aprotech Kaspersky IoT company

User Guide

OPC UA Client & MQTT Publisher

About OPC UA Client and MQTT Publisher

The OPC UA Client and MQTT Publisher applications (hereinafter also *applications*) are software designed to run on the Kaspersky IoT Secure Gateway 1000 cyberimmune system platform, which is based on the KasperskyOS operating system.

The full description of the applications and the instructions on how to work with them are provided below. For information about Kaspersky IoT Secure Gateway 1000, refer to the Kaspersky IoT Secure Gateway 1000 documentation.

OPC UA Client uses the OPC UA protocol to receive data from the OPC UA server residing in the internal enterprise network. MQTT Publisher forwards data received over the MQTT protocol to the MQTT broker with TLS encryption. Kaspersky IoT Secure Gateway 1000 provides secure data collection over OPC UA, data conversion from the OPC UA protocol to the MQTT protocol, and unidirectional data transfer from the OPC UA server to the MQTT broker.

The standard deployment pattern for Kaspersky IoT Secure Gateway 1000 as a unidirectional gateway (data diode) assumes the following:

1. The device is a software unidirectional gateway.
2. The internal and external network stacks are divided at the process level.
3. Data transfer between the internal and external networks is possible only through the special MessageConsumer programming interface.

   This provides unidirectional transmission of data from the internal network to information systems on the external network. The TLS protocol is used to ensure the confidentiality of information being transmitted.

   The MessageConsumer API is implemented in the following applications:

   - OPC UA Client for processing traffic from the internal network.
   - MQTT Publisher for processing traffic in the external network.
4. OPC UA Client is connected to the internal network.

For general information about the Kaspersky IoT Secure Gateway 1000 deployment scheme, refer to the Kaspersky IoT Secure Gateway 1000 documentation.

Installation and initial configuration of applications on Kaspersky IoT Secure Gateway 1000 is performed by experts of APROTECH LLC or its partners.

In this Help section

Distribution kit

The application distribution kit includes the following components:

- OPC UA Client
- MQTT Publisher
- File with information about third-party code `legal_notices.txt`

Hardware and software requirements

Application requirements

The applications work only with Kaspersky IoT Secure Gateway 1000.

The OPC UA server must be configured to receive data from the equipment and send data to the OPC UA Client. You can view the OPC UA protocol specification on the developer's website. The application supports only version 1.04 of the OPC UA protocol.

The MQTT broker must be configured to receive data from the MQTT Publisher. You can read the MQTT protocol specification on the developer's website. The application only supports version 3.1.1 of the MQTT protocol.

Requirements for configuring and troubleshooting the applications

A computer running a Windows operating system is required to configure and troubleshoot the applications.

The following applications must be installed on the computer:

- An application for editing simple text. It is recommended to use a text editor that supports highlighting of JSON syntax.
- Browser: Google™ Chrome™ version 118 or later or Mozilla™ Firefox™ version 118 or later for accessing the Kaspersky IoT Secure Gateway 1000 web interface

2. What's new

Kaspersky IoT Secure Gateway 1000 version 3.0 includes the following functions and capabilities, which are significant for the OPC UA Client and MQTT Publisher operation:

- Kaspersky IoT Secure Gateway 1000 acts as a software platform that supports edge computing. Applications are hosted on this software platform, run in an isolated environment, and managed using the platform.
- Kaspersky IoT Secure Gateway 1000 acts as a unidirectional gateway (data diode). OPC UA Client and MQTT Publisher start only when Kaspersky IoT Secure Gateway 1000 operates in the unidirectional gateway mode. Information on other operating modes is provided in the Kaspersky IoT Secure Gateway 1000 documentation.
- Applications can be managed using the web plug-in for Kaspersky Security Center 14.2 Web Console. Including such actions as:
    - Downloading and installation of applications, configuration, start, stop, and uninstallation of the applications.
    - Managing application certificates, including adding, updating, and deleting application certificates. An application certificate is a special digital signature file that ensures secure application operation in Kaspersky IoT Secure Gateway 1000.
    - Configuring data transfer routes between the applications, including creating, modifying, and deleting an application route.
- Ability to manually reconfigure Kaspersky IoT Secure Gateway 1000 using the web interface. This approach allows you to configure a restart of the applications.
- Capability to upload application operation logs using the Kaspersky IoT Secure Gateway 1000 web interface. Kaspersky IoT Secure Gateway 1000 logs events generated by installed applications and ensures the safety of these application logs when the system is restarted, turned off, or updated.
- Capability to manage the application logging level using the Kaspersky IoT Secure Gateway 1000 web interface. You can choose one of the 6 logging levels, which differ in the level of detail and performance impact.
- Ability to download applications uploaded to the Kaspersky Appicenter for Developers web portal and install them in Kaspersky IoT Secure Gateway 1000.

3. Managing apps

Configure Kaspersky IoT Secure Gateway 1000 before starting to work with applications. Before starting to work with applications, make sure that you have completed the steps described in the Kaspersky IoT Secure Gateway 1000 documentation to get started with the system. For more details, refer to the Prerequisites for working with applications on Kaspersky IoT Secure Gateway 1000 section.

Applications can be managed using the web interface or using the web plug-in for Kaspersky Security Center 14.2 Web Console. Application management scenarios differ depending on whether you use the web interface or Kaspersky Security Center 14.2 Web Console. Both scenarios are described below with references to the Kaspersky IoT Secure Gateway 1000 documentation.

