topic = '<topic>'.	20
Unable to unsubscribe from topic. Channel = '<channel>', Topic = '<topic>'.	20
Appendix A — Tag Addressing Examples	21
Index	23

MQTT Client Driver

Help version 1.012

CONTENTS

Overview

What is the MQTT Client Driver?

Setup

How do I configure a device for use with this driver?

Data Types Description


What data types does this driver support?

Address Descriptions

How do I address a data location on a MQTT client?

Event Log Messages

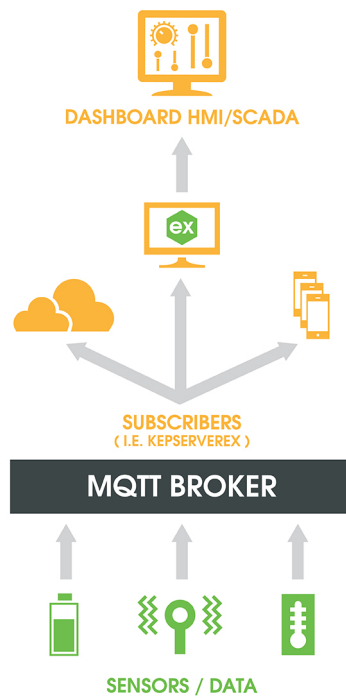
What messages are produced by the MQTT Client Driver?

 *Some of the messages displayed in the Event Log are forwarded from other protocols. Consult the vendor document for additional information.*

Overview

The MQTT Client Driver provides a way to get data from devices sent through the MQTT protocol. Many devices and sensors use diverse or proprietary communication protocols. Using MQTT can unify data collection and publication. This driver connects to an MQTT server (broker) and subscribes to specified topics. When updates on those topics are received, the payload is parsed and the values are set on OPC tags. Those tags can then be accessed through OPC clients to build meaningful data models and dashboards. The MQTT Client Driver supports:

- Parsing valid JSON data format
- Reading data from a MQTT server (writes are currently not supported)
- Authentication and TLS/SSL encryption
- Connecting to MQTT servers using MQTT 3.1 and 3.1.1 protocols
- User-level access based on the User Manager and Security Policies Plug-In



Setup

The maximum number of channels supported by this driver is 64.

Configure channels and devices by defining values for:

[Channel Properties](#)

[Device Properties](#)

• If the environment is secured with trusted certificates, establish those relationships through the server administration settings.

Channel Properties — General

This server supports the use of simultaneous multiple communications drivers. Each protocol or driver used in a server project is called a channel. A server project may consist of many channels with the same communications driver or with unique communications drivers. A channel acts as the basic building block of an OPC link. This group is used to specify general channel properties, such as the identification attributes and operating mode.

Property Groups	[-] Identification	
General	Name	
Write Optimizations	Description	
Advanced	Driver	
	[-] Diagnostics	
	Diagnostics Capture	Disable

Identification

Name: User-defined identity of this channel. In each server project, each channel name must be unique. Although names can be up to 256 characters, some client applications have a limited display window when browsing the OPC server's tag space. The channel name is part of the OPC browser information.

• For information on reserved characters, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in the server help.

Description: User-defined information about this channel.

• Many of these properties, including Description, have an associated system tag.

Driver: Selected protocol / driver for this channel. This property specifies the device driver that was selected during channel creation. It is a disabled setting in the channel properties.

• **Note:** With the server's online full-time operation, these properties can be changed at any time. This includes changing the channel name to prevent clients from registering data with the server. If a client has already acquired an item from the server before the channel name is changed, the items are unaffected. If, after the channel name has been changed, the client application releases the item and attempts to re-acquire using the old channel name, the item is not accepted. With this in mind, changes to the properties should not be made once a large client application has been developed. Utilize the User Manager to prevent operators from changing properties and restrict access rights to server features.

