

MQTTJSON Package Documentation

Revision 1.0 March 15, 2023

The MQTTJSON package allows you to sync your CSV data file in JSON format using a custom template over MQTT. One line of the CSV file results in a single MQTT message, allowing you to encapsulate the data in the way you need.

Installation

Upload the latest mqttjson_YYYYMMDD.tar.gz file to the root (/) folder of your Wattmon and extract it.

Next, go to Control Panel > Package Manager and click the Rescan Folder button.

Once the MQTTJSON package appears in the list, click the Action > Install link at the right of it.

Next, configure the package with Action > Configure

Configuration Options

Debug Mode

☒ Disabled
☐ Enabled

MQTT JSON debug mode (to log file)

MQTT JSON Publishing

Publish Channel /pub/test

MQTT Channel

Publish interval in seconds 15

This can be reduced below 60 seconds to push historic data. Since the package uses the CSV data, it will not push more than once per minute when all historic data is pushed

MQTT Template File /package/mqttjson/template.txt

Path and filename (i.e. /config/mqtt_template.txt)

[EDIT TEMPLATE](#)

[APPLY CHANGES](#)

Set the publish channel to push your data to in the Publish Channel field.

Set the interval to the number of seconds between pushes – this is only relevant for historic data.

Once all data has been synced, it will only push new data once the CSV file has been appended to.

Choose the template you wish – there is a default in the *package/mqttjson/template.txt* location.

The template format is described below:

Template Format

The template can be plain text or JSON. Use the {{ opening and }} closing tag to encapsulate functions.

The dataset for the CSV row can be accessed using specific functions:

rolebyname('str')

Get the value of a role name that is logged in the CSV. The only value that is not dynamically defined by the EZConfig settings is ts – the unix timestamp of the row.

Examples:

rolebyname('ts') – this gets the timestamp value for the row

rolebyname('inverter1_AC_Active_Power') would get the active power role value

rolebyindex(x)

Get the value of a role by its index. Refer to the CSV log file for a day to get the indexes.

Examples:

{{rolebyindex(0)}} – get the timestamp

rolename(x)

Get the name of a role name at column index x that is logged in the CSV.

Examples:

rolename(0) – this gets the timestamp value for the row

all_roles('format')

Get all roles as a comma separated list using the format specifier.

There are two valid formats:

'v' – this returns just the values

'n:v' – this returns the "name":value

Examples:

{{all_roles("n:v")}} – will return a comma separated list of all roles

all_roles_with('text','format')

Get all roles that contain the text string as a comma separated list using the format specifier.

There are two valid formats:

'v' – this returns just the values

'n:v' – this returns the "name":value

Examples:

`{{all_roles_with("inverter","n:v")}}` – will return a comma separated list of all roles that contain 'inverter'

Full examples:

```
{ "ts": {{rolebyname('ts')}} },  
  "roles": [ {{x=0}}  
    {{DO}} {{x=x+1}}  
    "rolename(x)={{rolename(x)}}":rolebyindex(x)={{rolebyindex(x)}}  
    {{WHILE x<10}}  
  ] }
```

This will loop through roles 1 to 10 and display the name and value – this is not very efficient and not recommended.

```
{ "ts": {{rolebyname('ts')}} },  
  "data": { {{all_roles('n:v')}}  
}  
}
```

This will produce a JSON template with all roles as key:value pairs in the data object.