The scenario of configuration of data transfer from the OPC UA server to the MQTT broker using OPC UA Client and MQTT Publisher consists of the following stages:

1. Configuring nodes for data transfer over the OPC UA protocol on the OPC UA server. You can read about the OPC UA protocol specification on the developer's website.
2. Installing OPC UA Client and MQTT Publisher on Kaspersky IoT Secure Gateway 1000.
3. Preparing an encryption key and certificates for connection via the MQTT protocol with TLS encryption.
    a. Preparing an encryption key and a file containing a certificate chain that was used to sign the MQTT broker certificate.
    b. Preparing an encryption key and a file containing a certificate chain that was used to sign the MQTT Publisher certificate. Optional for client authentication. For more details on working with certificates, see the Securing a connection over the MQTT protocol section.
4. Configuring applications using the web interface or Kaspersky Security Center. For detailed instructions on this step, see the Configuring applications section.
5. Configuring application routing for the correct data transfer from OPC UA Client to MQTT Publisher and, as a result, from the OPC UA server to the MQTT broker.

In this Help section

3.1 Prerequisites for working with applications on Kaspersky IoT Secure Gateway 1000

Before you start working with applications on Kaspersky IoT Secure Gateway 1000, make sure the following prerequisites are met. Prerequisites include:

- Installation and initial configuration of Kaspersky IoT Secure Gateway 1000.

- Preparing certificates required for the secure operation of applications in Kaspersky IoT Secure Gateway 1000, as well as for maintaining an encrypted data transfer channel from MQTT Publisher to the MQTT broker.
- Preparing additional software used for working with the applications (Kaspersky Security Center 14.2 Web Console and Kaspersky Appicenter).

Prerequisites are listed below in the recommended order:

1) Connect the Kraftway Rubezh-N device to the network and turn it on.

2) Prepare the Kraftway Rubezh-N device for installing Kaspersky IoT Secure Gateway 1000.

3) Install Kaspersky IoT Secure Gateway 1000 by selecting the unidirectional gateway as the network device type. Steps 2 and 3 are performed by the Aprotech experts.

4) Create and upload administrator certificates.

5) Configure the date and time on Kaspersky IoT Secure Gateway 1000.

6) Configure network settings for Kaspersky IoT Secure Gateway 1000.

7) Change the web server certificate to the one used in your organization.

8) Connect to the Kaspersky IoT Secure Gateway 1000 web interface. While connecting as an administrator, change your credentials: user name and password. When you change the password, the password entry form states that the password meets the requirements, even when the password does not actually meet the requirements. Make sure that the password meets the requirements yourself, not relying on the information from the password entry form.

9) Install the web plug-in for preparing Kaspersky Security Center 14.2 Web Console for interaction with Kaspersky IoT Secure Gateway 1000.

10) Add a Kaspersky IoT Secure Gateway 1000 device to managed devices of the Kaspersky Security Center 14.2 Web Console

11) Link Kaspersky IoT Secure Gateway 1000 with Kaspersky Security Center 14.2 Web Console for subsequent system management. Connecting Kaspersky IoT Secure Gateway 1000 to Kaspersky Security Center 14.2 also allows you to use Kaspersky Appicenter to download purchased applications in the future.

To work with applications on Kaspersky IoT Secure Gateway 1000, an administrator account is required. To manage applications, you need to use Kaspersky IoT Secure Gateway 1000 or the Kaspersky Security Center 14.2 Web Console. The capabilities of application management tools are presented in the table below:

Application management capabilities on Kaspersky IoT Secure Gateway 1000		
Function	Kaspersky IoT Secure Gateway 1000 web interface	Kaspersky Security Center 14.2 Web Console
Downloading and installing applications	Yes	Yes
Starting and stopping applications	See Known Limitations section	Yes
Managing application launch rules	Yes	Yes
Removing applications	See Known Limitations section	Yes
Configuring applications	Yes	Yes
Uploading application logs	Yes	No
Managing application certificates	No	Yes

Application management capabilities on Kaspersky IoT Secure Gateway 1000		
Routing applications	Yes	Yes

In addition to the differences in the set of functions, take into account that to work with the web interface, you need a computer with access to Kaspersky IoT Secure Gateway 1000 through the internal network. The management through Kaspersky Security Center 14.2 Web Console can be performed remotely.

3.2 Installing and removing applications

3.2.1 Downloading and installing applications

To download and install applications on Kaspersky IoT Secure Gateway 1000 using the web interface, follow the instructions in the corresponding section of the Kaspersky IoT Secure Gateway 1000 documentation. Follow the instructions. At step 5, click Install in the Action column next to OPC UA Client and MQTT Publisher.

To download and install applications on Kaspersky IoT Secure Gateway 1000 using Kaspersky Security Center 14.2 Web Console, follow the instructions in the corresponding section of the Kaspersky IoT Secure Gateway 1000 documentation. Follow the instructions. At step 7, select the check boxes for OPC UA Client and MQTT Publisher, and click Save at the bottom of the page.

The selected apps will be downloaded and installed in Kaspersky IoT Secure Gateway 1000. After Kaspersky IoT Secure Gateway 1000 is synchronized with Kaspersky Security Center, the Installed status will be displayed for these apps in the apps table. Information about successful or unsuccessful downloads and installations is saved in the event log.

Installed applications are not updated automatically. To update an application, first uninstall the currently installed version of the application and then install the new version. The delisted version of the application cannot be re-installed.