Diagnostics

Diagnostics Capture: When enabled, this option makes the channel's diagnostic information available to OPC applications. Because the server's diagnostic features require a minimal amount of overhead processing, it is recommended that they be utilized when needed and disabled when not. The default is disabled.

- **Note:** This property is disabled if the driver does not support diagnostics.
- *For more information, refer to "Communication Diagnostics" in the server help.*

Channel Properties — Advanced

This group is used to specify advanced channel properties. Not all drivers support all properties; so the Advanced group does not appear for those devices.

Property Groups	<input type="checkbox"/> Non-Normalized Float Handling	
General	Floating-Point Values	Replace with Zero
Write Optimizations	<input type="checkbox"/> Inter-Device Delay	
Advanced	Inter-Device Delay (ms)	0

Non-Normalized Float Handling: Non-normalized float handling allows users to specify how a driver handles non-normalized IEEE-754 floating point data. A non-normalized value is defined as Infinity, Not-a-Number (NaN), or as a Denormalized Number. The default is Replace with Zero. Drivers that have native float handling may default to Unmodified. Descriptions of the options are as follows:

- **Replace with Zero:** This option allows a driver to replace non-normalized IEEE-754 floating point values with zero before being transferred to clients.
- **Unmodified:** This option allows a driver to transfer IEEE-754 denormalized, normalized, non-number, and infinity values to clients without any conversion or changes.

● **Note:** This property is disabled if the driver does not support floating point values or if it only supports the option that is displayed. According to the channel's float normalization setting, only real-time driver tags (such as values and arrays) are subject to float normalization. For example, EFM data is not affected by this setting.

● *For more information on the floating point values, refer to "How To ... Work with Non-Normalized Floating Point Values" in the server help.*

Inter-Device Delay: Specify the amount of time the communications channel waits to send new requests to the next device after data is received from the current device on the same channel. Zero (0) disables the delay.

● **Note:** This property is not available for all drivers, models, and dependent settings.

Channel Properties — Connection

Property Groups	<input type="checkbox"/> Connection	
General	Host	localhost
Write Optimizations	Port	1883
Advanced	SSL/TLS	Disable
Connection	Client Certificate	Disable
Timing		
Authentication		

Host: The IP address or host name of the MQTT server to connect. The default is localhost.

Port: The port to use to connect to the MQTT server. The valid range is 1 to 65535. The default is 1883.

SSL/TLS: Enable to use a secure connection when connecting to the MQTT server. When enabled, all information is encrypted; this usually requires additional setup.

Client Certificate: Enable to allow client-side certificate validation with the MQTT server.

● **Notes:**

- To configure SSL/TLS certificates, use the **Certificate Store** tab in the Server Admin tool. Select the **MQTT Client** feature. More details may be found in the server help document under the **Administration | Settings | Certificate Store** section.
- The server runtime must be restarted when a new certificate is imported.

Channel Properties — Timing

Property Groups	Communication Timeouts	
General	Connect Timeout (s)	10
Write Optimizations	Reconnect Minimum (s)	10
Advanced	Reconnect Maximum (s)	10
Connection		
Timing		
Authentication		

Connect Timeout (s): Specify the number of seconds the client waits for the MQTT server to confirm the connection.

● **Tip:** This time is doubled when connecting to a server (broker) that uses the 3.1.0 specification. The valid range for the connect timeout is 1 second to 600 seconds.

Reconnect Minimum (s): Specify the minimum amount of time the MQTT Client driver waits before trying to reconnect to the MQTT Server. The valid range for the reconnect minimum is 1 second to 43200 seconds.

Reconnect Maximum (s): Specify the maximum amount of time the MQTT Client driver waits before trying to reconnect to the MQTT Server. This value must be equal to or greater than the reconnect minimum time. The valid range for the reconnect maximum is 1 second to 43200 seconds.

● **Tip:** To always have the same amount of time between retries set the minimum and maximum to the same value.

● **Note:** When a reconnection attempt fails, the current value is doubled. This continues until the maximum is reached. For example, if the minimum is 3 and maximum is 10, the first retry attempt has a three-second delay; the next attempt has a six-second delay; and any attempt after that has a ten-second delay.

