

IoT Gateway Plug-In

©2015 Kepware, Inc.

Table of Contents

Table of Contents	2
Internet of Things Gateway	5
Overview	5
Architectural Summary	5
External Dependencies	6
General Operation	7
User Interface	8
Configuration	9
Configuring the Gateway	9
Configuring a New Agent	10
Configuring an MQTT Connection	11
MQTT Client Message	12
MQTT Client Security	13
MQTT Last Will and Testament	14
Configuring a REST Client Connection	15
REST Client Header	15
REST Client Body	16
REST Client Security	17
Configuring a REST Server Connection	19
Working with a REST Server	20
Changing an Agent Configuration	21
Configuring a Gateway Certificate	21
Configuring a Self-Signed Certificate	22
Command Line Steps	22
Windows Console Steps	22
Licensing	27
Data	28
Data Format	28
Adding Tags to an Agent	29
Adding a Single Tag	29
Adding Multiple Tags	29
System Tags	31
Importing / Exporting CSV Files	31
Troubleshooting	32
Event Log Messages	32
Browse rejected: no user credentials were provided in the request and anonymous requests are currently disabled.	33
Browse rejected: the credentials for user <user> are invalid.	33
Connection restored to server: <gateway>. Reinitializing server configuration.	34
Data change event buffer overrun; dropping updates. Ensure that the IoT Gateway service is running or reduce the volume of data collected.	34
Error adding item <tag> to connection <agent>.	34

Error adding item <tag>. This item already exists in connection <agent>.	34
Error importing CSV data. Invalid CSV header.	35
Error importing CSV data. No item records found in CSV file.	35
Error importing CSV item record <tag>. Update rate <update rate> is out of range, setting to <valid update>.	35
Error importing CSV item record <tag>. No update rate found, setting to <update rate>.	35
Error importing CSV item record <tag>. Deadband <deadband rate is invalid. Deadband set to <valid deadband>.	36
Error importing CSV item record <tag>. No deadband value found, setting to <valid deadband>.	36
Failed to connect to server: <gateway>. Please verify this connection information is correct and that the host can be reached.	36
Failed to start IoT Gateway service.	37
Failed to start IoT Gateway service. Please ensure arguments <Java variables> are valid.	37
Failed to connect to server: <URL and port>. Please verify this connection information is correct and that the host can be reached.	37
Failed to create JVM using JRE at <path to JRE>.	37
Failed to import server instance cert: <agent>. Please use the Administration utility to re-issue the certificate.	38
Failed to initialize the JVM: insufficient memory available (requested initial=<MB>, max. =<MB>).	38
Failed to initialize the JVM: JNI error <Error>.	38
Failed to initialize the IoT Gateway.	38
Failed to launch IoT Gateway: no suitable 32-bit JRE was configured or found.	38
Failed to load XML project. Item <tag> already exists in connection <agent>.	39
Failed to load project: <agent URL> is not a valid address.	39
Failed to load agent <agent>: invalid payload specification.	39
IoT Gateway using JRE at <path to JRE>.	39
IoT Gateway failed to start. Failed to bind to port <port>.	39
Item <tag> on connection <agent> is now licensed and sending data.	40
Missing server instance certificate <agent>. Re-issue the certificate using the Administration utility.	40
MQTT agent <agent name> disconnected – reason: Connection lost.	40
MQTT agent <agent name> dropped data change events.	40
MQTT agent <agent name> failed to parse payload.	40
MQTT agent <agent name> failed to publish - reason: <Broker URL>.	41
MQTT agent <agent name> failed to publish - reason: Connection reset.	41
MQTT agent <agent name> failed to publish - reason: Unable to connect to server.	41
MQTT agent <agent name> is connected to broker <broker URL>.	41
Read rejected for item <tag>: no user credentials were provided in the request and anonymous requests are currently disabled.	41
Read rejected for item <tag>: the credentials for user <user> are invalid.	42
Read rejected for item <tag>. No user credentials were provided in the request and anonymous requests are currently disabled.	42
REST client <agent name> dropped data change events.	42
REST client <agent name> failed to parse payload.	42
REST client <agent name> processing update.	43
REST client <agent name> publish failed - reason: Connection refused: connect.	43
REST client <agent name> publish failed - reason: Read timed out.	43
REST client <agent name> publish failed - reason: SSL configuration error.	43

REST client <agent name> publish failed - reason: Unexpected EOF.43

REST client <agent name> returned HTTP error <HTTP error>, buffering records.44

REST client <agent name> started publishing to <REST server URL>. 44

REST server <agent name> started at <URL and port>. 44

REST server <agent name> - failed to start on <URL and port>, reason: Address already in use: bind.44

Running with Java <full Java version>. 44

The REST server certificate has been reissued. 44

The REST server certificate has been imported.45

The REST server certificate has expired. Please use the Administration utility to re-issue the certificate. 45

Unable to send data for item <tag> on connection <agent>. The licensed item count of <license count> items has been reached. 45

Unable to start secure REST server <agent name> at <URL and port>: missing or invalid certificate. .45

Unable to use network adapter <network adapter> for REST server <agent>. Binding to localhost only. 45

Unsupported JVM: please install or configure a 32-bit Java 1.6 or higher JRE or JDK. 46

Write request failed on item <tag>. The write data type <data type> cannot be converted to the tag data type <data type>. 46

Write rejected for item <tag>: no user credentials were provided in the request and anonymous requests are currently disabled. 46

Write rejected for item <tag>: the credentials for user <user> are invalid. 46

Index **47**

Internet of Things Gateway

Help version 0.013

CONTENTS

[Overview](#)

What is the Internet of Things (IoT) Gateway plug-in?
What can the plug-in do?
What is the data format?

[Configuration](#)

What other software is needed to run the IoT Gateway Plug-In?
How do I add an agent connection?
How do I add an IoT Gateway tag item?

[Troubleshooting](#)

How to I find and correct issues?
What messages does the IoT Gateway plug-in produce?

Overview

The IoT Gateway Plug-In is an optional feature of KEPServerEX that allows system and device tags to be published to third-party endpoints through industry standard IP based protocols. When the value for a configured tag changes or when a publish rate is met, an update is sent to the corresponding third-party endpoint with a configurable payload of tag ID, value, quality and timestamp in a standard JSON format. The IoT Gateway Plug-In offers the following features:

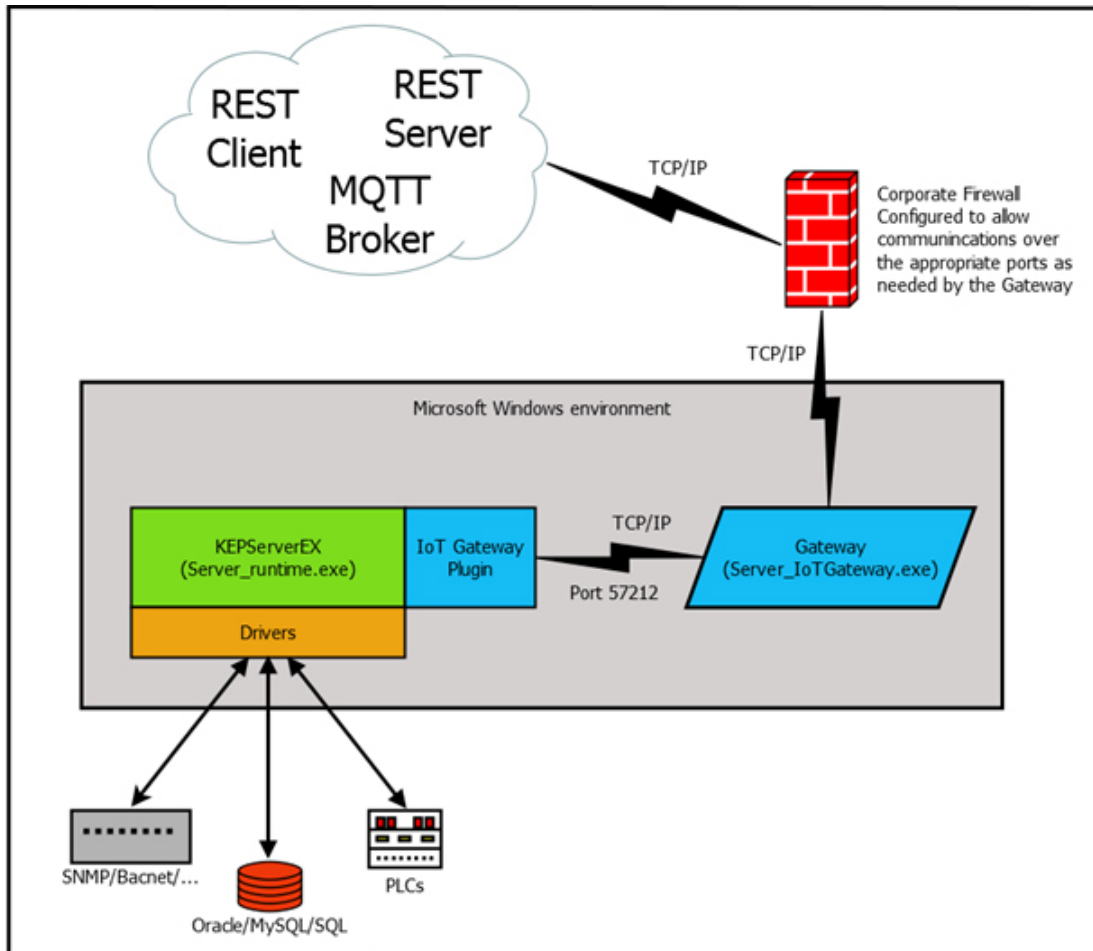
- Ability to publish data consisting of a name, value, quality, and timestamp from any data source in the server (e.g. drivers, plug-ins, or system tags)
- Standard human readable JSON data format
- Support for publish via MQTT, REST client, and two way communication via REST server agents
- Configurable data collection rate, as frequent as 10 milliseconds up to once per 27.77 hours (99999990 milliseconds) for the REST and MQTT Client
- Configurable data publish rate, as frequent as 10 milliseconds up to once per 27.77 hours (99999990 milliseconds) for the REST and MQTT Client
- Support for authentication and TLS / SSL encryption on all agents
- Support for user-level access based on the KEPServerEX User Manager and Security Policies Plug-In
- Configurable header and payload information for easy integration with different third party endpoints

Architectural Summary

The IoT Gateway Plug-In feature includes two main components:

- The server plug-in (IoT_Gateway.dll) is responsible for:
 - Configuration of the MQTT, REST client and, REST server agents
 - Data collection from the server runtime
 - Configuration of the Gateway settings
 - License enforcement
- The IoT gateway system service (server_iotgateway.exe), which:
 - Manages the connections to each third party endpoint
 - Buffers data collected from the plug-in
 - Provides the authentication and encryption layer to each agent

This diagram shows the layout of the IoT Gateway Plug-In and components. The plug-in and gateway install on the same machine with KEPServerEX. KEPServerEX uses drivers to connect to data sources. That data is collected in the plug-in and sent to the gateway. The gateway publishes that data to the configured endpoint(s). In this diagram, data flows from the device/data sources at the bottom up to the endpoints at the top.



External Dependencies

For the gateway to run, KEPServerEX requires a working 32-bit Java JRE or full JDK installation version 6 or higher. At this time, a 64-bit JRE or JDK is not supported. Kepware recommends the most current supported version of Java for use with the IoT Gateway. The current JRE may be downloaded and installed from Oracle at the following link:

<https://java.com/en/download/>

At the time of publication Java 8 with all updates has been tested and confirmed to be compatible.

Tip: Java does not need to be enabled in your browser for the gateway to run.

Note: As the IoT Gateway is a product that has the potential to push data across the Internet to third party endpoints, it is recommended that you configure your computer or corporate firewall appropriately to allow just the specific ports that those endpoints are configured to use.

To prevent the loss of data and to keep your KEPServerEX instance running properly, it is recommended when using Java version 7 or earlier that you do not try to do a Java update while in production. The Runtime service and Gateway service should be taken offline before a Java update is run. Java 8 has changed the way that it updates, allowing multiple versions of Java to exist on the computer at the same time. With Java 8 a new version is placed side by side with the existing version. In this scenario, the default Gateway behavior is to continue to use the old version of Java 8 until a reboot, an IoT Gateway restart or a change to the Java configuration in the Server Settings. If you have specified the JRE or JDK to use for the IoT Gateway in the Server Settings, it will continue to use that version even after an update. You may see what version of Java you are running by looking through the Event log entries in KEPServerEX. Please see our online Knowledge Base for specific recommendations for updating Java.

General Operation

This section explains how the two components described above work together to form the basis of the IoT Gateway. This discussion also serves as an introduction to the terminology used in the remainder of this document.

Initialization

An agent configuration is created using the IoT Gateway Plug-in from within the Server Configuration user interface. Details of this are covered later in the document. When an agent is configured and a runtime connection with the Server Configuration exists, the `server_iotgateway` service starts as directed by the Plug-in. At this time, the configuration is transferred from the Plug-in to the gateway where it is initialized. There may be multiple configurations for the same type of endpoint in the Plug-In. Each of these configurations creates its own instance on the gateway.

Startup

At system startup with a configured agent, the Server Runtime loads its project file (e.g. `default.opf`). Upon detecting that an agent is defined, the plug-in starts the `server_iotgateway` service. The plug-in establishes a connection to the IoT gateway service and transmits the active agent configuration(s).

Data Updates

Data updates are managed by the plug-in for the REST and MQTT Clients. The agent creates a server item reference from each configured tag and polls for data at the configured scan rate like any other client. Scan rates are configured on a per tag basis. The updates received are forwarded to the server gateway service, where they are buffered and eventually pushed to the third party endpoint at the configured publish rate.

Each data update persisted to the agent consists of four elements: ID, value, quality, and timestamp.