3.2.2 Removing apps

To uninstall the application from Kaspersky IoT Secure Gateway 1000 using the web interface, follow the instructions in the corresponding section of the Kaspersky IoT Secure Gateway 1000 documentation. Follow the instructions and in step 3 in the row of the application you want to remove, click the trash bin icon in the Delete column and confirm the deletion in the window that opens.

To uninstall the applications from Kaspersky IoT Secure Gateway 1000 using Kaspersky Security Center 14.2 Web Console, follow the instructions in the corresponding section of the Kaspersky IoT Secure Gateway 1000 documentation. Follow the instructions. At step 7, select the check box for OPC UA Client or MQTT Publisher, and click Save at the bottom of the page.

3.3 Configuring applications

The following sections contain information on how to configure OPC UA Client and MQTT Publisher.

To configure the applications via the web interface, you need information about the configuration structure of Kaspersky IoT Secure Gateway 1000. The application configuration is in the configContent line, located in the APP_CONFIGURATION object.

Note that if the application state is Started, it must be stopped and restarted to apply the new configuration settings.

In this Help section
Configuring OPC UA Client
Configuring MQTT Publisher

3.3.1 Configuring OPC UA Client

3.3.1.1 Configuring a connection over the OPC UA Client protocol

OPC UA Client receives data from the OPC UA server residing in the internal enterprise network over the OPC UA protocol, which is described by the OPC Unified Architecture specification. You can view the OPC UA protocol specification on the developer's website. The application supports only version 1.04 of the OPC UA protocol.

In this Help section
Configuring OPC UA Client using the web interface
Configuring OPC UA Client using the KSC Web Console
Description of PC UA Client settings
Special considerations when configuring OPC UA security settings

3.3.1.2 Configuring OPC UA Client using the web interface

To configure data acquisition using the OPC UA protocol:

1. Open the Kaspersky IoT Secure Gateway 1000 web interface.
2. Open the Settings section and then the Configuration tab.
3. Find the ru.aprotech.opcuaclient section in the text displayed on the tab.
4. Copy the Base64-encoded text with the application settings located in the configContent line.
5. Decode the text from Base64 to JSON (for example, using the https://www.base64decode.org/ site).
6. Copy the resulting text with the application settings into a separate file for subsequent editing.
7. Specify the OPC UA settings and their values using the JSON syntax.
8. Encode the resulting settings text back from JSON to Base64 (for example, using the https://www.base64encode.org/ site).

Before that, make sure that you comply with the JSON syntax, because the Kaspersky IoT Secure Gateway 1000 web interface does not indicate any errors in the Base64-encoded configuration. An application started with configuration errors will be stopped. A message informing that the application terminated with an error appears in the Kaspersky IoT Secure Gateway 1000 log.

9. Copy the resulting text into the configContent line in the ru.aprotech.mqttpublisher section on the Configuration tab.
10. Click the Save button.

An example of OPC UA Client settings in JSON format:

```
{
  "id": 0,
  "name": "OPC UA Client Example",
  "description": "KISG Applications Development (Example)",
  "url": "opc.tcp://192.168.1.254:4840",
  "readingCycle": 1,
  "userCredentials": null,
  "heartbeat": {
    "name": "Heartbeat",
    "timeout": 3
  },
  "nodes": [
    {
      "name": "Boolean",
      "nodeId": "ns=2;s=Boolean"
    },
    {
      "name": "Opcstress1",
      "nodeId": "ns=2;s=Opcstress1"
    },
    {
      "name": "Opcstress2",
      "nodeId": "ns=2;s=Opcstress2"
    }
  ]
}
```

Code Block 1. Example of OPC UA Client settings

To edit files in JSON format, we recommend using a text editor that supports JSON syntax highlighting. This will help avoid potential errors (for example, unbalanced braces).

3.3.1.3 Configuring OPC UA Client using the KSC Web Console

To configure OPC UA Client using Kaspersky Security Center 14.2 Web Console, follow the instructions in the corresponding section of the Kaspersky IoT Secure Gateway 1000 documentation. Follow the instructions. At step 8, specify the required application settings as stated in the Description of OPC UA Client settings. The settings are specified in the "Application configuration" text field; it is important to comply with the JSON syntax. After specifying the settings, click Save at the bottom of the settings panel, and then at the bottom of the installed applications page.

```
{
  "id": 0,
  "name": "OPC UA Client Example",
  "description": "KISG Applications Development (Example)",
  "url": "opc.tcp://192.168.1.254:4840",
  "readingCycle": 1,
  "userCredentials": null,
  "heartbeat": {
    "name": "Heartbeat",
    "timeout": 3
  },
  "nodes": [
    {
      "name": "Boolean",
      "nodeId": "ns=2;s=Boolean"
    },
    {
      "name": "Opcstress1",
      "nodeId": "ns=2;s=Opcstress1"
    },
    {
      "name": "Opcstress2",
      "nodeId": "ns=2;s=Opcstress2"
    }
  ]
}
```

Code Block 2. Example of OPC UA Client settings

3.3.1.4 Description of OPC UA Client settings

Parameters marked as required must be explicitly defined. The other parameters are optional. For optional parameters that are not included in the configuration, the default value prescribed by the OPC UA protocol may be used.

Specification defining the protocols and mechanism for data transfer in industrial networks as well as interaction between devices in these networks.