Channel Properties — Authentication

Property Groups	<input checked="" type="checkbox"/> Authentication	
General	Client ID	u6oxhrdXxZBGQYIbHwSF29
Write Optimizations	User Name	*****
Advanced	Password	*****
Connection		
Timing		
Authentication		

Client ID: The client identifier defines this MQTT client to the MQTT server. It defaults to a 22-character randomly generated value. Only characters from this list can be used: "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ".

Tip: If this value is left blank, the MQTT server assigns a unique value; this depends on the MQTT server.

This property value must be unique for each MQTT client connecting to a specific MQTT server. Using non-blank duplicates causes problems with communications, including disconnections and missing updates.

User Name: Enter a UTF-8 string for the authorized user to connect to the MQTT server. This cannot be blank if the password property has a value.

Password: The password to use when connecting to the MQTT server with the specified user name.

Tip: If SSL/TLS is not enabled, the password sent to the MQTT server can be viewed using a packet sniffing tool.

Device Properties — General

A device represents a single target on a communications channel. If the driver supports multiple controllers, users must enter a device ID for each controller.

Property Groups	<input checked="" type="checkbox"/> Identification	
General	Name	
Scan Mode	Description	
	Channel Assignment	
	Driver	
	Model	
	ID Format	Decimal
	ID	2
	<input checked="" type="checkbox"/> Operating Mode	
	Data Collection	Enable
	Simulated	No

Identification

Name: This property specifies the name of the device. It is a logical user-defined name that can be up to 256 characters long, and may be used on multiple channels.

Note: Although descriptive names are generally a good idea, some OPC client applications may have a limited display window when browsing the OPC server's tag space. The device name and channel name

become part of the browse tree information as well. Within an OPC client, the combination of channel name and device name would appear as "ChannelName.DeviceName".

• For more information, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in server help.

Description: User-defined information about this device.

• Many of these properties, including Description, have an associated system tag.

Channel Assignment: User-defined name of the channel to which this device currently belongs.

Driver: Selected protocol driver for this device. This property specifies the driver selected during channel creation. It is disabled in the channel properties.

Model: This property specifies the specific type of device that is associated with this ID. The contents of the drop-down menu depends on the type of communications driver being used. Models that are not supported by a driver are disabled. If the communications driver supports multiple device models, the model selection can only be changed when there are no client applications connected to the device.

• **Note:** If the communication driver supports multiple models, users should try to match the model selection to the physical device. If the device is not represented in the drop-down menu, select a model that conforms closest to the target device. Some drivers support a model selection called "Open," which allows users to communicate without knowing the specific details of the target device. For more information, refer to the driver help documentation.

ID: This property specifies the device's station / node / identity / address. The type of ID entered depends on the communications driver being used. For many drivers, the ID is a numeric value. Drivers that support a Numeric ID provide users with the option to enter a numeric value whose format can be changed to suit the needs of the application or the characteristics of the selected communications driver. The ID format can be Decimal, Octal, and Hexadecimal. If the driver is Ethernet-based or supports an unconventional station or node name, the device's TCP/IP address may be used as the device ID. TCP/IP addresses consist of four values that are separated by periods, with each value in the range of 0 to 255. Some device IDs are string based. There may be additional properties to configure within the ID field, depending on the driver.

Operating Mode

Data Collection: This property controls the device's active state. Although device communications are enabled by default, this property can be used to disable a physical device. Communications are not attempted when a device is disabled. From a client standpoint, the data is marked as invalid and write operations are not accepted. This property can be changed at any time through this property or the device system tags.

Simulated: This option places the device into Simulation Mode. In this mode, the driver does not attempt to communicate with the physical device, but the server continues to return valid OPC data. Simulated stops physical communications with the device, but allows OPC data to be returned to the OPC client as valid data. While in Simulation Mode, the server treats all device data as reflective: whatever is written to the simulated device is read back and each OPC item is treated individually. The item's memory map is based on the group Update Rate. The data is not saved if the server removes the item (such as when the server is reinitialized). The default is No.