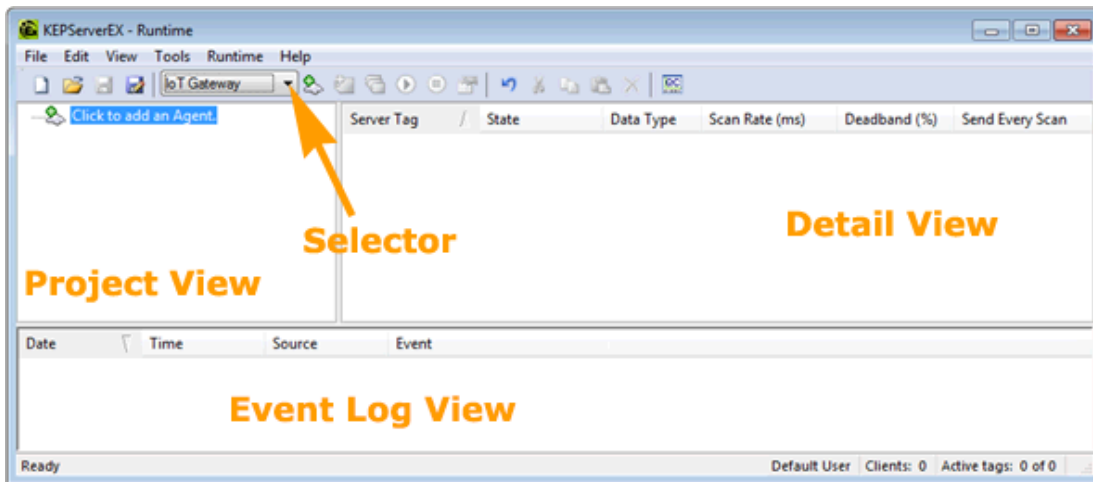
New data is pushed to the gateway when there is a change in value, unless the "every scan" box is checked at the tag level. This feature sends an update to the gateway to be published to an endpoint for every good-quality scan of the tag whether or not there was a data change. A bad-quality scan will be sent only once. When the "every scan" radio button is selected on a tag, the deadband setting for that tag is ignored.

Shutdown

When the Server Runtime receives a request to shutdown, the IoT Gateway plug-in is responsible for stopping data collection. After sending the final tag updates, the IoT Gateway plug-in uses the messaging interface to tell the Server Gateway Service to close any active TCP/IP connections to third party endpoints.

User Interface

Within the KEPServerEX Configuration window, the IoT Gateway plug-in is accessible from either the **View** menu or from the Selector drop down menu in the toolbar. Once the IoT Gateway is selected, the interface should appear as below:



Project View - the left pane where agents are configured.

Detail View - the upper right where tags are configured for the selected agent.

Event Log View - the lower pane where messages, warnings, and errors are presented.

Note: If the IoT Gateway is not available from the View menu or drop-down; re-run the setup, choose the modify option, and install the IoT Gateway plug-in.

Consult the main KEPServerEX help system for more detail about menu and button functions.

Configuration

Configuration of the IoT Gateway is accomplished in two places. The server_iotgateway service is configured from the Settings sections of the KEPServerEX Administration Icon in the system tray. This is where Java settings and gateway-level changes may be made. Generally these settings do not need to be adjusted for the IoT Gateway to function properly. The agents and tags themselves are configured from the IoT Gateway Plug-In section of the KEPServerEX configuration window. Click any of the following for more information about configuration.

[Configuring the Gateway Settings](#)

[Configuring a New Agent](#)

[Configuring an MQTT Connection](#)

[Configuring a REST Client Connection](#)

[Configuring a REST Server Connection](#)

[Changing an Agent Configuration](#)

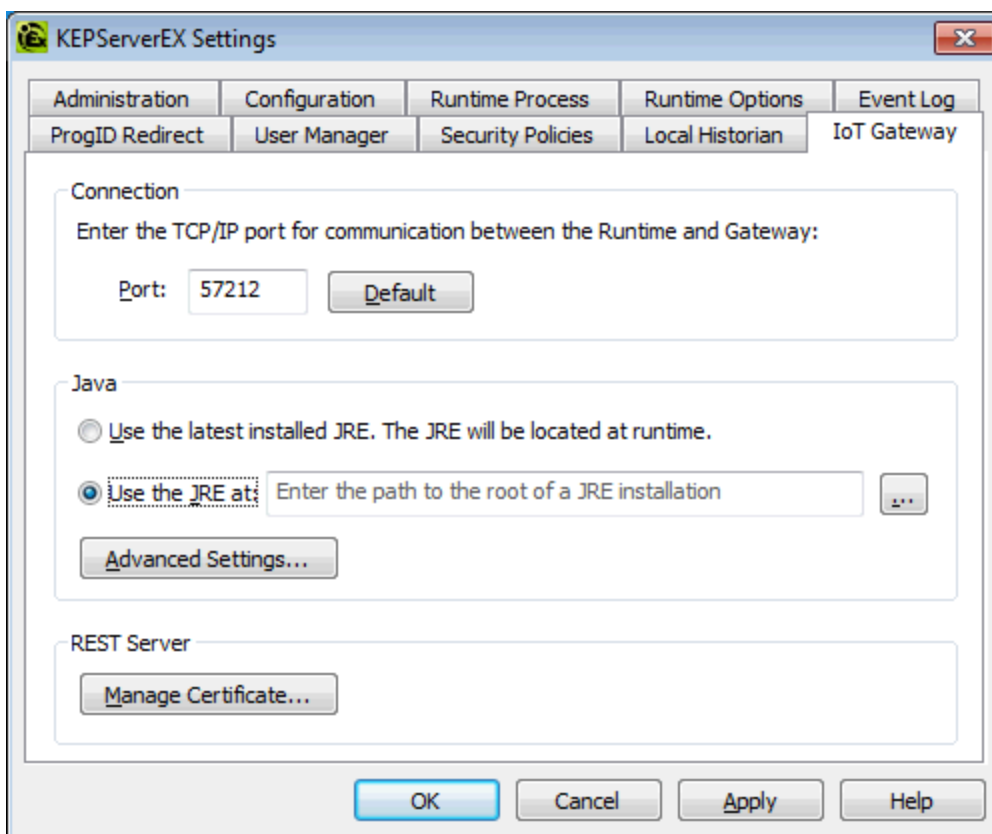
[License Configuration](#)

[Certificate Configuration](#)

Configuring the Gateway

The IoT Gateway administrative settings are automatically configured on installation. If the settings need to be adjusted, access the IoT Gateway system settings by right-clicking on the Administration icon located in the system tray and selecting **Settings | IoT Gateway**.

Tip: If the Administrative icon is not in the system tray, re-launch it by selecting **Start | All Programs | Kepware | KEPServerEX 5 | KEPServerEX 5 Administration | Settings**.



In the Connection area:

Port: specifies the TCP/IP port that the server runtime and configuration use to communicate with the gateway service. The valid range is 1024 to 65535. The **Default** button populates the field with the default port number of 57212, configured by the server.

Tips:

1. The default port is recommended unless there is a conflict with another local application using that port.
2. Before changing the port setting, verify there is no conflict with the new port number.
3. The gateway service does not accept remote connections, so there should be no firewall implications associated with this port assignment.

In the Java area:

Use latest installed version of the JRE locates and utilizes the newest 32-bit JRE installed on the system when the IoT Gateway starts.

To specify a specific JRE, de-select this option and enter the path to the JRE or use the Browse (...) button to locate the JRE.

Tip: If **Use latest installed version of the JRE** is selected and the Java version is updated on the machine, the gateway service automatically starts using the updated version the next time the IoT Gateway is started. If this option is disabled, the gateway service continues to use the specified version.

Advanced Settings... allow Java-specific settings to be used. These settings should only be changed if instructed by Technical Support.

In the REST Server area:

Click on the **Manage Certificate...** button to configure security certificate use for the REST server.

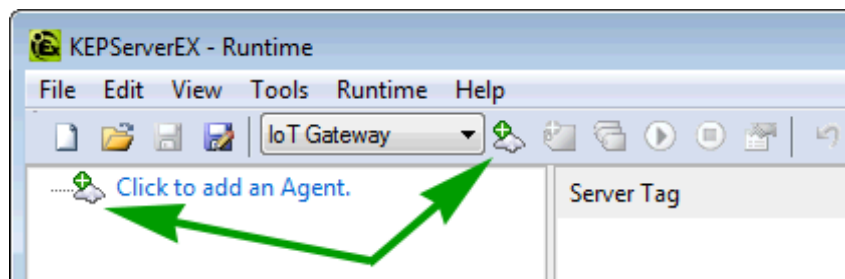
See Also:

[Configure Gateway Certificate](#)

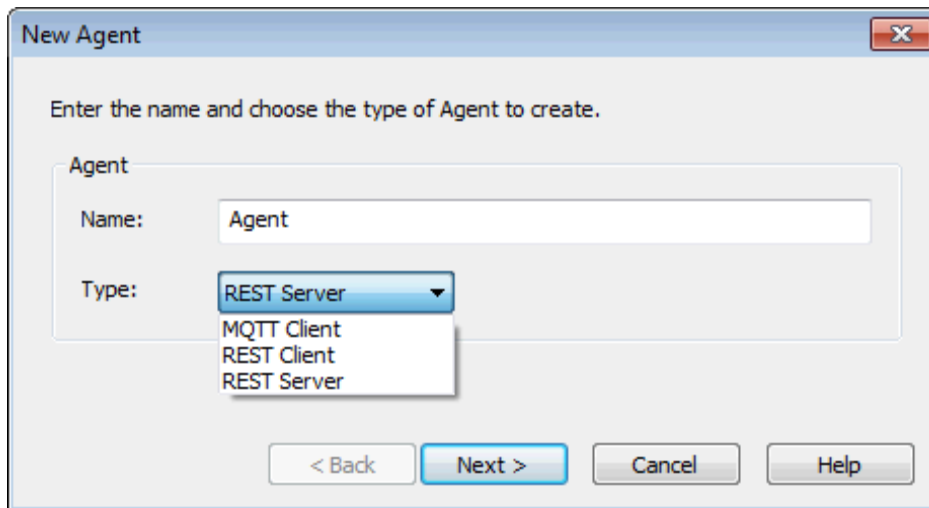
Configuring a New Agent

An agent configuration is required to begin publishing data to a third party endpoint. At least one agent needs to be configured with one active tag for the gateway service to start.

1. Click on the **"Click to add an Agent"** text or right-click any blank area in the agent pane and select **New Agent** from the pop up menu. Alternatively, click on the New Agent icon in the toolbar.



2. In the New Agent dialog, enter a name for the agent and select the type: MQTT Client, REST Client, or a REST Server agent.

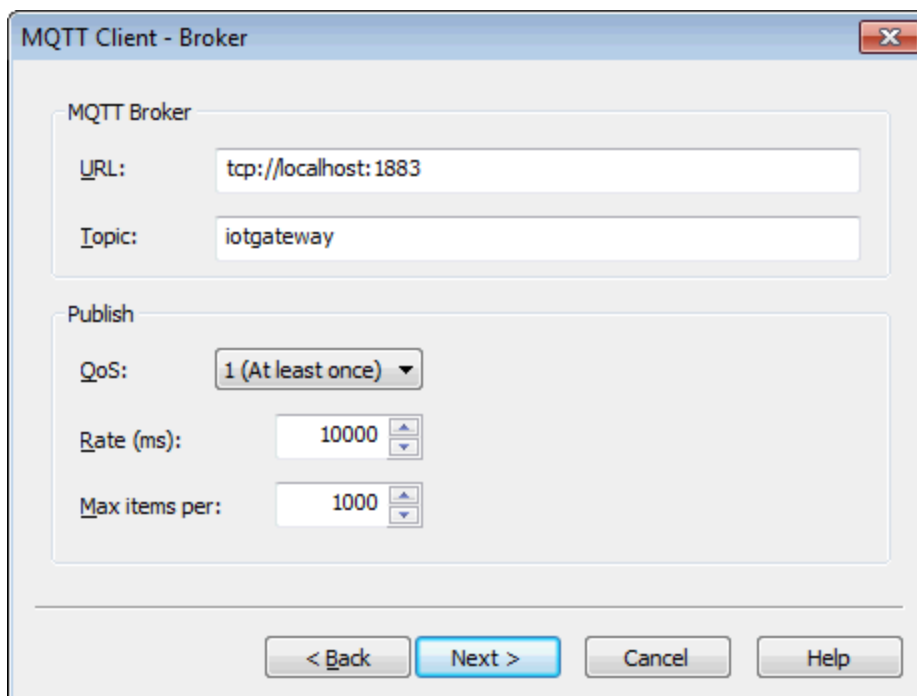


3. Click **Next >**.
4. Based on the type select, proceed to:
 - [Configure an MQTT Client](#)
 - [Configure a REST Client](#)
 - [Configure a REST Server](#)

Configuring an MQTT Connection

Once the [agent type](#) is selected as an MQTT client, follow the steps below to create a new MQTT agent connection:

1. In the MQTT Client dialog, name the broker and configure its settings.



URL: This is the IP address or URL and port of the endpoint for the agent connection.

Tip: If the endpoint uses an SSL connection, adjust the URL to use "ssl://" rather than "tcp://".

Topic: This is the name used to filter or organize data published on the broker.

QoS: this is the MQTT setting for publishing data Quality of Service. Choices include: 0 (at most once), 1 (At least once), 2 (exactly once).

Rate (ms): This sets the frequency of agent data pushes to the endpoint. The range is from 10 to 99,999,990 milliseconds.

Max. items per: This is the number of updates the Gateway packages into a single publish transmission.

