

RUTM10 Data to Server

[Main Page](#) > [RUTM Routers](#) > [RUTM10](#) > [RUTM10 Manual](#) > [RUTM10 WebUI](#) > [RUTM10 Services section](#) > **RUTM10 Data to Server**

The information in this page is updated in accordance with firmware version [RUTM_R_00.07.07.3](#).



Contents

- [1 Summary](#)
- [2 Data Senders](#)
 - [2.1 Sender Settings](#)
 - [2.1.1 General](#)
 - [2.1.2 Collection settings](#)
 - [2.1.3 Server configuration](#)

Summary

The **Data to Server** feature provides you with the possibility to set up data senders that collect data from various sources and periodically send it to remote servers.

If you're having trouble finding this page or some of the parameters described here on your device's WebUI, you should **turn on "Advanced WebUI" mode**. You can do that by clicking the "Advanced" button, located at the top of the WebUI.



Data Senders

A **Data Sender** is an instance that gathers and periodically sends collected data to a specified server. The Data Senders list is empty by default so, in order to begin configuration you must add a new data sender first. To add a new data sender, click the 'Add' button.



After this you should be redirected to the newly added data sender's configuration page.

Sender Settings

Refer to the table below for descriptions on data sender configuration fields.

Note: these tables have coloring schemes to indicate which fields can be seen with different

configuration.



General

Field	Value	Description
Name	string; default: none	Name of the data sender. Used for easier data senders management purposes only (optional).
Type	Base MNF info Modbus* Modbus Alarms Wifi scanner* DNP3* MQTT OPC UA DLMS* ; default: Base	Source of the data to be sent to server.
Format type	Json Custom ; default: Json	Arranges the format of the sent JSON segment.
Format string	string; default: none	Specifies custom format string.
Empty value	string; default: N/A	A string which will be placed if any value cannot be received.
Delimiter	string (Maximum length of value is 1 bytes); default: N/A	Specifies delimiters for multiple data segments.
Segment count	integer [1..64]; default: 1	Max segment count in one JSON string sent to server.
Send as object	off on; default: off	When turned on, sends JSON segment as object and not as an array element.
Data filtering	All Server ID Alarm ID Register number; default: All	If Data type: Modbus alarms data . Choose which data this sender will send to server.
Data filtering	All Server IP address Server ID Request name ; default: All	If Data type: Modbus data . Choose which data this sender will send to server.
Data filtering	All Name MAC address Signal strength ; default: All	If Data type: Wifi scanner . Choose which data this sender will send to server.
Data filtering	All Name ; default: All	If Data type: DLMS . Choose which data this sender will send to server.
Invert file	off on; default: off	If Data type: DLMS . Inverts filter condition.
Data filtering	All Address IP ; default: All	If Data type: DNP3 . Choose which data this sender will send to server.
Database	RAM Flash; default: RAM	Database location
Data filtering	All Name ; default: All	If Data type: OPC UA . Choose which data this sender will send to server.
Server address	Default: empty	Hostname or ip address of the broker to connect to.
Port	integer [0..65535]; default: 1883	Port number for connecting to MQTT.
Keepalive	integer [1..640]; default: 60	MQTT Keepalive period in seconds.
Topic	string; default: none	MQTT topic to be used for publishing the data.

Client ID	string; default: none	Client ID to send with the data. If empty, a random client ID will be generated
QoS	integer [0..2]; default: 0	MQTT Quality of Service. Allowed values: <ul style="list-style-type: none"> • 0 - when we prefer that the message will not arrive at all rather than arrive twice, • 1 - when we want the message to arrive at least once but don't care if it arrives twice (or more), • 2 - when we want the message to arrive exactly once. A higher QoS value means a slower transfer.
Enable secure connection	off on ; default: off	Enables the use of TLS certificates.
On: TLS type	Certificate based Pre-shared key based; default: Certificate based	Select type of TLS.
Certificate based: Allow insecure connection	off on ; default: off	Allow not verifying server authentication.
Certificate based: Certificate files from device	off on ; default: off	Specify where the certificates will be used from.
Certificate based: CA File	.ca file; default: none	Certificate authority is an entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate. Certificate file is a type of digital certificate that is used by client systems to make authenticated requests to a remote server. If client certificate is not needed, leave both client certificate and client key fields empty.
Certificate based: Client certificate	.crt file; default: none	
Certificate based: Client private Key	.key file; default: none	File containing private key for this client. This file needs to be not encrypted.
Pre-shared key based: Pre-Shared-Key	string; default: none	The pre-shared-key in hex format with no leading "0x".
Pre-shared key based: Identity	string; default: none	The identity of this client. May be used as the username depending on the server settings.
Username	string; default: none	Username used in authentication.
Require password	on off ; default: off	Enables password for authentication.
Password	string; default: none	Password used in authentication.

* This is additional software that can be installed from the **System** → [Package Manager](#) page.

Collection settings



Field	Value	Description
Enabled	off on ; default: on	Enables data to server collection instance.

Format type	Json custom ; default: Json	Data collection objects formatting.
Format string	Default: Instance name	Specifies custom format string
Empty value	Default: N/A	A string which will be placed if any value cannot be received
Period	Default: 60	Interval in seconds for collecting/sending data to destination.
Retry	off on ; default: off	In case of a failed attempt, retry to send the same data to destination later.
Retry count	Default: 10	Retry to send the same data N times
Timeout	Default: 1	Timeout in second between retry attempts

Server configuration



Field	Value	Description
Type	HTTP MQTT ; default: HTTP	Interval in seconds for collecting/sending data to destination.
Server address	Default: empty	Hostname or IP address of the broker to connect to.
HTTP headers	Default: empty	Allows to add custom headers to the HTTP requests.
Enable secure connection	on off; default: off	Enables the use of TLS certificates.
Port	integer [0..65535]; default: 1883	Port number for connecting to MQTT.
Keepalive	integer [1..640]; default: 60	MQTT Keepalive period in seconds.
Topic	string; default: none	MQTT topic to be used for publishing the data.
Client ID	string; default: none	Client ID to send with the data. If empty, a random client ID will be generated
QoS	integer [0..2]; default: 0	MQTT Quality of Service. Allowed values: <ul style="list-style-type: none"> • 0 - when we prefer that the message will not arrive at all rather than arrive twice, • 1 - when we want the message to arrive at least once but don't care if it arrives twice (or more), • 2 - when we want the message to arrive exactly once. A higher QoS value means a slower transfer.
Enable secure connection	off on ; default: off	Enables the use of TLS certificates.
On: TLS type	Certificate based Pre-shared key based; default: Certificate based	Select type of TLS.
Certificate based: Allow insecure connection	off on ; default: off	Allow not verifying server authentication.

Certificate based: Certificate files from device	off on; default: off	Specify where the certificates will be used from.
Certificate based: CA File	.ca file; default: none	Certificate authority is an entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate.
Certificate based: Client Certificate	.crt file; default: none	Certificate file is a type of digital certificate that is used by client systems to make authenticated requests to a remote server. If client certificate is not needed, leave both client certificate and client key fields empty.
Certificate based: Client Private Key	.key file; default: none	File containing private key for this client. This file needs to be not encrypted.
Pre-shared key based: Pre-Shared-Key	string; default: none	The pre-shared-key in hex format with no leading "0x".
Pre-shared key based: Identity	string; default: none	The identity of this client. May be used as the username depending on the server settings.
Use credentials	off on; default: off	Enables use of username and password for authentication.
On: Username	string; default: none	Username used in authentication.
On: Password	string; default: none	Password used in authentication.