Settings used to configure OPC UA Client

Parameter name	Required parameter	Data type	Title	Possible values and notes
`name`	Yes	string	Name of the OPC UA client that receives data from the OPC UA server.	`<OPC UA client name>`. Example: `"Kaspersky IoT Secure Gateway 1000 OPC UA Client"`.
`description`	No	string	Description of the OPC UA client that receives data from the OPC UA server.	`<OPC UA client description>`. Example: `"Collect data from CNC by Kaspersky IoT Secure Gateway 1000"`.
`url`	Yes	string	OPC UA server address.	`<scheme>://<host>:<port>`. Example: `"opc.tcp://192.168.177.7:4840"`.

Parameter name	Required parameter	Data type	Title	Possible values and notes
				Port 4840 is used by default.
`readingCycle`	No	int	Application data read frequency (in seconds).	`1`. Integer no less than `0`. `0` is a special value signifying the use of the maximum frequency available to the client and server.
`userCredentials`	No	object	Section containing the account credentials of the OPC UA client on the OPC UA server.	• `{username, password}` parameter block containing the user account credentials. • `null` is indicated if you want to allow an anonymous connection of the OPC UA client to the OPC UA server. In this case, you do not need to provide values for the `username` and `password`.
`username`	No	string	Name of the user account for authorization on the OPC UA server.	`"username"`.
`password`	No	string	Password of the user account for authorization on the OPC UA server.	`password`
`heartbeat`	No	object	This parameter block is generated by the OPC UA Client. It contains the parameters for the Kaspersky IoT Secure Gateway 1000 heartbeat signal.	• `{id, name, timeout}` parameter block. • `null`. If you do not add the `heartbeat` parameter or if you enter the `null` value, heartbeat signals will not be sent.
`name`	No	string	Data node name.	`<heartbeat node name>`. Example: `"Heartbeat"`.
`timeout`	No	int	Interval (in seconds) between the generation of	`60`. An integer no less than `0` must be entered. The default value is `30`.

Parameter name	Required parameter	Data type	Title	Possible values and notes
			heartbeat signals.	
`nodes`	Yes	array	Parameter block for data nodes.	`{name, nodeId}` parameter block. Completed for each data node.
`name`	Yes	string	Source connection point name. Used for configuring application routing.	`<node name>`. Example: `"Temperature"`. The value of each name parameter in the nodes sections in the OPC UA Client configuration must be unique.
`nodeId`	Yes	string	Data node ID.	`<namespace>, <nodeID>`.
`ns`	Yes	string	ID of the OPC UA server namespace.	`"namespace".`
`nodeId`	Yes	string	ID of the data node in the OPC UA server namespace.	`<nodeID>`. Two types of IDs are possible: • `s` (string) is a string value for the data node ID. For example, `"nodeId": "ns=1;s=Variable temperature"`. • `i` (numeric) is a numerical value for the data node ID. For example, `"nodeId": "ns=2;i=2045"`.

3.3.1.5 Special considerations when configuring OPC UA security settings

The current OPC UA Client version does not support a secure connection via the OPC UA protocol.

3.3.2 Configuring MQTT Publisher

MQTT Publisher sends data to the MQTT broker via the MQTT protocol. You can read the MQTT protocol specification on the developer's website. MQTT Publisher supports only version 3.1.1 of the MQTT protocol.

In this Help section

3.3.2.1 Securing a connection over the MQTT protocol

To transfer data using MQTT Publisher with TLS encryption, upload the following files to Kaspersky IoT Secure Gateway 1000. The list of files is presented below; some options are marked as optional:

1. File containing the certificate chain used to digitally sign the MQTT broker certificate.
2. File containing the certificate chain used to digitally sign the MQTT Publisher client certificate. Optional for client authentication.
3. File for the MQTT Publisher private encryption key. Optional for client authentication.

The certificate files must also be uploaded to the MQTT broker server:

1. File containing the certificate chain used to digitally sign the MQTT broker certificate.
2. File for the MQTT broker private encryption key.
3. File containing the certificate chain that authenticates the MQTT Publisher client certificate. Optional for client authentication.

A certificate chain can consist of a single self-signed certificate.

For more details on working with application certificates, refer to the Kaspersky IoT Secure Gateway 1000 documentation. Certificates and cryptographic keys used by MQTT Publisher must be in CRT, CER, DER, or PEM format. The key length of the application certificate must be at least 2048 bits.

Note that when an MQTT broker certificate is revoked, you will need to obtain a new certificate from the MQTT broker administrator and replace the revoked certificate in the MQTT broker. If you do not do this, Kaspersky IoT Secure Gateway 1000 will trust both the revoked certificate and the new certificate until the revoked certificate expires. This could lead to a situation in which a connection established over a secure channel is not actually secure.

Each time the MQTT broker certificate is reissued, the full certificate chain, which includes the MQTT broker leaf certificate, must be updated in Kaspersky IoT Secure Gateway 1000.

3.3.2.2 Configuring MQTT Publisher using the web interface

To configure sending data:

1. Open the Kaspersky IoT Secure Gateway 1000 web interface.
2. Open the Settings section and then the Configuration tab.
3. Find the ru.aprotech.mqttpublisher section in the text displayed on the tab.

4. Copy the Base64-encoded text with the application settings located in the configContent line.
5. Decode the text from Base64 to JSON (for example, using the https://www.base64decode.org/ site).
6. Copy the resulting text with the application settings into a separate file for subsequent editing.
7. Specify the MQTT settings and their values in accordance with JSON syntax.
8. Encode the resulting settings text back from JSON to Base64 (for example, using the https://www.base64encode.org/ site).
   Before that, make sure that you comply with the JSON syntax, because the Kaspersky IoT Secure Gateway 1000 web interface does not indicate any errors in the Base64-encoded configuration. An application started with configuration errors will be stopped. A message informing that the application terminated with an error appears in the Kaspersky IoT Secure Gateway 1000 log.
9. Copy the resulting text into the configContent line in the ru.aprotech.mqttpublisher section on the Configuration tab.
10. Click the Save button.