2. Click **Next >**.
3. [Add tags](#) to the configuration.

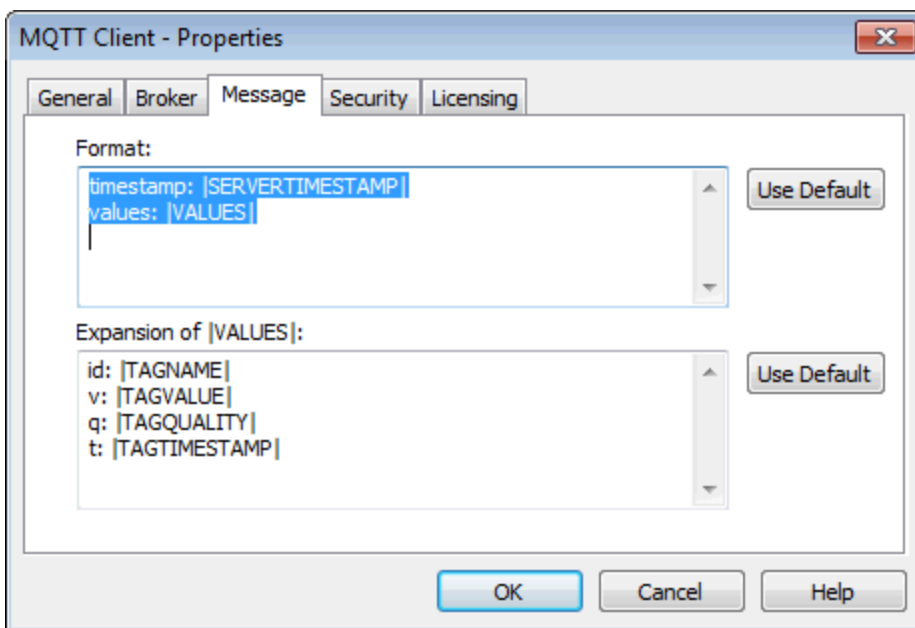
See Also:

[Adding Tags](#)
[Configuring Message Security](#)

MQTT Client Message

To change the order of the JSON data load or to remove data items, follow these steps:

1. Double-click on the agent name or right-click on the agent and select **Properties** to open the properties dialog.
2. From the Properties dialog, select the **Message** tab.
3. Make any desired changes (according to the guidelines below) and click **OK**.



Format: The format box contains any data to be sent before or after the JSON payload. This data is sent once per publish. Each variable may be preceded by any word or name, followed by a colon, then the variable. Key value pairs of static text may be included. Valid variables are:

|SERVERTIMESTAMP| the time and date when the gateway published the data to the endpoint, in UNIX or POSIX time format

|SERVERDATE| the same date and time as the timestamp, but in human-readable form

|VALUES| the combined payload of the following box

Click the **Use Default** button to reset the contents to the data and format set with the product before any changes are made.

Expansion of |VALUES|: This box contains the format for the JSON payload delivered to the endpoint. These values may be re-ordered or removed. A single publish event to the endpoint may include multiple instances of the data in this box. For example, if a tag had four data changes between publish events, there would be four complete JSON strings for that tag within this array. As with the Format box, each variable may be preceded by any desired word or descriptor. A colon is needed to separate the name from the variable. Valid variables are:

|TAGNAME| The name of the selected tag

|TAGVALUE| The value of the tag

|TAGQUALITY| This denotes if the tag was read as good or bad

|TAGTIMESTAMP| This is the time when the TAGVALUE was received

Click the **Use Default** button to reset the contents to the data and format set with the product before any changes are made.

Note: Each expression in this box needs to include a name-value pair.

MQTT Client Security

Follow the steps below to configure MQTT agent authorization and access:

1. In the Security dialog, enter the credentials below

Client ID: This is the unique identity for this agent's communication with the broker. Most brokers do not need a Client ID to connect.

Username: This is the authorized user for authentication on the broker.

Password: This is the password for basic authentication on the broker.

Note: If not using an SSL encrypted connection, the username and password are sent as plain text to the broker. This is a limitation of the protocol.

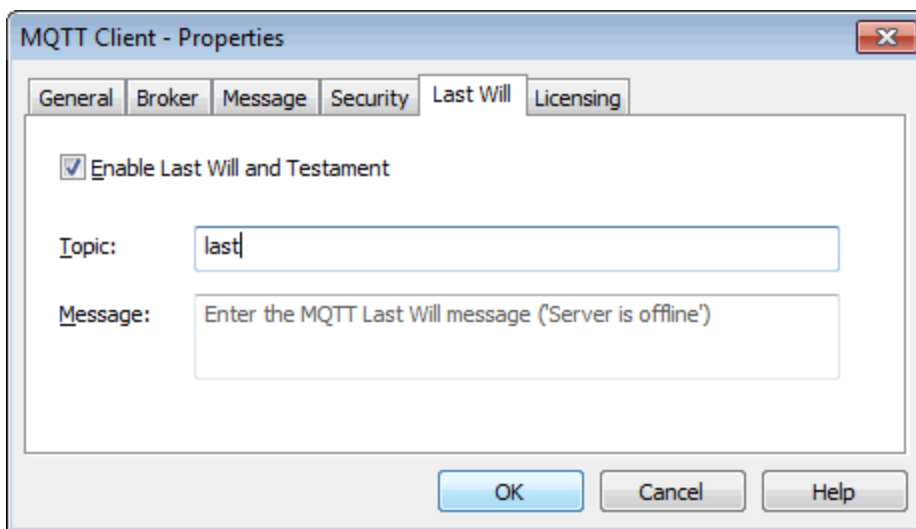
2. Click **Finish**.
3. Proceed to [Add tags](#) to the configuration.
4. The MQTT connection is now publishing to the broker. Verify the broker is receiving. If not, check the Event Log for errors.

See Also:
[Adding Tags](#)

MQTT Last Will and Testament

The "Last Will and Testament" (LWT) is a convention for MQTT to notify subscribed clients when a client has disconnected unexpectedly without a "DISCONNECT" notice. To enable and configure an MQTT Last Will and Testament, follow these steps:

1. Double-click on the agent name or right-click on the agent and select **Properties** to open the properties dialog.
2. From the Properties dialog, select the **Last Will** tab.
3. Click in the Enable Last Will and Testament check box to turn on a "last will" message.
4. Name the topic or path to serve as the LWT.
5. Enter the text that subscribed clients will receive in the LWT message. This is typically an explanation of the ungraceful disconnect, such as "offline" or "unexpected exit" to help the client.
6. Click **OK**.



In addition to the Last Will message, the MQTT agent publishes a message to the Last Will topic when the first data publish succeeds and when the MQTT agent is shut down due to a project edit, server reinitialize, or server shutdown.

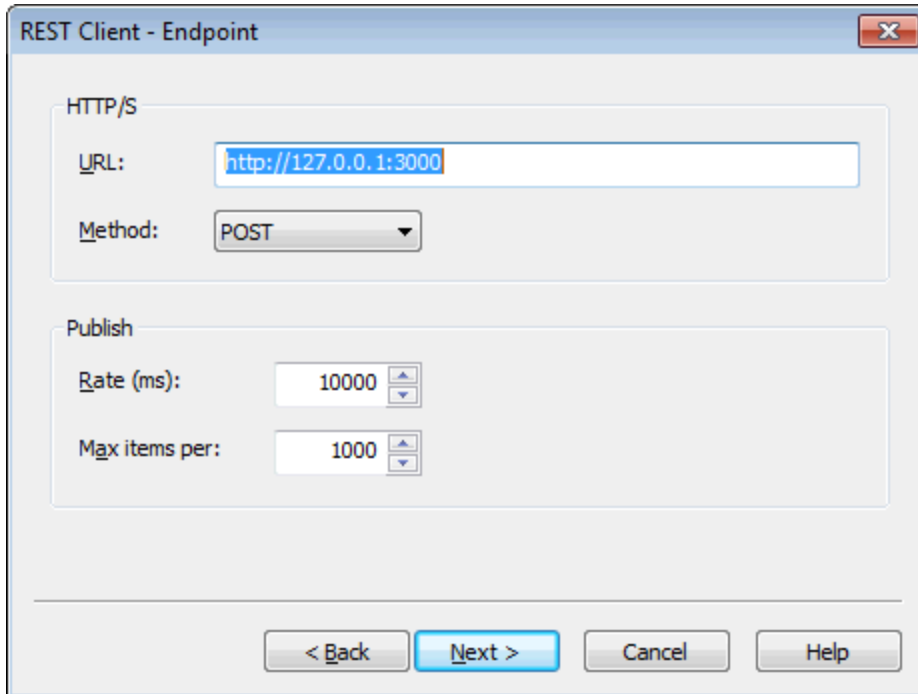
On first successful publish, the following text is published to the Last Will topic when the feature is enabled: "Server is online."

On graceful shutdown, the following text is published to the Last Will topic when the feature is enabled: "Server is shutting down."

Configuring a REST Client Connection

Once the [agent type](#) is selected a REST Client, follow the steps below to create a new REST Client agent connection

1. Define the REST Client Endpoint with the below fields



The screenshot shows a dialog box titled "REST Client - Endpoint". It is divided into two main sections: "HTTP/S" and "Publish".

- HTTP/S Section:**
 - URL:** A text input field containing "http://127.0.0.1:3000".
 - Method:** A dropdown menu currently set to "POST".
- Publish Section:**
 - Rate (ms):** A spin box set to "10000".
 - Max items per:** A spin box set to "1000".

At the bottom of the dialog, there are four buttons: "< Back", "Next >", "Cancel", and "Help". The "Next >" button is highlighted in blue.

URL: the IP address or URL and port of the endpoint for the agent connection. If the endpoint uses an SSL connection, adjust the URL in this box to use "https://"

Method: the way that the agent publishes data to the endpoint. It may be through a POST or PUT command.

Publish Rate: the frequency at which the agent pushes data to the endpoint.

Max. items per: This is the number of updates the Gateway packages into a single publish transmission.

2. Click [Next >](#).

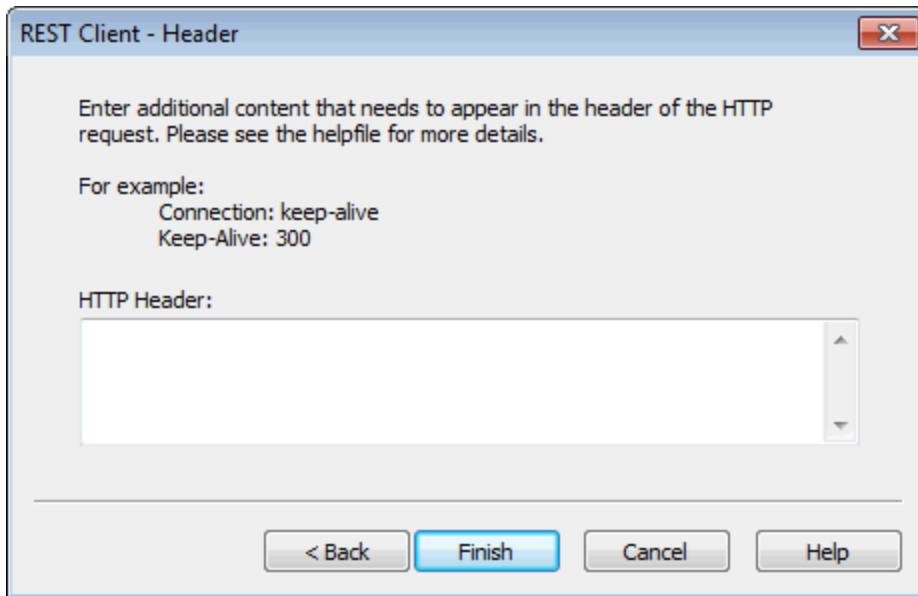
See Also:

[Adding Tags](#)
[REST Data Header](#)
[REST Data Body](#)
[REST Security](#)
[Licensing](#)

REST Client Header

Once the REST Client connection is created, the data header must be defined.

1. In the HTTP Header field, add name-value pairs to be sent to the REST server endpoint. This information is static and is sent with each connection to the endpoint.



2. Click **Finish**.
3. A REST client agent has been added. Once [tags are added](#) to this client, it begins publishing to the endpoint as long as the agent is enabled.

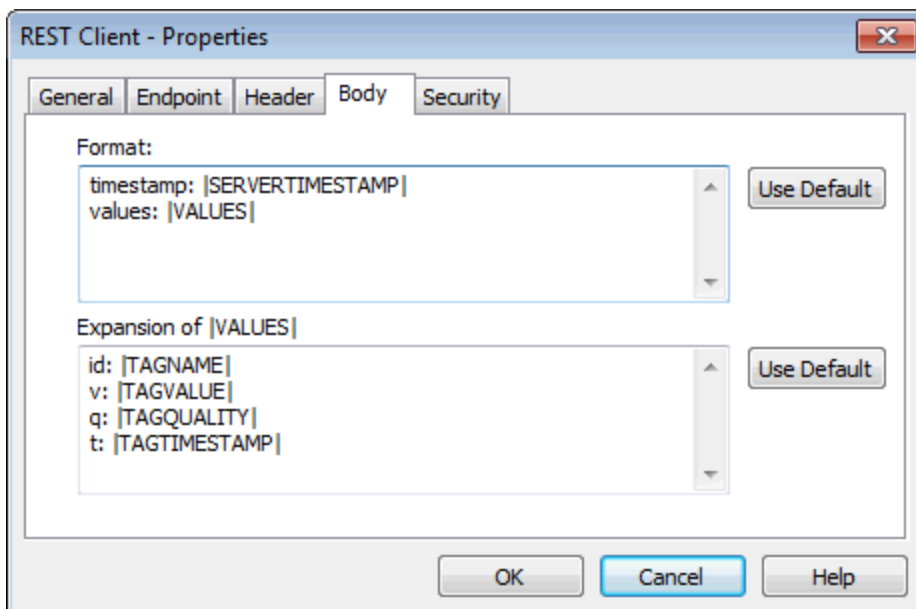
See Also:

[Licensing](#)

REST Client Body

To change the order of the JSON data load or to remove data items, follow these steps:

1. Double-click on the agent name or right-click on the agent and select **Properties** to open the properties dialog.
2. From the Properties dialog, select the **Body** tab.
3. Make any desired changes (according the guidelines below) and click **OK**.