• **Notes:**

1. This System tag (`_Simulated`) is read only and cannot be written to for runtime protection. The System tag allows this property to be monitored from the client.
2. In Simulation mode, the item's memory map is based on client update rate(s) (Group Update Rate for OPC clients or Scan Rate for native and DDE interfaces). This means that two clients that reference the same item with different update rates return different data.

🔴 Simulation Mode is for test and simulation purposes only. It should never be used in a production environment.

Device Properties — Scan Mode

The Scan Mode specifies the subscribed-client requested scan rate for tags that require device communications. Synchronous and asynchronous device reads and writes are processed as soon as possible; unaffected by the Scan Mode properties.

Property Groups	☐ Scan Mode	
General	Scan Mode	Respect Client-Specified Scan Rate ▼
Scan Mode	Initial Updates from Cache	Disable

Scan Mode: specifies how tags in the device are scanned for updates sent to subscribed clients. Descriptions of the options are:

- **Respect Client-Specified Scan Rate:** This mode uses the scan rate requested by the client.
- **Request Data No Faster than Scan Rate:** This mode specifies the maximum scan rate to be used. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
 - **Note:** When the server has an active client and items for the device and the scan rate value is increased, the changes take effect immediately. When the scan rate value is decreased, the changes do not take effect until all client applications have been disconnected.
- **Request All Data at Scan Rate:** This mode forces tags to be scanned at the specified rate for subscribed clients. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
- **Do Not Scan, Demand Poll Only:** This mode does not periodically poll tags that belong to the device nor perform a read to get an item's initial value once it becomes active. It is the client's responsibility to poll for updates, either by writing to the `_DemandPoll` tag or by issuing explicit device reads for individual items. *For more information, refer to "Device Demand Poll" in server help.*
- **Respect Tag-Specified Scan Rate:** This mode forces static tags to be scanned at the rate specified in their static configuration tag properties. Dynamic tags are scanned at the client-specified scan rate.

Initial Updates from Cache: When enabled, this option allows the server to provide the first updates for newly activated tag references from stored (cached) data. Cache updates can only be provided when the new item reference shares the same address, scan rate, data type, client access, and scaling properties. A device read is used for the initial update for the first client reference only. The default is disabled; any time a client activates a tag reference the server attempts to read the initial value from the device.

Data Types Description

Data Type	Description
Boolean	Single bit
Byte	Unsigned 8-bit value bit 0 is the low bit bit 7 is the high bit
Char	Signed 8-bit value bit 0 is the low bit bit 6 is the high bit bit 7 is the sign bit
Word	Unsigned 16-bit value bit 0 is the low bit bit 15 is the high bit
DWord	Unsigned 32-bit value bit 0 is the low bit bit 31 is the high bit
QWord	Unsigned 64-bit value bit 0 is the low bit bit 63 is the high bit
Short	Signed 16-bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
Long	Signed 32-bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
LongLong	Signed 64-bit value bit 0 is the low bit bit 62 is the high bit bit 63 is the sign bit
Float	32-bit floating point value The driver interprets two consecutive 16-bit registers as a floating point value by making the second register the high word and the first register the low word.
Double	64-bit floating point value
String	Zero-terminated character array

Address Descriptions

The Tag Address is made up of the topic to subscribe to and the path of the value in the JSON payload. The two parts are separated by a '+' sign. For example, if the topic "Device/Home/LivingRoom" publishes payloads like this {"temp":72.3, "Light1":"On", "Light2":"Off", "TV-On": true}, the tag addresses might look like this:

Device/Home/LivingRoom+temp

Device/Home/LivingRoom+Light1

Device/Home/LivingRoom+Light2

Device/Home/LivingRoom+TV-On

● **See Also:** For more examples of how to define tags for different payloads see [Appendix A](#).

