

MQTT Configuration Guide

RMC-100

Additional information

Additional free publications are available for download at www.abb.com/upstream.

Table 0-1: Related documents

Document	Document number
MQTT Data Interpretation Code Guide	2107649 (request from product manager)
RMC-100 Startup Guide	2105551

Cyber security

Products with embedded Message Queue Telemetry Transport (MQTT) capability are designed to be connected, and communicate information and data, via a network interface. All ABB Totalflow products should be connected to a secure network. It is the customer's sole responsibility to provide and continuously ensure a secure connection between the product and the customer network or any other network (as the case may be). The customer shall establish and maintain appropriate measures (such as, but not limited to, the installation of firewalls, application of authentication measures, encryption of data, or installation of antivirus programs) to protect this product, the network, its system and interfaces against any kind of security breaches, unauthorized access, interference, intrusion, leakage and/or theft of data or information. ABB Inc. and its affiliates are not liable for damages and/or losses related to such security breaches, any unauthorized access, interference, intrusion, leakage and/or theft of data or information.

Although ABB provides functionality testing on the products and updates that it releases, the customer should institute their own testing program for any product updates or other major system updates (to include, but not limited to, code changes, configuration file changes, third party software updates or patches, hardware change out) to ensure that the security measures that the customer has implemented have not been compromised and that system functionality in the customer's environment is as expected.

Malware prevention

Recommendation: As with any downloaded software, scan ABB embedded software packages using a malware prevention solution.

Contents

Additional information.....	1
Cyber security.....	1
Malware prevention.....	1
1 Overview	4
1.1 MQTT-enabled field devices.....	4
1.1.1 MQTT 3.1.1 support	4
1.1.2 Sparkplug support	4
1.1.3 MQTT configuration interface	5
1.2 MQTT broker	6
2 Application data publishing	6
3 Prepare for configuration	7
3.1 Prerequisites for device	7
3.2 Prerequisites for the laptop used in configuration	8
3.3 Prerequisites for authentication.....	9
3.3.1 Determine authentication methods	9
3.3.2 Prepare for authentication configuration	9
3.4 Preparing for Sparkplug.....	10
3.4.1 ABB device role in Sparkplug infrastructure.....	10
3.4.2 ABB device topic namespace.....	11
4 Enable MQTT services on the device.....	13
4.1 Enable MQTT from terminal mode	13
4.2 Enable MQTT from PCCU Entry mode (recommended).....	14
5 Access the MQTT configuration interface	16
6 Configure MQTT/Sparkplug parameters	17
6.1 Configure for MQTT 3.1.1.....	18
6.2 Configuration for Sparkplug.....	23
7 Verify device-broker connection status	28
8 Configure applications.....	29
8.1 Application configuration page	29
8.2 Measurement and control applications.....	30
8.3 Holding Registers application (private networks only)	30
9 Configure registers	31
9.1 Number of registers published	31
9.2 Register configuration page.....	31
9.3 Configure measurement and control applications	33
9.4 Configure Holding Registers	34
9.5 Holding Register renaming (Ignition Designer® users).....	35
10 Using the MQTT configuration on another device.....	35
11 Troubleshooting during initial configuration	36
11.1 Troubleshooting user-device connection	36
11.2 Troubleshooting device-broker connection	38
11.3 Troubleshooting authentication	38

12 Device security	39
12.1 Security guidelines	39
12.2 MQTT services	40
12.3 Secure connections.....	41
12.4 Manage users from the web interface.....	41
12.4.1 Default account users and role privileges	41
12.4.2 Access the User Management web page.....	42
12.4.3 Change default passwords.....	43
12.4.4 Add a user.....	43
12.4.5 Update a user	45
12.4.6 Delete a user	47
12.5 Upload valid certificates (for secure user connection).....	48
12.6 Disable MQTT Rest service	51
13 Monitor device audit logs	51
13.1 Audit Logging web page overview	51
13.2 Access the Audit Logging web page	53
14 Monitor devices statistics	54
14.1 Access the Statistics web page	55
14.2 Device configuration statistics.....	57
14.3 Device-broker connection statistics.....	58
14.4 Sparkplug statistics	61
15 Useful terms	64

1 Overview

This guide provides basic steps to enable an ABB Totalflow device for connection to an MQTT broker on a private network. ABB Totalflow devices implement native **MQTT 3.1.1** and **Sparkplug B** protocols. Consult with your system administrator for specific requirements and configuration based on your implementation.



NOTICE – Cybersecurity risk: ABB Totalflow MQTT-enabled devices are designed to connect to MQTT servers on a private network. Connection to a MQTT server on a public network such as the Internet should only be done indirectly through an Edge gateway.

The following sections provide an overview of the components for enabling MQTT communication for ABB Totalflow devices.

1.1 MQTT-enabled field devices

ABB Totalflow MQTT-enabled devices are flow measurement or control devices with embedded MQTT support. Typically, this is a remote controller such as the RMC-100.

These devices support standards-based native MQTT functionality: MQTT-enabled devices act as MQTT clients. As such, they can connect to an MQTT server (broker) to exchange MQTT messages. MQTT messages transmit data in the payload. The data format for the payload is defined by the ABB Ability information model.

1.1.1 MQTT 3.1.1 support

ABB Totalflow MQTT-enabled devices support MQTT 3.1.1. Refer to online resources for the MQTT standard documentation at <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf>.

1.1.2 Sparkplug support

ABB Totalflow MQTT-enabled devices support Sparkplug B to connect to SCADA or IIoT systems. Sparkplug enhances the MQTT protocol to better support the real-time requirements of these systems. For detailed information on the standard, see <https://sparkplug.eclipse.org/specification/version/3.0>.

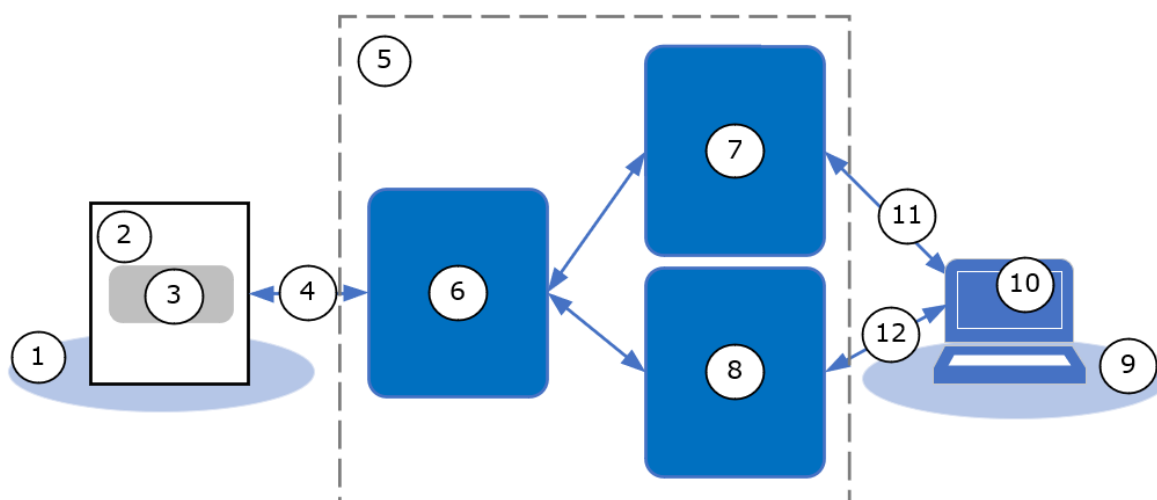


IMPORTANT NOTE: The implementation of the SCADA or IIoT system depends on specific customer requirements and available network topologies. Customers that implement end-to-end solutions on their own private networks manage their own MQTT broker/server.

[Figure 1-1](#) shows a simplified diagram of the functional blocks of a sample Sparkplug architecture implemented on a corporate network (5). The SCADA or IIoT system (7), and the MQTT server are installed at the customer network. The MQTT broker (6) is the intermediary for MQTT communication between the device (2) and the SCADA system applications (7, 8).

When Sparkplug is selected as the device's protocol for connecting with the MQTT broker, the device establishes an MQTT connection (4) and performs both the MQTT device and Edge of Node functionality, as per the Sparkplug specification. As an Edge of Node (EoN), the device supports the Sparkplug session management, topic name space, and payload definitions. This additional support enhances communication and provides better support for real-time data. Sparkplug message payload from the device reflects both roles: the device and Edge of Node roles. For details on monitored Sparkplug packets, see section [14.4](#).

Figure 1-1: Sparkplug high level architecture



Legend for Figure 1-1: Sparkplug high level architecture

ID	Field site	ID	Customer network	ID	Customer access
1	Field Local Area Network	5	Customer corporate network (VPN)	9	Field office network with secure access
2	Totalflow device	6	MQTT broker (server/distributor)	10	Client system: PC/Laptop with browser as client to SCADA/IIoT application
3	MQTT client and Sparkplug Device/Edge of Node (EoN) functionality	7	SCADA/IIoT Host (Primary Application)	11	Connection to primary application
4	MQTT connection	8	Other backend application (non-primary SCADA/IIoT client application)	12	Connection to other backend application



IMPORTANT NOTE: For simplicity, [Figure 1-1](#) does not show any databases or other services. Databases are typically implemented on-premise for data storage. Customer implementations are proprietary and specific to their systems and requirements.



IMPORTANT NOTE: MQTT servers supporting Sparkplug must be MQTT v3.1.1 compliant. MQTT servers may be referred to by other names, depending on the vendor implementing them. This manual uses the generic term “server” to indicate the main functionality or role of this component in the overall architecture. For details, consult your vendor documentation and architectures.

1.1.3 MQTT configuration interface

ABB Totalflow MQTT-enabled devices support the configuration of MQTT parameters and application data publishing options from a browser-based interface.



IMPORTANT NOTE: The web-based user interface is specifically implemented for the MQTT configuration. PCCU is still used for full device and application configuration, for enabling MQTT services and user access from web browsers.

The MQTT configuration pages support:

- MQTT/Sparkplug Configuration: MQTT protocol selection, MQTT-related and device parameter configuration, target MQTT broker configuration and connection verification, authentication method for device-broker connection, security configuration for user-device connection (see [section 6](#) [Configure MQTT/Sparkplug parameters](#).)
- Application Configuration: Selection of applications for which data will be published (see [section 8](#))

- [Configure applications.](#))
- Register Configuration: Selection of specific application registers for which data will be published (see section [9 Configure registers.](#))

1.2 MQTT broker

ABB Totalflow MQTT-enabled devices connect to MQTT brokers (servers). MQTT brokers provide many configuration options for secure connections with field devices. Customers must be aware of the broker configuration to configure the ABB Totalflow devices correctly.

A private MQTT broker enables secure MQTT-protocol-based connection of field devices to a private customer wide area network.

The successful device-MQTT broker connection allows the device to send (publish) data to the MQTT broker.



IMPORTANT NOTE: On a private customer network, customers must install and configure their own broker and integrate it with their data collection/management systems. This document assumes that a broker is already available and ready for device connection requests. Installation, configuration, and integration of private brokers is beyond the scope of the document.

2 Application data publishing

[Table 2-1](#) describes the type of data that ABB Totalflow devices publish on the MQTT broker for each of the applications supported.

Table 2-1: Type of data published by devices

Data type	Description
Application records or Snapshot data	General device or application information sent by the device at first-time boot or after reboot. It includes information about: <ul style="list-style-type: none"> — Device resources, etc. — Enabled registers (registers the device publishes data for) — Various records (also referred to as snapshot data) such as Alarm, Trend, Daily Logs, Custom Logs and Events
Application register data	Specific application register values for the supported ABB Totalflow applications (See Table 2-2). Application registers or variables, store application-specific data.

[Table 2-2](#) lists the specific ABB Totalflow applications supported. To access data for these applications, they must be fully configured from PCCU and then selected from the device MQTT configuration interface.



IMPORTANT NOTE: The Alarm System and Trend System applications are not listed on the application configuration web page (section [8 Configure applications](#)). Alarm data and trends are automatically published for all supported applications if defined and configured from PCCU.

Table 2-2: ABB Totalflow applications supporting publishing

Application	Description	Use
Alarm system	Alarm detection, logging, and reporting	Obtain or display alarms and alarm definition data.
Trend system	Data trending	Obtain or display trend definition data. Defined trend variables can be used to generate graphical displays.

Application	Description	Use
Holding Registers	Holding Registers allow the user to custom define how to store values of interest in specific device register ranges. This application is customized per user requirements. The registers are not pre-defined.	Obtain or display data from selected register ranges. Data stored is defined by the user, therefore displayed data depends on user-selections (custom). The holding register application can be configured for publishing on a broker, but customers must develop their own application to retrieve the holding register data.
AGA 3	Orifice gas measurement	Obtain or display data
AGA 7	Linear gas measurement	Obtain or display data
API Liquid SU	Linear liquid measurement	Obtain or display data
Plunger control	Control of a plunger on a production well	Obtain or display data. It provides some basic control also. The level of control functionality depends on the customer implementation.
Gas lift	Artificial lift for wells with liquid loading problems	Obtain or display data
Shutdown System	Shut down a well or site	Obtain or display data



IMPORTANT NOTE: The RMC-100's Sparkplug metrics (variables) do not have time stamps. Each Sparkplug message has a 32-bit device timestamp without a time zone, instead of the 64-bit UTC time stamp required by the Sparkplug protocol.

3 Prepare for configuration

This section describes device configuration requirements for connection and data publishing on the MQTT broker. Review requirements and associated tasks prior to configuration.

First time MQTT connection of an in-service device requires device restart. Follow your company guidelines to schedule configuration of in-service devices. Obtain required parameters from your administrator prior to configuration.



NOTICE – Cybersecurity risk. Review section [12 Device security](#) for general security guidelines before you enable devices for MQTT and connect them to a network.



IMPORTANT NOTE: ABB Totalflow application configuration is beyond the scope of this manual. This document assumes the application configuration is complete and operational in existing devices. For new installations, first instantiate, then enable, and last configure applications from PCCU.



IMPORTANT NOTE: This document assumes that the MQTT broker on the customer private network is already configured and available for connection.

3.1 Prerequisites for device

This section describes the minimum requirements to support device configuration. [Table 3-1](#) lists requirements for the RMC-100 as an example. Review the requirement lists and their associated tasks.

Table 3-1: Prerequisites for RMC-100

Requirement	Description	Tasks
MQTT-ready device OS and flash	The device embedded software with the MQTT client functionality	<ul style="list-style-type: none"> — Obtain customer package 2105452-045 or later for the RMC-100. — Upgrade the device. See PCCU help files or refer to Additional information for a link to RMC-100 documentation. — Enable MQTT functionality on the device as described in section 4 — Enable MQTT services on the device.
Valid IP configuration for cloud connection	IP configuration must include a valid IP address, subnet mask, and default gateway.	<ul style="list-style-type: none"> — Obtain valid IP configuration from your IT administrator if configuring a new device or an existing device without IP parameters assigned. — Configure the device's IP parameters (address, mask, and gateway) from PCCU.
Unique Device ID	Device ID, or name that uniquely identifies the ABB Totalflow device on the broker	<ul style="list-style-type: none"> — Obtain device ID from MQTT server Administrator. <p>Note: Not to be confused with the Device ID as configured in PCCU. The device ID configured from PCCU may be different than the ID used for MQTT Communications.</p>
Authentication certificates and keys	<p>Files generated by third-party certificate or security key generators</p> <p>Optional for private network implementations</p>	<p>MQTT servers can support several authentication options.</p> <ul style="list-style-type: none"> — Determine the authentication method. — Generate or obtain certificate and authentication keys as necessary. <p>Note: The MQTT configuration interface also supports server authentication with usernames and passwords. If a private MQTT server is configured to authenticate connection requests with these credentials, then obtain the required credentials from the server administrator. Customers must configure their authentication methods in their MQTT server. If they choose to use certificates, they are solely responsible for certificate generation and management.</p>



IMPORTANT NOTE: If planning on using an MQTT configuration in multiple RMC-100s, review section [10](#) [Using the MQTT configuration on another device.](#) before loading the configuration.

3.2 Prerequisites for the laptop used in configuration

[Table 3-2](#) provides requirements for the laptop or PC used to configure the device. Review the requirement lists and their associated tasks.

Field device configuration for MQTT requires IP communication. Ensure that both the ABB Totalflow device and the system used to configure the device each have the required IP configuration for successful communication.



IMPORTANT NOTE: Access to the device from mobile devices is also supported. Access interfaces adapt their display to the type of device.

Table 3-2: Configuration system (laptop) prerequisites

Requirement	Description	Task
Chrome browser	The Chrome browser provides access to the device's MQTT configuration web pages.	— Download and install up-to-date version of the Chrome Internet browser per your corporate policy.
PCCU	PCCU is required to add, enable, and optimize all ABB Totalflow applications.	— Obtain and install PCCU 7.76 or later. — It is assumed that all application configuration is complete prior to MQTT configuration. Note: PCCU 7.69, 7.69.1, 7.70, and 7.71 have been discontinued and are no longer supported. If you have these versions, obtain and install the latest PCCU version from the ABB library.
Valid IP configuration	The MQTT configuration requires IP communication between the laptop and the device. The laptop's IP configuration must be compatible with the device's IP configuration.	— Obtain a valid IP address from the system administrator. — Configure the laptop with the valid IP address.

3.3 Prerequisites for authentication

Secure device-broker connection requires authentication. Authentication might require access credentials, public/private key pairs or security certificates, depending on the authentication method or standard used. When using a private MQTT broker, it is the customer's responsibility to select the preferred authentication method and generate or manage certificates if this form of authentication is configured in their server.

3.3.1 Determine authentication methods

The ABB Totalflow device supports two types of authentication options:

- Authentication using valid username/password. The device embeds a valid username and password in its connection requests. The MQTT Broker verifies that the credentials match those provided and authorized for the customer.
- Authentication using the X.509 standard format. This standard defines the format of public key certificates used in the communication protocols for secure device-broker connections. There are two types of X.509 authentication:
 - Self-signed X.509 authentication uses a self-signed identity certificate.
 - Certification Authority (CA)-signed X.509 authentication uses a certificate signed by a third-party authority trusted by both the customer and the cloud service provider.

IMPORTANT NOTE: CA-signed X.509 certificates are preferred over self-signed certificates. Administrators must choose and configure the authentication policy in their private servers. If using a service provider, they must verify the service provider's policy and support to generate the appropriate certificates.

3.3.2 Prepare for authentication configuration

The authentication method is a required parameter for field configuration. [Table 3-3](#) provides high level tasks to prepare for authentication configuration.

IMPORTANT NOTE: Customers are fully responsible for certificate management. Administrators must follow company policies and procedures to maintain and save certificates, keys, fingerprints, verification codes, usernames, and passwords in a safe location.

Table 3-3: Obtain authentication parameters

Requirement	Description	Tasks
Authentication method	Format used for validation of field devices before connection to the broker. The authentication method in the device and the broker must match.	<ul style="list-style-type: none"> — Obtain the preferred method from the MQTT server administrator.
Username/password	Credentials that may be required by the MQTT broker to grant connection requests from field devices. These may be required in addition to certificates or be the sole authentication method.	<ul style="list-style-type: none"> — Obtain credentials from the MQTT server administrator. — The RMC requires these credentials to attempt connection.
Certificates, Keys	Required for X.509 authentication. Digital files with certificate, key and fingerprint that certify the device authenticity for acceptance by the broker. Optional for private MQTT servers.	<ul style="list-style-type: none"> — Obtain from MQTT server/system administrator the three required files for X.509 authentication. — Administrators must obtain the common Root Certificate (for all devices) — Administrators must obtain (or generate) the device-specific files: Client Certificate and Client Key. — Have certificate files available on the system the device is configured from. The certificate files must be copied to the device during configuration.

3.4 Preparing for Sparkplug

Values for Sparkplug related parameters must be ready prior to device configuration. Configuration should conform to the Sparkplug B specification which is the version supported by the RMC-100.

3.4.1 ABB device role in Sparkplug infrastructure

[Figure 3-1](#) shows a diagram from the Sparkplug B specification which identifies the generic components of a Sparkplug infrastructure. The Host application (SCADA, etc.) and the MQTT EoN Node are MQTT clients subscribing to the server. The Node publishes its own data and the aggregate data for devices, sensors, etc.

Figure 3-1: Sparkplug infrastructure components

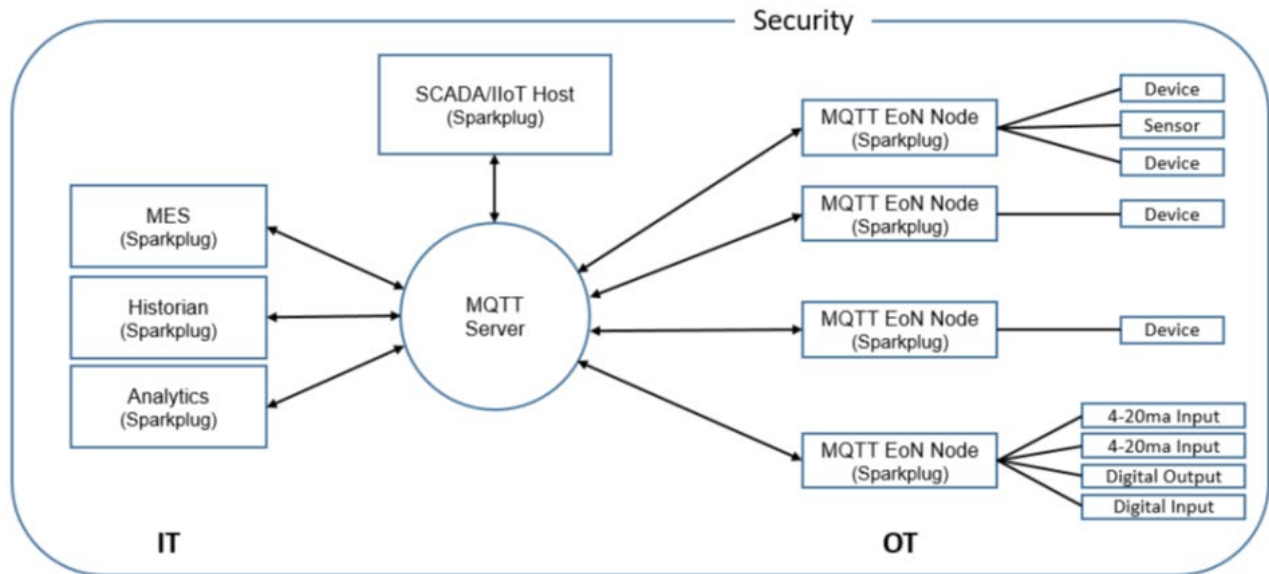
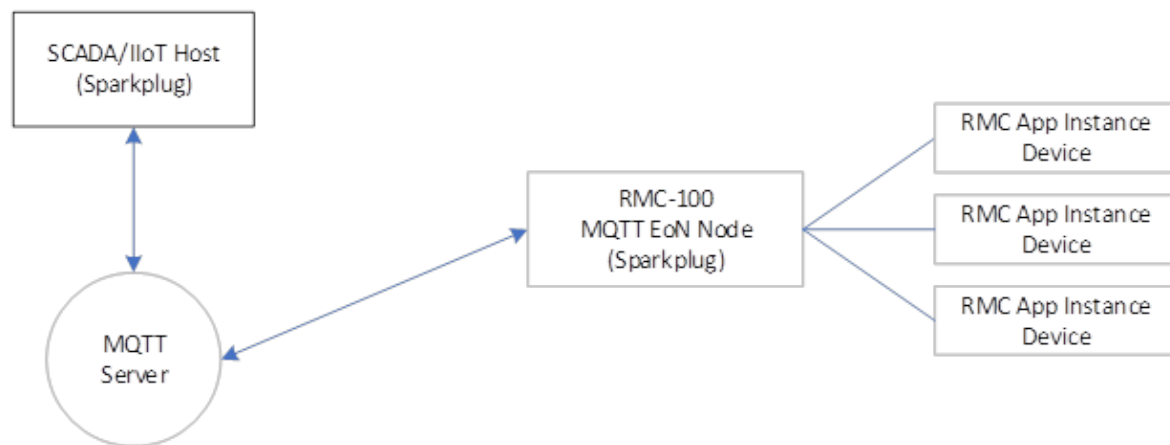


Figure 3-2 shows the components of the Sparkplug implementation with the RMC-100 (an implementation would include multiple RMCs, but only one is shown for simplicity)

- Each RMC-100 acts a logical MQTT EoN Node.
- Each RMC-100 Application Instance (AGA3-1, AGA3-2, AGA7-1, etc.) is a logical representation of an MQTT Device.