Format: The format box contains any data to be sent before or after the JSON payload. This data is sent once per publish. Each variable may be preceded by any word or name, followed by a colon, then the variable. Key value pairs of static text may be included. Valid variables are:

|SERVERTIMESTAMP| the time and date when the gateway published the data to the endpoint, in UNIX or POSIX time format

|SERVERDATE| the same date and time as the timestamp, but in human-readable form

|VALUES| the combined payload of the following box

Click the **Use Default** button to reset the contents to the data and format set with the product before any changes are made.

Expansion of |VALUES|: This box contains the format for the JSON payload delivered to the endpoint. These values may be re-ordered or removed. A single publish event to the endpoint may include multiple instances of the data in this box. For example, if a tag had four data changes between publish events, there would be four complete JSON strings for that tag within this array. As with the Format box, each variable may be preceded by any desired word or descriptor. A colon is needed to separate the name from the variable. Valid variables are:

|TAGNAME| The name of the selected tag

|TAGVALUE| The value of the tag

|TAGQUALITY| This denotes if the tag was read as good or bad

|TAGTIMESTAMP| This is the time when the TAGVALUE was received

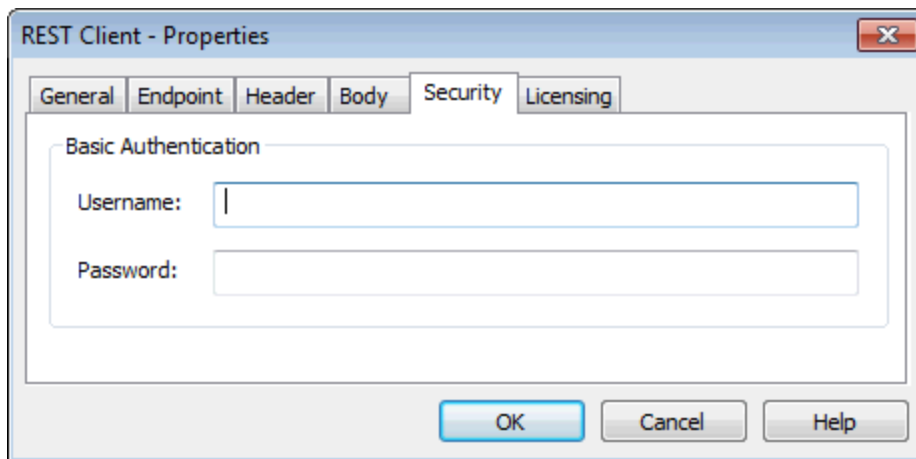
Click the **Use Default** button to reset the contents to the data and format set with the product before any changes are made.

Note: Each expression in this box needs to include a name-value pair.

REST Client Security

If the REST client endpoint needs basic authentication, follow these steps:

1. Double-click on the agent name or right-click on the agent and select **Properties** to open the properties dialog.
2. From the Properties dialog, select the **Security** tab.



Username: enter a valid user identity string for basic HTTP authentication.

Password: enter the password associated with the specified username.

3. Make the appropriate changes and click **OK**.

See Also:

[Adding Tags](#)

[REST Data Body](#)

[Working with a REST Server](#)

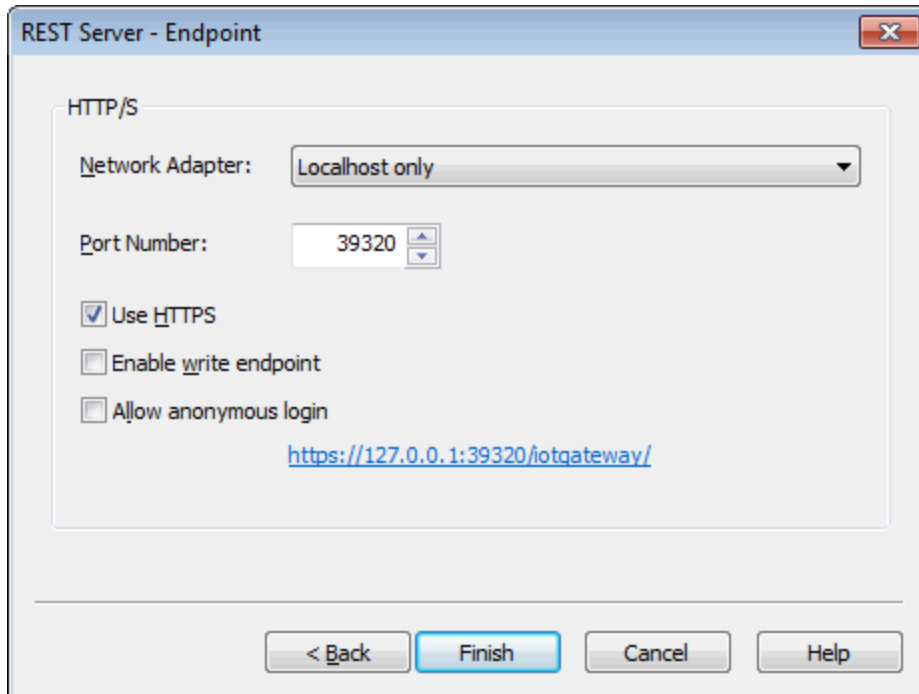
[Licensing](#)

Please also consult the documentation for KEPServerEX, User Manager, and Security Policies Plug-in

Configuring a REST Server Connection

Once the [agent type](#) is selected as a REST Server, follow the steps below to create a new REST Server connection

1. In the Endpoint dialog, set the network adapter, the port, the endpoint, and security.



- **Network Adapter:** This sets the Ethernet connection where the REST server will respond. The default, **Localhost only**, has the server respond on either localhost or 127.0.0.1 and is only accessible from the computer where KEPServerEX is installed. The drop-down list includes the network cards configured for the local computer. Select a specific card for the REST server if there is more than one. For the REST server to respond on all network connections, select **Default**.
- **Port Number:** This is the port to which the REST server binds. If there are multiple REST server agents configured on the same network adapter, they each need a different port number.
- **Use HTTPS:** This function encrypts the data between the remote client and this REST server. Upon installation, a self-signed certificate is created to allow this functionality. If you would like to use your own certificate you may import a PFX file in the IoT Gateway section of the Server Administrator settings. This is enable by default.
- **Enable write endpoint:** This allows or prevents the ability to write to any tags, regardless of the logged-in user's access level. When enabled, tags that are designated as read/write tags may be written based on user access level. If anonymous login is allowed, all accessing users may write to read/write tags. If anonymous login is not allowed, user credentials are respected regarding write permissions (based on the User Manager and or the Security Policy Plug-in). This options is disabled / unchecked by default.
- **Allow anonymous login:** By default, any client connection must have authentication credentials in the header that match a valid account in the User Manager or Security Policy Plug-in. If this option is enabled / checked, no look up for access is performed and connections are allow unauthenticated access. The User Manager and Security Policies Plug-in are both accessed from the server Administrator settings. This options is disabled / unchecked by default.
- The URL at the bottom of the dialog accesses the REST server. It is dynamic and changes as settings in this window change.

Note: The live URL link shows the address once changes are applied. Clicking on the URL before changes are applied may result in a failure to load the page.

2. Once the settings are configured, click **Finish**.
3. Typically, [Adding Tags](#) is the next operation.
4. Begin [working with the REST server](#) by using standard HTTP browser requests, Browse and Read REST commands, formatted as:

`http://localhost:39320/iotgateway/browse`

`http://localhost:39320/iotgateway/read?Tags=<TagName>`

See Also:

[Adding Tags](#)

[Working with a REST Server](#)

[Licensing](#)

Please also consult the documentation for KEPServerEX, User Manager, and Security Policies Plug-in.

Working with a REST Server

Once a REST server agent is created; a client may connect to the endpoint to browse, read, and write tags configured under that agent.

All REST server agent connections are checked against the User Manager to validate credentials unless the **Allow anonymous access** option is enabled on the agent.

For information about setting up the User Manager, refer to the KEPServerEX help guide. A quick review of the available commands may be found by using a web browser to navigate to the endpoint.

If the agent is configured with all the default selections, the link is `https://localhost:39320/iotgateway/`.

If the agent is not configured with the default selections, determine the specific link by opening the properties of the REST server agent and clicking on the **Endpoint** tab.

The IoT Gateway plug-in supports more than one REST server agent as long as they use different ports.

The REST server supports the following commands:

- Browse
- Read
- Write

The following GET commands may be tested in most web browsers.

Please note that current versions of Internet Explorer will no longer parse JSON in the browser and prompts to download it.

A Browse command will list all tags that are configured under this REST Server instance in a JSON array format. The format of the command is:

<https://localhost:39320/iotgateway/browse>

A Read command will return the tag or tags requested in a JSON array format. The format of the command is:

<https://localhost:39320/iotgateway/read?ids=Channel1.Device1.Tag1>

For multiple reads repeat the tags listing separated by &:

<https://localhost:39320/iotgateway/read?ids=Channel1.Device1.Tag1&ids=Channel2.Device2.Tag2>

The following are POST commands and will require the use of a more specialized client.

A Read post command will return the tag or tags requested in a JSON array format. The format of the command is:

<https://localhost:39320/iotgateway/read>

With the body of the POST containing the desired tags in the following format:

```
["Channel1.Device1.Tag1"]
```

Or for multiple tags:

```
["Channel1.Device1.Tag1","Channel2.Device2.Tag2","Channel3.Device3.Tag3"]
```

A write command allows a client to write to one or more tags that are configured under the client. To be able to write to the tag, the agent must be configured to allow writes, and the tag must also be writable. The format of the command is:

```
https://localhost:39320/iotgateway/write
```

With the body of the POST containing the desired tag or tags and values in the following format:

```
[{"id": "Channel1.Device1.Tag1","v": "123"}]
```

For multiple tags:

```
[{"id": "Channel1.Device1.Tag1","v": "123"}, {"id": "Channel2.Device2.Tag2","v": "456"}, {"id": "Channel3.Device3.Tag3","v": "789"}]
```

For the body, "id": is the tag name and "v": is the value write.

Tip: Directing a browser to <https://localhost:39320/iotgateway/> or <http://localhost:39320/iotgateway/>, if you have disabled HTTPS, will provide you a brief description of these commands.

Notes:

1. The port and Tag names in the above examples need to match those configured in the REST server settings.
2. Computer names with underscores are not seen as valid endpoints and result in HTTP 500 errors.

Changing an Agent Configuration

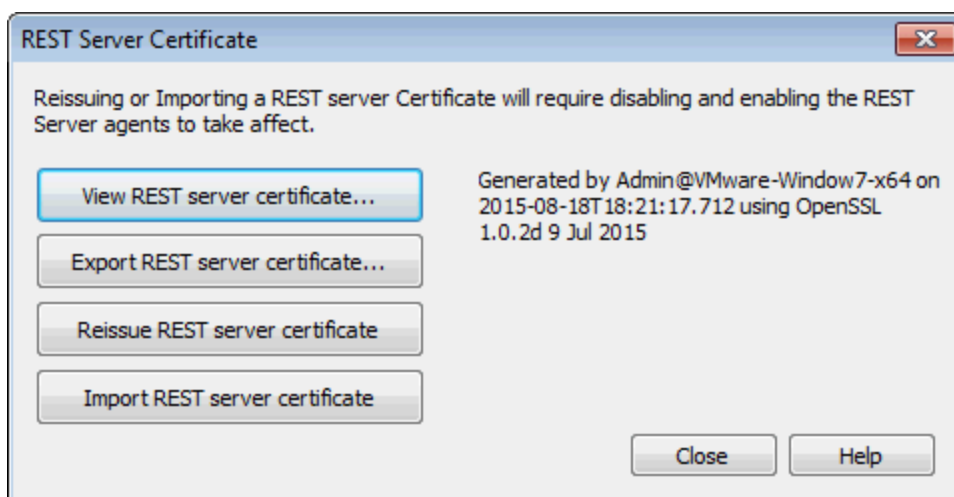
Agent settings can be updated after configuration. To access the settings, double-click the agent name or right-click on the agent and select Properties. Changes take effect immediately once you click **OK**. This causes the gateway to reload the agent configuration.

Notes:

1. If there are any events in the agent's buffer when a property change is made, those events are not lost; they are pushed to the updated configuration.
2. Disabling an agent causes its buffer to be dropped.

Configuring a Gateway Certificate

Through the **Administration | Settings...** menu; a certificate for the gateway can be viewed, exported, imported, or re-issued.



View REST server certificate... This allows you to view the details of the current certificate.

Export REST server certificate... Use this button to save the current certificate in a .DER format for importing into third-party REST clients.

Reissue REST server certificate This creates a new certificate, replacing the current certificate.

Import REST server certificate Use this button to import a certificate in .PFX format, only necessary using a custom-created certificate.

Note: When reissuing or importing a certificate, the new certificate does not take effect until the REST server endpoint(s) are stopped and restarted by disabling and re-enabling them or by reinitializing the server runtime.

See Also:

[Configuring the Gateway](#)

Configuring a Self-Signed Certificate

The IoT Gateway Plug-in supports the use of self-signed certificates with the MQTT and REST clients. These agents use the Microsoft Windows, computer-level, trusted certificate store to keep track of these certificates. By using this store, most recognized certificate authorities are already approved. To import a certificate, use the below instructions.