The application settings are applied after Kaspersky IoT Secure Gateway 1000 is restarted.

An example of MQTT Publisher settings in JSON format:

```
{
  "name": "MQTT Publisher Example",
  "description": "KISG Applications Development (Example)",
  "clientId": "KisgApplicationsDevelopmentExample0",
  "serverUri": "mqtts://192.168.2.1:8883",
  "userCredentials": null,
  "lastWill": {
    "topicName": "LastWill",
    "message": "LastMessage"
  },
  "topics": [
    {
      "name": "Heartbeat",
      "topicName": "Heartbeat"
    },
    {
      "name": "Consumer 1",
      "topicName": "FirstConsumer"
    },
    {
      "name": "Consumer 2",
      "topicName": "SecondConsumer"
    }
  ]
}
```

Code Block 3. Example of MQTT Publisher settings

To edit files in JSON format, we recommend using a text editor that supports JSON syntax highlighting. This will help avoid potential errors (for example, unbalanced braces).

3.3.2.3 Configuring MQTT Publisher using the KSC Web Console

To configure MQTT Publisher using Kaspersky Security Center 14.2 Web Console, follow the instructions in the corresponding section of the Kaspersky IoT Secure Gateway 1000 documentation. Follow the instructions. At step 8, specify the required application settings as stated in the MQTT Publisher settings description. The settings are specified in the "Application configuration" text field; it is important to comply with the JSON syntax. After specifying the settings, click Save at the bottom of the settings panel, and then at the bottom of the installed applications page.

```
{
  "name": "MQTT Publisher Example",
  "description": "KISG Applications Development (Example)",
  "clientId": "KisgApplicationsDevelopmentExample0",
  "serverUri": "mqtts://192.168.2.1:8883",
  "userCredentials": null,
  "lastWill": {
    "topicName": "LastWill",
    "message": "LastMessage"
  },
  "topics": [
    {
      "name": "Heartbeat",
      "topicName": "Heartbeat"
    },
    {
      "name": "Consumer 1",
      "topicName": "FirstConsumer"
    },
    {
      "name": "Consumer 2",
      "topicName": "SecondConsumer"
    }
  ]
}
```

Code Block 4. Example of MQTT Publisher settings

3.3.2.4 MQTT Publisher settings description

Settings marked as required must be configured. The other parameters are optional. For optional parameters that are not included in the configuration file, the default value prescribed by the MQTT protocol may be used.

Settings used to configure MQTT Publisher

Parameter name	Required parameter	Data type	Title	Possible values and notes
`name`	Yes	string	Name of MQTT Publisher that sends data to the MQTT broker.	`<MQTT Publisher name>`. Example: `"Kaspersky IoT Secure Gateway 1000 MQTT Publisher"`.
`description`	No	string	Description of MQTT Publisher that sends data to the MQTT broker.	`<MQTT Publisher description>`. Example: `"Transfer data to MQTT Broker by Kaspersky IoT Secure Gateway 1000"`.
`clientId`	Yes	string	Unique ID of MQTT Publisher.	`"1"`. The `clientId` value must be unique among all clients connected to the MQTT broker.
`serverUri`	Yes	string	Address of the server that MQTT	`<scheme>://<host>:<port>`. Example: `"ssl://192.168.188.8:8883"`.

Parameter name	Required parameter	Data type	Title	Possible values and notes
			Publisher connects to.	`ssl,` tls, wss, mqtts are architecture-prescribed schemes for querying a resource. `8883` is the default port.
`userCredentials`	Yes	object	Settings responsible for MQTT Publisher authentication on the server.	• `{username, password}` parameter block containing the user account credentials. • `null` is indicated if you want to allow an anonymous connection of the MQTT client to the MQTT broker. In this case, you do not need to fill in the `username` and `password` fields.
`username`	No	string	Name of the user account for authorization on the MQTT server.	`"username"`.
`password`	No	string	Password of the user account for authorization on the MQTT server.	`"password"`.
`lastWill`	No	object	Parameter block for configuring a message informing that the client was improperly disconnected (LWT message).	`{topicName, message}` parameter block. The application can specify the LWT message when connecting to the MQTT broker for the first time. The MQTT broker will store this message until it detects an improper disconnection of the application. Upon detection of an improper disconnection, it will send the LWT message to all clients that have subscribed to this type of message. The MQTT broker does not send this message when the application is disconnected properly.
`topicName`	No	string	Name of the MQTT topic that determines the	`<topicName>`. Example: `"LastWill"`.

Parameter name	Required parameter	Data type	Title	Possible values and notes
			information channel where the LWT message will be published.	
`message`	No	string	Contents of the LWT message.	`<message>`. Example: `"LastMessage"`.
`keepAlive`	No	int	Time that the MQTT broker can wait to receive a message from the MQTT Publisher before terminating the connection due to inactivity.	`800`. The default value is `120`. Available values: `0`–`65535`. If the `keepAlive` value is equal to zero, the server will not be obligated to disconnect a client based on inactivity of the client. If the server deems a client to be inactive or if the client is not responding to queries, the server can disconnect the client at any time, irrespective of the `keepAlive` value provided by the client.
`qualityOfService`	No	int	Setting defining the guarantee to receive messages.	`1`. Agreement between the message sender (publisher) and message recipient (subscriber) that defines a guarantee to deliver a specific message. The MQTT specification defines the following three levels of `qualityOfService`: • `0` is no more than one time: the client publishes messages without verifying whether they are delivered to the broker. Messages may be lost or duplicated. • `1` is at least one time: the broker confirms delivery. Messages may be duplicated, but delivery is guaranteed. • `2` is exactly one time: message delivery is guaranteed, and any potential duplication is eliminated. The default value is `1`.