● **Notes:**

1. Even though '+' is a wildcard in MQTT, wildcards are not supported in the topic of the address. The '+' in the tag address is used as the separator between the topic and the path of the value in the payload.
2. The 'topic#Payload' tag always supplies the last payload received from that topic even if it is not valid JSON. This is ideal for trouble shooting issues or doing client side parsing. For example, to see the entire payload for the example above define a tag whose address is :
Device/Home/LivingRoom#Payload.

Event Log Messages

The following information concerns messages posted to the Event Log pane in the main user interface. Consult the server help on filtering and sorting the Event Log detail view. Server help contains many common messages, so should also be searched. Generally, the type of message (informational, warning) and troubleshooting information is provided whenever possible.

Datatype conversion failed. | Device = '<device>', Address = '<address>', Expected = '<datatype>', Actual = '<datatype>'.

Error Type:

Error

Possible Cause:

The datatype on the tag does not match the type of data from the payload. This can happen for overflow conditions or if the conversion is invalid.

Possible Solution:

Correct the datatype of the tag to match the data coming from the payload.

Unexpected character found while parsing MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>.

Error Type:

Warning

Possible Cause:

The payload is not valid JSON. Expected a token, got something else.

Possible Solution:

Reconfigure the payload to be valid JSON.

Invalid Unicode character found while parsing MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>.

Error Type:

Warning

Possible Cause:

Found an incomplete or invalid Unicode character escape sequence.

Possible Solution:

Reconfigure the payload to be valid JSON.

Missing matching quote while parsing MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>.

Error Type:

Warning

Possible Cause:

Found the End of doc before the end of a string

Possible Solution:

Reconfigure the payload to be valid JSON.

Invalid value found while parsing MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>.

Error Type:

Warning

Possible Cause:

Found an invalid value.

Possible Solution:

Reconfigure the payload to be valid JSON.

Invalid number found while parsing MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>.

Error Type:

Warning

Possible Cause:

A numerical value in the payload is not correct. It may contain invalid characters.

Possible Solution:

Reconfigure the payload to be valid JSON.

Unexpected token found within an array while parsing MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>.

Error Type:

Warning

Possible Cause:

An array within the payload is not formatted correctly.

Possible Solution:

Reconfigure the payload to be valid JSON.

Unexpected token found within an object while parsing MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>.

Error Type:

Warning

Possible Cause:

An object within the payload is not formatted correctly.

Possible Solution:

Reconfigure the payload to be valid JSON.

Unexpected token found while looking for the end of the MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>.

Error Type:

Warning

Possible Cause:

Unexpected token while looking for the end of the document.

Possible Solution:

Reconfigure the payload to be valid JSON.

Failed to connect to MQTT server. | Channel = '<channel>', Server = '<hostname:port>'.

Error Type:

Warning

Possible Cause:

1. The driver was unable to connect to the MQTT server over TCP/TLS due to an unspecified reason, but will continue to attempt connection.
2. The channel is configured to use a DNS host name for the MQTT server rather than an IP address. The host name cannot be resolved by the server to an IP address.

Possible Solution:

1. Verify that the MQTT server is started and online.
2. Verify that the correct MQTT port was specified.
3. Verify that the MQTT server IP is within the subnet of the IP to which the server is bound.
4. Verify that the MQTT server is registered with the domain.
5. Correct any connectivity issues with the MQTT server.

Failed to connect to MQTT server. | Channel = '<channel>', Server = '<hostname:port>', Reason = '<non-localized reason>'.

Error Type:

Warning

Possible Cause:

1. The channel is configured to use a DNS host name for the MQTT server rather than an IP address. The host name cannot be resolved by the server to an IP address.
2. The driver was unable to connect to the MQTT server over TCP/TLS due to the reason provided, but will continue to attempt connection.