It is important to understand the role of the ABB device and its applications. Their role determines the correct way to define topic namespaces that adhere to the Sparkplug B specification. See section [3.4.2 ABB device topic namespace](#).

Figure 3-2: Sparkplug infrastructure components using RMC-100



3.4.2 ABB device topic namespace

Topic namespaces determine the data structure held by the broker for accurate data flow from/to subscribers. The namespace defines where data is published by devices and where data is retrieved from by the subscribing applications (SCADA system, etc.).

Sparkplug B specifies the following elements for the topic namespace:

namespace/group_id/message_type/edge_node_id/[device_id]

The namespace element identifies the Sparkplug specification version. For version B it is specified as: spBv1.0.

The RMC-100 implementation of the topic namespace maps RMC-100 parameters to the Sparkplug B topic namespace elements as follows:

spBv1.0/{Group ID}/{Message Type}/{Device ID}/{App Name}

Where:

- **Group ID:** Maps to the Sparkplug B **group_id** element. This element is user-configurable in the MQTT configuration interface (See [Table 6-2](#)).
- **Message Type:** It is the same as defined by the Sparkplug B **message_type** element. It defines the type of information carried by the message. For example, if the message contains the device's register data, the message type is indicated by **DDATA**.
- **Device ID:** Maps to the Sparkplug B **edge_node_id** element. This is the unique identification for each RMC-100 connecting to the MQTT server. Not to be confused with the Device ID as configured in PCCU. This element is user-configurable in the MQTT configuration interface (See [Table 6-2](#)).
- **App Name:** Maps to the Sparkplug B **device_id** element. It is the name of an RMC-100 application instance for which data is being published.

An example to customize the namespace can be as follows:

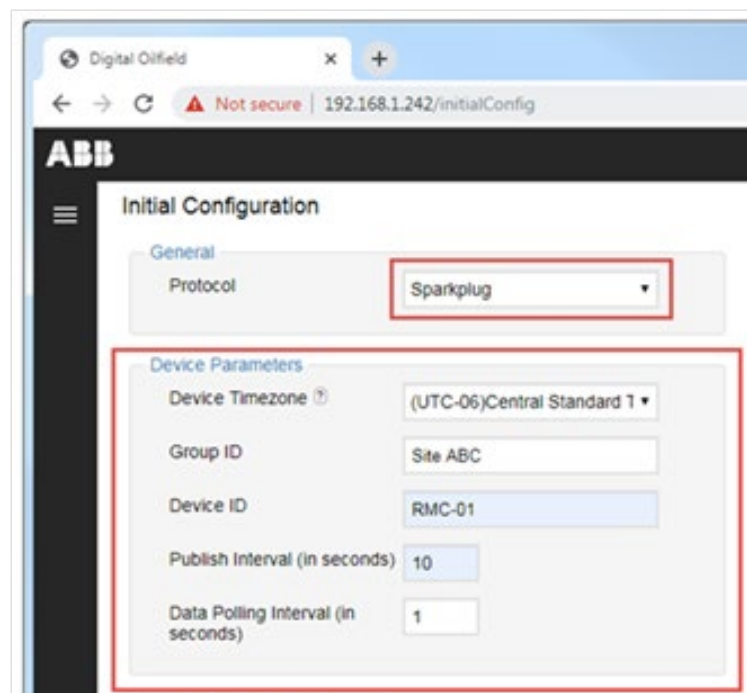
- Assume that there are multiple sites with RMC-100s. Define Group IDs to uniquely identify each field site (For example one site is named: "Site ABC"). Configure this parameter in the RMC-100.
- Assume that there are multiple RMC-100s in each site. Define Device IDs to uniquely identify each RMC-100 at each site (For example one device is named: "RMC-01"). The Group ID/Device ID combination must be unique. Configure the Device ID in the RMC-100.
- Assume that the RMC-01 has several AGA3 application instances for which it will publish data (For example there are several AGA3 instances named: "AGA3-1", "AGA3-2", "AGA3-3", etc.)

For register data from the first instance of the AGA3 application, the topic namespace would be:

spBv1.0/Site ABC/DDATA/RMC-01/AGA3-1

[Figure 3-3](#) shows the **Device Parameter** section in the MQTT configuration page. Group ID and Device ID are required configuration parameters and must match the configuration on the MQTT broker. Note that the configuration of the Group ID, Device ID, and the actual application instance name will construct the topic namespace to publish to. The broker holds that structure of data for subscribers to access. A client such as SCADA would need to point to that topic namespace structure to obtain the data.

Figure 3-3: Configuring Device parameter (reflect topic namespace)



4 Enable MQTT services on the device

The MQTT functionality is disabled by default. There are two ways to enable MQTT, depending on the software build:

- To enable MQTT in RMC-100s with software versions prior to 2105452-045 (or 2106260-015 for RMC-100 LITE), see section [4.1 Enable MQTT from terminal mode](#).
- To enable MQTT in RMC-100s with software versions 2105452-045 (flash 2105457-042) or 2106260-015 (flash 2106229-015) or later, see section [4.2 Enable MQTT from PCCU Entry mode](#).



IMPORTANT NOTE: Enabling MQTT does not require device restart.

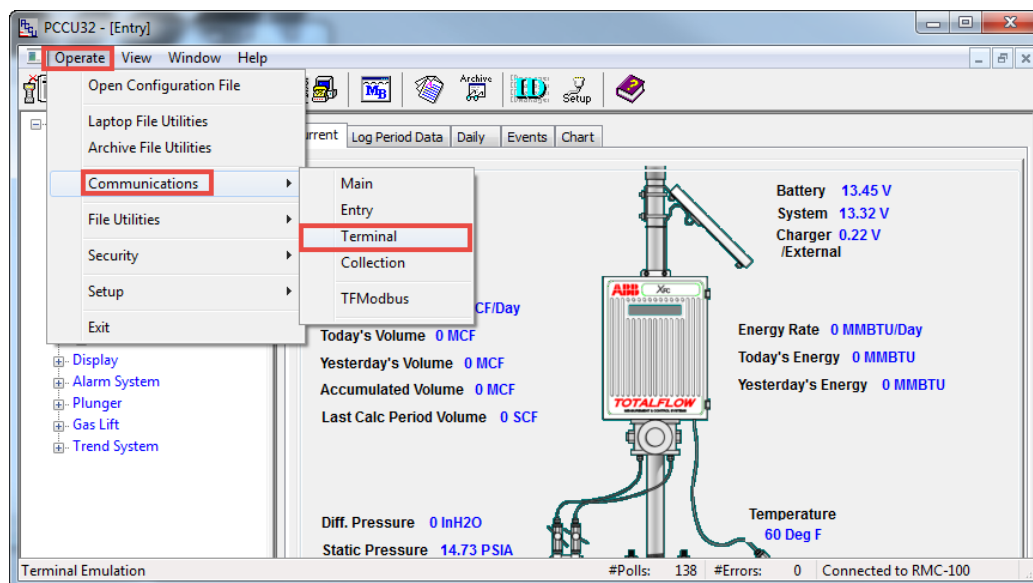
4.1 Enable MQTT from terminal mode

Terminal mode is available from PCCU after connection with the device. Use the USB port for local connection.

To enable MQTT on the device from terminal mode:

1. Connect the laptop to the USB port on the device.
2. Start PCCU.
3. Click the Entry icon to connect with the device.
4. Select **Operate** > **Communications** > **Terminal** from the top menu to go to terminal mode.

Figure 4-1: Access terminal mode on device



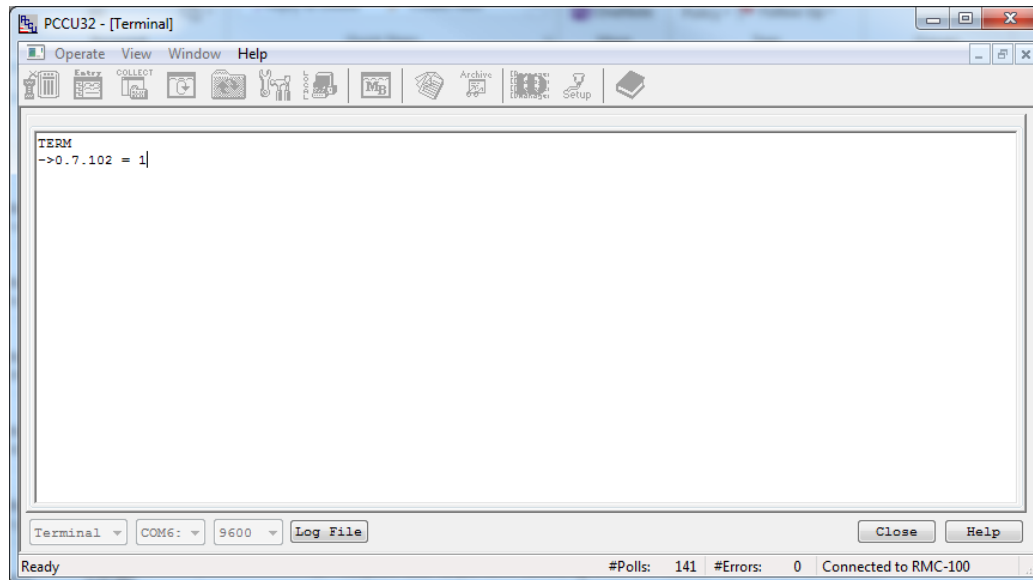
5. To enable MQTT, type **0.7.102 = 1** at the prompt, then press **Enter**.



IMPORTANT NOTE: The terminal mode does not issue a confirmation message after MQTT is enabled. It takes 5 to 7 seconds for the change to take effect.

Enabling MQTT on the device also enables the MQTT configuration interface.

Figure 4-2: Enable MQTT functionality from terminal mode



6. Click **Close** to exit terminal mode.

4.2 Enable MQTT from PCCU Entry mode (recommended)

This procedure is applicable to RMC-100s with customer software package version 2105452-034 (which contains Flash version 2105457-031) or later.

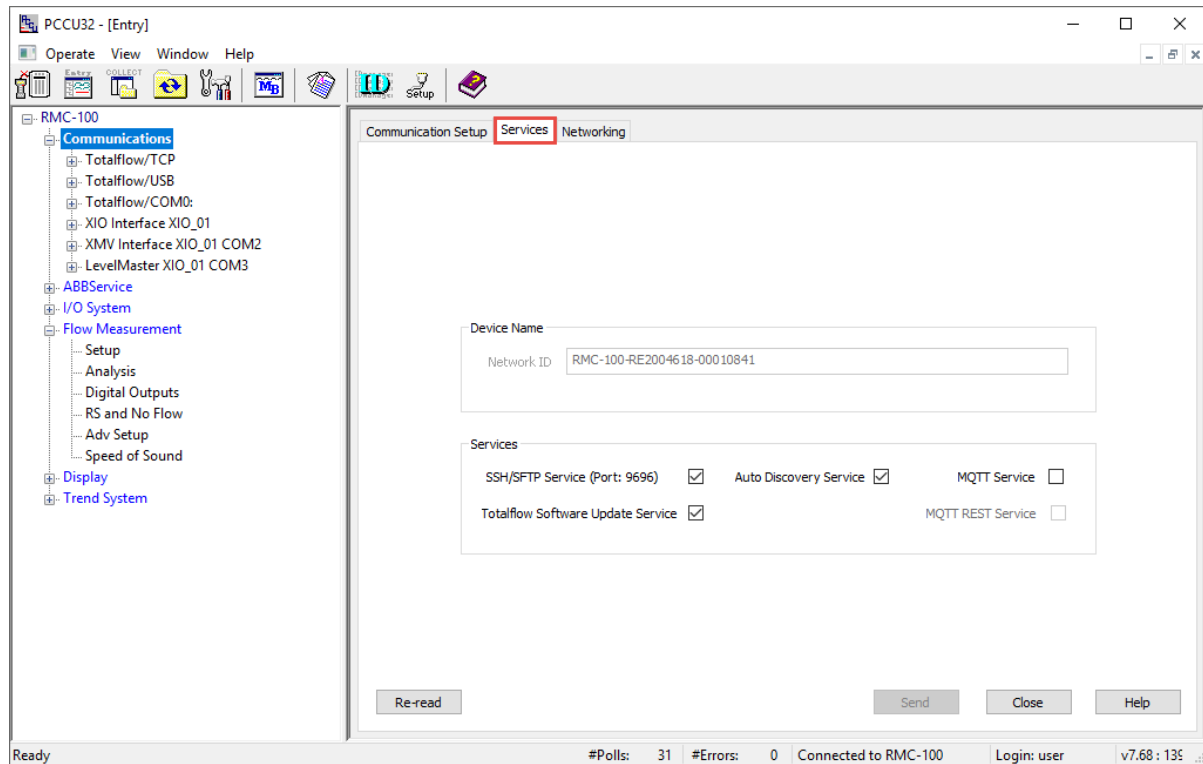
This procedure enables the MQTT services from the **Communication** > **Services** tab. The MQTT services are separated into 2 options:

- The **MQTT Service**: Enables the MQTT (client) process managing the connection to the MQTT broker(s). It should always be enabled if the device is part of an MQTT implementation.
- The **MQTT REST server**: Enables browser-based user access to the MQTT configuration interface on the device. This option can be enabled for configuration and verification purposes only. It is recommended to be disabled after configuration is complete.

To enable MQTT services on the device using PCCU Entry mode:

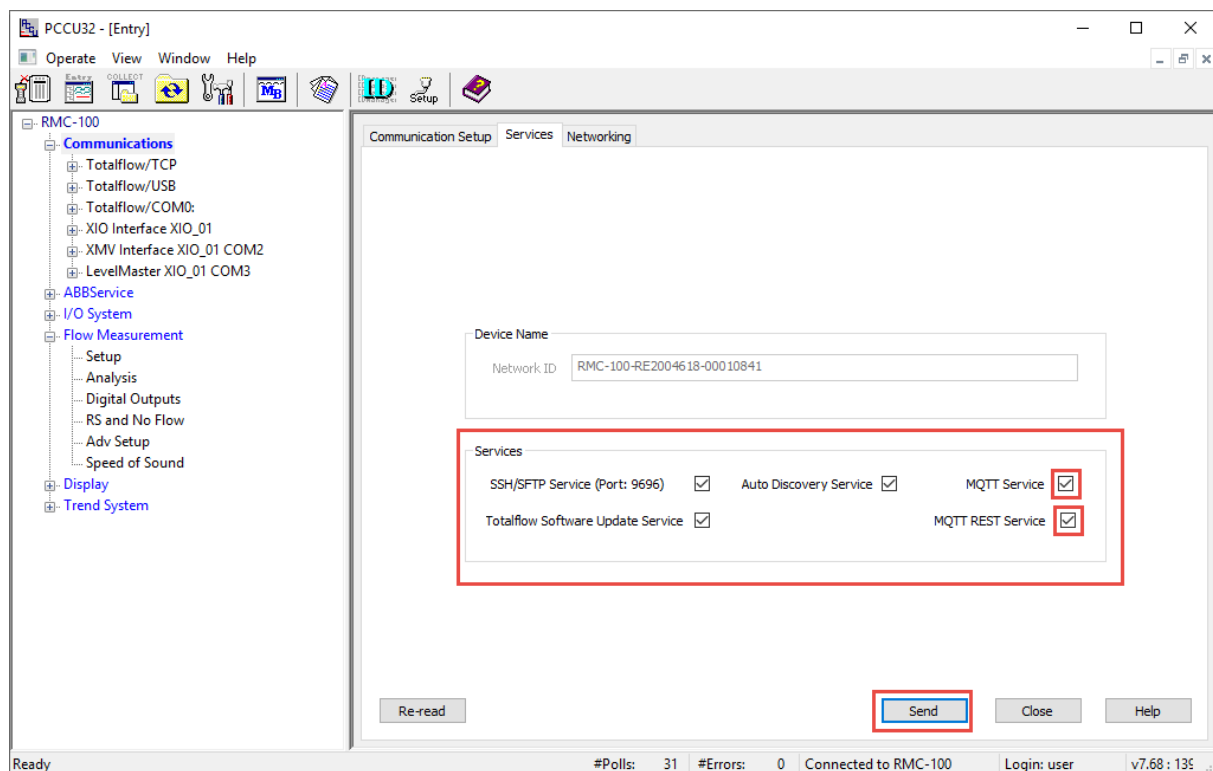
1. Connect the laptop to the USB port on the device.
2. Start PCCU.
3. Click the Entry icon to connect with the device.
4. On the navigation tree, select **Communications**. The Communication Setup screen displays.
5. Select the **Services** tab ([Figure 4-3](#)).

Figure 4-3: Communication Services tab on the RMC-100



6. In the Services section ([Figure 4-4](#)):
 - a. Select **MQTT Service**.
 - b. Select **MQTT REST Service**.
7. Click **Send**.

Figure 4-4: Enable MQTT and REST Services from PCCU Services tab



5 Access the MQTT configuration interface

The MQTT configuration interface supports only configuration for MQTT-related parameters. Perform all other device and application configuration from PCCU.



IMPORTANT NOTE: Local access to the MQTT configuration interface requires that both the laptop and the device are connected to the field network (through Ethernet). Both must be configured with valid IP addresses for this communication. MQTT parameters cannot be configured using USB. You must have the IP address of the device to connect with it.

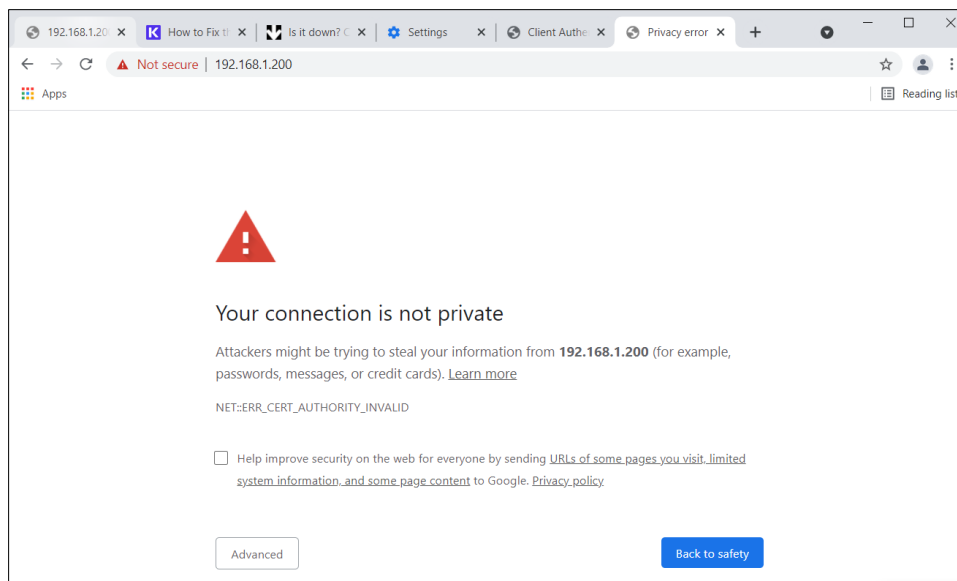
To access the MQTT configuration interface:

1. Connect the laptop and the device to the local field network (Ethernet).
2. On the laptop, start the Chrome browser.
3. Go to the URL address: **https://<Totalflow Device's IP address>:443**. For example, **https://10.127.133.220:443**. A security warning displays.



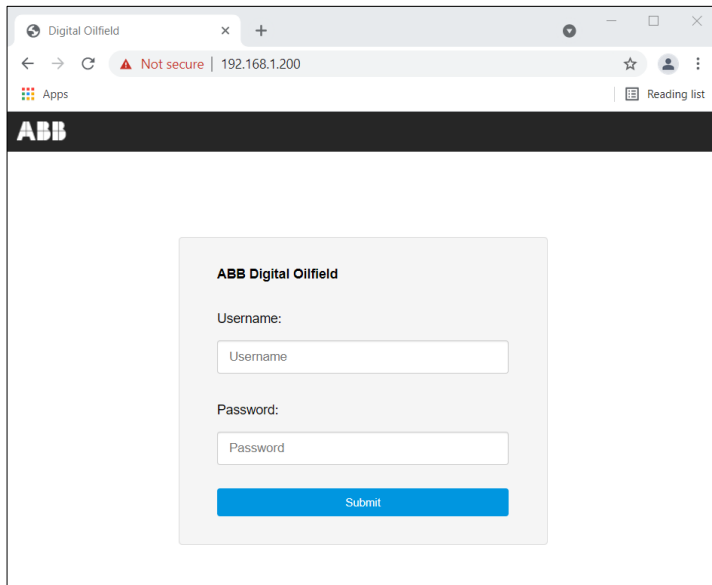
IMPORTANT NOTE: Security warnings displays at first-time login when the end-user browser does not have valid certificates to access the device ([Figure 5-1](#)). The “Not secure” warning in the URL field displays because the browser does not establish the connection on secure mode. The device supports secure connections from end users, but the end user must have valid certificates. New devices shipped from the factory will have certificates and the warning may not appear. The browser can be configured to trust certificates on devices for secure connections. Device certificates must be valid and uploaded on the device. See section [12.5 Upload valid certificates \(for secure user connection\)](#) for more details. **Please note that certificates for secure connection to the device are not the same as the certificates used by a device to connect to a broker.**

Figure 5-1: Security warning message



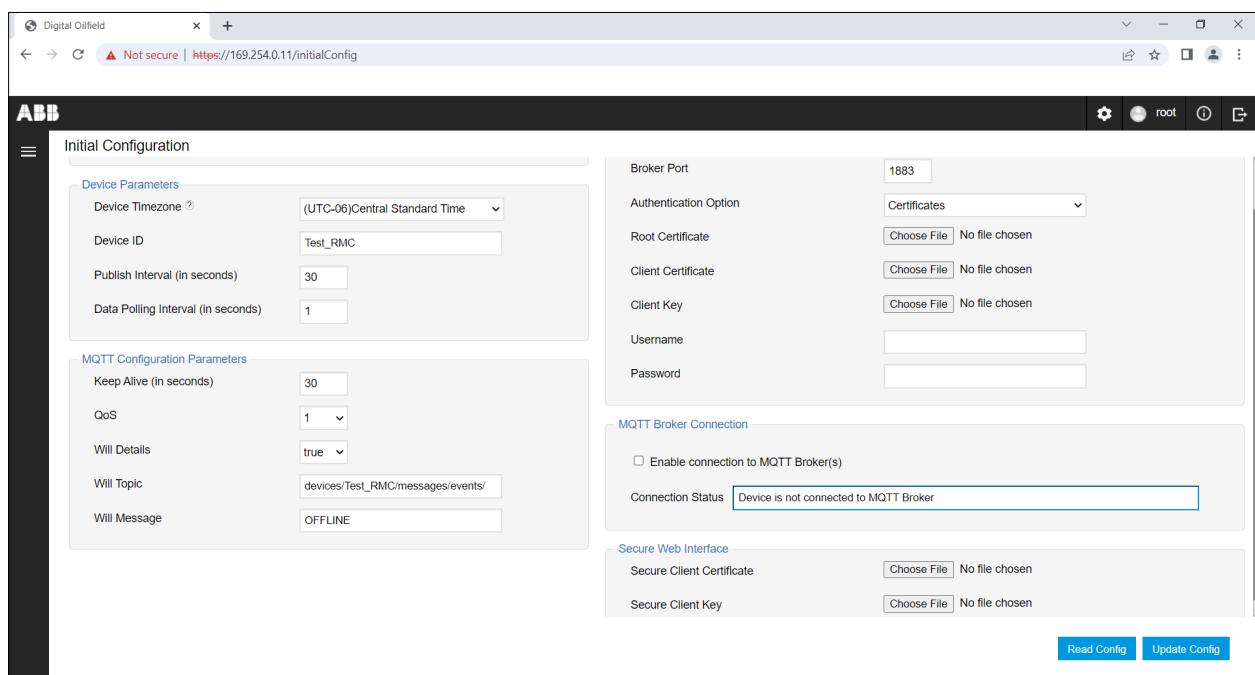
4. Click **Advanced** at the bottom of the screen. Additional security information displays. A link to the device is provided to proceed.
5. Click **Proceed to <device IP address>**. The login screen displays.