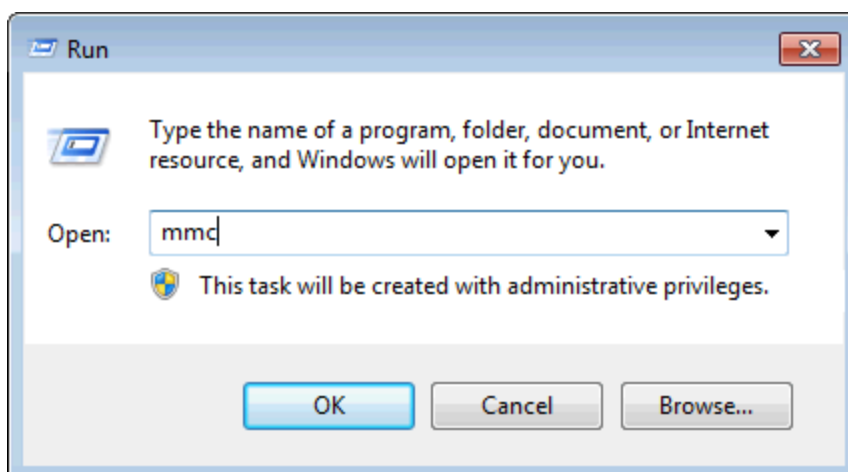
Note: If is necessary to log in to the computer with an account that is part of the Local Administrators user group to add certificates to the appropriate Windows certificate store.

Command Line Steps

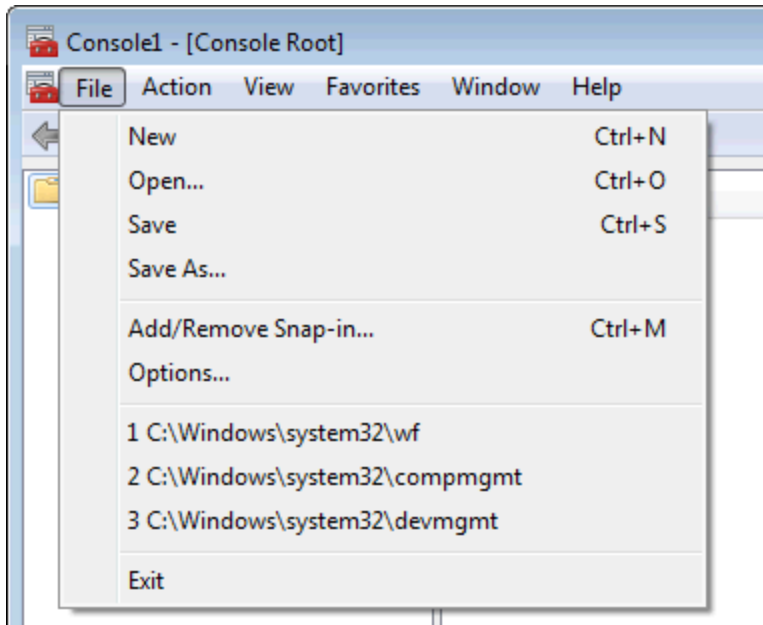
1. From the **Start** menu, select **All Programs**.
2. Choose **Accessories** then right-click on **Command Prompt** and select **Run as Administrator** from the menu.
3. In the command prompt window, navigate to the location of the certificate.
4. Enter the command: `certutil -addstore "Root" <CertificateName>` where the <CertificateName> is the name of the .cer or .crt file.
5. Press Enter to execute the command.
6. Verify the import is complete when several lines of output appear ending with: `CertUtil: -addstore command completed successfully.`

Windows Console Steps

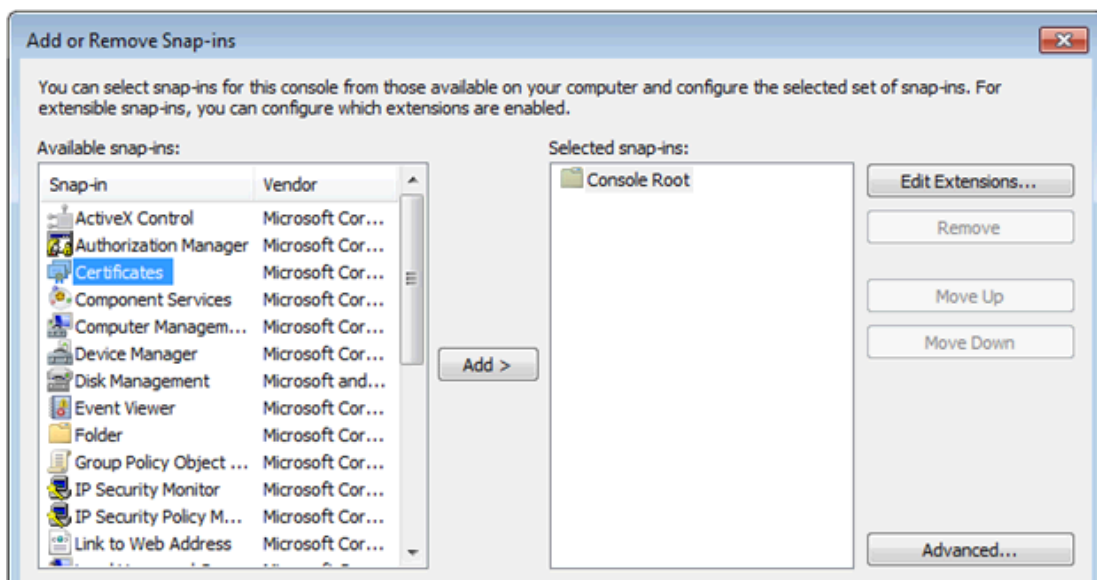
1. From the **Start** menu, select **All Programs**.
2. Choose **Accessories** | **Run**.
3. In the Run box, type "mmc" and click **OK**.



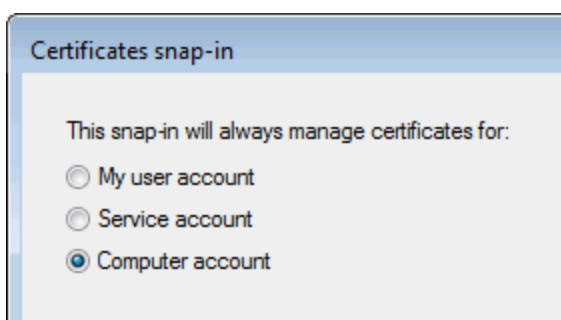
4. In the console window, choose **File | Add/Remove Snap-in...**



5. Select **Certificates** on the left.

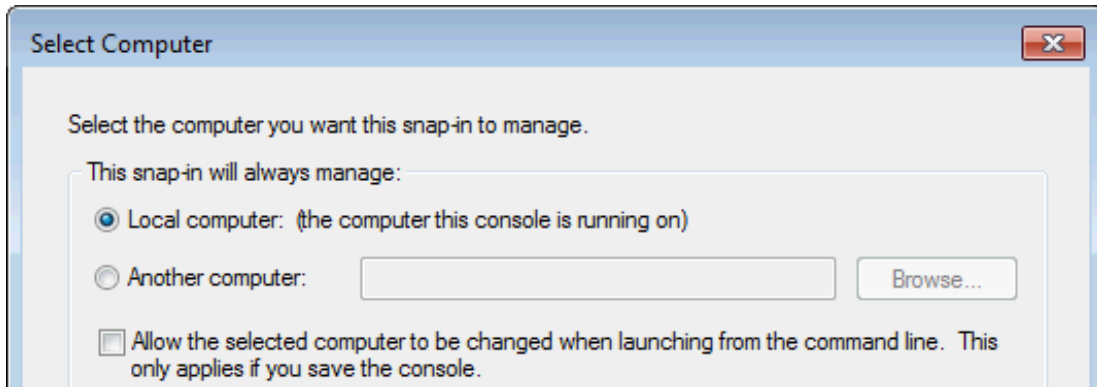


6. Click the **Add** button.

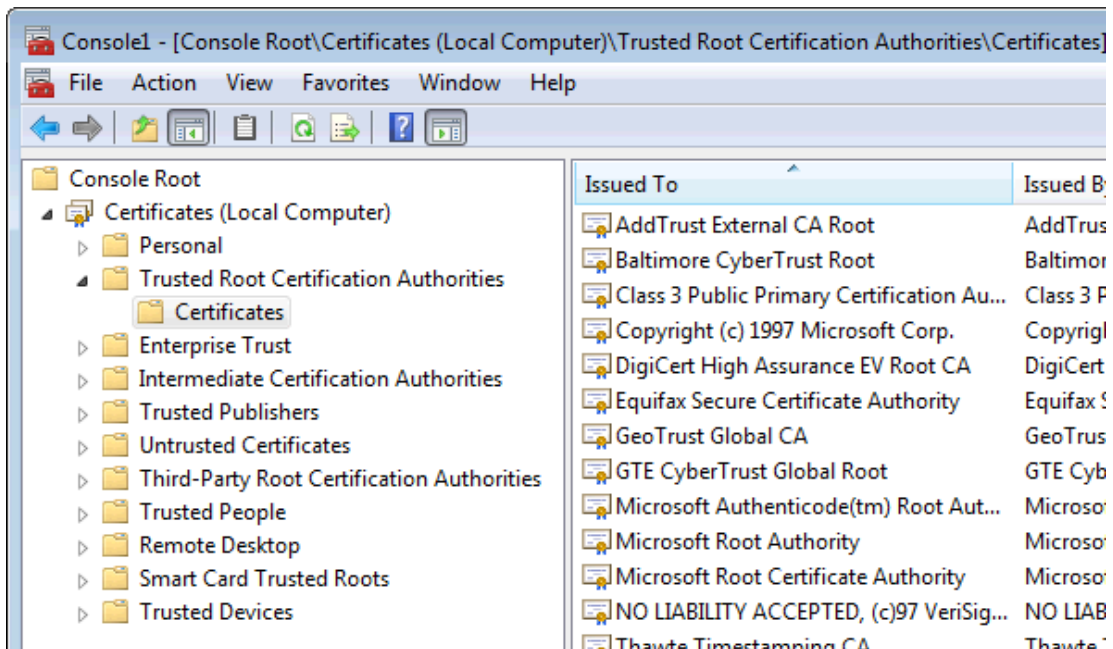


7. Select **Computer account** and then click **Next >**.

8. Select **Local Computer** and click **Finish**.

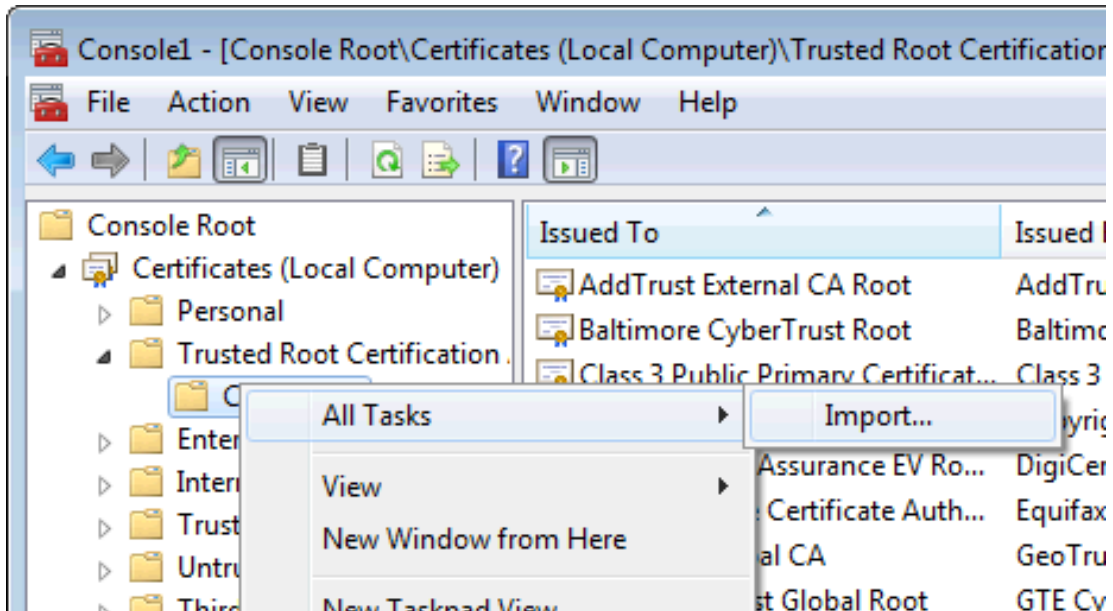


9. Back in the Add or Remove Snap-ins window, click **OK**.
10. Verify there is a Certificates (Local Computer) listing in the Console window.

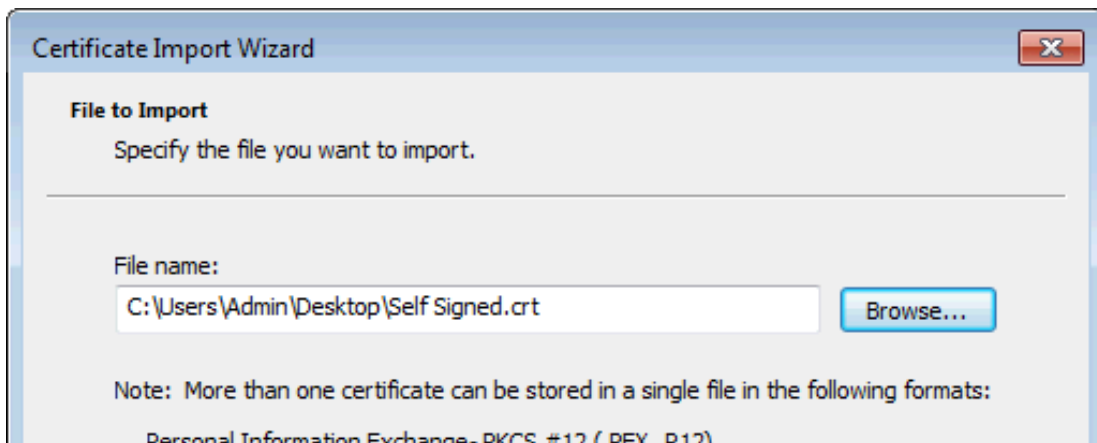


11. Expand the Certificates listing, then expand Trusted Root Certification Authorities.
12. Click on Certificates and verify a listing of all the Root certificates appears.
13. Locate the self-signed certificate to import for the MQTT and REST client.