Possible Solution:

1. Verify that the MQTT server is started and online.
2. Verify that the correct MQTT port was specified.
3. Verify that the MQTT server IP is within the subnet of the IP to which the server is bound.
4. Verify that the MQTT server is registered with the domain.
5. Correct any connectivity issues with the MQTT server.

Failed to connect to MQTT server. | Channel = '<channel>', Server = '<hostname:port>', Error Code = '<code>'.

Error Type:

Warning

Possible Cause:

1. The channel is configured to use a DNS host name for the MQTT server rather than an IP address. The host name cannot be resolved by the server to an IP address.
2. The driver was unable to connect to the MQTT server over TCP/TLS due to the error code provided, but will continue to attempt connection.

Possible Solution:

1. Verify that the MQTT server is started and online.
2. Verify that the correct MQTT port was specified.
3. Verify that the MQTT server IP is within the subnet of the IP to which the server is bound.
4. Verify that the MQTT server is registered with the domain.

Failed to connect to MQTT server. | Channel = '<channel>', Server = '<hostname:port>', Reason = '<reason>', Error Code = '<code>'.

Error Type:

Warning

Possible Cause:

1. The channel is configured to use a DNS host name for the MQTT server rather than an IP address. The host name cannot be resolved by the server to an IP address.

2. The driver was unable to connect to the MQTT server over TCP/TLS due to the reason and error code provided, but will continue to attempt connection.

Possible Solution:

1. Verify that the MQTT server is started and online.
2. Verify that the correct MQTT port was specified.
3. Verify that the MQTT server IP is within the subnet of the IP to which the server is bound.
4. Verify that the MQTT server is registered with the domain.

Failed to connect to MQTT server. | Channel = '<channel>', Server = '<hostname:port>', Reason = 'Connection refused; MQTT Version not supported'.

Error Type:

Warning

Possible Cause:

The driver was able to connect to the MQTT server over TCP/TLS but the server refused our MQTT CONNECT request.

Possible Solution:

Verify the MQTT version specified matches the MQTT server's supported versions.

Failed to connect to MQTT server. | Channel = '<channel>', Server = '<hostname:port>', Reason = 'Connection refused; Client ID rejected'.

Error Type:

Warning

Possible Cause:

The driver was able to connect to the MQTT server over TCP/TLS but the server refused our MQTT CONNECT request.

Possible Solution:

Verify a valid Client ID has been specified. Empty Client IDs or Client IDs longer than 23 characters are not supported in MQTT version 3.1.

Failed to connect to MQTT server. | Channel = '<channel>', Server = '<hostname:port>', Reason = 'Connection refused; MQTT Server unavailable'.

Error Type:

Warning

Possible Cause:

The driver was able to connect to the MQTT server over TCP/TLS but the server refused our MQTT CONNECT request.

Possible Solution:

Verify that the MQTT server is started and online.

Failed to connect to MQTT server. | Channel = '<channel>', Server = '<hostname:port>', Reason = 'Connection refused; Bad Username or Password'.

Error Type:

Warning

Possible Cause:

The driver was able to connect to the MQTT server over TCP/TLS but the server refused our MQTT CONNECT request.

Possible Solution:

Verify a valid username and password, according to the MQTT server, have been specified.

Failed to connect to MQTT server. | Channel = '<channel>', Server = '<hostname:port>', Reason = 'Connection refused; Not authorized to connect'.

Error Type:

Warning

Possible Cause:

The driver was able to connect to the MQTT server over TCP/TLS but the server refused our MQTT CONNECT request.

Possible Solution:

Verify a valid username and password, according to the MQTT server, have been specified.

Failed to connect to MQTT server. | Channel = '<channel>', Server = '<hostname:port>', Reason = 'Connection refused', Error Code = '<code>'.

Error Type:

Warning

Possible Cause:

The driver was able to connect to the MQTT server over TCP/TLS but the server refused our MQTT CONNECT request.

Possible Solution:

Error code is undocumented. Please refer to latest MQTT specification or MQTT server for error code description.