Parameter name	Required parameter	Data type	Title	Possible values and notes
`topics`	Yes	array of objects	Array from the parameter blocks of MQTT topics.	Array of `[{name, topicName}]` blocks. A separate parameter block in the array is completed for each MQTT topic.
`name`	Yes	string	Destination connection point name. Used for configuring application routing.	`<name>`. Example: `"Temperature"`. Each value of the name setting in the topic objects of the MQTT Publisher configuration must be unique.
`topicName`	Yes	string	Name of the MQTT topic.	`<topicName>`. Example: `"Heartbeat"`. See also: Special considerations when generating names of MQTT topics.

3.3.2.5 Special considerations when generating names of MQTT topics

When filling in the values for `topicName`, please adhere to the following guidelines:

- Wildcard characters cannot be used in the names of MQTT topics: # and +. We also do not recommend using the `$` character in the names of MQTT topics.
- The name of an MQTT topic cannot be blank (it must contain at least one character).
- The names of MQTT topics are case sensitive.
- The names of MQTT topics can contain a blank space character.
- MQTT topics that are separated by only a `/` character at the beginning or end of the name are considered to be different MQTT topics.
- An MQTT topic name consisting of only a `/` character is permitted.
- The name of an MQTT topic must not contain the null character (NUL).
- Names of MQTT topics are UTF-8 strings of size that does not exceed 65535 bytes.

3.4 Configuring application routing

Configuring application routing is required for the correct data transfer from OPC UA Client to MQTT Publisher and, as a result, from the OPC UA server to the MQTT broker. Before you start configuring application routing, make sure that you have correctly configured the applications.

To create a new route using Kaspersky Security Center 14.2 Web Console, follow the instructions in the corresponding section of the Kaspersky IoT Secure Gateway 1000 documentation. Follow the instructions. At step 7:

- In the Source app drop-down list, select OPC UA Client.
- In the Source connection point drop-down list, select the connection point indicated by the name setting in the nodes section of the OPC UA Client configuration.
- In the Destination app drop-down list, select MQTT Publisher.
- In the Destination connection point drop-down list, select the connection point indicated by the name setting in the topics section of the MQTT Publisher configuration.
- Click Save in the lower part of the pane.

The route for apps will be created and displayed in the table. A new route is created in the active state by default. The applications apply the created routes after restarting Kaspersky IoT Secure Gateway 1000.

The key condition for the correct application routing is the correct mapping of the name setting of the OPC UA data nodes and MQTT topics. Note that the names of the connection points must be unique within the same application, but can be the same between applications.

Similarly, you can modify previously created application routes by following the instructions in the Kaspersky IoT Secure Gateway 1000 documentation, You can also delete previously created routes.

You can also create routes by using the web interface of Kaspersky IoT Secure Gateway 1000. Follow the instructions below:

1. Open the Kaspersky IoT Secure Gateway 1000 web interface.
2. Open the Settings section and then the Configuration tab.
3. Find the section dedicated to routing in the text presented on the tab: APPS_ROUTING (APPLICATIONS object —> APPS_ROUTING object).
4. In the applications section, in the endpoints setting, list all connection points.
    a. For OPC UA Client, these are the name settings in the nodes section that you specified in the application configuration.
    b. For MQTT Publisher, these are the name settings in the topics section that you specified in the application configuration.
5. In the routes section, specify the settings for all routes that must be created.
    a. In the destination section, in the application_id setting, specify ru.aprotech.mqttpublisher.
    b. In the endpoint setting, specify the required connection point indicated by the name setting in the topics section of the MQTT Publisher configuration.
    c. In the source section, in the application_id setting, specify ru.aprotech.opcuaclient.
    d. In the endpoint setting, specify the necessary connection point indicated by the name setting in the nodes section of the OPC UA Client configuration.
    e. Specify true in the active setting.
6. Repeat step 5 for all routes you want to create. An example of a completed APPS_ROUTING section is shown below.

7. Click the Save button.

```
"APPS_ROUTING": {
    "applications": [
      {
        "application_id": "ru.aprotech.opcuaclient",
        "endpoints": [
          "Provider 1",
          "Provider 2"
        ],
        "name": "OPC UA Client",
        "subtype": "Input",
        "type": "Network protocol converter"
      },
      {
        "application_id": "ru.aprotech.mqttpublisher",
        "endpoints": [
          "Heartbeat",
          "Consumer 1",
          "Consumer 2"
        ],
        "name": "MQTT Publisher",
        "subtype": "Output",
        "type": "Network protocol converter"
      }
    ],
    "routes": [
      {
        "active": true,
        "destination": {
          "application_id": "ru.aprotech.mqttpublisher",
          "endpoint": "Consumer 1"
        },
        "source": {
          "application_id": "ru.aprotech.opcuaclient",
          "endpoint": "Provider 1"
        }
      },
      {
        "active": true,
        "destination": {
          "application_id": "ru.aprotech.mqttpublisher",
          "endpoint": "Consumer 2"
        },
        "source": {
          "application_id": "ru.aprotech.opcuaclient",
          "endpoint": "Provider 2"
        }
      }
    ]
  }
},
```

Code Block 5. Example of APPS_ROUTING configuration section

Special considerations for configuring application routing in the current version of Kaspersky IoT Secure Gateway 1000:

- After making any changes to the Kaspersky IoT Secure Gateway 1000 configuration using the web interface, the active setting of the existing application routes is automatically set to false. For the routes to work correctly, specify the true value for the active settings again.
- The capability of deactivating routes is not provided. Routes can be created, modified, or deleted. At the same time, routes cannot have any value other than true in the active setting.