- Right-click on Certificates and select **All Tasks | Import...**

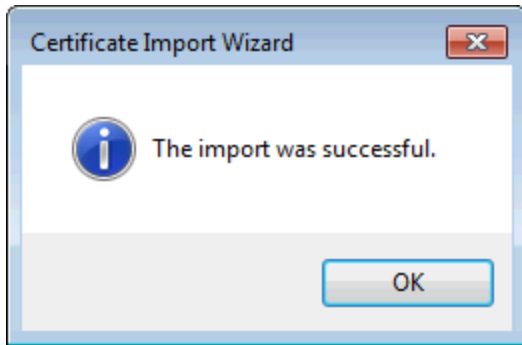


- In the Certificate Import wizard, click **Next** on the initial page.
- Click **Browse...** to locate and select the certificate to import (most client certificates are in a .cer or .crt format).



- Click **Next >**.
- Verify that **Place all Certificates in the following store** is selected and that store is the Trusted Root Certification Authorities.
- Click **Next >**.
- On the final page of the wizard, click **Finish**.

21. In the a pop-up message that the import was successful, click **OK** i



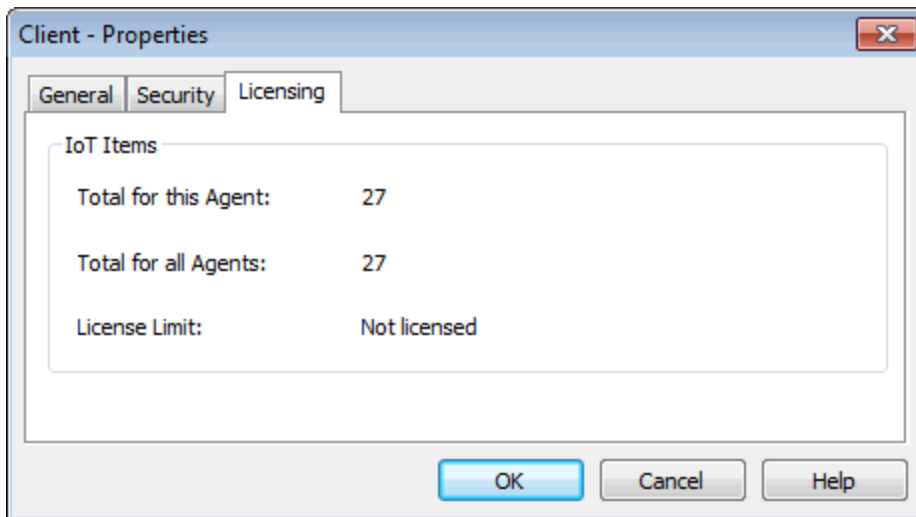
22. Close the Console window.

Licensing

The IoT Gateway Plug-in uses a tiered, count-based licensing model. A license may be purchased that enables the product to run for an unlimited amount time for a fixed maximum number of tags. The license limit does not prevent the addition of new tags beyond the tag count, nor does it signal the product to enter Demo Mode, but it does prevent updates from any tags added beyond that count.

The number of tags licensed is shown on the Licensing tab under the properties of each agent. There is a listing for the number of tags for that agent as well as the total number of tags licensed in the IoT Gateway as well as the license limit.

The licensing tab lists the total number of tags configured under this agent as well as the total number configured under all the agents and the license level. Each agent has a licensing tab under properties. In this example, the client is the only agent configured and it has 27 tags configured in the tag window.



Exceeding the Limit

An event log message is posted each time a new tag is created in excess of the license limit. If the license limit is exceeded, existing tags can be deleted to make license counts available for the new ones.

Notes:

1. When the license limit is exceeded, IoT Gateway processes the licensed number of tags only; the exact tags can vary based on the project loading order. The Event Log messages indicate the exact tags that did not load.
2. For licensing, tags are counted on a per-agent, not per-tag basis. For example; if the same tag is added to both a REST client and an MQTT client, it counts as two tags against the license.

Unlicensed Operation

When no license is installed, the entire product enters Demo Mode and runs for two hours before the runtime service must be stopped and restarted, which is accomplished through the administration icon found in the system tray.

Data

The IoT Gateway pushes data in a standard JSON format via the REST and MQTT Clients. This format may then be consumed by the third-party endpoint and broken down in an appropriate way. The data structure for these agents looks like the following sample by default:

[Data Format](#)
[Adding Tags](#)
[CSV Import and Export](#)
[System Tags](#)

Data Format

The IoT Gateway pushes data in a standard JSON format via the REST and MQTT Clients. This format may then be consumed by the third party endpoint and broken down in an appropriate way. The data structure for these agents looks like the following sample by default:

```
{ "timestamp": 1438011255230,
  "values":
  [
    { "id": "Channel1.Device1.Tag1", "v": "250", "q": true, "t": 1438011254668 }
  ]
}
```

where:

id = The unique name of the tag in KEPServerEX

v = The value of the tag as a string

q = True means good quality update, false means bad (i.e. lost communications to the underlying device or invalid configuration)

t = The time the tag data was sampled in Unix or POSIX time format

The REST server uses a similar format with additional identifiers as detailed below.

For a Browse request, a JSON list of the IoT item names available with a "succeeded" and "reason" field is returned. The reason field remains empty if succeeded is true.

```
{ "browseResults":
  [
    { "id": "Channel1.Device1.Tag1" }
  ],
  "succeeded": true, "reason": ""
}
```

For a Read request, a JSON list of the IoT item names and values with a "s" for success and "r" for reason is returned, such as:

```
{ "readResults":
  [
    { "id": "Channel1.Device1.Tag1", "s": true, "r": "",
      "v": 4878, "t": 1444307548259 }
  ]
}
```

For a Write request, a JSON list of the IoT items with a "s" for success and "r" for reason is returned, such as:

```
{ "writeResults":
  [
    { "id": "Channel1.Device1.Tag1", "s": true, "r": "" }
  ]
}
```

Adding Tags to an Agent

Once you have an agent configured you may add tags to it. Follow the steps below to add [single](#) or [multiple](#) tags.

Adding a Single Tag

1. In the configuration window, select the agent to which to add the tag.
2. Right-click in the upper right pane and select **Add IoT Item**.

Server Tag: Enter the full channel.device.name of the tag or browse to locate the single tag.

Scan Rate: the frequency, in milliseconds, at which the tag is checked for updates in value.

Only on Data Changes: sets the agent to only publish data for this tag when the value changes.

Deadband: the percentage of value change that defines the threshold of change to trigger publication. Change is based on the full range of the tag data type. A deadband of 0 means all data changes are published.

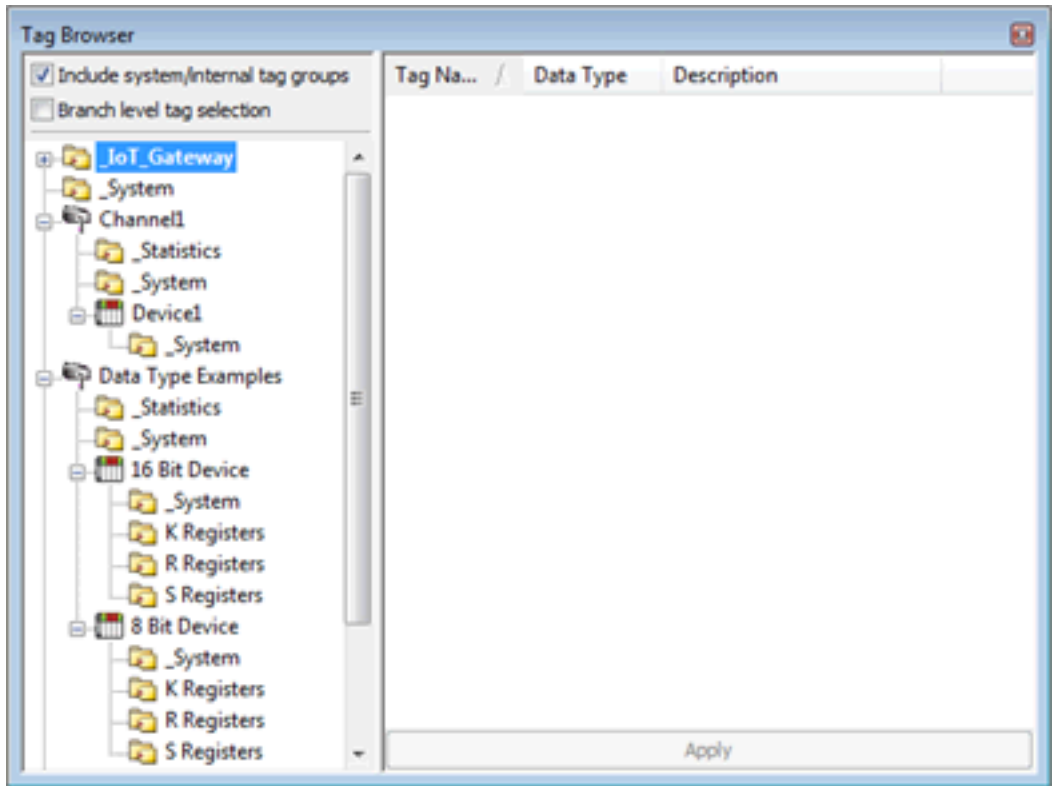
Every Scan: This forces the agent to publish data for this tag to the endpoint even if there was no change in the tag value.

Enabled: allows or prevents data for this tag to be published. Tags that are not enabled still count against the license count.

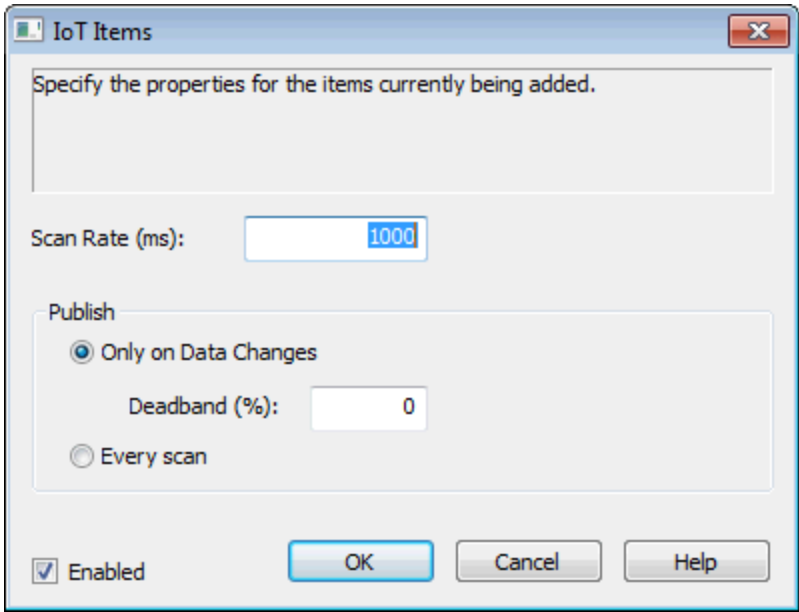
3. Click **OK**.

Adding Multiple Tags

1. In the configuration window, select the agent to which to add tags.
2. Right-click in the upper right pane and select **New IoT Items** or click on the **New IoT Items** button in the tool bar.
3. In the Tag Browser, select the tags to publish by this agent. These tags are only available on this particular agent; not on any others configured.



4. Once the tags are selected, click the **Apply** button.
5. In the IoT Items dialog, define the properties for the tags being added:



Scan Rate: the frequency at which the tag(s) are checked for updates

Publish **Only on Data Changes:** Limits the data published to value changes.

Deadband (%) the percentage of value change that defines the threshold of change to trigger publication. Change is based on the full range of the tag data type. A deadband of 0 means all data changes are published.

Publish **Every scan**: forces the agent to publish data from this tag to the endpoint even if there was no change in the tag value.

Enabled: allows or prevents the tag(s) to be monitored, collected, and published. Tags are enabled by default.

Note: Tags that are not enabled still consume a license count.

6. Once configured, click **OK**.
7. Verify the tags are listed in the upper right pane of the configuration window and in the target agent or broker. (Consult the Event Log for errors if no data appears.)

Note: Tags configured to publish on one agent are only published there unless they are also added separately to other agents.

System Tags

The IoT Gateway Plug-in exposes some status information through IoT Gateway system tags. These tags update at a five-second interval. They may be reset to zero by writing any value to them. When configured as a tag under an IoT Gateway agent, system tags count against the overall licensed number of tags. The following list contains the system tags and a brief description:

_DroppedEvents: the number of tag updates that were not successfully published due to the agent buffer being full. This can occur if the configured endpoint is not up or responding

_PublishesSent: the total number of data push events an agent has successfully made on an endpoint. Each successful publish may encompass a single or many tag updates.

Importing / Exporting CSV Files

Configuring tags for each agent may be done through the import of a comma delineated file. This is done on a per agent basis. Exporting the list of tags of an already configured agent may also be done this way. To import or export a list of tags, right click on the agent you wish to work with and select either "Import CSV..." or "Export CSV...". Selecting "Export CSV..." will bring up a dialog box asking what to name and where to save the CSV file. Once saved you may open and edit the CSV file as you wish. This file may then be re-imported to this agent or any agent as long as the formatting is maintained. To start with an appropriately formatted file, it is recommended that a single tag is added to an agent. Once added export the CSV file for that agent for use as a template. This will provide you with the format needed to add any additional tags that are needed.

Note: Importing a CSV file removes existing tags under the agent and adds what is in the CSV file. To add to the existing tags, use the export option first, add the new tags to that CSV file, and then Import that complete CSV file.

Troubleshooting

Event Log

The KEPServerEX Event log will provide information pertaining to each of your agent connections and the status of the gateway service.

Please refer to it and the [message list](#) to resolve issues.

Data Loss

While rare under normal circumstances, errors reported in the event log can indicate dropped data. This can be due to one of the following conditions:

- Continuous incoming data rate is too fast (>100,000 updated tags per second).
- Unable to communicate with the third party endpoint and the gateway buffer is full.