Figure 5-2: Device MQTT configuration interface login screen



6. Type **root** into the Username field. Type the default root password, **root@123**, into the Password field.
7. Click **Submit**. The Initial Configuration screen displays ([Figure 5-3](#)).

Figure 5-3: Initial Configuration page



6 Configure MQTT/Sparkplug parameters

Configure the MQTT and Sparkplug parameters on the Initial Configuration page ([Figure 6-1](#)). These parameters are required for correct protocol configuration and successful device-broker connection/data publishing. Some parameters vary depending on the protocol used.

- To retrieve and view the current configuration in the device, click **Read Config**.
- To save new configuration or updates in the device, click **Update Config**.

Figure 6-1: MQTT/Sparkplug configuration (Initial Configuration page)

The screenshot displays the ABB Initial Configuration page for MQTT/Sparkplug. The interface is organized into several sections:

- General:** Protocol is set to 'Standard MQTT Protocol'.
- Device Parameters:** Device Timezone is '(UTC-06)Central Standard Time', Device ID is 'Test_RMC', Publish Interval (in seconds) is '30', and Data Polling Interval (in seconds) is '1'.
- MQTT Configuration Parameters:** Keep Alive (in seconds) is '30', QoS is '1', Will Details is 'true', Will Topic is 'devices/Test_RMC/messages/events/', and Will Message is 'OFFLINE'.
- MQTT Broker Details:** Broker IP/Hostname is '0.0.0.0', Broker Port is '1883', Authentication Option is 'Certificates'. Fields for Root Certificate, Client Certificate, and Client Key are present, each with a 'Choose File' button and 'No file chosen' status.
- MQTT Broker Connection:** The checkbox 'Enable connection to MQTT Broker(s)' is unchecked. The 'Connection Status' field shows 'Device is not connected to MQTT Broker'.

At the bottom right, there are 'Read Config' and 'Update Config' buttons.

The parameters in the Initial Configuration screen set the device for authentication and connection with the MQTT broker. This procedure assumes you have all the required values for those parameters and authentication certificates or credentials.

When using authentication certificates, certificate and key files are copied from the laptop to the device during this procedure. After configuration, these files reside in the device and ensure automatic re-authentication in case of disconnection from the broker. Certificate files do have expiration dates and must be maintained. Certificate generation and management are the sole responsibility of the customer. Ensure you have the certificate files ready before starting configuration.

6.1 Configure for MQTT 3.1.1

MQTT 3.1.1 is the default protocol in the ABB Totalflow device, but associated parameters require configuration specific to your implementation:



IMPORTANT NOTE: When changing the protocol, clear the **Enable Connection to MQTT Broker(s)** option, then select **Update Config**. This disables the device from attempting connection while changing the protocol configuration. Changing protocol when the device is connected to the broker could crash the device's MQTT Service.

1. Refer to [Table 6-1](#) to configure parameters applicable to the MQTT 3.1.1 protocol.
2. When configuration is complete, click **Update Config** to save the configuration in the device.
3. When you enable the device to connect to the MQTT broker, go to section [7 Verify device-broker connection status](#).

Table 6-1: Configuration parameters when using MQTT 3.1.1 protocol

Parameter	Description	Values
General		
Protocol	Specific MQTT protocol used by the device to communicate with the MQTT broker. The protocol specification defines the packet format and types for connection requests and responses between the field device and MQTT server.	Select MQTT Standard Protocol . This option instructs the device to use the MQTT 3.1.1 protocol to communicate with the MQTT broker.
Device Parameters		
Device Time Zone	Standard time associated with the device's geographical location. The Time Zone selected determines the time stamp used for device data and communication messages with the broker. Options available are standard times for several geographical locations, offsets from the Coordinated Universal time (UTC).	Default: Central Standard Time (UTC-6). Select the time zone applicable to the device's location. Ensure the selected option is consistent with the "Time" settings of the device. See the device configuration of the Date/Time parameter in the Station Setup tab using PCCU.
Device ID	Unique identification or name assigned to the device. The MQTT broker keeps track of MQTT clients with this unique ID.	Obtain from the administrator. Type the name or ID assigned to the device. It must match what is defined in the MQTT server. Note: Not to be confused with the Device ID as configured in PCCU.
Publish Interval (In seconds)	The frequency at which the device publishes its application register data to the MQTT Broker	30 (default) Range: 10 to 120
Data Polling Interval (in seconds)	The frequency at which the device reads its application register data.	1 (default) Range: 1 to the Publish Interval value The Polling Interval value affects the age of the published data and the device's CPU utilization: — A shorter interval results in the most recent published data, but a higher CPU utilization. For example, setting the polling interval to 1 (default) results in the most up-to-date published data, but a high CPU use. — A longer interval results in older published data, but lower CPU utilization. For example, setting the polling interval to a value larger than 1 and up to the publish interval results in published data that can be as old as the polling interval.

Parameter	Description	Values
MQTT Configuration parameters		
Keep Alive (in seconds)	Number of seconds after which the broker should send a PING message to the client if no other messages have been exchanged in that time. Configuring this parameter with a non-zero value helps to keep the customer informed of device disconnection from the broker.	<p>30 (default) Range: 0 to 65,535 Shorter keep alive values result in detecting device disconnections early.</p> <p>Notes:</p> <ul style="list-style-type: none"> — It may take up to 1.5 x (Keep Alive value) for the broker to report the disconnection from the device. For example, if the keep alive is 30 seconds, it may take up to 45 seconds for the broker to report the issue. This accounts for processing time on the part of the broker. — A value of 0 deactivates the Keep Alive mechanism. Small values (1-5 seconds for example) may not be effective due to network delays or broker processing times. Configure the value that accounts for your network conditions and server configuration to ensure this mechanism helps you detect disconnections.
QoS	Quality of Service Level on the device-broker connection. It is the agreement between the device and the broker that defines the guarantee of delivery for data the device publishes. Selection of QoS depends on the reliability of the network the devices connect to. The device exchanges messages with the MQTT broker according to the QoS levels defined by the MQTT specification and supported in the device.	<p>Select one of the following values:</p> <ul style="list-style-type: none"> — 0: Best effort delivery. No guarantee of delivery. The broker does not acknowledge receipt of the data and the device does not retransmit the data. — 1: Default. Guarantees at least one-time data delivery. The broker must acknowledge receipt of data message. The device stores the message sent and retransmits it until the broker acknowledges receipt.
Will Details	Feature which allows the device to indicate if it wants the MQTT broker to send a will message, Last Will and Testament (LWT) message, on its behalf. The device sends the LWT message to the broker while connected to the broker specifying details. The broker receives and retains the LWT message. It sends it to other MQTT clients only when it detects the ungraceful disconnection of the device.	<p>Select one of the following:</p> <ul style="list-style-type: none"> — True (default, recommended): The device requests the broker to send the LWT message upon ungraceful device disconnection. — False: The device does not request the broker to send the LWT message upon ungraceful device disconnection.

Parameter	Description	Values
Will Topic	<p>Topic where the broker publishes the Will message after ungraceful device disconnection.</p> <p>Other MQTT clients (applications consuming the device's data) subscribed to this topic receive this notification and are aware of the device disconnection.</p>	<p>The Will topic depends on the definitions set in MQTT server. Obtain this information from the administrator.</p> <p>Type the Will topic string using the correct format [the topic consists of one or more topic levels. Each topic level is separated by a forward slash (topic level separator)]. For example:</p> <p>devices/<device ID>/messages/events/</p> <p>Ensure the device ID in the topic matches the Device ID configured in the Device Parameters section above.</p> <p>The hierarchical structure of the topic allows subject-based filtering by the MQTT broker.</p>
Will Message	Last Will and Testament (LWT) message the broker sends to other MQTT clients on behalf of the device when the device disconnects ungracefully from the MQTT broker (connection loss).	OFFLINE (default)
MQTT Broker Details		
Broker IP/Hostname	The IP address or hostname of the target MQTT server.	Obtain from IT or server administrator.
Broker Port	TCP port assigned to MQTT connections.	<p>Obtain from IT or server administrator. This port is usually one of the standard ports reserved by Internet authorities for MQTT connections:</p> <ul style="list-style-type: none"> — 1883: Reserved port for MQTT over TCP — 8883 (recommended for security): Reserved port for MQTT over TLS (Transport Layer Security protocol). <p>For non-standard ports, verify with your administrator or service provider.</p>
Authentication Option	Defines the method to identify the device as a valid client to the MQTT server.	<p>The device must use the authentication method configured in the server. Select one of the following:</p> <ul style="list-style-type: none"> — Username/Password: The device must present a valid username and a password to request and establish connection with server. — Certificates: X.509-based authentication. The device must present valid certificates (root, client, and key) to request and establish connection with the server. <p>Selecting the Certificates option at first-time configuration requires the upload of valid certificate files to the device. Once uploaded and saved in the device, these files are used automatically every time authentication is required.</p>

Parameter	Description	Values
Root Certificate	Required when Certificates is the selected authentication method.	Click Choose File to locate and select the root certificate file to upload to the device. Note: Be sure to generate file names with the correct extension. An example of a root certificate file name is root.ca.cer.cer .
Client Certificate	Required when Certificates is the selected authentication method.	Click Choose File to locate and select the client certificate file to upload to the device. Note: Be sure to generate file names with the correct extension. An example of a client certificate file name is client-cert.pem .
Client Key	Required when Certificates is the selected authentication method.	Click Choose File to locate and select the client key certificate file to upload to the device. Note: Be sure to generate file names with the correct extension. An example of certificate file name is client-key.pem .
Username	Credential for authentication: — Required when Username/Password is the selected authentication method. — Optional when Certificates is the selected authentication method. Provide if the server is configured to require it.	Type username obtained from administrator.
Password	Credential for authentication: — Required when Username/Password is the selected authentication method. — Optional when Certificates is the selected authentication method. Provide if the server is configured to require it.	Type password obtained from administrator.
MQTT Broker Connection		
Enable Connection to MQTT Broker(s)	Enables the device to request and establish connection with the MQTT server.	— Select option to enable. — Click Update Config and verify connection status.
Connection Status	(Read-only) Displays the state of the device-broker connection when the device is enabled to attempt and establish connection with the target server.	See Table 11-2 for list of status messages.
Secure Web Interface		

Parameter	Description	Values
Secure Client Certificate	<p>Part of the certificate/key file pair used to establish secure user connections to the device's MQTT configuration pages. The certificate file is stored in the device.</p> <p>This certificate file is not the same as the certificate used by the device to authenticate with an MQTT broker.</p>	<p>Default certificate/key file pairs may be already in the device from the factory, but they can expire. When the certificate expires (Refer to section 12.5 Upload valid certificates (for secure user connection)):</p> <ul style="list-style-type: none"> — Obtain or generate new certificate/key file pair. — Click Choose File to locate and select the new certificate file. — Click Update Config to save the file in the device. <p>Note: Be sure to generate certificate file names with the correct extension. An example of a client certificate file name is: <device- name>.cert.pem.</p>
Secure Client Key	<p>Part of the certificate/key file pair used to establish secure user connections to the device's MQTT configuration pages. The key file is stored in the device.</p> <p>This key file is not the same as the key used by the device to authenticate with an MQTT broker.</p>	<p>Default certificate/key file pairs may be already in the device from the factory, but they can expire.</p> <p>When the certificate expires (Refer to section 12.5 Upload valid certificates (for secure user connection)):</p> <ul style="list-style-type: none"> — Obtain or generate new certificate/key pair. — Click Choose File to locate and select the new client key file. — Click Upload Config to save the key file in the device. <p>Note: Be sure to generate certificate file names with the correct extension. An example of a client key file name is: <device- name>.key.pem.</p>

6.2 Configuration for Sparkplug



IMPORTANT NOTE: When changing the protocol, clear the **Enable Connection to MQTT Broker(s)** option, then select **Update Config**. This disables the device from attempting connection while changing the protocol configuration. Changing protocol when the device is connected to the broker could crash the device's MQTT Service.

Change the default MQTT configuration to support Sparkplug:

1. Refer to [Table 6-2](#) to configure parameters applicable to the Sparkplug B protocol.
2. When configuration is complete, click **Update Config** to save the configuration in the device.
3. When you enable the device to connect to the MQTT broker, go to section [7 Verify device-broker connection status](#).



IMPORTANT NOTE: The Sparkplug configuration does not allow user configuration of the QoS. Refer to [Table 6-3](#) for QoS per message type.

Table 6-2: Configuration parameters when using Sparkplug B

Parameter	Description	Values
General		
Protocol	Specific MQTT protocol used by the device to communicate with the MQTT broker. The protocol specification defines the packet format and types for connection requests and responses between the field device and MQTT server.	Select Sparkplug . This option instructs the device to use the Sparkplug B protocol to communicate with the MQTT broker. Sparkplug B optimizes the MQTT protocol for use with SCADA.
Device Parameters		
Device Time Zone	Standard time associated with the device's geographical location. The Time Zone selected determines the time stamp used for device data and communication messages with the broker. Options available are standard times for several geographical locations, offsets from the Coordinated Universal time (UTC).	Default: Central Standard Time (UTC-6) Select the time zone applicable to the device's location. Ensure the selected option is consistent with the Time settings of the device (See the device configuration of the Date/Time parameter in the Station Setup tab using PCCU.)
Group ID	Name assigned to a logical grouping of Sparkplug Edge Nodes. It maps to the group_id element in the topic namespace defined in your implementation.	Type the character string obtained from the server administrator. It is assumed that IT or MQTT server administrators have used appropriate guidelines for naming conventions for the Sparkplug topic namespace structure elements. The Sparkplug specification states that the Group ID (element) should be descriptive, but as small as possible to minimize bandwidth usage.
Device ID	Name assigned to a Sparkplug Edge Node. It maps to the edge_node_id element in the topic namespace defined for your implementation. Note: Not to be confused with the Device ID as configured in PCCU.	Type the character string obtained from the server administrator. The Sparkplug specification states that the edge_node_id is included in every message published and therefore should be as short as possible.
Publish Interval (In seconds)	The frequency at which the device publishes its application register data to the MQTT Broker	30 (default) Range: 10 to 120

Parameter	Description	Values
Data Polling Interval (In seconds)	The frequency at which the device reads its application register data.	<p>1 (default) Range: 1 to the Publish Interval value</p> <p>The Polling Interval value affects the age of the published data and the device's CPU utilization:</p> <ul style="list-style-type: none"> — A shorter interval results in the most recent published data, but a higher CPU utilization. For example, setting the polling interval to 1 (default) results in the most up-to-date published data, but a high CPU use. — A longer interval results in older published data, but lower CPU utilization. For example, setting the polling interval to a value larger than 1 and up to the publish interval results in published data that can be as old as the polling interval.
MQTT Configuration parameters		
Keep Alive (In seconds)	Number of seconds after which the broker should send a PING message to the client if no other messages have been exchanged in that time.	<p>30 (default) Range: 0 to 65,535 Shorter keep alive values result in detecting device disconnections early.</p> <p>Notes:</p> <ul style="list-style-type: none"> — It may take up to 1.5 x (Keep Alive value), for the broker to report the disconnection from the device. For example, if the keep alive is 30 seconds, it may take up to 45 seconds for the broker to report the issue. This accounts for processing time on the part of the broker. — A value of 0 deactivates the Keep Alive mechanism. Small values (1-5 seconds for example) may not be effective due to network delays or broker processing times. Configure the value that accounts for your network conditions and server configuration to ensure this mechanism helps you detect disconnections.
MQTT Broker Details		
Broker IP/Hostname	The IP address or hostname of the target MQTT server.	<p>Obtain from IT or server administrator.</p> <ul style="list-style-type: none"> — Select the Create Additional MQTT Broker link. — Type IP/Port. — Configure based on the selected authentication method
Broker Port	TCP port assigned to MQTT connections.	<p>Obtain from IT or server administrator. This port is usually one of the standard ports reserved by Internet authorities for MQTT connections:</p> <ul style="list-style-type: none"> — 1883: Reserved port for MQTT over TCP — 8883 (recommended for security): Reserved port for MQTT over TLS (Transport Layer Security protocol). <p>For non-standard ports, verify with your administrator or service provider.</p>

Parameter	Description	Values
Authentication Option	Defines the method to identify the device as a valid client to the MQTT server.	<p>The device must use the authentication method configured in the server. Select one of the following:</p> <ul style="list-style-type: none"> — Username/Password: The device must present a valid username and password to request and establish connection with server. — Certificates: X.509-based authentication. The device must present valid certificates (root, client, and key) to request and establish connection with the server. <p>Selecting the Certificates option at first-time configuration requires the upload of valid certificate files to the device. Once uploaded and saved in the device, these files are used automatically every time authentication is required.</p>
Root Certificate	Required when Certificates is the selected authentication method.	<p>Click Choose File to locate and select the root certificate file to upload to the device.</p> <p>Note: Be sure to generate file names with the correct extension. An example of a root certificate file name is root.ca.cer.cer.</p>
Client Certificate	Required when Certificates is the selected authentication method.	<p>Click Choose File to locate and select the client certificate file to upload to the device.</p> <p>Note: Be sure to generate file names with the correct extension. An example of a client certificate file name is client-cert.pem.</p>
Client Key	Required when Certificates is the selected authentication method.	<p>Click Choose File to locate and select the client key file to upload to the device.</p> <p>Note: Be sure to generate file names with the correct extension. An example of certificate file name is client-key.pem.</p>
Username	<p>Credential for authentication:</p> <ul style="list-style-type: none"> — Required when Username/Password is the selected authentication method. — Optional when Certificates is the selected authentication method. Provide if the server is configured to require it. 	Type username obtained from administrator.
Password	<p>Credential for authentication:</p> <ul style="list-style-type: none"> — Required when Username/Password is the selected authentication method. — Optional when Certificates is the selected authentication method. Provide if the server is configured to require it. 	Type password obtained from administrator.
MQTT Broker Connection		
Enable Connection to MQTT Broker(s)	Enables the device to request and establish connection with the MQTT server.	<ul style="list-style-type: none"> — Select option to enable. — Click Update Config and verify connection status.

Parameter	Description	Values
Connection Status	(Read-only) Displays the state of the device-broker connection when the device is enabled to attempt and establish connection with the target server.	See Table 11-2 for a list of status messages.
Secure Web Interface		
Secure Client Certificate	<p>Part of the certificate/key file pair used to establish secure user connections to the device's MQTT configuration pages. The certificate file is stored in the device.</p> <p>This certificate file is not the same as the certificate used by the device to authenticate with an MQTT broker.</p>	<p>Default certificate/key file pairs may be already in the device from the factory, but they can expire. When the certificate expires (Refer to section 12.5 Upload valid certificates (for secure user connection)):</p> <ul style="list-style-type: none"> — Obtain new certificate/key file pair. — Click Choose File to locate and select the new certificate file. — Click Update Config to save the file in the device. <p>Note: Be sure to generate file names with the correct extension. An example of a client certificate file name is: <device- name>.cert.pem.</p>
Secure Client Key	<p>Part of the certificate/key file pair used to establish secure user connections to the device's MQTT configuration pages. The key file is stored in the device.</p> <p>This key file is not the same as the key used by the device to authenticate with an MQTT broker.</p>	<p>Default certificate/key file pairs may be already in the device from the factory, but they can expire. When the certificate expires (Refer to section 12.5 Upload valid certificates (for secure user connection)):</p> <ul style="list-style-type: none"> — Obtain new certificate/key file pair. — Click Choose File to locate and select the new certificate file. — Click Update Config to save the file in the device. <p>Note: Be sure to generate file names with the correct extension. An example of a client key file name is: <device- name>.key.pem.</p>

[Table 6-3](#) shows the QoS level used per message type by the Sparkplug protocol where applicable.

Table 6-3: QoS and Retain flag rules in Sparkplug

Message type	QoS Level	Retain flag	Notes
Connecting			
WILL (NDEATH)	1	False	
Publishing			
NBIRTH	0	False	
NDATA	0	False	
NDEATH	0	False	The QoS level for this message type is not defined in the Sparkplug specification. It is assumed that it is the same level as the one used for the NBIRTH message. The RMC-100 does not publish this topic
NCMD	0	False	The RMC-100 does not publish this topic

Message type	QoS Level	Retain flag	Notes
DDATA	0	False	
DDEATH	0	False	The RMC-100 publishes this topic when an application is removed from the list of applications to publish data for (Application Configuration page), or when the group/device ID changes.
DCMD	0	False	RMC-100 does not publish this topic
Subscribing			
NCMD	1	Not Applicable	
DCMD	1	Not Applicable	
STATE	1	Not Applicable	The Sparkplug specification does not define the QoS level for subscription to the STATE topic. The Host Application publishes the STATE message using QoS=1. The RMC-100 subscribes to the STATE topic using the same QoS (1).

7 Verify device-broker connection status

This procedure assumes that the correct target MQTT server details are configured. On the Initial Configuration screen, in the **MQTT Broker Connection** section:

1. Verify that **Enable Connection to MQTT Broker(s)** is selected. If not, select the option ([Figure 7-1](#)) and then click **Update Config**.

Figure 7-1: Enable the connection with MQTT broker

MQTT Broker Connection

☒ Enable connection to MQTT Broker(s)

Connection Status: Device is not connected to MQTT Broker

2. Observe the **Connection Status** field.
3. Verify that the field displays: "Device is connected to MQTT Broker", and that it displays the broker's IP and port. If the connection is not successful, see section [11.1 Troubleshooting user-device connection](#)

Figure 7-2: Verify device-broker connection status

MQTT Broker Connection

☒ Enable connection to MQTT Broker(s)

Connection Status: Device is connected to MQTT Broker (IP: 192.168.1.71, Port: 1883)

4. When the connection is successful, proceed to section [8 Configure applications](#) then proceed to section [9 Configure registers](#).



