

# Managing historic data with MQTT Modules

## MQTT Store and Forward

MQTT Store and Forward allows data to be buffered locally at a client when connections are down to the MQTT Server infrastructure and deliver that data when the connection is restored. This feature is critical in most applications because if we lose connection to the MQTT Server we will lose data if it is not buffered locally.

### [MQTT Store and Forward Overview](#)

- Provides an overview of Store and Forward within an MQTT environment

## Storing data with MQTT Transmission

When Store and Forward is configured at MQTT Transmission and the edge node detects a disconnect to the MQTT Server, tags will be stored locally in a history store. When the edge node can reconnect to the MQTT Server, it will publish any stored tags.

### [Determining the settings for an MQTT Transmission History Store](#)

- Shows how to determine the settings for an MQTT Transmission History Store

### [MQTT Transmission History Store - Rolling History Buffer](#)

- Describes how the MQTT Transmission History Store Rolling History Buffer works

### [Minimizing data loss when using MQTT Store and Forward](#)

- Describes the use of Keep Alive and Primary Host ID by MQTT Transmission and MQTT Engine within a Store and Forward system

## Processing historic data with MQTT Engine

MQTT Engine has two methods of historical event processing resulting in historical inserts into [Ignition's Tag Historian Module](#).

Engine can be configured to write historical events directly to the database, via the Historian, bypassing the Tag or it can be configured to write historical events to the Tag instead of directly to the Historian.

### [Configuring history on MQTT Engine tags](#)