- If you specify false for a route's active setting using the Kaspersky IoT Secure Gateway 1000 web interface, then when this route is viewed using Kaspersky Security Center 14.2 Web Console, it will have the Active value. In this case, Kaspersky Security Center 14.2 Web Console offers you to save changes. When the configuration changes are saved, and the configuration is opened in the web interface, the active setting of the route is set to true.
- Restart Kaspersky IoT Secure Gateway 1000 for the created or modified routes to start working.

3.5 Starting and stopping applications

The applications must be in the Started state to perform their functions.

To start the application using the web interface, follow the instructions in the corresponding section of the Kaspersky IoT Secure Gateway 1000 documentation. Follow the instructions. At step 3, click Start in the Management column, in the OPC UA Client or MQTT Publisher line.

Application launch conditions:

- The application is installed and configured without errors and has a Stopped status.
- The Manual start or the Autostart rule is selected for the application.

To stop the application using the web interface, follow the instructions in the corresponding section of the Kaspersky IoT Secure Gateway 1000 documentation. Follow the instructions. At step 3, click Stop in the Management column, in the OPC UA Client or MQTT Publisher line.

You can stop the application if configured without errors and has a Run status. For example, you need to stop the application if its configuration must be updated. After making and saving configuration changes, you can start the application again.

To start or stop the application using Kaspersky Security Center 14.2 Web Console, follow the instructions in the corresponding section of the Kaspersky IoT Secure Gateway 1000 documentation. Follow the instructions. At step 7 select the check box next to OPC UA Client and MQTT Publisher, and click Start/Stop at the top of the table.

3.5.1 Changing startup rules and configuring application restart

You can use the Kaspersky IoT Secure Gateway 1000 web interface or Kaspersky Security Center 14.2 Web Console to configure how the application starts in Kaspersky IoT Secure Gateway 1000 (automatically or manually) or prevent the application from starting. If you have changed the launch rule for a running application, the modified launch rule will be applied only after the applications are stopped.

To configure the application restart, manually reconfigure Kaspersky IoT Secure Gateway 1000 using the web interface. The restart_on_failure key in the Kaspersky IoT Secure Gateway 1000 configuration is responsible for the application restart. This key activates the application restart mode in case of an abnormal termination.

3.6 Managing application logs

The sections below contain information about downloading application logs using the Kaspersky IoT Secure Gateway 1000 web interface and managing logging levels.

In this Help section
Downloading application logs using the web interface
Managing logging levels

3.6.1 Downloading application logs using the web interface

Kaspersky IoT Secure Gateway 1000 logs events generated by installed applications. Application logs are required to diagnose application operation and contact technical support.

The instructions on how to download application log files using the Kaspersky IoT Secure Gateway 1000 web interface are provided in the Kaspersky IoT Secure Gateway 1000 technical documentation.

3.6.2 Managing logging levels

Kaspersky IoT Secure Gateway 1000 supports six logging levels. Logging levels are described in the table below and are ranged by the level of detail of the information saved in the logs.

Logging level	Level of detail	Title	Example
0	Critical	Only messages about abnormal situations that lead to an emergency application termination are logged.	[04-09-2023 13:48:19][Critical][ru.aprotech.jsonreceiver][3562:3563][][MessageRouterImpl.cpp:GetMessageConsumerConnectionInfo:147]* Message Router found 0 Message Consumers, but expected 1 exactly
1	Error	Messages about abnormal situations interrupting the operation (for example, data transfer between applications) are logged.	[04-09-2023 13:48:19][Error][ru.aprotech.jsonreceiver][3562:3563][][MessageRouterImpl.cpp:GetMessageConsumerConnectionInfo:147]* Message Router found 0 Message Consumers, but expected 1 exactly
2	Warning	Messages about abnormal situations that do not interfere with the operati	[04-09-2023 13:48:18][Warning][ru.aprotech.mqttpublisher][3981:4200][][Socket.cpp:Connect:90]* Failed to connect to 192.168.2.1:8883

Logging level	Level of detail	Title	Example
		on are logged.	
3	Info	Information about the standard operation is logged.	[04-09-2023 13:48:19][Info][ru.aprotech.jsonreceiver][3562:3563][][MessageRouterImpl.cpp:DoUp:65]* Message Router will make next attempt after timeout
4	Debug	Detailed technical information about the operation is logged.	[04-09-2023 13:48:19][Debug][ru.aprotech.jsonreceiver][3562:3563][][MessageRouterImpl.cpp:GetMessageConsumerConnectionInfo:147]* Message Router found 0 Message Consumers, but expected 1 exactly
5	Trace	The maximum possible volume of information is logged and used for the most detailed debugging. When enabled, it may significantly affect the performance.	[04-09-2023 13:49:40][Trace][ru.aprotech.jsonreceiver][3562:4345][][Client.cpp:OnNewLine:116] CRT {"source": {"name": "JSON Receiver example", "port": "Generator"}, "dataItem": {"timestamp": "2023-09-04T14:29:31.503Z", "timestampSource": null, "value": "123", "status": "00000000"}}

Thus, you can select the desired logging level depending on the existing need for information about the application operation and the tasks of diagnosing the application status. By default, all applications use logging level 4 (Debug). If you need the log to contain data item traces, set the logging level to 5 (Trace). The logging level is set for each application separately.