IMPORTANT NOTE: The device's MQTT implementation is designed to automatically re-establish connection to the broker in the event of a restart, network failure or disconnection.

8 Configure applications

Configure the applications that the device publishes data for. The device transfers or sends application data to through the broker. Data is kept on the service provider or private network database(s) depending on the implementation.

IMPORTANT NOTE: The procedures in this section assume that the required applications are already instantiated, configured, and enabled on the device. Use PCCU to add and configure additional applications or instances if necessary. The device application configuration in this section does not provide the ability for full application configuration.

IMPORTANT NOTE: In PCCU, check Application and Holding Register names to ensure names do not include or start with special characters (.,\|@#\$%^&*()[]{}|!`~:;'\ "<>?+). These characters will be replaced by the `_` character.

8.1 Application configuration page

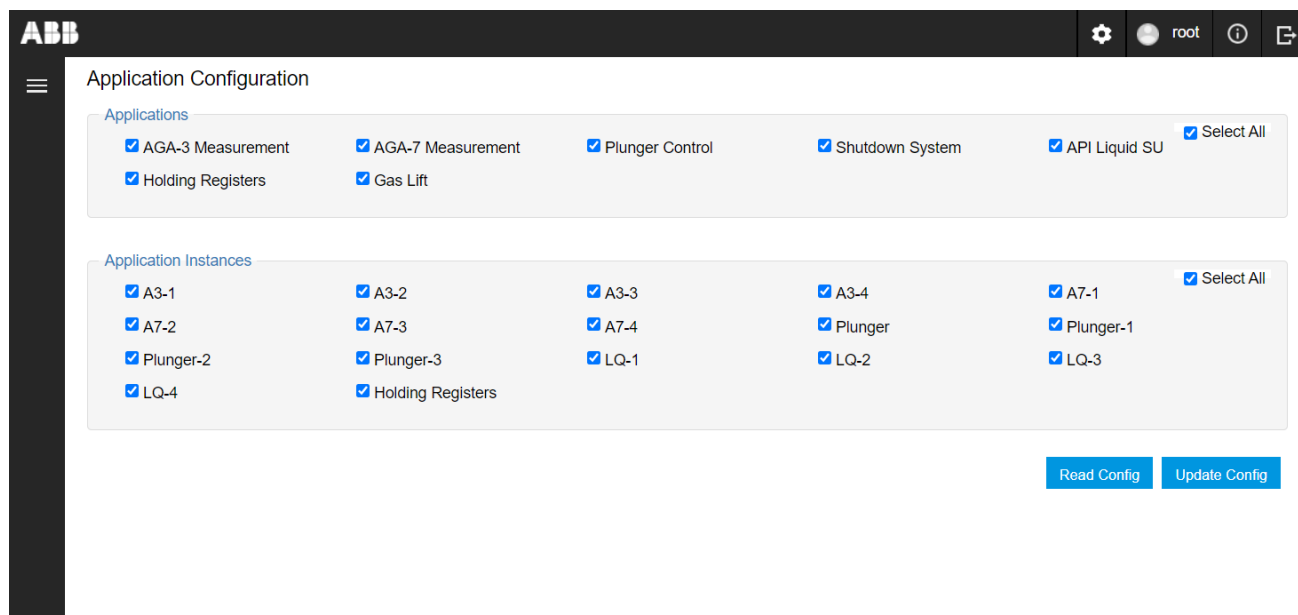
[Figure 8-1](#) shows the Application Configuration web page with the list of applications and application instances configured in the field device. Use checkboxes to configure preferences:

- Check **Select All** to publish data for all application and application instances shown in the list.
- Check an individual application or instance to enable the device to publish that data.
- Clear an individual application or instance to disable the device from publishing that data.

Function buttons are available to view and update configuration:

- **Read Config:** retrieves and displays the current applications and instances and their setting for data publishing.
- **Update Config:** saves new data publishing settings for the current application and instances after updates.

Figure 8-1: Application Configuration web page



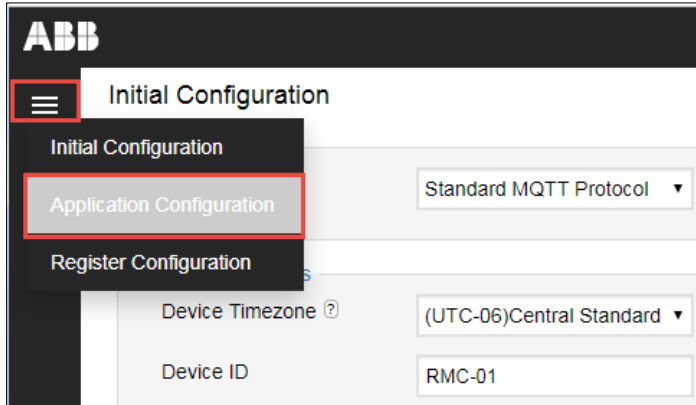
IMPORTANT NOTE: The Applications section in the Application Configuration page displays the applications supported by the cloud interface, even if not instantiated. The Application Instances section displays only those instances instantiated from PCCU.

8.2 Measurement and control applications

To select applications data is published for:

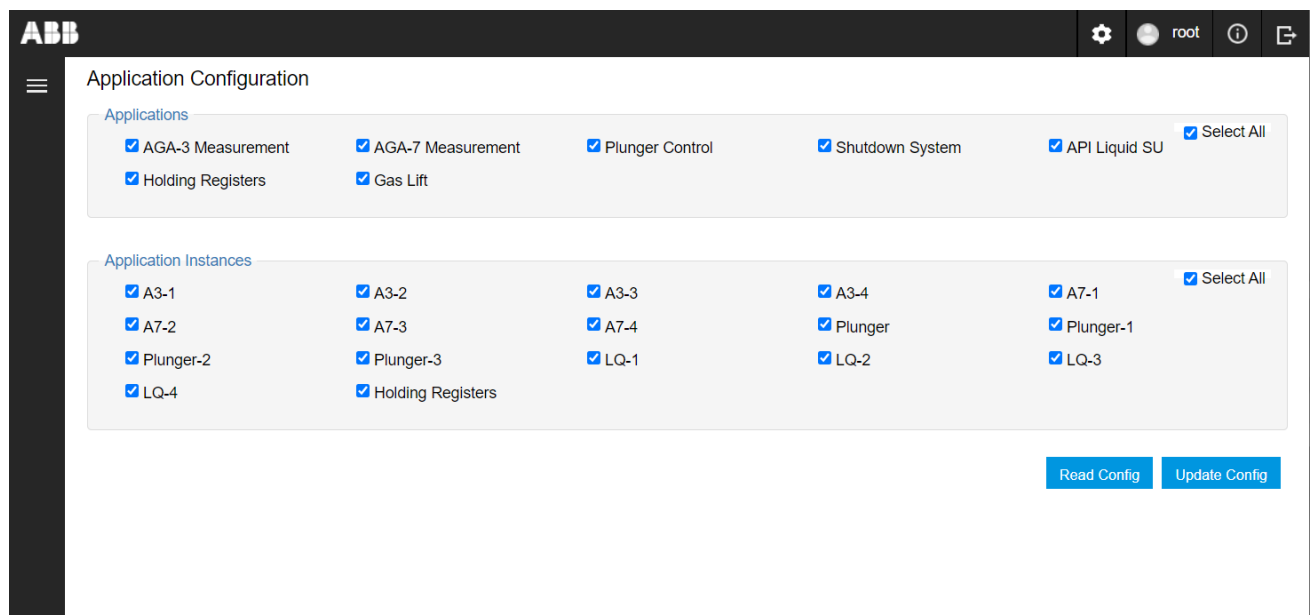
1. Click the menu icon on the left of the Initial Configuration screen and select **Application Configuration**.

Figure 8-2: Navigate to Application Configuration



2. On the Application Configuration screen, select specific applications and instances or keep the **Select All** option checked.

Figure 8-3: Application Configuration page

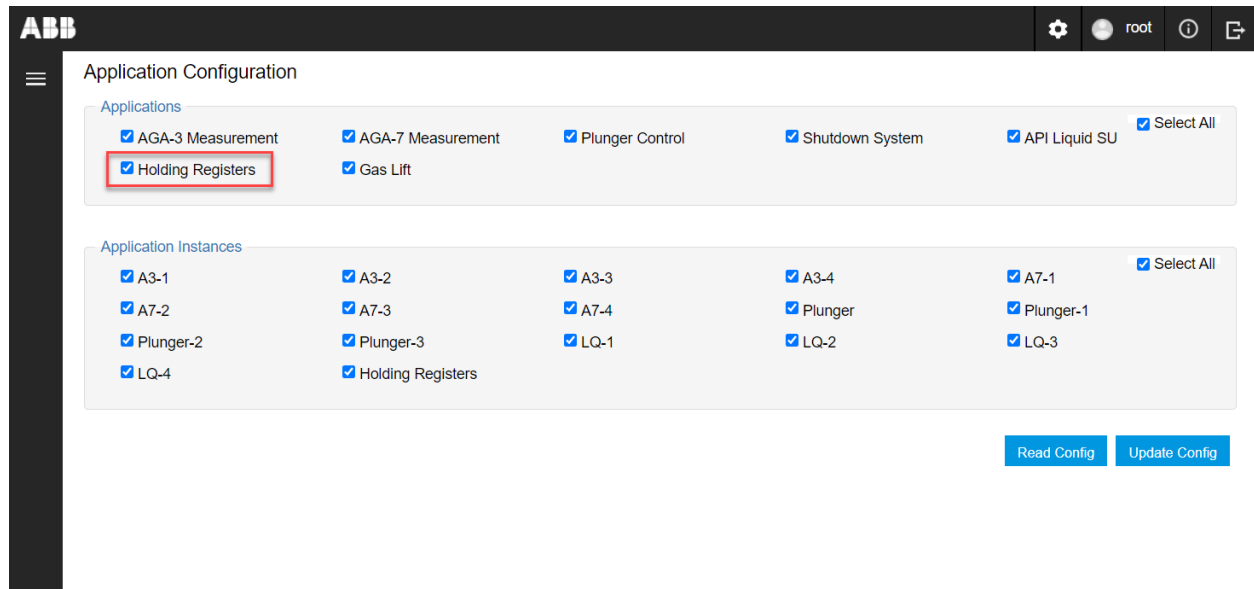


3. Click **Update Config** if you changed default selections. A confirmation for the update displays.
4. Click **Close** when the configuration update is successful.

8.3 Holding Registers application (private networks only)

The Holding Registers application allows the user to custom define how to store values of interest in specific device register ranges. This application is customized per user requirements. The registers are not pre-defined as with the other supported applications.

Figure 8-4: Support for Holding Registers (private network only)



IMPORTANT NOTE: At time of this writing, this application is supported only in private customer implementations, not on service provider clouds or on systems connected to the internet. If you are connecting to a MQTT broker on the cloud, clear the Holding Register check box from the application configuration page to disable publishing by this application.

9 Configure registers

Configure the registers that the device publishes data for. The device sends specific application register data through the broker.

IMPORTANT NOTE: In PCCU, check Application and Holding Register names to ensure names do not include or start with special characters (.,\|@#\$\$%^&*()[\]{}|!`~:;'\ "<>?+). These characters will be replaced by the `_` character.

9.1 Number of registers published

When planning register data publishing on the MQTT or Sparkplug broker, consider the limits for the applications:

For the **Holding Registers** application, the number of registers (variables) that the device can publish for is less than 2500. To calculate the number of registers:

- If:
 - x = Number of variables with Persistent or Non-Persistent type
 - z = Number of variables with Indirect or Indirect Change type
 - t = Number of tabs in the Holding Register application
- Then, the number of variables (n) that the device can publish is:
$$n = x + z$$
- Where:
$$(4t + 1 + 2x + 3z) < 2500$$

For the other supported applications (listed in section [2 Application data publishing](#)), the device can publish data for 2500 registers.

9.2 Register configuration page

[Figure 9-1](#) shows the Register Configuration web page. The page displays the register list for the selected application and specific instance. The first application and its first instance are selected by default. Select the application and instances of interest to view other registers.

The page automatically classifies and displays the registers in categories. These categories might vary based on the application type. For example, for measurement applications, register options are organized in categories such as aggregate, application, and composition registers. These register categories might combine parameters available across different tabs in PCCU or reflect the same parameters as the PCCU tabs.

Options to configure register data publishing:

- Select the application and the app instance of interest to display the specific register list.
- Check **Select All Registers** to publish data for all registers for the selected application and instance or select the individual required registers.
- Use Search to configure specific registers or the pagination buttons at the bottom of the screen to see registers per alphabetical order. The screen also has a scroll bar to navigate while searching for specific registers.

Function buttons are available to view and update configuration:

- **Read Config**: retrieves and displays current register selections for publishing.
- **Update Config**: saves new register selections for publishing after updates.

Figure 9-1: Register configuration web page

The screenshot shows the ABB Register Configuration web page. At the top, there is a search bar. Below it, there are sections for 'Select Application' and 'Select App Instance'. The 'Aggregate' section contains checkboxes for 'Today's Volume', 'Today's Mass', 'Today's Energy', 'Yesterday's Volume', 'Yesterday's Mass', 'Yesterday's Energy', 'Accumulated Volume', 'Accumulated Mass', and 'Accumulated Energy'. The 'Application' section contains checkboxes for 'Static Pressure', 'Differential Pressure', 'Flow Rate', 'Flowing Temperature', 'Energy Rate', 'Mass Rate', 'Device/APP ID', and 'Tube Description'. At the bottom, there is a pagination bar with buttons labeled 'ALL', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', and '>'. There are also 'Read Config' and 'Update Config' buttons on the right side of the pagination bar.



IMPORTANT NOTE: Some of the registers on the register configuration page are required and will always be enabled. The configuration interface does not allow users to disable the publishing of those registers. Required registers display grayed-out check boxes (See highlighted examples in [Figure 9-2](#)).

Figure 9-2: Required registers examples (read-only)

ABB

Register Configuration

Select Application : ☒ AGA-3 Measurement ☐ AGA-7 Measurement ☐ Plunger Control ☐ Shutdown System
☐ API Liquid SU ☐ Holding Registers ☐ Gas Lift

Select App Instance : ☒ A3-1 ☐ A3-2 ☐ A3-3 ☐ A3-4

☒ Select All Registers

Aggregate

☒ Today's Volume ☒ Today's Mass ☒ Today's Energy ☒ Yesterday's Volume ☒ Select All
☒ Yesterday's Mass ☒ Yesterday's Energy ☒ Accumulated Volume ☒ Accumulated Mass
☒ Accumulated Energy

Application

☒ Static Pressure ☒ Differential Pressure ☒ Flow Rate ☒ Flowing Temperature ☒ Select All
☒ Energy Rate ☒ Mass Rate ☒ Device/APP ID ☒ Tube Description

ALL A B C D E F G H I J K L M N >

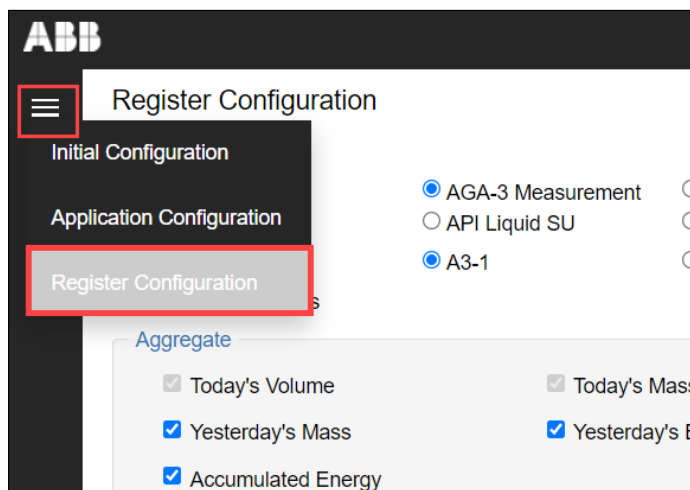
Read Config Update Config

9.3 Configure measurement and control applications

To select registers data is published for:

1. Click the menu icon and select **Register Configuration** ([Figure 9-3](#)).

Figure 9-3: Navigate to Register Configuration



2. On the Register Configuration page ([Figure 9-4](#)), select the application and instance.
3. Select specific registers or click **Select All**. Use the search or pagination filters to locate specific registers if not selecting all registers.
4. Repeat steps 2-3 for each application instance.
5. Click **Update Config** if defaults are changed.
6. Click **Close** after update confirmation.

Figure 9-4: Register Configuration page

ABB

Register Configuration

Select Application : ☒ AGA-3 Measurement ☐ AGA-7 Measurement ☐ Plunger Control ☐ Shutdown System
☐ API Liquid SU ☐ Holding Registers ☐ Gas Lift

Select App Instance : ☒ A3-1 ☐ A3-2 ☐ A3-3 ☐ A3-4

☒ Select All Registers

Aggregate

☒ Today's Volume ☒ Today's Mass ☒ Today's Energy ☒ Yesterday's Volume ☒ Select All
☒ Yesterday's Mass ☒ Yesterday's Energy ☒ Accumulated Volume ☒ Accumulated Mass
☒ Accumulated Energy

Application

☒ Static Pressure ☒ Differential Pressure ☒ Flow Rate ☒ Flowing Temperature ☒ Select All
☒ Energy Rate ☒ Mass Rate ☒ Device/APP ID ☒ Tube Description

ALL A B C D E F G H I J K L M N >

Read Config Update Config

9.4 Configure Holding Registers

Holding Registers publish data based on the custom definitions. To enable Holding Registers for publishing:

1. Select **Holding Registers** from the Register Configuration page.
2. Select the Holding Register application instance.
3. Leave the **Select All Registers** option checked to publish all data, or clear to select specific registers.

Figure 9-5: Configure Holding Registers

ABB

Register Configuration

Select Application : ☐ AGA-3 Measurement ☐ AGA-7 Measurement ☒ Holding Registers ☐ Plunger Control ☐ Shutdown System
☐ API Liquid SU ☐ Gas Lift

Select App Instance : ☒ Holding Registers

☒ Select All Registers

Application

☒ Number of Arrays ☒ Select All

Byte

☒ Byte-0 ☒ Byte-1 ☒ Byte-2 ☒ Byte-3 ☒ Select All
☒ Byte-4 ☒ Byte-5 ☒ Byte-6 ☒ Byte-7
☒ Byte-8 ☒ Byte-9

ALL A B C D E F G H I J K L M N >

Read Config Update Config

4. Select custom register definitions to publish specific data. Use the search or pagination buttons to locate specific registers or use the scroll bar to navigate the screen. In the example below, several custom definitions display. They can be selected to publish their values.

Figure 9-6: Select to publish

The screenshot shows the ABB Register Configuration web interface. The browser address bar indicates the URL is 192.168.1.200/registerConfig. The interface includes a search bar, a sidebar with the ABB logo, and a main content area. The 'Select Application' section has radio buttons for AGA-3 Measurement, AGA-7 Measurement, Plunger Control, and Shutdown System. The 'Select App Instance' section has radio buttons for API Liquid SU and Holding Registers. The 'Select All Registers' checkbox is checked. The 'Float' section lists registers: dp alarm max, sp alarm max, temp alarm max, Float-3, Float-4, Pulse, Constant "1", Float-7, Float-8, Float-9, and Select All. The 'Int16' section lists registers: Int16-0, Int16-1, Int16-2, Int16-3, and Select All. The 'Read Config' and 'Update Config' buttons are at the bottom right.

5. Click **Update Config**.
6. Click **Close** after the update confirmation.

IMPORTANT NOTE: Once all application and register configuration is completed, and the permanent broker is available for connection, make sure to configure the correct broker parameters before leaving the field. Verify that the connection to the broker is successful.

9.5 Holding Register renaming (Ignition Designer® users)

When a holding register is renamed, the register node is not automatically renamed in Ignition Designer. The node's tag remains with the previous name as a "Good" tag. Remove the node from the system to remove the tag. This allows the Ignition Designer to automatically repopulate the node with renamed tag.

10 Using the MQTT configuration on another device

An ABB device configuration file stores the MQTT configuration. Once the MQTT configuration is completed and verified successfully, you can plan to use the configuration file on other device(s) to save configuration time. Before you load a configuration file in another device, review the following:

- A configuration file retains the MQTT Device ID.
- The MQTT Device ID must be unique for each device and must be correctly configured before the device is allowed to establish connection to the broker.
- A device begins to publish data as soon as connection is established.

NOTICE – Loss of data. Data publishing with a duplicate Device ID will cause data corruption and must be prevented.

Follow standard procedures to save and upload configuration files using PCCU. If not familiar with this process, see the product user manuals.

To save and reuse MQTT configuration on another device:

1. Save the source device configuration:
 - a. Make sure to complete and successfully verify the MQTT configuration from the device's MQTT web interface.

- b. Use the standard process to save the device configuration from PCCU's entry mode and 32-bit loader.
 - c. Take note of the location of the saved configuration file to be uploaded on the target device next.
2. Use standard process (32-Bit Loader) to upload source configuration on the target device(s).
3. Update the MQTT configuration of the target device(s):
 - a. Connect to the device's MQTT web interface. Follow the steps described in section [6 Configure MQTT/Sparkplug parameters](#) to update configuration specific for the device, making sure to assign a unique Device ID.
 - b. When all parameters are correctly configured, select the **Enable Connection to MQTT Broker(s)** option.
 - c. Click **Update Config**.
 - d. Verify that the device can connect to the broker.

11 Troubleshooting during initial configuration

The following sections provide details to help you troubleshoot and resolve issues for:

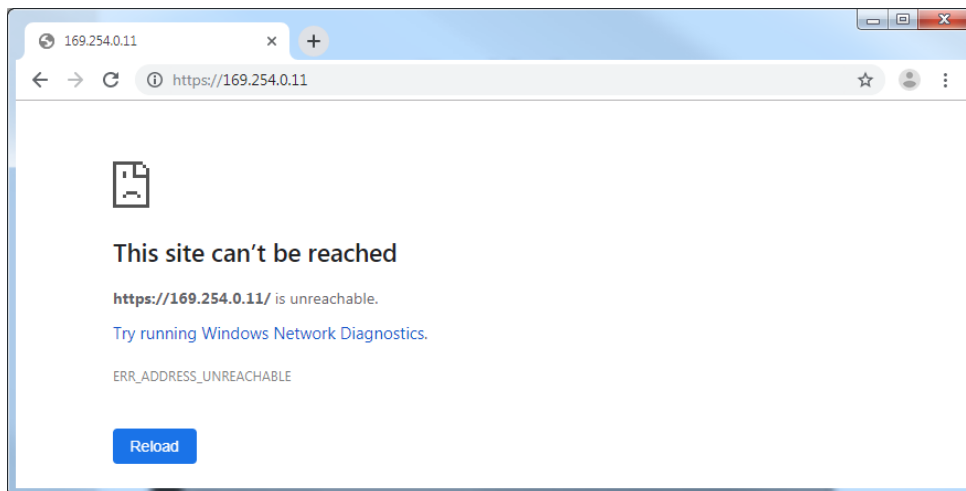
- User-device connections
- Device-MQTT broker connections
- Device-MQTT broker authentication

11.1 Troubleshooting user-device connection

User-device connection failure is the inability to connect to the device from the browser. There can be many reasons for failure.