The gateway buffer is configured on a per agent basis. Once the buffer maximum has been reached the gateway will drop new data coming in to the gateway for that agent. Older data is retained in the buffer until it may be pushed to the third-party endpoint.

Note: If there are any tags in an agents' buffer when a change is made to its configuration, the tags are lost.

See Also:

[Event Log Messages](#)

Event Log Messages

The following messages may be generated. Click on the link for a description of the message.

[Browse rejected: no user credentials were provided in the request and anonymous requests are currently disabled.](#)

[Browse rejected: the credentials for user <user> are invalid.](#)

[Connection restored to server: <gateway>. Reinitializing server configuration.](#)

[Data change event buffer overrun; dropping updates. Ensure that the IoT Gateway service is running or reduce the volume of data collected.](#)

[Error adding item <tag>. This item already exists in connection <agent>.](#)

[Error adding item <tag> to connection <agent>.](#)

[Error importing CSV data. Invalid CSV header.](#)

[Error importing CSV data. No item records found in CSV file.](#)

[Error importing CSV item record <tag>. Update rate <update rate> is out of range, setting to <valid update>.](#)

[Error importing CSV item record <tag>. No update rate found, setting to <update rate>.](#)

[Error importing CSV item record <tag>. Deadband <deadband rate is invalid. Deadband set to <valid deadband>.](#)

[Error importing CSV item record <tag>. No deadband value found, setting to <valid deadband>.](#)

[Failed to connect to server: <gateway>. Please verify this connection information is correct and that the host can be reached.](#)

[Failed to start IoT Gateway service.](#)

[Failed to start IoT Gateway service. Please ensure arguments <Java variables> are valid.](#)

[Failed to connect to server: <URL and port>. Please verify this connection information is correct and that the host can be reached.](#)

[Failed to create JVM using JRE at <path to JRE>.](#)

[Failed to import server instance cert: <agent>. Please use the Administration utility to re-issue the certificate.](#)

[Failed to initialize the JVM: insufficient memory available \(requested initial=<MB>, max. =<MB>\).](#)

[Failed to initialize the JVM: JNI error <Error>.](#)

[Failed to initialize the IoT Gateway.](#)

[Failed to launch IoT Gateway: no suitable 32-bit JRE was configured or found.](#)

[Failed to load XML project. Item <tag> already exists in connection <agent>.](#)

[Failed to load project: <agent URL> is not a valid address.](#)

[Failed to load agent <agent>: invalid payload specification.](#)

[IoT Gateway failed to start. Failed to bind to port <port>.](#)

[IoT Gateway using JRE at <path to JRE>.](#)

Item <tag> on connection <agent> is now licensed and sending data.

Missing server instance certificate <agent>. Re-issue the certificate using the Administration utility.

MQTT agent <agent name> disconnected – reason: Connection lost.

MQTT agent <agent name> dropped data change events.

MQTT agent <agent name> failed to parse payload.

MQTT agent <agent name> failed to publish - reason: <Broker URL>.

MQTT agent <agent name> failed to publish - reason: Connection reset.

MQTT agent <agent name> failed to publish - reason: Unable to connect to server.

MQTT agent <agent name> is connected to broker <broker URL>.

Read rejected for item <tag>: no user credentials were provided in the request and anonymous requests are currently disabled.

Read rejected for item <tag>: the credentials for user <user> are invalid.

Read rejected for item <tag>. No user credentials were provided in the request and anonymous requests are currently disabled.

REST client <agent name> dropped data change events.

REST client <agent name> failed to parse payload.

REST client <agent name> processing update.

REST client <agent name> publish failed - reason: Connection refused: connect.

REST client <agent name> publish failed - reason: Read timed out.

REST client <agent name> publish failed - reason: SSL configuration error.

REST client <agent name> publish failed - reason: Unexpected EOF.

REST client <agent name> returned HTTP error <HTTP error>, buffering records.

REST client <agent name> started publishing to <REST server URL>.

REST server <agent name> started at <URL and port>.

REST server <agent name> - failed to start on <URL and port>, reason: Address already in use: bind.

Running with Java <full Java version>.

The REST server certificate has been reissued.

The REST server certificate has been imported.

The REST server certificate has expired. Please use the Administration utility to re-issue the certificate.

Unable to send data for item <tag> on connection <agent>. The licensed item count of <license count> items has been reached.

Unable to start secure REST server <agent name> at <URL and port>: missing or invalid certificate.

Unable to use network adapter <network adapter> for REST server <agent>. Binding to localhost only.

Unsupported JVM: please install or configure a 32-bit Java 1.6 or higher JRE or JDK.

Write request failed on item <tag>. The write data type <data type> cannot be converted to the tag data type <data type>.

Write rejected for item <tag>: no user credentials were provided in the request and anonymous requests are currently disabled.

Write rejected for item <tag>: the credentials for user <user> are invalid.

Browse rejected: no user credentials were provided in the request and anonymous requests are currently disabled.

Message Type:

Security

Possible Cause:

Anonymous access is disabled, but no credentials were sent with the request.

Solution:

Enable anonymous access on the REST Server agent or enter a valid username and password.

Browse rejected: the credentials for user <user> are invalid.

Message Type:

Security

Possible Cause:

The credentials sent with the request are invalid or do not have browse permissions. Anonymous access is disabled, but invalid credentials were sent with the request.

Solution:

Verify the username and password are correct and have adequate rights before trying the request again.

Connection restored to server: <gateway>. Reinitializing server configuration.**Message Type:**

Informational

Possible Cause:

This message is logged when the plug-in reconnects to the Gateway service, such as when there is a java change or the runtime is re-initialized.

Data change event buffer overrun; dropping updates. Ensure that the IoT Gateway service is running or reduce the volume of data collected.**Message Type:**

Warning

Possible Cause:

The plug-in is unable to communicate with the Gateway service and has started to lose data.

Solution:

1. Verify the Gateway service is running.
2. Verify the gateway is not disabled in the Windows Services.
3. Verify the configured gateway port is not in use by another service.

Error adding item <tag> to connection <agent>.**Message Type:**

Error

Possible Cause:

The tag or tag name is invalid.

Solution:

- Correct the name of the item or tag for which data is desired and re-try the request.
- Create a new tag with the same address and a different name.

Error adding item <tag>. This item already exists in connection <agent>.**Message Type:**

Error

Possible Cause:

A tag with this name already exists under this agent.

Solution:

- Verify the name of the item or tag for which data is desired and correct the request.
- Create a new tag with the same address, but a different name, to import it under the same agent.

Error importing CSV data. Invalid CSV header.

Message Type:

Error

Possible Cause:

The header information or format in the CSV file import is missing data or is invalid.

Solution:

- Export a new CSV file from an existing agent and use that as a template.
- Correct the header information or format according to the instructions on headers.

See Also:

[Importing / Exporting CSV Files](#)

Error importing CSV data. No item records found in CSV file.

Message Type:

Error

Possible Cause:

The CSV file was not a valid format or contained no data.

Solution:

- Export a new CSV file from an existing agent and use that as a template.
- Correct the information or format of the CSV file.

See Also:

[Importing / Exporting CSV Files](#)

Error importing CSV item record <tag>. Update rate <update rate> is out of range, setting to <valid update>.

Message Type:

Warning

Possible Cause:

Tags in the import file included invalid update rate information.

Solution:

Verify all fields have valid data in the CSV files.

See Also:

[Importing / Exporting CSV Files](#)

Error importing CSV item record <tag>. No update rate found, setting to <update rate>.

Message Type:

Warning

Possible Cause:

Tags in the import file are missing the update rate information.

Solution:

Verify all fields have valid data in the CSV files.

See Also:

[Importing / Exporting CSV Files](#)

Error importing CSV item record <tag>. Deadband <deadband rate is invalid. Deadband set to <valid deadband>.

Message Type:

Warning

Possible Cause:

Tags in the import file include invalid deadband information.

Solution:

Verify all fields have valid data in the CSV files.

See Also:

[Importing / Exporting CSV Files](#)

Error importing CSV item record <tag>. No deadband value found, setting to <valid deadband>.

Message Type:

Warning

Possible Cause:

Tags in the import file are missing deadband information.

Solution:

Verify all fields have valid data in the CSV files.

See Also:

[Importing / Exporting CSV Files](#)

Failed to connect to server: <gateway>. Please verify this connection information is correct and that the host can be reached.

Message Type:

Error

Possible Cause:

The server cannot communicate with the Gateway service.

Solution:

- Verify the Gateway service is running.
- Verify the configured gateway port is not in use by another service.

Failed to start IoT Gateway service.

Message Type:

Error

Possible Cause:

The IoT gateway service is set to Manual or there is no appropriate Java JRE installed.

Solution:

- Under Windows Services, verify that the IoT Gateway service is set to Manual.
- Verify that a valid 32-bit Java JRE version 6 or higher is installed.
- Install a new version of Java that meets the system requirements.

Failed to start IoT Gateway service. Please ensure arguments <Java variables> are valid.

Message Type:

Error

Possible Cause:

Invalid JRE arguments were added to the advanced settings.

Solution:

Remove or correct any advanced settings from **Administration | Settings | IoT Gateway | Advanced Settings**.

Failed to connect to server: <URL and port>. Please verify this connection information is correct and that the host can be reached.

Message Type:

Error

Possible Cause:

The port configured for communications between the plug-in and gateway is in use by another process.

Solution:

Change the connection port in **Administration | Settings | IoT Gateway** tab.

Failed to create JVM using JRE at <path to JRE>.

Message Type:

Error

Possible Cause:

The installed JRE is unable to create the JVM instance.

Solution:

- Re-install the latest version of Java.
- Verify that the gateway is set to use an appropriate JRE in the **Administration | Settings | IoT Gateway** tab.

Failed to import server instance cert: <agent>. Please use the Administration utility to re-issue the certificate.

Message Type:

Security

Possible Cause:

The REST server SSL certificate could not be imported.

Solution:Re-issue the certificate through **Administration | Settings | IoT Gateway | Manage Certificates**.

Failed to initialize the JVM: insufficient memory available (requested initial=<MB>, max. =<MB>).

Message Type:

Error

Possible Cause:

The computer has insufficient memory to start the JVM.

Solution:The initial and maximum memory levels in the **Administration tool | Settings | IoT Gateway | Advanced settings** should be removed.

Failed to initialize the JVM: JNI error <Error>.

Message Type:

Error

Possible Cause:

There is an issue with the Java JRE.

Solution:

- Reinstall a valid 32-bit Java JRE version 6 or higher.
- Verify that the gateway is set to use an appropriate JRE in the **Administration tool | Settings | IoT Gateway** tab.

Failed to initialize the IoT Gateway.

Message Type:

Error

Possible Cause:

The IoT Gateway is unable to start.

Solution:

Re-run the installation and choose to repair the setup.

Failed to launch IoT Gateway: no suitable 32-bit JRE was configured or found.

Message Type:

Error

Possible Cause:

A valid 32-bit Java JRE version 6 or higher was not found on the computer.

Solution:

- Re-install the latest version of Java.
- Verify that the gateway is set to use an appropriate JRE in the **Administration | Settings | IoT Gateway** tab.

Failed to load XML project. Item <tag> already exists in connection <agent>.**Message Type:**

Warning

Possible Cause:

There is a duplicate tag under an agent in the XML project.

Solution:

Remove or correct the duplicate tag in the XML before attempting to import again.

Failed to load project: <agent URL> is not a valid address.**Message Type:**

Error

Possible Cause:

The agent specified has an invalid URL format.

Solution:

Correct the URL to a valid format and try importing the file again.

Failed to load agent <agent>: invalid payload specification.**Message Type:**

Error

Possible Cause:

The JSON payload contains invalid or disallowed content.

Solution:

Adjust the Payload section of the XML to be a valid cdata section with correct name-value pairs.

IoT Gateway using JRE at <path to JRE>.**Message Type:**

Informational

Possible Cause:

This message appears when the gateway starts, indicating the JRE being used.

IoT Gateway failed to start. Failed to bind to port <port>.**Message Type:**

Error

Possible Cause:

The Gateway service was unable to use the port assigned in the Administration tool.

Solution:

Change the port in **Administration | Settings | IoT Gateway** tab to an available unused port.

Item <tag> on connection <agent> is now licensed and sending data.

Message Type:

Informational

Possible Cause:

The referenced tag was disabled due to license limitation, but is now sending data.

Missing server instance certificate <agent>. Re-issue the certificate using the Administration utility.

Message Type:

Security

Possible Cause:

The REST server SSL certificate is missing or invalid.

Solution:

Re-issue the certificate through **Administration | Settings | IoT Gateway | Manage Certificates**.

MQTT agent <agent name> disconnected – reason: Connection lost.

Message Type:

Error

Possible Cause:

The broker is unreachable.

Solution:

- Verify that the broker is online and the network connection is functioning properly.
- Check the configured URL for the agent and verify a properly configured broker exists at that address.
- Verify that communication is not being blocked by a hardware or software firewall or filter.

MQTT agent <agent name> dropped data change events.

Message Type:

Warning

Possible Cause:

The broker is unreachable.

Solution:

- Verify that the broker is online and the network connection is functioning properly.
- Check the configured URL for the agent and verify a properly configured broker exists at that address.
- Verify that communication is not being blocked by a hardware or software firewall or filter.

MQTT agent <agent name> failed to parse payload.

Message Type:

Error

Possible Cause:

The JSON payload has invalid or disallowed content in it.

Solution:

Adjust the Format and Expansion of [VALUES] boxes on the Message tab to remove the incorrect information.

MQTT agent <agent name> failed to publish - reason: <Broker URL>.**Message Type:**

Error

Possible Cause:

The broker is unreachable.

Solution:

- Verify that the broker is online and the network connection is functioning properly.
- Check the configured URL for the agent and verify a properly configured broker exists at that address.
- Verify that communication is not being blocked by a hardware or software firewall or filter.

MQTT agent <agent name> failed to publish - reason: Connection reset.**Message Type:**

Error

Possible Cause:

The agent is configured for an SSL connection and the broker does not support SSL or is not configured for SSL connections.