Connection with MQTT server lost. | Channel = '<channel>', Server = '<hostname:port>'.

Error Type:

Warning

Possible Cause:

The driver lost communications with MQTT server, but will attempt to reconnect at the reconnect intervals configured.

Possible Solution:

1. Verify that the MQTT server is started and online.
2. Correct any connectivity issues with the MQTT server.

Unable to subscribe to topic. | Channel = '<channel>', Topic = '<topic>'.

Error Type:

Warning

Possible Cause:

The driver lost communications with MQTT server during a subscribe request, but will attempt to reconnect at the reconnect intervals configured.

Possible Solution:

1. Verify that the MQTT server is started and online.
2. Correct any connectivity issues with the MQTT server.

Unable to unsubscribe from topic. | Channel = '<channel>', Topic = '<topic>'.

Error Type:

Warning

Possible Cause:

The driver lost communications with MQTT server during an unsubscribe request, but will attempt to reconnect at the reconnect intervals configured.

Possible Solution:

1. Verify that the MQTT server is started and online.
2. Correct any connectivity issues with the MQTT server.

Appendix A — Tag Addressing Examples

While working with MQTT one of the more difficult things is to define the mapping between content within a payload and OPC tags. This document is to help with that process with the MQTT Client driver.

Tag Address

Tag addresses are of the form `mqtt_topic+payload_item`. The `mqtt_topic` is the topic in which the desired data value is to be published. The `payload_item` is a path, in the payload, to the specific item for which data is desired. Here are some examples:

Topic: BB/123456ABCDEF/data

Sample Payload:

```
{
  "s":4,
  "t":"2017-09-29T19:52:19Z",
  "q":192,
  "c":6,
  "tempint":67.1,
  "vbatt":3.28,
  "ai1":8.92,
  "ai2":0.03,
  "temp1":46.4,
  "temp2":68.0
}
```

Tag addresses and their values would be:

```
BB/123456ABCDEF/data+s = 4
BB/123456ABCDEF/data+t = 2017-09-29T19:52:19Z
BB/123456ABCDEF/data+q = 192
BB/123456ABCDEF/data+c = 6
BB/123456ABCDEF/data+tempint = 67.1
BB/123456ABCDEF/data+vbatt = 3.28
BB/123456ABCDEF/data+ai1 = 8.92
BB/123456ABCDEF/data+ai2 = 0.03
BB/123456ABCDEF/data+temp1 = 46.4
BB/123456ABCDEF/data+temp2 = 68.0
```

Topic: Wago/1234

Sample Payload:

```
{
  "FormatId": "DeviceState",
  "ApiVersion": 1,
  "CurrentTime": "2012-06-11T14:26:59.690+02:00",
  "UserSwitch": "State:Run",
  "Leds": [
    {"Name": "IO", "State": "Blinking", "Color": "Red"},
    {"Name": "SYS", "State": "On", "Color": "Green"},
    {"Name": "USR", "State": "On", "Color": "Off"}
  ]
}
```

Tag addresses and their values would be:

```
Wago/1234+FormatId = DeviceState
Wago/1234+ApiVersion = 1
Wago/1234+CurrentTime = 2012-06-11T14:26:59.690+02:00
Wago/1234+UserSwitch = State:Run
Wago/1234+Leds[0]/Name = IO
Wago/1234+Leds[0]/State = Blinking
Wago/1234+Leds[0]/Color = Red
Wago/1234+Leds[1]/Name = SYS
Wago/1234+Leds[1]/State = On
Wago/1234+Leds[1]/Color = Green
Wago/1234+Leds[2]/Name = USR
Wago/1234+Leds[2]/State = On
```

Wago/1234+Leds[2]/Color = Off

Topic: Sample/NestedArrays

Sample Payload:

```
{
  "name": "John",
  "age": 30,
  "cars": [
    { "name": "Ford", "models": [ "Fiesta", "Focus", "Mustang" ] },
    { "name": "BMW", "models": [ "320", "X3", "X5" ] },
    { "name": "Fiat", "models": [ "500", "Panda" ] }
  ]
}
```