[Figure 11-1](#) shows a typical browser error message displayed after an attempt to access a device that may be blocked by a firewall, disconnected from the network, or not correctly configured.

Figure 11-1: User-device connection failure (failed device network connection)



[Figure 11-2](#) shows the error message displayed when attempting to connect to a device that has MQTT functionality disabled.

Figure 11-2: User-device connection failure (MQTT services disabled on device)

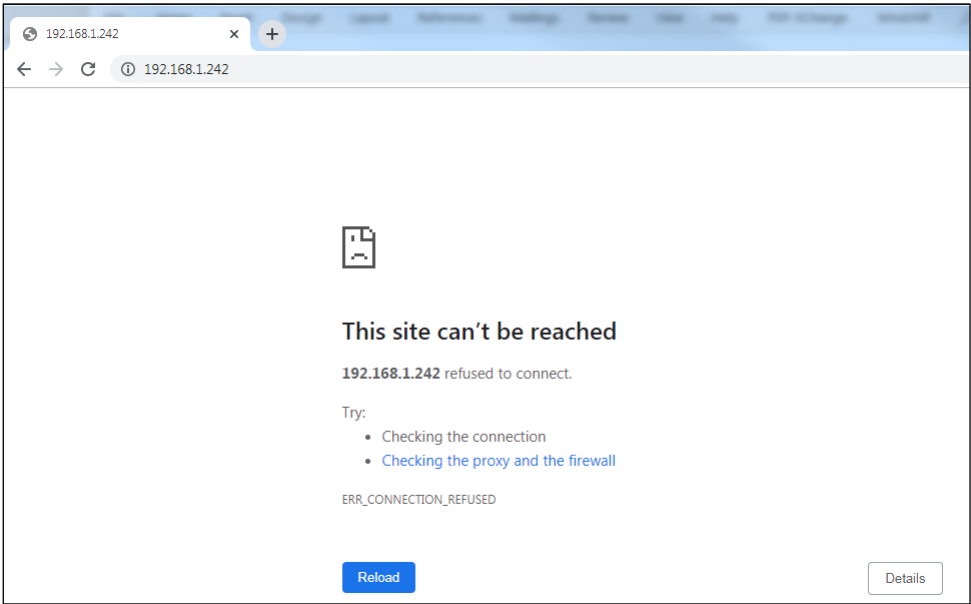


Table 11-1 displays a basic checklist to troubleshoot and resolve failure to connect to the device’s web interface.

Table 11-1: Causes for user failure to connect with device

Problem	Cause	Resolution
Connection failure	Disabled MQTT	— Enable MQTT as described in section 4 .
	Incorrect URL	— Verify the URL required for the device. It should include the device IP address and the TCP port. For example: <code>http://<device’s IP address>:443</code> — Type the correct URL on the browser and retry connection
	Incorrect IP configuration on the field device	— Verify the device’s IP parameter configuration. — Update to a valid configuration. — Restart device as necessary for IP configuration to take effect.
	Incorrect IP configuration on the laptop	— Obtain compatible IP parameters or verify the laptop has an IP address (if using DHCP). — Verify the laptop’s IP configuration and correct as necessary.
	Access to the device’s network port may be blocked or network is experiencing heavy traffic	— Verify field network connection or equipment configuration for possible port blocking and change configuration as required. — If port access is open, verify network performance on the Ethernet port. Use PCCU to monitor or troubleshoot Ethernet port performance/traffic. See Ethernet help topics.
	Connection failure on the field device network	Onsite: — Verify Ethernet cabling or connectors are intact. Ensure the Ethernet cable connects the device and network equipment. — Ensure network equipment is operational and the network link is up. — Ping the device from the laptop. The device with a good network connection responds to the ping.

11.2 Troubleshooting device-broker connection

[Table 11-2](#) shows the messages displayed in the Connection Status field in the Initial Configuration screen. If the device-broker connection is not successfully established, follow the resolution instructions to correct the issue. Be sure to obtain correct configuration parameters from the broker administrator. The device must have the correct target broker parameters for successful connection.

Table 11-2: Connection Status messages

Connection Status message	Description	Resolution
MQTT Service not responding	The MQTT service is busy or stuck and not able to respond to the REST interface.	Enable or Restart MQTT client in the ABB device: On PCCU entry mode, go to Communication>Services tab Verify that MQTT Service is selected. If not, select and then click Send .
Device is not connected to MQTT Broker	The device is not able to connect to the broker.	Verify that the IP address and TCP port for the destination broker are configured correctly. Verify that the authentication method is configured correctly. If certificates are used, ensure the certificates are valid and uploaded on the device.
Device is connected to MQTT broker (IP: nnn.nnn.nnn.nnn, Port:1883)	The device has successfully established a connection with the MQTT broker. The broker IP address and the TCP port it listens for the MQTT client, are displayed. The TCP port can be 1883 or 8883 (when using certificates as authentication method)	No action required on the device. A successful connection allows the device to publish data as configured.
Trying to reconnect to MQTT Broker	The device is attempting to re-establish connection with the MQTT broker.	Check for network connectivity. Check that the broker is still connected to the network. Check that the broker (system) is not down.
Device is connected to MQTT Broker but the STATE of the Primary Host Application is OFFLINE or unpublished.	Displays when the application processing the data published by the device (for example, the customer's SCADA system) is offline, that is, not actively receiving the data published by the device.	No action required on the device. The ABB Totalflow MQTT client implementation does not support the "Primary Host Affinity" feature. Device publishing activity will not be affected by the state of the Primary Host Application (ONLINE or OFFLINE). The lack of this feature does not prevent the device from publishing the data on the broker. Data will be published at the configured interval if the client-broker connection is maintained. Data retrieval by the Primary Host Application, once ONLINE, is an independent process and does not affect the device publishing activity.

11.3 Troubleshooting authentication

[Table 11-3](#) shows error messages that may display during the configuration of the authentication method. Incorrect credentials or certificates will prevent successful device-broker connection.

- For authentication using Username/Password, errors will display when the incorrect credentials are configured. Be sure to obtain the correct credentials from the IT or Broker administrator.
- For authentication using Certificates, errors will display when the incorrect certificate files are loaded. If certificates are generated in-house, they must have the correct format and be valid. This responsibility may fall on the IT or broker administrator, who must provide the correct files to those configuring the device. Make sure you know which files are the root, client, and key files. Each file is different in name and file extension.

Table 11-3: Authentication configuration errors

Message	Description	Resolution
Invalid- Root Certificate	Message displays when using Certificates as the Authentication option and you have tried to load a certificate file. The root certificate provided is incorrect or in the incorrect format.	<ul style="list-style-type: none"> — Click Close to close error message. Obtain or verify location of the correct root certificate. — Click Choose File again and load the correct file. — Continue configuration. — Click Update Configuration.
Invalid- Root Certificate, Client, Key	Message displays when using Certificates as the Authentication option and you have tried to load the certificate files. The files provided are incorrect or invalid.	<ul style="list-style-type: none"> — Click Close to close error message. Obtain or verify location of all the certificate files (root, client, and key). — Click Choose File again for each of the three files and load the correct files. — Continue configuration. — Click Update Configuration.
Invalid-Username	Message displays when using Username/Password as the Authentication Option and you have provided credentials that do not match what is configured on the MQTT server.	<ul style="list-style-type: none"> — Click Close to close error message. Obtain or verify Username is correct. — Type correct Username. — Continue configuration. — Click Update Configuration.
Incorrect file type	Message displays when using Certificates as the Authentication option and you have tried to load a certificate file. The file provided is incorrect or in the incorrect format.	<ul style="list-style-type: none"> — Click Close to close error message. Verify the file name or extension. Ensure you are loading the correct file. — Click Choose File again and load the correct file. — Continue configuration. — Click Update Configuration.

12 Device security

The following sections include information regarding security for MQTT-enabled devices. Review guidelines, recommendations, and additional device details prior to configuring and connecting devices to customer networks.



IMPORTANT NOTE: Refer to the device user manual for detailed guidelines and procedures to secure physical access to the device or access from PCCU. This manual only includes procedures relevant to the MQTT functionality.

12.1 Security guidelines

[Table 12-1](#) lists recommended MQTT-specific configuration and operation guidelines to secure access to devices.

Table 12-1: Guidelines for MQTT device configuration user interface and operation

Recommendation	Description
Secure network connection	Connect the device only to a firewall-protected private network. Do not connect directly to the Internet. See section 12.3 Secure connections .
Secure access to the device user interface (web interface)	<ul style="list-style-type: none"> — Change default passwords to private passwords on user accounts created at the factory. The device enforces a strong password policy which allows defining passwords with a minimum and maximum length, the use of special characters and upper- and lower-case letters, etc. See section 12.4.3 Change default passwords. — Add new user accounts and assign appropriate roles and private credentials. See sections 12.4 Manage users from the web interface. — The web interface should only be enabled when needed for initial configuration, update, or troubleshooting. Once configuration is complete, disable access. See section 12.6 Disable MQTT Rest service.
Manage configuration interface credentials	Store all private device interface credentials in safe locations. Share private device interface credentials only with properly trained and authorized personnel. Change or update private credentials as needed.
Secure connection with the MQTT broker	Select MQTT broker (servers) that support secure MQTT connections (TLS connections on port 8883).
Manage MQTT credentials and authentication certificates	Generate and upload valid certificates to the device. Store all authentication certificates in safe locations. Change or update authentication certificates as needed.

12.2 MQTT services

[Table 12-2](#) shows the services or processes that activate on the device when enabled for MQTT operation. MQTT-related services should provide access only to authorized third-party devices, such as MQTT brokers, or to users for device configuration. These services should not be enabled until guidelines for secure user-device or user-broker connection have been met.

Table 12-2: Services required for MQTT operation

Service/Process	Default state	Description	Security features available
MQTT Service	Disabled	Process that performs the MQTT client function for communication with the MQTT broker. The client initiates communication with the MQTT broker by sending a connection request.	<ul style="list-style-type: none"> — Security features inherent to the secure (TLS) connection standard used on the device-MQTT broker connection. — Authentication certificates
MQTT REST Service	Disabled	Serves connection requests for client access to the device configuration web pages (configuration interface for MQTT related parameters). The service listens to TCP port 443 for connection requests.	<ul style="list-style-type: none"> — Access to the device is protected by role-based access control: Access requires credentials-based authentication. The device configuration interface supports user management to add users and assign roles. — Users can replace factory default credentials with private credentials for authentication of authorized personnel only.

Service/ Process	Default state	Description	Security features available
			— For security, this service can be disabled after configuration is completed. See section 12.6 Disable MQTT Rest service .

12.3 Secure connections



NOTICE – Cybersecurity risk: ABB Totalflow MQTT-enabled devices are designed to connect to MQTT servers on a private network. Connection to an MQTT server on a public network such as the Internet should only be done indirectly through an Edge gateway.

General guidelines:

- Remote connections to and from the device must be established over the corporate network for security.
- Authorized remote web users should have access to a secure connection from the customer premises or use the corporate VPN.
- Field local area network equipment access for local operator connections and device-to-device connections should be protected.

12.4 Manage users from the web interface

The web interface supports role-based access. Users are configured on the User Management page on the web interface.

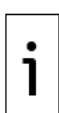
Devices store the defined users and their credentials in an encoded file (SHA-1 storage).



IMPORTANT NOTE: Users defined in this section access the device configuration interface for MQTT operation. These users are different from those defined for device access using PCCU.

12.4.1 Default account users and role privileges

[Table 12-3](#) lists the default users, roles, and credentials in the MQTT-enabled device.



IMPORTANT NOTE: Change factory default passwords to private passwords at first-time login. Do not leave devices with default passwords after installation and commissioning or after flash upgrade to MQTT-enabled flash. Be sure to set strong passwords. The device enforces strong password attributes: it ensures the password is within the minimum and maximum password length and allows the use of special characters, numbers, upper- and lower-case letters, etc.

Table 12-3: Default user accounts on MQTT-enabled device

User Name	Role	Password
AbbCustomer	guest	root@123
AbbDeveloper	admin	root@123
AbbTester	user	root@123
root	admin	root@123

[Table 12-4](#) lists roles and access levels available on the MQTT-enabled device.

Table 12-4: Role privileges on MQTT-enabled device

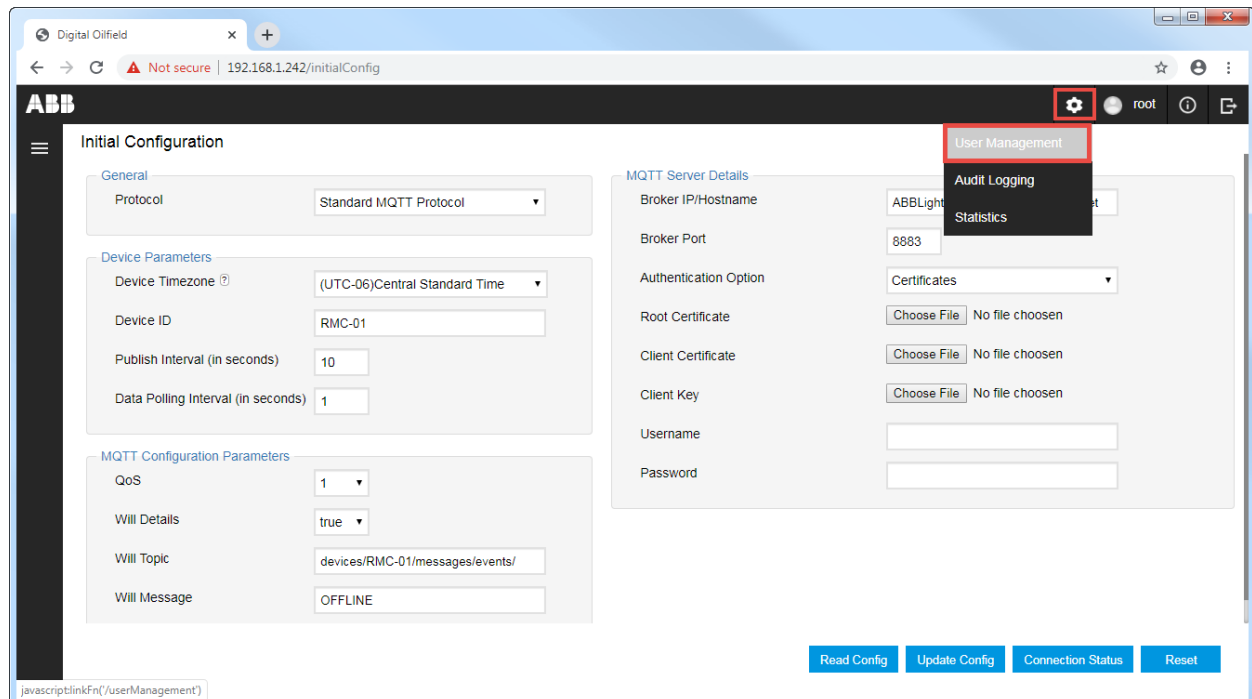
Role	Access level	Description
admin	Read and write (update) Manage users: add, delete or update users	The admin role has the following privileges: <ul style="list-style-type: none"> — View (read) and update device parameters (if applicable) in all device configuration pages (Initial, Application and Register configurations pages) — Access the device Audit Logging and Statistics pages — Add new users, delete existing users and update user attributes in the device's User Management page
user	Read and write (update)	The user role has the following privileges: <ul style="list-style-type: none"> — View (read) and update device parameters (if applicable) in all device configuration pages (Initial, Application and Register configurations pages) — Access the device Audit Logging and Statistics pages
guest	Read-only access	The guest has minimum privileges: <ul style="list-style-type: none"> — View (read) device parameters in all device configuration pages (Initial, Application and Register configurations pages)

12.4.2 Access the User Management web page

IMPORTANT NOTE: The user management web page is available only for users with the admin role. To complete the procedures in this section, you must log into the device as an administrator. Access the device with the default username and password used for the initial device configuration in previous sections.

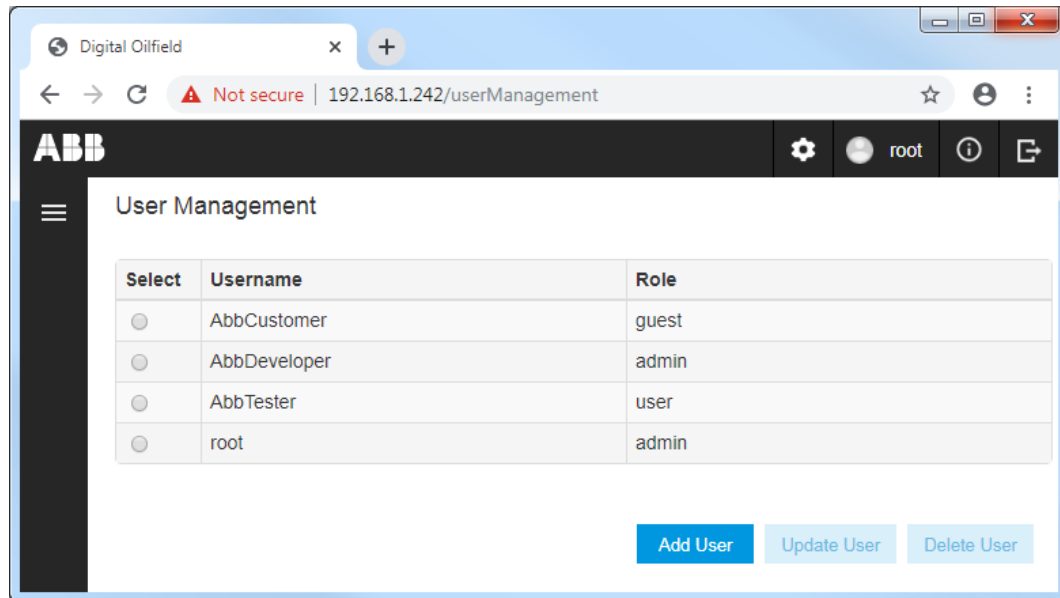
To access the User Management page:

1. Log into the device web interface with username: **root**, and password: **root@123**.
1. Navigate to the **Initial Configuration** page.
2. Click the settings icon and then **User Management** from the drop-down list ([Figure 12-1](#)).

Figure 12-1: Access the User Management web page

The User Management web page displays ([Figure 12-2](#))

Figure 12-2: User Management web page



12.4.3 Change default passwords

You can use default credentials for initial configuration. But be sure to change default passwords to private passwords for all default users as soon as initial configuration is completed. Use the **Update User** function to change passwords.

To update passwords on default accounts:

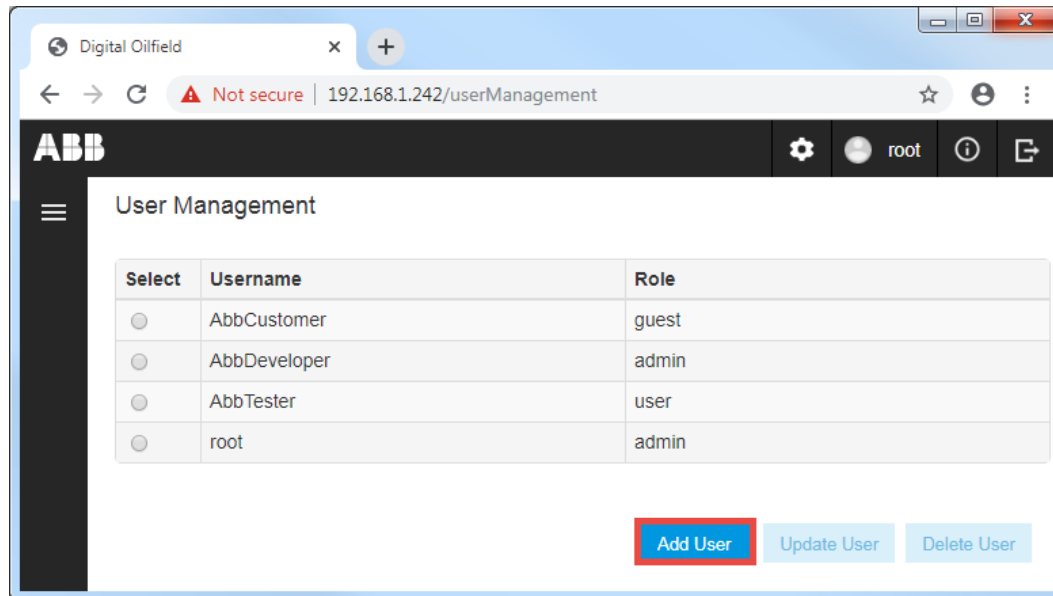
1. Select the user from the list on the User Management web page. The **Update User** and **Delete User** buttons activate.
2. Click **Update User**.
3. Update the password at the **Update User** dialog box. Take note of the new password and store this information in a safe location.
4. Click **Update**.
5. Repeat steps 1-4 for each default user.

12.4.4 Add a user

Add additional users to the defined defaults:

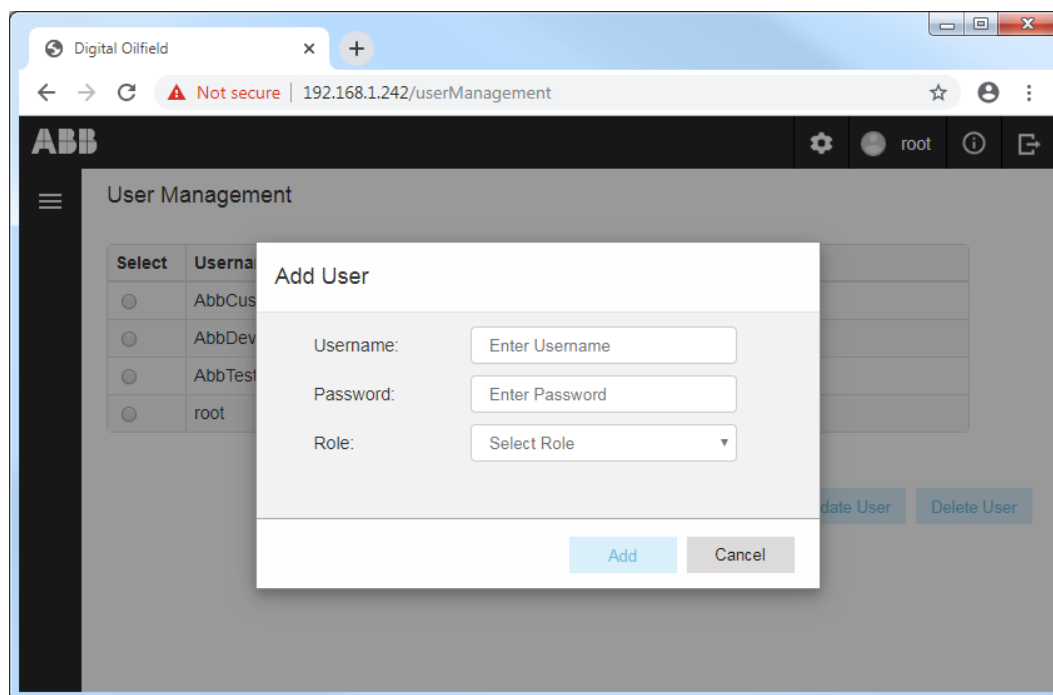
1. Click **Add User** on the User Management web page.