Solution:

- Check the URL and port that the agent is using and verify that it is an SSL-enabled endpoint.
- Change the agent URL to use TCP rather than SSL and try again.

MQTT agent <agent name> failed to publish - reason: Unable to connect to server.**Message Type:**

Error

Possible Cause:

No valid MQTT broker at the URL provided or communication is blocked.

Solution:

- Verify that the broker is online and the network connection is functioning properly.
- Check the configured URL for the agent and verify a properly configured broker exists at that address.
- Verify that communication is not being blocked by a hardware or software firewall or filter.

MQTT agent <agent name> is connected to broker <broker URL>.**Message Type:**

Informational

Possible Cause:

This message is posted once a successful connection is made with an MQTT broker.

Read rejected for item <tag>: no user credentials were provided in the request and anonymous requests are currently disabled.**Message Type:**

Security

Possible Cause:

Anonymous access is disabled or no credentials were sent from the client.

Solution:

- Enable anonymous access on the REST server agent.
- Send valid credentials in the authentication section of GET or POST.

Read rejected for item <tag>: the credentials for user <user> are invalid.**Message Type:**

Security

Possible Cause:

- The credentials sent with the request are invalid or do not have read permissions.
- Anonymous access is disabled, but invalid credentials were sent with the request.

Solution:

Verify the username and password are correct and have adequate rights before trying the request again.

Read rejected for item <tag>. No user credentials were provided in the request and anonymous requests are currently disabled.**Message Type:**

Security

Possible Cause:

Anonymous access is disabled, but no credentials were sent with the request.

Solution:

Enable anonymous access on the REST Server agent or enter a valid username and password.

REST client <agent name> dropped data change events.**Message Type:**

Warning

Possible Cause:

The REST server is unreachable.

Solution:

- Verify that the REST server endpoint is online and the network connection is functioning properly.
- Check the configured URL for the agent and verify a properly configured REST server exists at that address.
- Verify that communication is not being blocked by a hardware or software firewall or filter.

REST client <agent name> failed to parse payload.**Message Type:**

Error

Possible Cause:

The JSON payload has invalid or disallowed content in it.

Solution:

Adjust the Format and Expansion of |VALUES| boxes on the Message tab to remove the incorrect information.

REST client <agent name> processing update.

Message Type:

Informational

Possible Cause:

This message is posted when a change is made to the REST client configuration.

REST client <agent name> publish failed - reason: Connection refused: connect.

Message Type:

Error

Possible Cause:

A valid REST server endpoint has gone offline.

Solution:

- Verify that the REST server endpoint is online and the network connection is functioning properly.
- Check the configured URL for the agent and verify a properly configured REST server exists at that address.
- Verify that communication is not being blocked by a hardware or software firewall or filter.

REST client <agent name> publish failed - reason: Read timed out.

Message Type:

Error

Possible Cause:

Publishing to a HTTP endpoint with HTTPS enabled.

Solution:

Edit the agent endpoint URL to use HTTP rather than HTTPS and try again.

REST client <agent name> publish failed - reason: SSL configuration error.

Message Type:

Error

Possible Cause:

The client certificate has not been imported into the Microsoft computer-level root trusted certificate store.

Solution:

Import the proper client certificate into the certificate store. These instructions are listed in the [Configuring a Self-Signed Certificate](#) section.

REST client <agent name> publish failed - reason: Unexpected EOF.

Message Type:

Error

Possible Cause:

Publishing to a HTTPS endpoint without HTTPS enabled in the URL.

Solution:

Edit the agent endpoint URL to use HTTPS rather than HTTP and try again.

REST client <agent name> returned HTTP error <HTTP error>, buffering records.

Message Type:

Warning

Possible Cause:

The configured endpoint URL is incorrect or not accepting connections. The Gateway will buffer data until the endpoint comes online or is corrected in the configuration.

Solution:

The HTTP error returned from the endpoint should indicate how to establish a connection.

Examples:

A 404 error indicates the URL is incorrect.

A 401 error indicates the username or password is incorrect.

REST client <agent name> started publishing to <REST server URL>.

Message Type:

Informational

Possible Cause:

This message is posted once a successful publish is made with the configured REST server.

REST server <agent name> started at <URL and port>.

Message Type:

Informational

Possible Cause:

This message is posted when the REST server is activated on the gateway.

REST server <agent name> - failed to start on <URL and port>, reason: Address already in use: bind.

Message Type:

Error

Possible Cause:

An existing REST server agent is already using this port or another service on the computer is using this port.

Solution:

Edit the port setting in the REST server agent properties to an available port.

Running with Java <full Java version>.

Message Type:

Informational

Possible Cause:

This message appears when the gateway starts, indicating the full version of Java being used.

The REST server certificate has been reissued.

Message Type:

Security

Possible Cause:

A new REST server certificate has been successfully issued from the certificate manager.

The REST server certificate has been imported.

Message Type:

Security

Possible Cause:

A new REST server certificate has been successfully imported.

The REST server certificate has expired. Please use the Administration utility to re-issue the certificate.

Message Type:

Security

Possible Cause:

The current REST server SSL certificate is expired.

Solution:

Re-issue the certificate from **Administration | Settings | IoT Gateway | Manage Certificates**.

Unable to send data for item <tag> on connection <agent>. The licensed item count of <license count> items has been reached.

Message Type:

Warning

Possible Cause:

More tags are configured than the license allows.

Solution:

Remove unused tags from any configured agents or apply a license that allows for more tags.

Unable to start secure REST server <agent name> at <URL and port>: missing or invalid certificate.

Message Type:

Error

Possible Cause:

The SSL certificate is missing is or invalid.

Solution:

- Verify that certificate exists in the product path at: C:\ProgramData\Kepware\KEPServerEX\V5\IoT Gateway\RESTServer\cert
- Re-issue the REST server certificate from **Administration | Settings | IoT Gateway | Manage Certificates**.

Unable to use network adapter <network adapter> for REST server <agent>. Binding to localhost only.

Message Type:

Warning

Possible Cause:

The network adapter in the project does not match any found on the current machine.

Solution:

Use the Endpoint tab of the REST server to adjust the network adapter.

Unsupported JVM: please install or configure a 32-bit Java 1.6 or higher JRE or JDK.

Message Type:

Error

Possible Cause:

There is no valid version of Java installed for the gateway.

Solution:

Install a valid 32-bit Java JRE version 6 or higher.

Write request failed on item <tag>. The write data type <data type> cannot be converted to the tag data type <data type>.

Message Type:

Warning

Possible Cause:

The write payload was of a data type that cannot be written to the selected tag.

Solution:

Verify that the tag data type being written is correct and that the data being written matches acceptable values for that data type.

Write rejected for item <tag>: no user credentials were provided in the request and anonymous requests are currently disabled.

Message Type:

Security

Possible Cause:

Anonymous access is disabled, so credentials must be provided, but none were sent from the client.

Solution:

Enable anonymous access on the REST server agent or enter a valid username and password.

Write rejected for item <tag>: the credentials for user <user> are invalid.

Message Type:

Security

Possible Cause:

- The credentials sent with the request are invalid or do not have write permissions.
- Anonymous access is disabled, but invalid credentials were sent with the request.

Solution:

Verify the username and password are correct and have adequate rights before trying the request again.

Index

A

Add or Remove Snap-ins 24
Adding Tags to an Agent 29
Advanced Settings 10
Anonymous 19
Architectural Summary 5
Authentication 5, 13

B

Broker 12
Browse 20
Browse rejected
 no user credentials were provided in the request and anonymous requests are currently disabled. 33
 the credentials for user <user> are invalid. 33

C

Certificate 22
Certificate Import 25
Changing an Agent Configuration 21
Client ID 13
Collection rate 5
Command Line 22
Commands 20
Configuration 9
Configuring a Gateway Certificate 21
Configuring a New Agent 10
Configuring a REST Client Connection 15
Configuring a REST Server Connection 19
Configuring a Self-Signed Certificate 22
Configuring an MQTT Connection 11
Configuring the Gateway 9
Connection restored to server
 <gateway>. Reinitializing server configuration. 34
CSV file 31

D

Data 28
Data change event buffer overrun 34
Data Format 28

Data Loss 32
Data structure 28
Data Updates 7
Deadband 30
Detail View 8
DroppedEvents 31
dropping updates. Ensure that the IoT Gateway service is running or reduce the volume of data collected. 34

E

Enable Write Endpoint 19
Endpoint 15
Error adding item <tag> to connection <agent>. 34
Error adding item <tag>. This item already exists in connection <agent>. 34
Error importing CSV data. Invalid CSV header. 35
Error importing CSV data. No item records found in CSV file. 35
Error importing CSV item record <tag>. Deadband <deadband rate is invalid. Deadband set to <valid deadband>. 36
Error importing CSV item record <tag>. No deadband value found, setting to <valid deadband>. 36
Error importing CSV item record <tag>. No update rate found, setting to <update rate>. 35
Error importing CSV item record <tag>. Update rate <update rate> is out of range, setting to <valid update>. 35
Event Log 32
Event Log Messages 32
Event Log View 8
Exceeding the Limit 27
External Dependencies 6

F

Failed to connect to server
 <gateway>. Please verify this connection information is correct and that the host can be reached. 36
 <URL and port>. Please verify this connection information is correct and that the host can be reached. 37
Failed to create JVM using JRE at <path to JRE>. 37
Failed to import server instance cert
 <agent>. Please use the Administration utility to re-issue the certificate. 38
Failed to initialize the IoT Gateway. 38
Failed to initialize the JVM
 insufficient memory available (requested initial=<MB>, max. =<MB>). 38
 JNI error <Error>. 38
Failed to launch IoT Gateway
 no suitable 32-bit JRE was configured or found. 38
Failed to load agent <agent>
 invalid payload specification. 39
Failed to load project
 <agent URL> is not a valid address. 39
Failed to load XML project. Item <tag> already exists in connection <agent>. 39
Failed to start IoT Gateway service. 37
Failed to start IoT Gateway service. Please ensure arguments <Java variables> are valid. 37

G

General Operation 7

GET command 20

H

Help Contents 5

HTTP Heade 15

HTTPS 19

I

Importing / Exporting CSV Files 31

Initialization 7

IoT Gateway failed to start. Failed to bind to port <port>. 39

IoT Gateway using JRE at <path to JRE>. 39

IoT Item 29

IoT_Gateway.dll 5

Item <tag> on connection <agent> is now licensed and sending data. 40

J

Java 10

Java 8 6

JDK installation 6

JRE 10

JSON data load 12

JSON format 28

L

Licensing 27

Localhost 19

M

Manage Certificate 10

Max. items per 15

Menu 8

Method 15

Missing server instance certificate <agent>. Re-issue the certificate using the Administration utility. 40

MQTT agent 11
MQTT agent <agent name> disconnected – reason: Connection lost. 40
MQTT agent <agent name> dropped data change events. 40
MQTT agent <agent name> failed to parse payload. 40
MQTT agent <agent name> failed to publish - reason
 <Broker URL>. 41
 Connection reset. 41
 Unable to connect to server. 41
MQTT agent <agent name> is connected to broker <broker URL>. 41
MQTT Client Message 12
MQTT Client Security 13
MQTT Last Will and Testament 14

N

Network 19

O

Overview 5

P

Plain text 13
Port 9, 19
POST 15
Project View 8
Publish rate 5
Publish Rate 15
PublishesSent 31
Publishing 10
PUT 15

Q

QoS 12

R

Read 20
Read rejected for item <tag>
 no user credentials were provided in the request and anonymous requests are currently disabled. 41
 the credentials for user <user> are invalid. 42
Read rejected for item <tag>. No user credentials were provided in the request and anonymous requests are currently disabled. 42

REST client <agent name> dropped data change events. 42
REST client <agent name> failed to parse payload. 42
REST client <agent name> processing update. 43
REST client <agent name> publish failed - reason
 Read timed out. 43
 SSL configuration error. 43
 Unexpected EOF. 43
REST client <agent name> publish failed - reason: Connection refused: connect. 43
REST client <agent name> returned HTTP error <HTTP error>, buffering records. 44
REST client <agent name> started publishing to <REST server URL>. 44
REST Client Body 16
REST Client Header 15
REST Client Security 17
REST command 20
REST server <agent name> - failed to start on <URL and port>, reason: Address already in use: bind. 44
REST server <agent name> started at <URL and port>. 44
Running with Java <full Java version>. 44

S

Scan Rate 30
Security 13, 17
server_iotgateway.exe 5
SERVERDATE 12, 17
SERVERTIMESTAMP 12, 17
Shutdown 7
SSL encrypted 13
Startup 7
System Tags 31

T

TAGNAME 13, 17
TAGQUALITY 13, 17
TAGTIMESTAMP 13, 17
TAGVALUE 13, 17
TCP/IP 9
The REST server certificate has been imported. 45
The REST server certificate has been reissued. 44
The REST server certificate has expired. Please use the Administration utility to re-issue the certificate. 45
Toolbar 8, 10
Topic 12
Troubleshooting 32
Trusted 22
Trusted Root Certification Authorities 24

U

Unable to send data for item <tag> on connection <agent>. The licensed item count of <license count> items has been reached. 45

Unable to start secure REST server <agent name> at <URL and port> missing or invalid certificate. 45

Unable to use network adapter <network adapter> for REST server <agent>. Binding to localhost only. 45

Unix or POSIX time 28

Unlicensed 27

Unsupported JVM

please install or configure a 32-bit Java 1.6 or higher JRE or JDK. 46

URL 15

User Interface 8

V

VALUES 12, 17

Version 10

W

Windows Console 22

Working with a REST Server 20

Write 20

Write rejected for item <tag>

no user credentials were provided in the request and anonymous requests are currently disabled. 46
the credentials for user <user> are invalid. 46

Write request failed on item <tag>. The write data type <data type> cannot be converted to the tag data type <data type>. 46