Tag addresses and their values would be:

```
Sample/NestedArrays+name = John
Sample/NestedArrays+age = 30
Sample/NestedArrays+cars[0]/name = Ford
Sample/NestedArrays+cars[0]/models[0] = Fiesta
Sample/NestedArrays+cars[0]/models[1] = Focus
Sample/NestedArrays+cars[0]/models[2] = Mustang
Sample/NestedArrays+cars[1]/name = BMW
Sample/NestedArrays+cars[1]/models[0] = 320
Sample/NestedArrays+cars[1]/models[1] = X3
Sample/NestedArrays+cars[1]/models[2] = X5
Sample/NestedArrays+cars[2]/name = Fiat
Sample/NestedArrays+cars[2]/models[0] = 500
Sample/NestedArrays+cars[2]/models[1] = Panda
```

Index

A

Address Descriptions 13
Advanced Channel Properties 7
Appendix 21
Authentication 5, 8

B

Bad Username or Password'. 19
Boolean 12
Broker 5
Byte 12

C

Channel Assignment 10
Channel Properties - General 6
Char 12
Client Certificate 8
Client ID 9
Client ID rejected'. 18
Connect Timeout 8
Connection 7
Connection with MQTT server lost. | Channel = '<channel>', Server = '<hostname port>'. 20

D

Data Collection 10
Data Types Description 12
Datatype conversion failed. | Device = '<device>', Address = '<address>', Expected = '<datatype>', Actual = '<datatype>'. 14
Description 10
Device Properties — General 9
Diagnostics 7

Do Not Scan, Demand Poll Only 11

Double 12

Driver 6, 10

DWord 12

E

Event Log Messages 14

F

Failed to connect to MQTT server. | Channel = '<channel>', Server = '<hostname
port>', Error Code = '<code>'. 17
port>', Reason = '<non-localized reason>'. 16
port>', Reason = '<reason>', Error Code = '<code>'. 17
port>', Reason = 'Connection refused 18-19
port>', Reason = 'Connection refused', Error Code = '<code>'. 19
port>'. 16

Float 12

H

Host 8

I

ID 10

IEEE-754 floating point 7

Initial Updates from Cache 11

Invalid number found while parsing MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line =
<line>, Col = <col>. 15

Invalid Unicode character found while parsing MQTT payload. | Channel = '<channel>', Topic = '<topic>',
Line = <line>, Col = <col>. 14

Invalid value found while parsing MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line = <line>,
Col = <col>. 15

J

JSON 5

L

Long 12

LongLong 12

M

Missing matching quote while parsing MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>. 14

Model 10

MQTT 5

MQTT Server 5

MQTT Server unavailable'. 18

MQTT Version not supported'. 18

N

Name 9

Non-Normalized Float Handling 7

Not authorized to connect'. 19

O

Overview 5

P

Password 9

Payload 13

Port 8

Q

QWord 12

R

Reconnect Maximum 8

Reconnect Minimum 8
Request All Data at Scan Rate 11
Request Data No Faster than Scan Rate 11
Respect Client-Specified Scan Rate 11
Respect Tag-Specified Scan Rate 11

S

Scan Mode 11
Setup 6
Short 12
Simulated 10
SSL/TLS 8
String 12

T

Tag Addressing Examples 21
Timing 8

U

Unable to subscribe to topic. | Channel = '<channel>', Topic = '<topic>'. 20
Unable to unsubscribe from topic. | Channel = '<channel>', Topic = '<topic>'. 20
Unexpected character found while parsing MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>. 14
Unexpected token found while looking for the end of the MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>. 16
Unexpected token found within an array while parsing MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>. 15
Unexpected token found within an object while parsing MQTT payload. | Channel = '<channel>', Topic = '<topic>', Line = <line>, Col = <col>. 15
User Name 9

W

Wildcard 13
Word 12