Figure 12-3: Add User



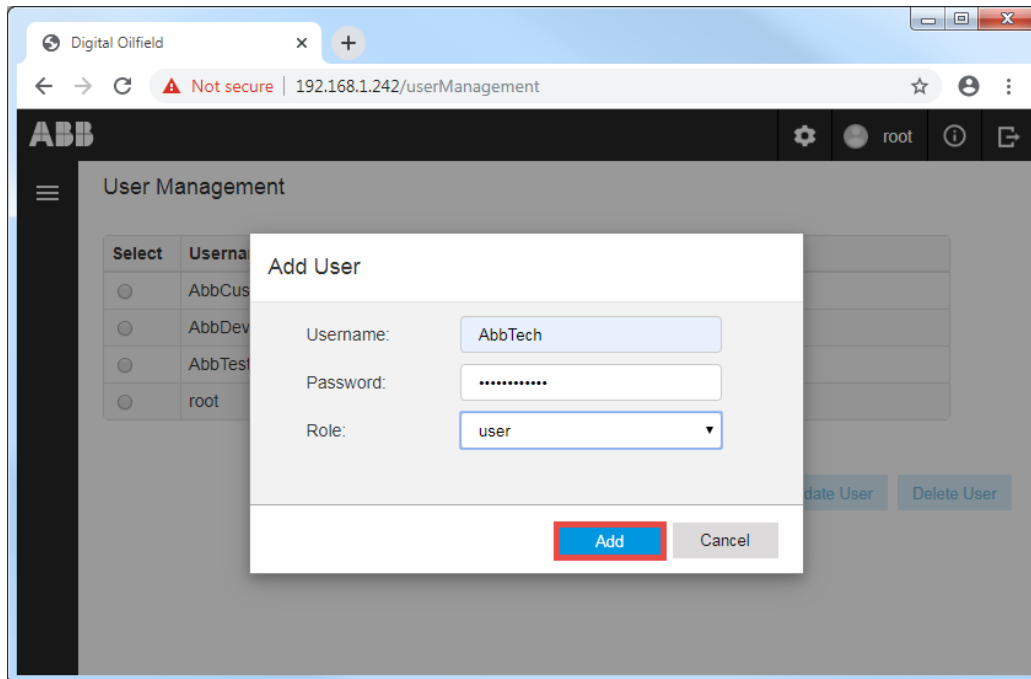
The Add User dialog displays.

Figure 12-4: Add user dialog box



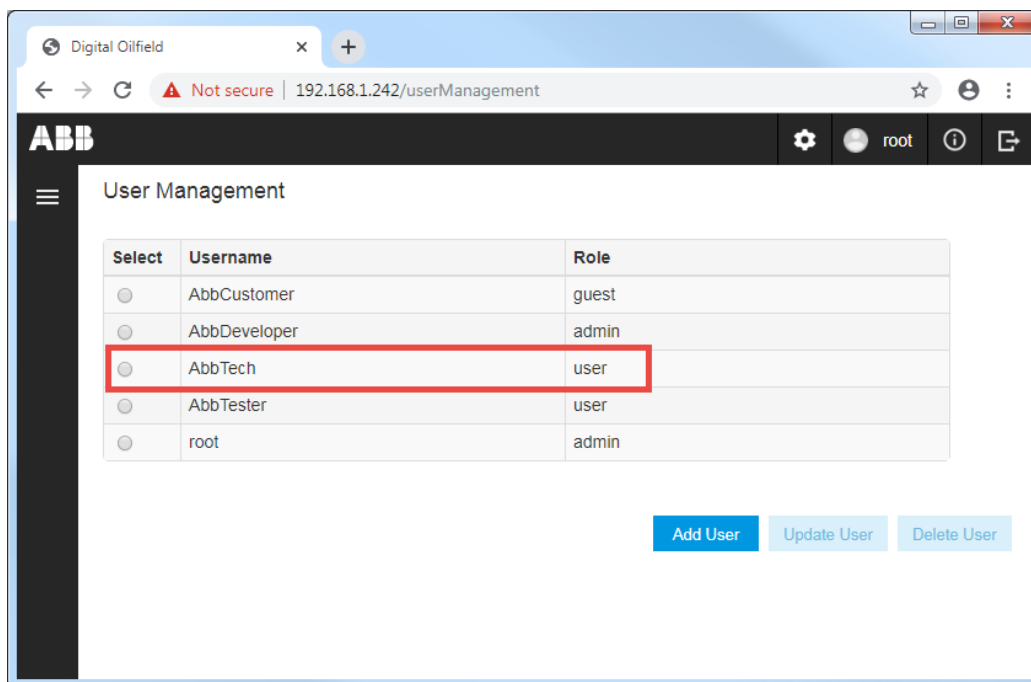
2. Type credentials (Username and Password). Take note of the credentials and store in safe location.
3. Select the Role from the drop-down menu and click **Add** ([Figure 12-5](#)). In this example, the new user AbbTech is assigned the user role.

Figure 12-5: Add new user credentials and role



4. Verify that the new user displays in the list ([Figure 12-6](#)).

Figure 12-6: Verify new user



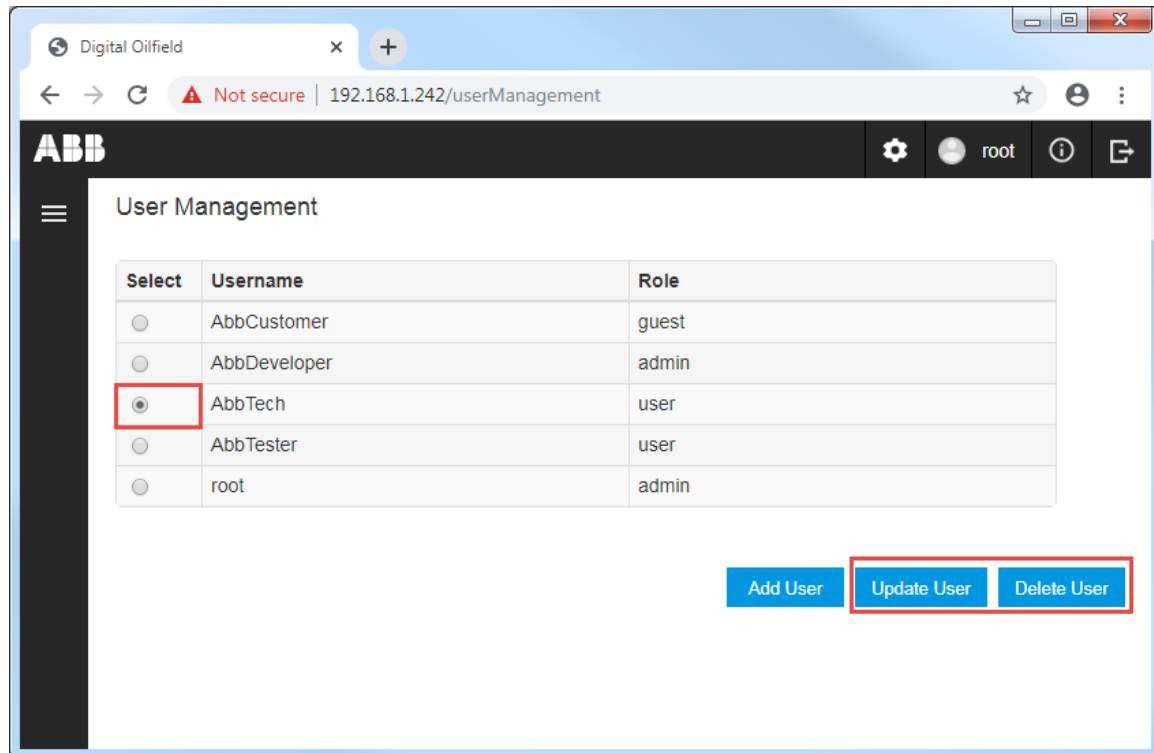
12.4.5 Update a user

The Update User function allows the change of the password or role assigned to an existing user. Username change is not supported. Create new accounts and deleted unwanted ones if required.

To update an existing user account:

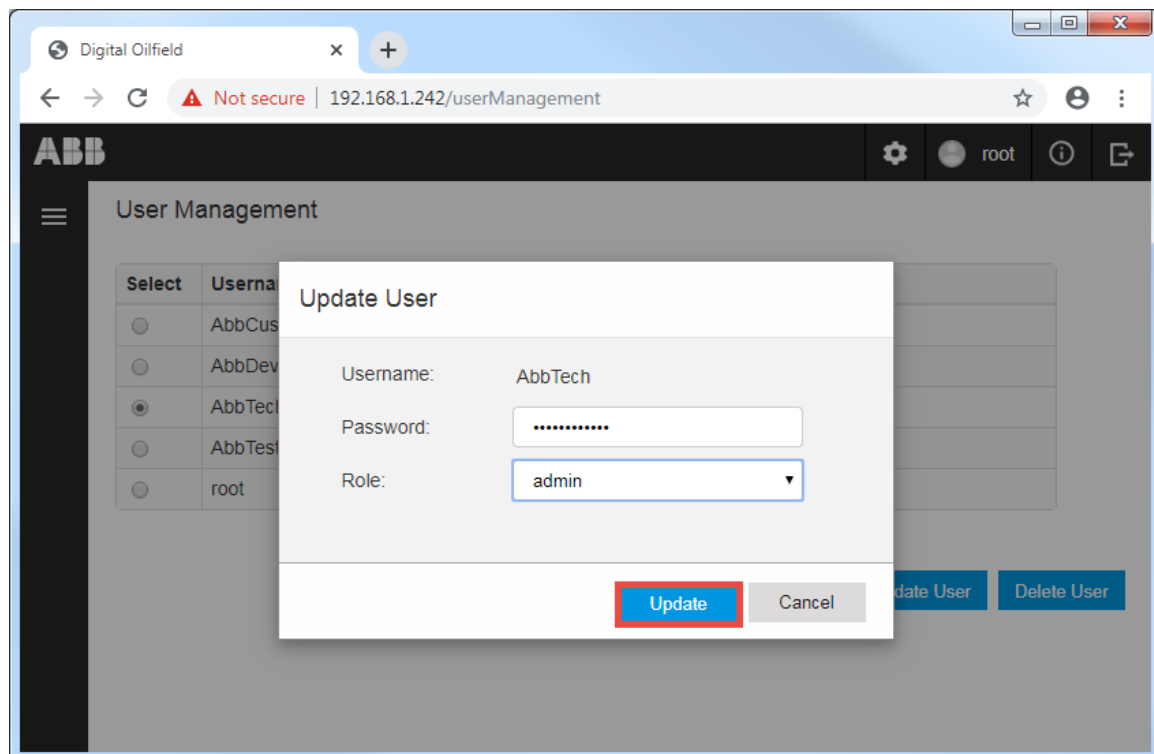
1. Select the user from the list on the User Management web page. The Update User and Delete User buttons activate.

Figure 12-7: Select existing user to update



2. Click **Update User**.
3. Update the password or role at the Update User dialog box. In this example, the password is the same, but the role is updated from user to admin.
4. Click **Update**.

Figure 12-8: Update password or role for an existing user



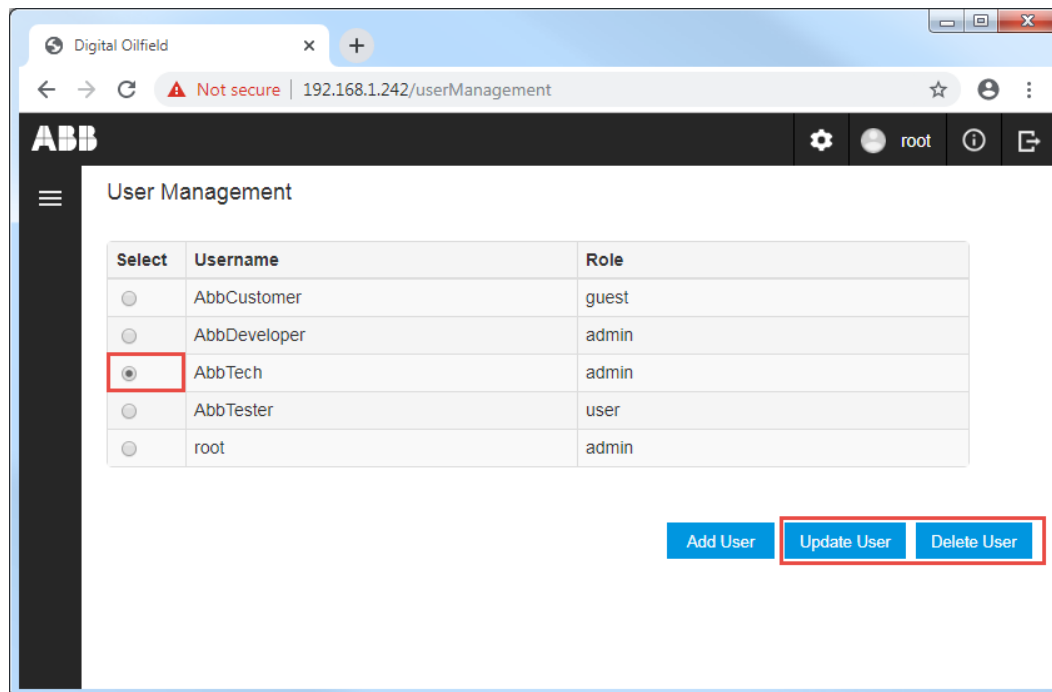
12.4.6 Delete a user

The Delete User function removes an existing user.

To delete a user:

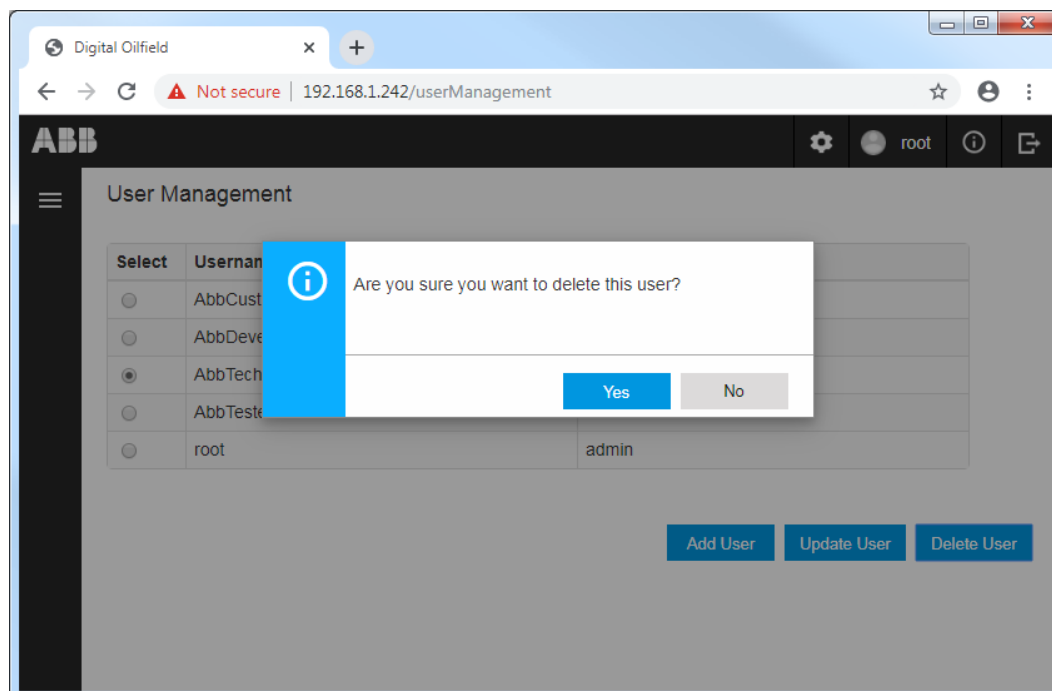
1. Select the user from the list on the User Management web page. The Update User and Delete User buttons activate.

Figure 12-9: Select a user to delete



2. Click **Delete User**.
3. Click **Yes** when prompted to confirm.

Figure 12-10: Confirm message to delete user



4. Verify that the user no longer displays in the User Management page.

12.5 Upload valid certificates (for secure user connection)

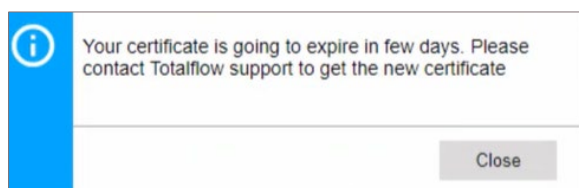
The device MQTT configuration interface supports secure connections from the end user browser with the use of valid certificates. Initial certificates to authenticate connections are included with the embedded software in the device itself, but they will need to be updated if they expire or if the customer prefers to have their own certificates. Certificate update requires the user to login with the **admin** role.

Factory default certificates are valid for one year. The Initial Configuration interface allows certificate update from the **Secure Web Interface** section on the main configuration page ([Figure 12-11](#)).

It is the customer responsibility to generate or obtain new valid certificates and upload them to the device. If you need to have your own certificates or if the expiration warning displays ([Figure 12-12](#)), follow the procedures in this section to update certificates.

Figure 12-11: Initial Configuration screen displaying the Secure MQTT REST Interface

Figure 12-12: Certificate expiration warning



To update certificates:

1. Login to the device as admin.
1. Generate or obtain valid certificates to update certificates on the device. Certificates should be generated for the IP address range assigned to the field devices. There are two files required: client-cert.pem and client-key.pem.
2. Download or copy the new certificate files to the laptop or PC used to connect with the device(s).
3. From the **Secure Web Interface** section of the Initial Configuration screen ([Figure 12-13](#)):

Figure 12-13: Use Secure MQTT REST Interface section to update certificates

The screenshot shows the ABB Initial Configuration page in a web browser. The page is divided into several sections: Device Parameters, MQTT Configuration Parameters, Broker Port, Authentication Option, Root Certificate, Client Certificate, Client Key, Username, Password, MQTT Broker Connection, and Secure Web Interface. The Secure Web Interface section is highlighted with a red box, showing fields for Secure Client Certificate and Secure Client Key, both with 'Choose File' buttons and 'No file chosen' text. Below this section are 'Read Config' and 'Update Config' buttons.

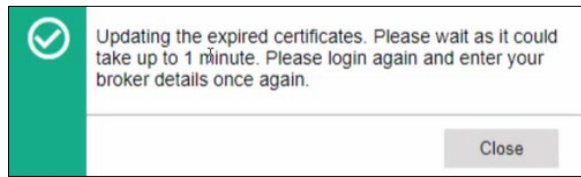
- a. Click **Choose File** for the Secure Client Certificate ([Figure 12-14](#)).
- b. Locate and select the file from the file browser window.
- c. Click **Open**. The name of the file displays on the screen.
- d. Click **Choose File** for the Secure Client Key ([Figure 12-14](#)).
- e. Locate and select the file from the file browser window.
- f. Click **Open**. The name of the file displays on the screen.
4. Click **Update Config** ([Figure 12-14](#)).

Figure 12-14: Upload new certificates using Update Config

The screenshot shows the ABB Initial Configuration page with the Secure Web Interface section. The 'Secure Client Certificate' field now displays 'client-cert.pem' and the 'Secure Client Key' field displays 'client-key.pem', both with red arrows pointing to the file names. The 'Update Config' button is highlighted with a red border. The 'Device ID' field is now 'RMC-100' and the 'Will Topic' is 'devices/RMC-100/messages/events/'.

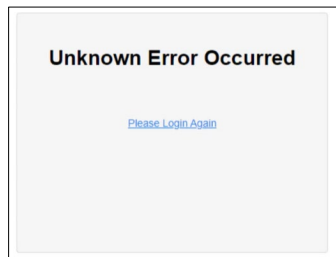
5. Wait for the certificate upload to complete. The following message should display:

Figure 12-15: Certificate update



6. Click **Close**.
7. Refresh the Initial Configuration screen. An error message display:

Figure 12-16: Error message after certificate update



8. Click **Please Login Again**.

With valid certificates, a secure browser-device connection can be configured. Follow Chrome's instructions to configure the browser to trust device certificates. The valid certificates uploaded to the device are detected by the browser and must also be imported into the browser certificate store. Browser configuration is beyond the scope of this document. Verify Chrome settings and follow recommended steps.

When browser access is secure, the "Not secure" warning ([Figure 12-17](#)) no longer displays ([Figure 12-18](#)).

Figure 12-17: Security warning (non-secure device access)

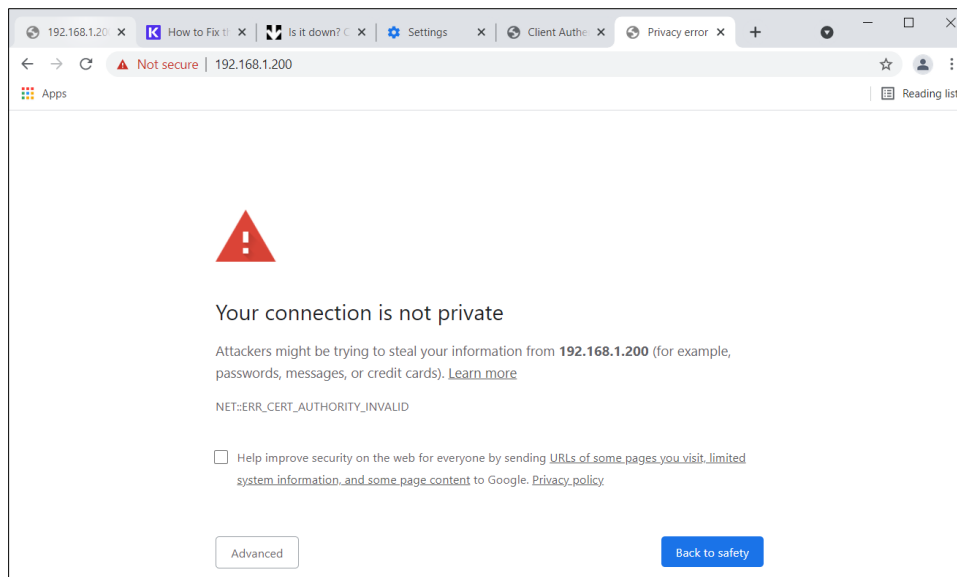
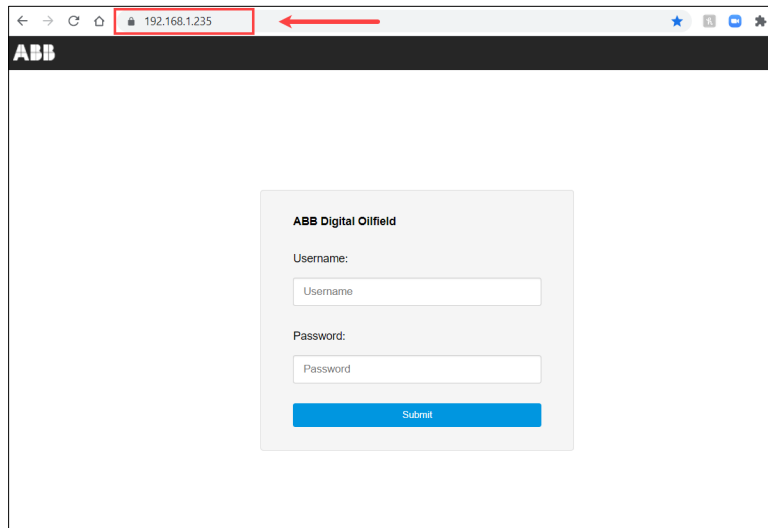


Figure 12-18: Secure device access (“Not Secure” warning no longer displays)



12.6 Disable MQTT Rest service

After successful MQTT configuration is completed, protect access to the configuration interface by disabling the MQTT REST Service.

To disable the MQTT REST Service from the PCCU Entry mode:

1. Connect the laptop to the USB port on the device.
2. Start PCCU.
3. Click the Entry icon to connect with the device.
4. On the navigation tree, select **Communications**. The Communication Setup screen displays.
5. Select the **Services** tab.
6. Clear the **MQTT REST Service** checkbox.
7. Click **Send** to confirm change. Access to the MQTT configuration interface is blocked.

13 Monitor device audit logs



IMPORTANT NOTE: Access to the Audit Logging web page is available for user and admin roles.

13.1 Audit Logging web page overview

The Audit Logging web page displays device configuration update activity ([Figure 13-1](#)). The logs record the parameter change and its value before (old) and after (new) the update. Each log has a time stamp and records the user and role at the time of the update.

The device stores up to 100 logs. When the number of logs reaches this limit, the device overwrites the older logs to continue to store and display the most current information.

Figure 13-1: Audit Logging web page

TimeStamp	S.No	Username	Role	Request Type	Old Value	New Value	Req. Status
11/10/2019 11:19:46 AM	7	root	admin	Update Register Configuration	InstanceName:AGA3-1 model:Aggregate	InstanceName:AGA3-1 model:Aggregate	Success
11/08/2019 12:41:43 PM	6	root	admin	Update Application Configuration	AGA-7 Measurement:Disable Plunger Control:Disable	AGA-7 Measurement:Enable Plunger Control:Enable	Success
11/08/2019 12:40:52 PM	5	root	admin	Update Register Configuration	InstanceName:AGA3-1 model:Aggregate	InstanceName:AGA3-1 model:Aggregate	Success
11/08/2019 12:37:35 PM	4	root	admin	Update Application Configuration	AGA3-2 :Disable	AGA3-2 :Enable	Success
11/08/2019 12:35:05 PM	3	root	admin	Update Application Configuration	API Liquid SU:Disable	API Liquid SU:Enable	Success
11/08/2019 12:31:54 PM	2	root	admin	Update Application Configuration	AGA-7 Measurement:Enable Plunger Control:Enable	AGA-7 Measurement:Disable Plunger Control:Disable	Success
11/05/2019 09:27:50 AM	1	root	admin	Update Initial Configuration	TimeZone:330 Device_Id:RMCPinyonSite	TimeZone:-360 Device_Id:RMC-01	Success

Download Report

[Table 13-1](#) describes the attributes on the audit logging page.

Table 13-1: Device audit logging parameter description

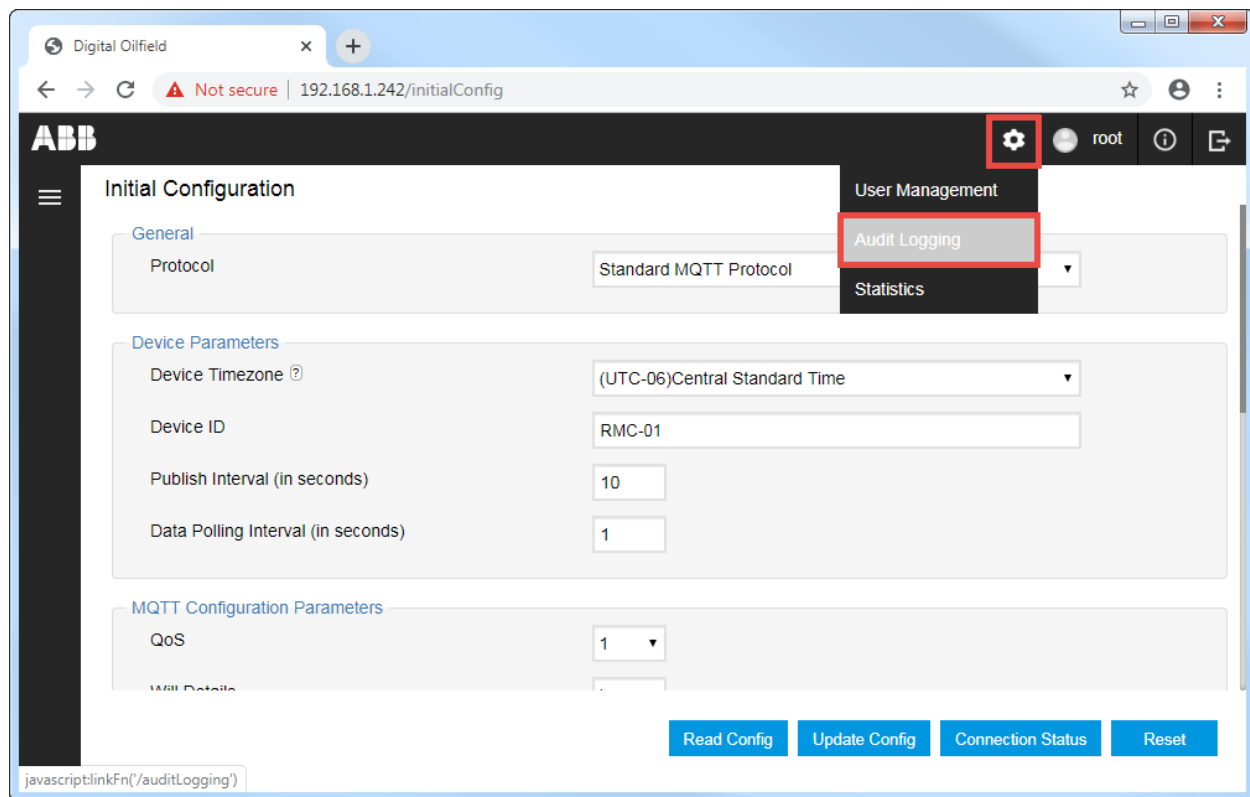
Field	Description	Values
Time Stamp	Date and time of the update by the logged-in user	Date and time match the date and time kept by the device
S. No	Serial number of the audit log	Logs are numbered sequentially with decimal numbers beginning at 1 for the first log. Serial numbers do not restart when the number of logs reaches its limit of 100.
Username	Identifies the logged-in user at the time of the update	Any user already defined in the User Management web page
Role	Identifies the role of the logged-in user	Role assigned to logged-in user (admin, user, guest)
Request type	Identifies the device configuration page that the update originated from	Update Initial Configuration Update Application Configuration Update Register Configuration Reset Statistics
Old Value	Name and value of parameter or configuration option prior to the update request from the logged-in user	Values applicable to the parameter type Values might be user-defined or selected from drop-down menus.
New Value	Name and value of parameter or configuration option after the device completes update request by the logged-in user	Values applicable to the parameter type Values might be user-defined or selected from drop-down menus.
Re. Status	Request Status Indicates the status of the update request by the logged-in user	Success – The update request message is validated and is being applied by software. Failure - The update request message validation has failed.

13.2 Access the Audit Logging web page

To access the audit logging page:

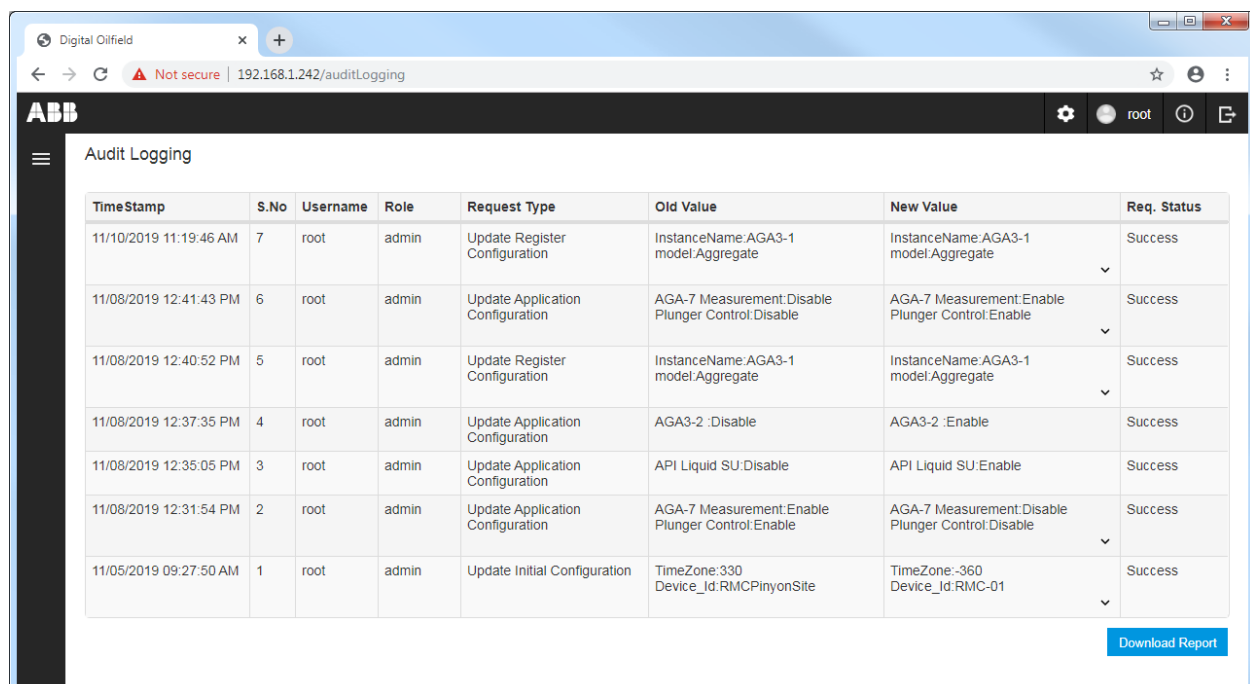
1. Click on the settings icon and select **Audit logging** from the drop-down list (Figure 13-2).

Figure 13-2: Access the Audit logging page



The Audit Logging web page displays.

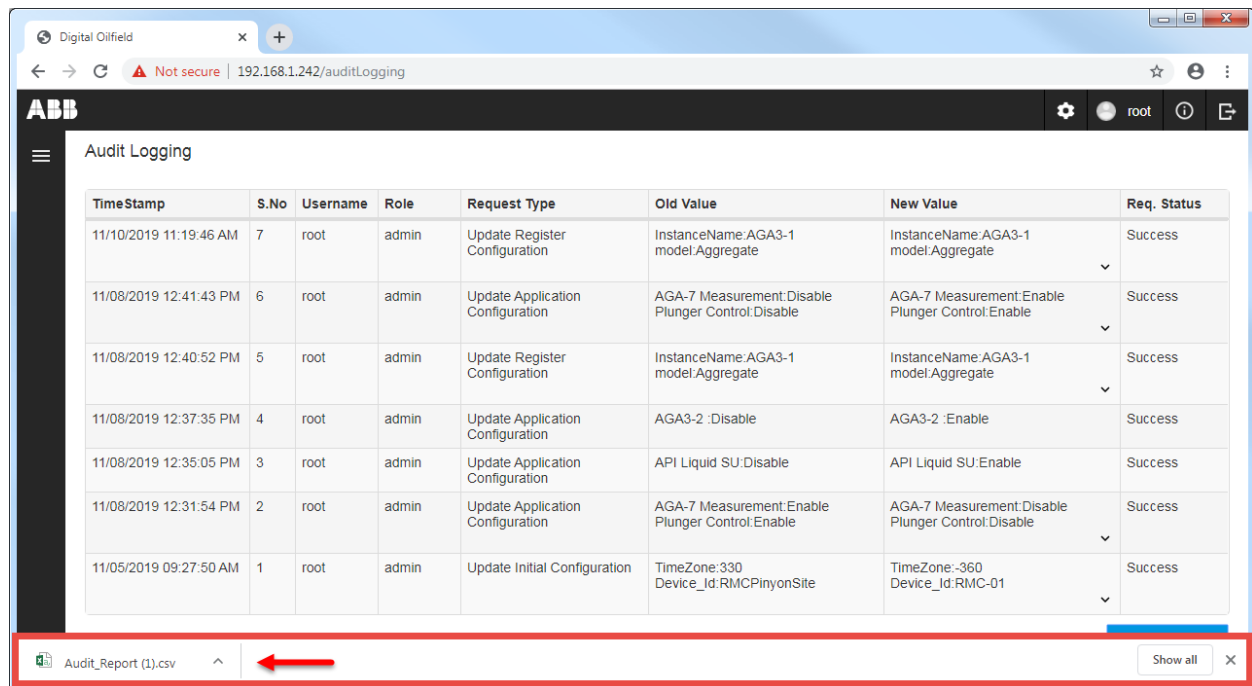
Figure 13-3: Audit logging page



2. Locate the log of interest or list review logs as necessary.

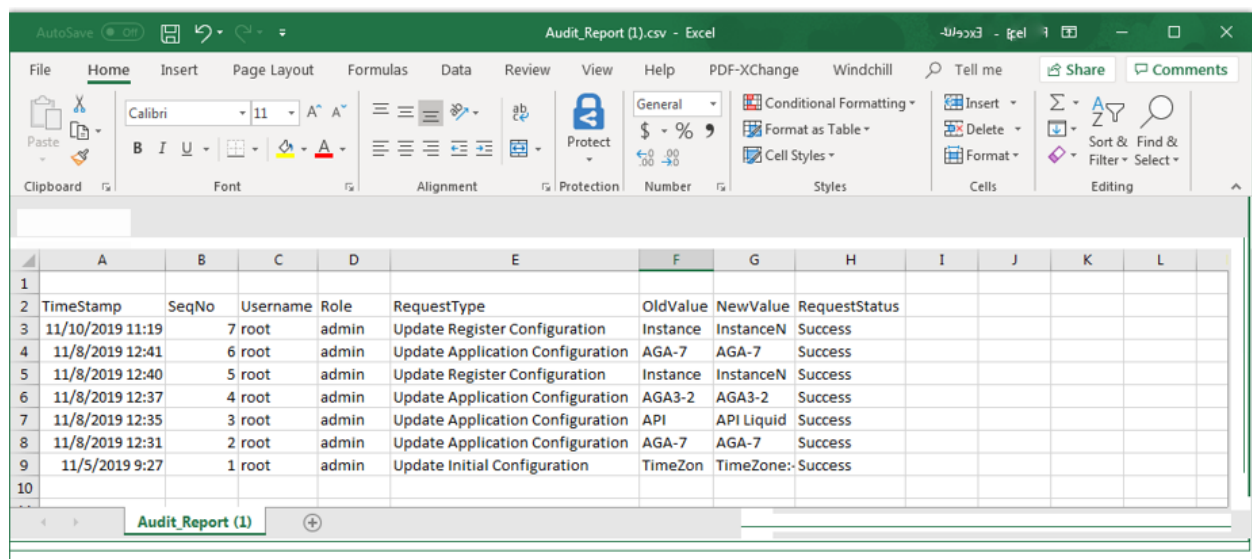
- To generate and save a copy of the logs, click **Download Report**. A file with .csv extension saves automatically in the download folder of your laptop or PC ([Figure 13-4](#)).

Figure 13-4: Audit Report file generated and downloaded to local laptop



- Click **Show All**.
- Select the Audit_Report file from the download list. The file opens ([Figure 13-5](#)).

Figure 13-5: Audit report downloaded from the cloud



- Save the file in the desired folder to keep a backup copy.

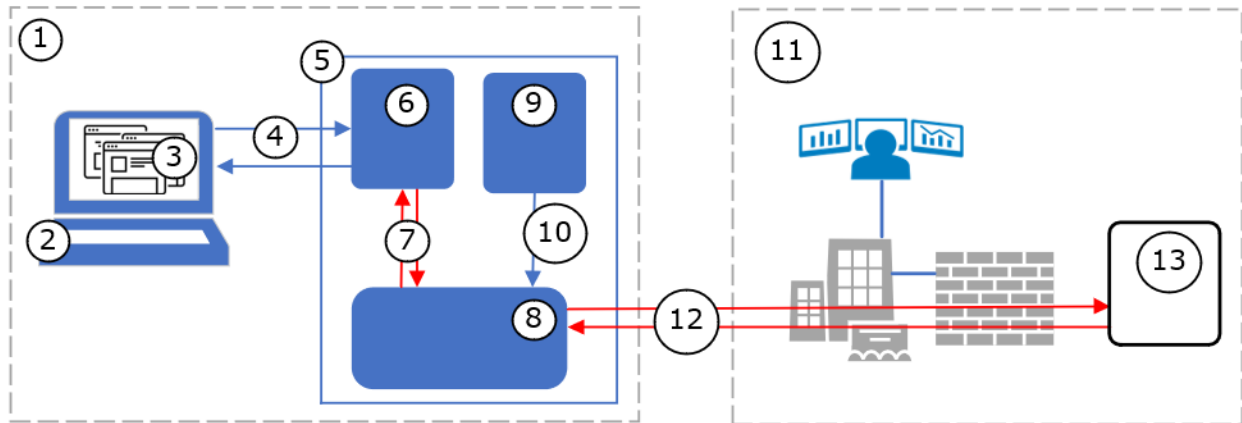
14 Monitor devices statistics

The Statistics web page provides valuable information to monitor the device configuration process and the device-MQTT broker communication. The device tracks certain parameters such as the number and type of MQTT packets sent or received, disconnection events, etc.

The high-level diagram in [Figure 14-1](#) illustrates a simplified view of the intra-process communication and the device-broker communication for which the device keeps statistics:

- Configuration-related statistics keep track of the internal communication between the processes that handle MQTT-related configuration in the device. This communication consists of configuration requests/responses (7) exchanged between the user interface/REST server (6) and the MQTT stack (8). This example shows a local user connected to the ABB MQTT-enabled device at the site. Statistics are also logged for configuration through a remote connection. Details for these statistics are described in section [14.2 Device configuration statistics](#).
- Device-broker connection statistics keep track of the communication between the device (5) and the broker (13) over the network connection (12). Details for these statistics are described in section [14.3 Device-broker connection statistics](#).

Figure 14-1: Intra-process and device-broker communication



Legend: Intra-process and device-broker communication

ID	Field device on site	ID	Customer private network
1	Field Local Area Network	11	Corporate network
2	Configuration client	12	Device-broker communication, data flow
3	Configuration web pages	13	MQTT broker
4	Configuration update requests		
5	ABB device (RMC-100)		
6	REST server		
7	MQTT configuration intra-process communication		
8	MQTT stack processes		
9	Application data collector		
10	Publish application data		

14.1 Access the Statistics web page

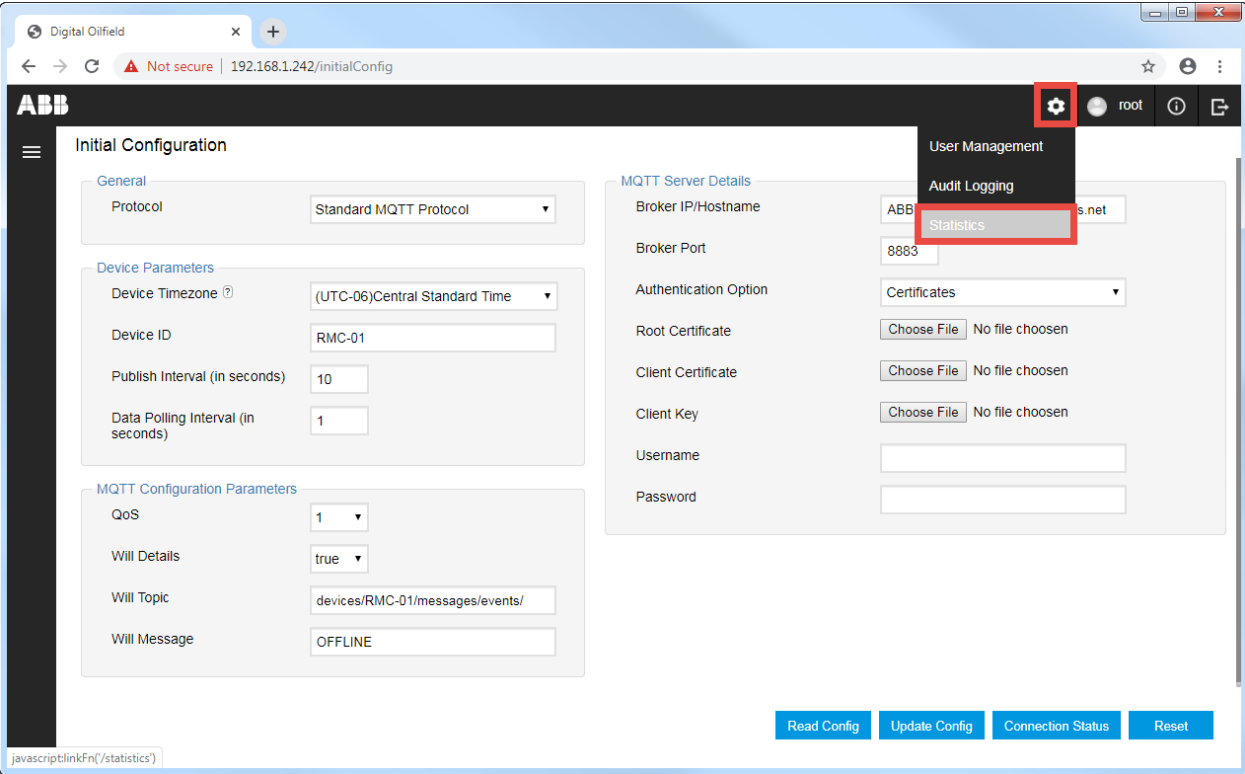


IMPORTANT NOTE: Access to the Statistics web page is available for user and administrator roles.

To view the Statistics page:

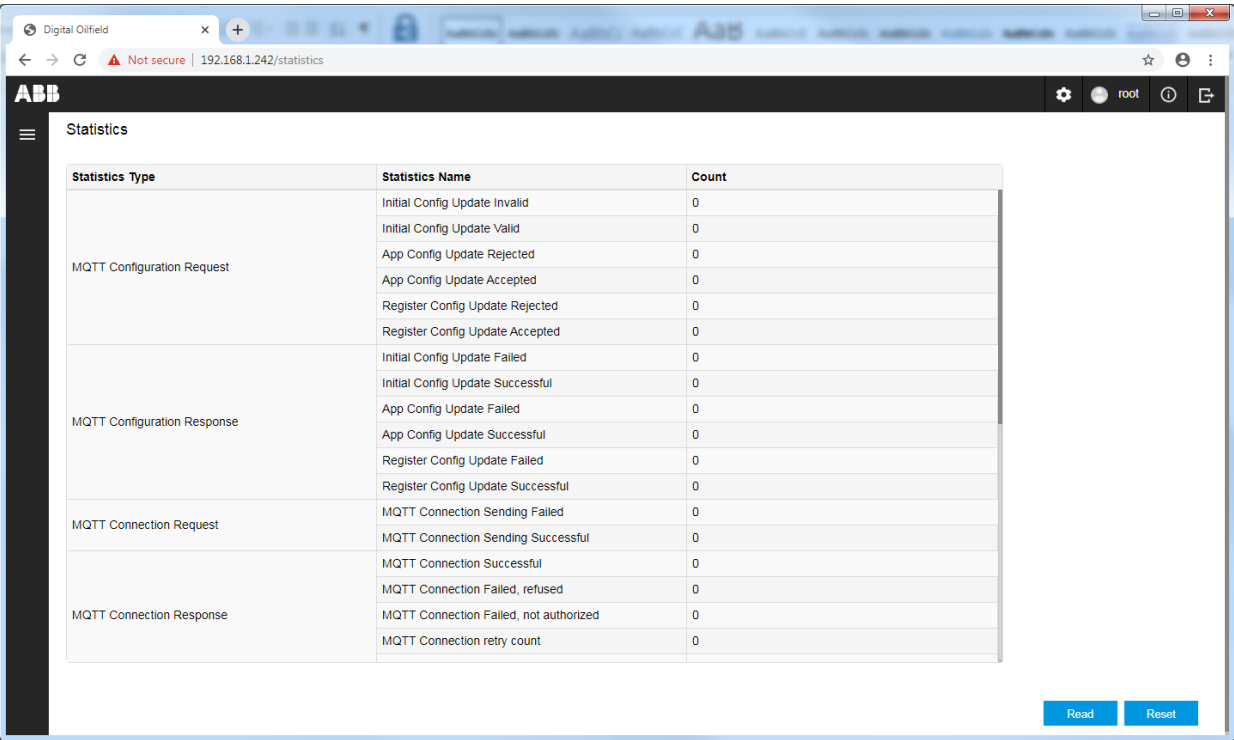
1. Click the settings icon and select **Statistics** from the drop-down menu ([Figure 14-2](#)).

Figure 14-2: Access the device Statistics page



The statistics web page displays.

Figure 14-3: Statistics for standard MQTT protocol



2. Review the information as necessary.
3. If monitoring activity, click **Reset** any anytime to set the counts to zero. Packets counts that do not increase when expected may indicate connection issues.



IMPORTANT NOTE: If sparkplug is the selected protocol, sparkplug-related statistics display in addition to the MQTT statistics. See section [14.4 Sparkplug statistics](#) for additional information.

14.2 Device configuration statistics

The device supports the configuration of MQTT-related parameters from client's web browsers. [Table 14-1](#) lists the device configuration statistics sets that track the internal communication between the processes that handle configuration requests and the user interface. [Table 14-2](#) and [Table 14-3](#) provide details for each set.

Table 14-1: Device configuration interface statistics

Type	Description	Monitored packets
MQTT Configuration Request	Packets that the MQTT stack receives from the device user interface for configuration updates	Initial Config Update Invalid Initial Config Update Valid App Config Update Rejected App Config Update Accepted Register Config Update Rejected Register Config Update Accepted
MQTT Configuration Response	Packets that the MQTT stack sends to the device user interface in response to configuration requests	Initial Config Update Failed Initial Config Update Successful App Config Update Failed App Config Update Successful Register Config Update Failed Register Config Update Successful

[Table 14-2](#) describes each of the monitored packets in the configuration requests statistic set. These statistics keep track of packets the MQTT stack receives when a user submits updates from the initial, application, or register configuration pages.

Table 14-2: Configuration requests statistics

Packet name	Description
Initial Config Update Invalid	Packets received with invalid Initial Configuration update values. For example, the user may have submitted an update request with incorrect MQTT device or broker parameters or invalid certificates.
Initial Config Update Valid	Packets received with valid Initial Configuration update values. Values submitted on the Initial Configuration page such as MQTT device and broker parameters and certificates are correct or valid.
App Config Update Rejected	Packets received with invalid updates on the application configuration page. These may include attempts to enable applications that may have been deleted from the device.
App Config Update Accepted	Packets received with valid updates on the Application Configuration page. These may include enabling or disabling some or all the applications configured in the device.
Register Config Update Rejected	Packets received with invalid updates on the register configuration page. These may include attempts to enable registers for applications that may have been deleted from the device.

Packet name	Description
Register Config Update Accepted	Packets received with valid updates on the Register Configuration page. These may include enabling or disabling some or all the user-configurable registers for the supported applications. Some registers are mandatory and do not allow selection by the user. Mandatory registers remain enabled through any update request.

[Table 14-3](#) describes each of the monitored packets in the configuration response statistic set. These statistics keep track of packets the device sends in response to configuration update requests submitted from the initial, application, or register configuration pages.

Table 14-3: Configuration response statistics

Name	Description
Initial Config Update Failed	Packets that MQTT stack sends to the user interface to notify that it could not apply an update request submitted from the initial configuration page. For example, the device cannot update the configuration because the MQTT broker did not accept certificates submitted or the request had other invalid broker parameter values.
Initial Config Update Successful	Packets that MQTT stack sends to the user interface to notify that it successfully applied the update request submitted from the initial configuration page. This packet is also generated when connection to a new broker has been established successfully in the case where the configuration update involved certificate or MQTT broker hostname change.
App Config Update Failed	Packets that the MQTT stack sends to the user interface to notify that it could not apply an update request submitted from the application configuration page. For example, the device fails to apply the selection (enabling) of an application that is no longer instantiated on the device. This packet type is also generated when the device is not connected to the MQTT broker.
App Config Update Successful	Packets that the MQTT stack sends to the user interface to notify that it successfully applied the update request submitted from the application configuration page. For example, the device accepts the selection of an application that is instantiated and enabled in the device.
Register Config Update Failed	Packets that the MQTT stack sends to the user interface to notify that it could not apply an update request submitted from the register configuration page. For example, the device could not apply the enabling or disabling of a register for an application that is no longer instantiated or enabled in the device. This packet type is also generated when the device is not connected to the MQTT broker.
Register Config Update Successful	Packets that the MQTT stack sends to the user interface to notify that it successfully applied the update request submitted from the register configuration page.

14.3 Device-broker connection statistics

[Table 14-4](#) lists the device-broker connection statistics sets that keep track of the communication between the device and the MQTT broker. Some of these statistics apply to both the MQTT 3.1.1 protocol and Sparkplug. Others apply only to the MQTT 3.1.1 protocol. For statistics that are specific to Sparkplug, see also section [14.4 Sparkplug statistics](#).

Table 14-4: Device-broker connection statistics

MQTT packet type	Description	Packets
MQTT Connection Request	(MQTT 3.1.1 and Sparkplug) Packets generated by the MQTT stack to indicate if it was able to send a connection request to the MQTT broker.	MQTT Connection Sending Failed MQTT Connection Sending Successful
MQTT Connection Response	(MQTT 3.1.1 and Sparkplug) Packets the device receives from the MQTT broker in response to a connection request.	MQTT Connection Successful MQTT Connection Failed, refused MQTT Connection Failed, not authorized MQTT Connection retry count MQTT Connection reconnect count
MQTT Packet Received	(MQTT 3.1.1 protocol only) Packets the device receives from the broker over the device-broker connection.	Register Write Request
MQTT Packet Sent	(MQTT 3.1.1 protocol only) Packets the device sends to the broker over the device-broker connection.	Device Packet Count Application Structure Packet Count Trend Definition Packet Count Alarm Definition Packet Count Register Packet Count Trend Packet Count Daily_Log Packet Count Custom_Log Packet Count Event Packet Count Alarm Packet Count Plunger Cycles Packet Count Gaslift Events Packet Count References Packet Count Plunger Events Packet Count Shutdown Events Packet Count Device Packet Count
MQTT disconnect	(MQTT 3.1.1 and Sparkplug) Packets the device sends or receives prior to device-broker connection graceful termination.	MQTT Connection disconnected by device MQTT Connection disconnected by broker

[Table 14-5](#) describes the monitored packets in the MQTT connection request statistic set. These statistics keep track of packets that indicate if connection requests have reached the MQTT broker.

Table 14-5: MQTT connection request (MQTT 3.1.1 and Sparkplug)

Name	Description
MQTT Connection Sending Failed	Packet generated by the MQTT stack connection manager process when it is unable to send a connection request to the broker on behalf of the device. The connection request never reached the broker. This can be caused by invalid broker parameters, network error, or an unreachable broker.
MQTT Connection Sending Successful	Packet generated by the MQTT stack connection manager process when it is able to send a connection request to the broker on behalf of the device. The request has reached the device.

[Table 14-6](#) describes the monitored packets in the MQTT connection response statistic set. These statistics keep track of packets that the device receives from the broker after it has issued connection requests.

Table 14-6: MQTT Connection response (MQTT 3.1.1 and Sparkplug)

Name	Description
MQTT Connection Successful	Packet that the broker sends to the device when the device successfully establishes a connection with the broker. The broker validates and accepts the connection request from the device.
MQTT Connection Failed, refused	Packet that the broker sends to the device to reject a connection request. The device-broker connection is not established.
MQTT Connection Failed, not authorized	Packet that the broker sends to the device to reject a connection request due to invalid or unauthorized certificates. The device-broker connection is not established.
MQTT Connection retry count	Number of times the device tries to reconnect with the broker since the last successful connection. The device triggers automatic retries as soon as it loses connection with the broker.
MQTT Connection reconnect count	Number of times the device reconnects with a broker

[Table 14-7](#) below describes the packets the device receives from the broker on the device-broker connection.

Table 14-7: MQTT packets received (MQTT 3.1.1 protocol only)

Name	Description
Register Write Request	PUBLISH packets received from the MQTT broker that request an application register update on the device. Register update requests are submitted from the cloud user interface on specific application pages. The MQTT broker forwards those requests to the appropriate device.

[Table 14-8](#) below describes the monitored packets in the MQTT packet sent statistic set. These statistics keep track of packets that the device sends to the MQTT broker. Packets sent counts depend on the applications the device is publishing data for and the configured publish interval or frequency.

Table 14-8: MQTT packet sent (MQTT 3.1.1 protocol only)

MQTT packet type	Description
Device Packet Count	Any of the PUBLISH packets the device sends to the broker
Application Structure Packet Count	Packets sent with the device structure in the payload
Trend Definition Packet Count	Packets sent with trend definitions in the payload
Alarm Definition Packet Count	Packets sent with alarm definitions in the payload
Register Packet Count	Packets sent with payloads containing information required for register data updates. These packets identify the variable names associated with the register number, old and new values, etc.
Trend Packet Count	Packets sent with trend logs in the payload
Daily_Log Packet Count	Packets sent with a daily log in the payload
Custom_Log Packet Count	Packets sent with a custom log in the payload
Event Packet Count	Packets sent with an event in the payload
Alarm Packet Count	Packets sent with alarm logs in the payload

MQTT packet type	Description
Plunger Cycles Packet Count	Packets sent with plunger cycles in the payload
Gaslift Events Packet Count	Packets sent with gaslift events in the payload
References Packet Count	Packets sent with reference data in the payload (applicable to Gas Lift, Liquid and shutdown applications only). Example of reference data include meter factors or multipoint K factors configured for these applications.
Plunger Events Packet Count	Packets sent with plunger events in the payload
Shutdown Events Packet Count	Packets sent with shutdown events in the payload

[Table 14-9](#) below describes the monitored packets for device-broker disconnection. Disconnection notifications can be triggered from the device or from the broker. These packets contain the DISCONNECT notification in the payload.

Table 14-9: MQTT disconnect (MQTT 3.1.1 and Sparkplug)

Name	Description
MQTT Connection disconnected by device	Packets the device sends to the broker to notify it will disconnect from the broker. The device can send this packet before a graceful device shutdown.
MQTT Connection disconnected by broker	Packets the broker sends to the device to notify it will disconnect from the device.

14.4 Sparkplug statistics

Sparkplug statistics display when sparkplug is the communication protocol selected for the device-broker connection. These statistics keep track of the packets or messages that flow between the device and the MQTT server.

[Figure 14-4](#) shows a simplified diagram for Sparkplug-specific messages. The message set for which statistics are tracked are sparkplug packets sent (5) or received (6) by the device through the MQTT connection with the server. The SCADA/IIoT primary application (9) acts as an MQTT client and establishes a connection with the MQTT server. Requests for data update are issued in command messages (11) sent by the application to the MQTT server (8).

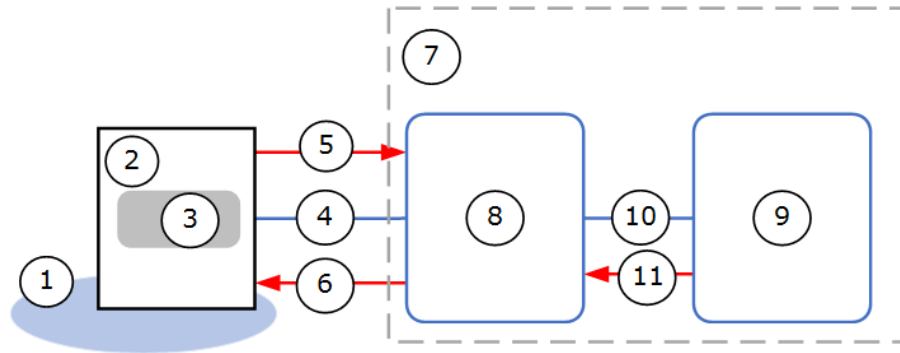


IMPORTANT NOTE: MQTT servers may be referred to by other names depending on the vendor implementing them. This manual uses the generic term “server” to indicate the main functionality or role of this component in the overall architecture. For details, consult your vendor documentation and architectures. Component functionality may be implemented as standalone or as software modules in some solutions.



IMPORTANT NOTE: The MQTT-enabled device performs both the device and Edge of Node (EoN) function. Refer to section [3.4 Preparing for Sparkplug](#) for details.

Figure 14-4: Sparkplug-specific message flow



Legend: Sparkplug-specific message flow

Field site	Customer network
1 Field Local Area Network	7 Customer corporate network (VPN)
2 Totalflow device	8 MQTT server/distributor
3 MQTT client and Sparkplug Device/Edge of Node (EoN) functionality	9 SCADA/IIoT Host (Primary Application)
4 Device - MQTT connection	10 Application-MQTT server connection
5 Sparkplug packets sent: NBIRTH, NDATA or DDATA messages	11 Sparkplug messages sent: NCMD, DCMD, STATE messages
6 Sparkplug packets received: NCMD, DCMD, STATE messages	

IMPORTANT NOTE: The statistics screen for a device using sparkplug also shows the MQTT Connection Request, MQTT Connection Response and MQTT disconnect statistics sets. These are the same statistic types as for the MQTT 3.1.1 protocol. See section [14.4 Sparkplug statistics](#) for details.

IMPORTANT NOTE: For additional details on sparkplug message types and device-MQTT server message flow, refer to the following link: <https://sparkplug.eclipse.org/specification/version/3.0/>.

[Table 14-10](#) shows the sparkplug-specific statistic sets and message types with payloads defined by the sparkplug specification.

Table 14-10: Sparkplug-specific statistics

Type	Description	Monitored packets
Sparkplug Packet Received	Packets with sparkplug-specific payloads that the device receives from the MQTT server	Sparkplug NCMD Message Count Sparkplug DCMD Message Count Sparkplug STATE Message Count
Sparkplug Packet sent	Packets with sparkplug-specific payloads that the device sends to the MQTT server	Sparkplug NBIRTH Message Count Sparkplug NDATA Message Count Sparkplug DDATA Message Count

[Table 14-11](#) describes each of the packet types the device receives from the MQTT server.

Table 14-11: Sparkplug Packet Received statistics

Name	Description
Sparkplug NCMD Message Count	<p>Number of Node Command (NCMD) messages.</p> <p>This message type is used to send commands to the Edge of Node (EoN) to update data values related to node control messages like REBIRTH requests. These requests are initiated by the Ignition servers when they receive corrupted data. The SCADA/IIoT primary application publishes the NCMD message to the MQTT server. The MQTT server ensures that the device receives the command and updates the data values as required.</p> <p>Note that the MQTT-enabled Totalflow device acts as an Edge of Node (EoN) when sparkplug is used. No separate Edge of Node device (gateway) is required for sparkplug support as this functionality is implemented as part of the MQTT stack in the Totalflow device.</p>
Sparkplug DCMD Message Count	<p>Number of Device Command (DCMD) messages.</p> <p>This message type is used to send commands to the device to update data values related to device information primarily sent in DDATA. The SCADA/IIoT primary application publishes the DCMD message to the MQTT server. The MQTT server ensures that the device receives the command and updates the data values as required. Totalflow MQTT-enabled devices support register writes requests only for the Plunger Application.</p>
Sparkplug STATE Message Count	<p>Number of critical application state messages.</p> <p>This message type is used to indicate the state of the primary SCADA/IIoT host application(s). The SCADA/IIoT system acts as an MQTT client and must publish its state for its own connection and session with the MQTT server. The STATE of the application can be:</p> <ul style="list-style-type: none"> — OFFLINE: The application is not connected, and the device/EoN tries to connect to the next registered MQTT server provided in the initial configuration page. — ONLINE: The application is connected. After reception of this message only, the device/EoN starts sending messages to the MQTT server (Birth and delta messages are not required). <p>The MQTT server ensures that the device is aware of the application state.</p>

[Table 14-12](#) describes each of the packet types the device sends to the MQTT server.

Table 14-12: Sparkplug Packet sent statistics

Name	Description
Sparkplug NBIRTH Message Count	<p>Number of Node birth certificate (NBIRTH) messages.</p> <p>The device sends this message type to communicate that it has established a session with the MQTT broker and is ready to start publishing its data. The NBIRTH message is the first message that the device publishes upon establishing a session.</p>
Sparkplug NDATA Message Count	<p>Number of Edge of Node (EoN) Data (NDATA) messages.</p> <p>The device sends this message type to enable the continuous session awareness that monitors the state of the Edge of Node connection to the cloud.</p> <p>The device sends this message type to send the latest values of node parameters like CPU, memory usage, etc. to the SCADA System via MQTT Broker based on publish interval.</p> <p>Note that the MQTT-enabled Totalflow device acts as an Edge of Node (EoN) when sparkplug is used. No separate Edge of Node device (gateway) is required for sparkplug support since this functionality is implemented as part of the MQTT stack in the Totalflow device.</p>

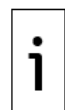
Name	Description
Sparkplug DBIRTH Message Count	Number of device birth certificate (DBIRTH) messages. The device sends this message type to communicate that it has established a session with the MQTT broker and is ready to start publishing its application/device data. The DBIRTH message is the message sent just after NBIRTH message.
Sparkplug DDATA Message Count	Number of Device Data (DDATA) messages. The device sends this message type to send the changed values of device/application parameters, like application registers to the SCADA System via MQTT Broker.

15 Useful terms

Table 15-1 provides a general description of the terms used in this manual for quick reference. For technical details on protocol implementation, infrastructure components or cloud architecture definitions, consult standard committees' websites or other online resources.



IMPORTANT NOTE: Refer to online resources for the MQTT standard documentation at this link: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf>.



IMPORTANT NOTE: Refer to online resources for the Sparkplug protocol at this link: <https://sparkplug.eclipse.org/specification/version/3.0/>.

Table 15-1: Useful terms

Term/Acronym	Description
ABB Ability™	A set of tools, software processes and data models available for each ABB cloud-based domain-specific solution. The Totalflow web applications are solutions specific for oil and gas upstream production and constitute one of the many ABB solutions offered on the cloud.
Cloud/Cloud Services	Hardware and software infrastructure enabling connectivity of devices, systems and processes across a large geographical area. Cloud solutions can be offered over proprietary vendor-owned infrastructures or over third-party service providers. ABB offers solutions over the Microsoft® Azure Platform or cloud services.
IoT	Internet of Things Hardware and software platforms supporting remote device integration for web-based access to device data and control capabilities
IIoT	Industrial Internet of Things The use of the Internet of Things platforms to support and enhance industrial and manufacturing processes such as factory or plan-floor control, automation, and other complex systems.
IoT Hub (device)	System on the cloud service platform processing communication with field MQTT clients (MQTT-enabled field devices)
MQTT	Message Queue Telemetry Transport (Standard MQTT) A client-server publish-subscribe messaging protocol for use on top of the TCP/IP protocol. This protocol enables connectivity and integration of field devices into the cloud. Packet payload for the standard MQTT protocol supports the ABB Ability format.
MQTT client	Functionality that performs the client role in MQTT communication. Typically implemented on field devices.

Term/Acronym	Description
MQTT server	Functionality that performs the server role in MQTT communication. Typically implemented on systems serving as IoT hubs or MQTT brokers.
MQTT-enabled field device	ABB Totalflow devices with embedded capability to connect and communicate with an MQTT broker. These devices support the MQTT client functionality which requests connections to the broker and establishes the communication links for data transfer to and from the broker.
MQTT Broker	The system with the MQTT server functionality that authenticates and accepts connection requests, establishes communication links, and allows data transfer for MQTT clients.
MQTT Control packets	MQTT communication packets sent by client to server or server to client to establish the connection for the data transfer between the device and the cloud. MQTT has several types of control packet types: CONNECT, SUBSCRIBE, PUBLISH. Each of these packets has a specific function and format.
CONNECT packet	The first packet sent by the MQTT client to the MQTT server after the connection between the two is successfully established.
PINGREQ (Ping request) packet	<p>Packet is sent from a client to the server to:</p> <ul style="list-style-type: none"> — Indicate to the Server that the Client is alive in the absence of any other MQTT Control Packets being sent from the Client to the Server — Request that the Server responds to confirm that it is alive — Exercise the network to indicate that the Network Connection is active <p>This packet is used in Keep Alive processing.</p>
PINGRESP (PING response) packet	<p>Packet is sent by the server to the client in response to a PINGREQ packet. It indicates that the server is alive.</p> <p>This packet is used in Keep Alive processing.</p>
DISCONNECT (Disconnect notification packet)	Final MQTT Control Packet sent from the client or the server before device-broker connection is closed
SUBSCRIBE (request) packet	Packet sent from the client to the server to create one or more subscriptions. Each subscription registers a Client's interest in one or more Topics. The Server sends PUBLISH packets to the Client to forward Application Messages that were published to Topics that match these Subscriptions. The SUBSCRIBE packet also specifies (for each Subscription) the maximum QoS with which the Server can send Application Messages to the Client.
UNSUBSCRIBE packet	Packet sent by the Client to the Server to unsubscribe from topics
PUBLISH packet	A PUBLISH packet is sent from a Client to a Server or from a Server to a Client to transport an Application Message.
Payload	The actual data in a packet or file minus all headers attached for transport and minus all descriptive meta-data. The payload format depends on the communication protocol used: MQTT or Sparkplug.
Topic	Topic Name that identifies the information channel to which payload data is published.
MQTT TCP port	TCP port number assigned for the MQTT protocol. TCP ports 8883 and 1883 are registered with IANA for MQTT Transport Layer Security (TLS) and non-TLS communication respectively. Port 8883 is recommended for secure connection.

Term/Acronym	Description
Sparkplug	Communication protocol that enhances the standard MQTT protocol to support field device connection with real-time SCADA or IIoT systems. The Sparkplug packet payload format is different from the format used by standard MQTT. Sparkplug requires specific payload format definitions.

ABB Inc.

Measurement & Analytics

Quotes: US-IAMA.inquiry@us.abb.com

Orders: US-IAMA.order@us.abb.com

Training: US-IAMA.training@us.abb.com

Support: upstream.support@us.abb.com

+1 800 442 3097 (opt. 2)

Additional free publications are available for download at:

www.abb.com/upstream

Main Office - Bartlesville

7051 Industrial Blvd
Bartlesville, OK 74006
Ph: +1 918 338 4888

Texas Office - Houston

3700 W. Sam Houston
Parkway S., Suite 600
Houston, TX 77042
Ph: +1 713 587 8000

We reserve the right to make technical changes or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.

We reserve all rights in this document and in the subject matter and illustrations contained therein. Any reproduction, disclosure to third parties or utilization of its contents - in whole or in parts - is forbidden without prior written consent of ABB.

Ignition Designer® is a registered trademark of Inductive Automation LLC.

2106521MNAC

Copyright© 2024 ABB all rights reserved