To set the required logging level for an application:

1. Open the Kaspersky IoT Secure Gateway 1000 web interface.
2. Open the Settings section and then the Configuration tab.
3. Find in the text on the tab the logLevel line (APPLICATIONS object —> list of applications objects —> logLevel key).
4. Select the required logging level from the list above. For example, "logLevel":"Warning"
5. Click the Save button.

4 Diagnostics and contacting Technical Support

If you cannot configure the applications and you did not find a solution to your issue in the documentation or you have encountered any problems in the operation of applications, or if you needed to reinstall Kaspersky IoT Secure Gateway 1000 on a Kraftway Rubezh-N device, contact APROTECH LLC technical support by email: support@aprotech.ru. Attach the following to your message:

- Detailed description of the issue.
- OPC UA Client and MQTT Publisher settings, as well as the OPC UA server and MQTT broker settings.
- Application log files.
- Your organization name and contact details for feedback.

5 Licensing

The application's terms of use of the are set forth in the End User License Agreement or in a similar document regulating usage of the application.

6 Data provision

OPC UA Client and MQTT Publisher do not collect, use, or process user personal data.

7 Known limitations

7.1 General limitations

The following limitations apply to both applications:

- If one of the applications (OPC UA Client or MQTT Publisher) is deleted from the device using the Kaspersky IoT Secure Gateway 1000 web interface, the second application is deleted automatically.
- Kaspersky IoT Secure Gateway 1000 supports data transfer by applications on no more than 256 routes simultaneously.
- When starting or stopping applications manually, it is necessary to start or stop the OPC UA Client and MQTT Publisher applications only as a pair. To run, you must first run MQTT Publisher, then the OPC UA Client. Stop in reverse order: first OPC UA Client, then MQTT Publisher. The recommended application startup configuration is automatic startup for both applications.
- When connecting to the Kaspersky IoT Secure Gateway 1000 web interface, when changing the password, the password entry form will indicate that the password meets the requirements, including in cases where the password does not actually meet the requirements. Make sure that the password meets the requirements yourself, without relying on the information from the password entry form.
- In case of a data transfer error between the OPC UA Client and MQTT Publisher applications, there is no reliable way to understand which data was delivered correctly and which was not. There may be situations when some data will be marked as lost in the log, even if in fact it was transmitted correctly.
- The size of the storage space for the logs of the OPC UA Client and MQTT Publisher applications has a limit of 50 MB for each of the applications.
- Kaspersky IoT Secure Gateway 1000 is not equipped with an integrated uninterruptible power supply, so we recommend using an external UPS to avoid data loss in the event of an unintentional power outage.
- Changing the configuration of any of the applications (OPC UA Client or MQTT Publisher) disables data transmission routes. In such a situation, Kaspersky Security Center 14.2 Web Console will automatically switch all routes to the "Active" state and offer to "Save changes".
- The disability (the transfer of routes from the "Active" state) is of a notification nature. Kaspersky It Secure Gateway 1000 notifies applications about route invalidation, but does not prohibit data transmission over them.
- Kaspersky Security Center 14.2 Web Console does not provide the ability to download files uploaded by the user to the application configuration page (for example, certificate files).
- Kaspersky Security Center 14.2 Web Console when trying to install, uninstall or update an application on a Kaspersky IoT Secure Gateway 1000 managed with it, the list of available applications does not display in the "Programs" menu in the "Program Settings" tab in the "Application Manager" submenu.

7.2 OPC UA Client limitations

OPC UA Client has the following limitations of the OPC UA protocol support:

- There is no secure connection over the OPC UA protocol. The connection is established when the None security policy is used. Authentication on the OPC UA server is performed using a user name and password. The credentials are transmitted in clear format. It is also possible to connect anonymously by specifying null in the userCredentials section.
- Only the following data types described in the OPC UA specification are supported:
    - Boolean
    - SByte
    - Byte
    - Int16
    - UInt16
    - Int32
    - UInt32
    - Int64
    - UInt64
    - Float
    - Double
    - String
    - DateTime
    - XmlElement
    - NodeId (only numeric and string)
    - ExpandedNodeId (only numeric and string)
    - StatusCode
    - QualifiedName
    - LocalizedText (partially)
    - Variant
- Double- and Float-type data received over the OPC UA protocol is rounded to the nearest six significant digits.
- To transmit data over OPC UA, the server must support the MonitoredItem and Subscription service sets.
- Only one OPC UA client connection to one OPC UA server is available.

7.3 MQTT Publisher limitations

MQTT Publisher has the following limitations of the MQTT protocol support:

- Only one MQTT client connection to one MQTT broker is available.
- MQTT Publisher uses the "1" value for the Clean Session flag each time it connects to the MQTT broker.
- The value of the qualityOfService setting is common for all published messages from MQTT Publisher to topics (the topics setting), including service topics (heartbeat, lastWill).
- The qualityOfService setting cannot be configured for every published message from MQTT Publisher to topics (the topics setting).

- The MQTT client does not use the `retain` flag when sending messages nor for the LWT message (message informing that the client was improperly disconnected).
- Setting the `keepAlive` parameter of the MQTT client to `0` will not disable the "keep alive" mechanism (this mechanism disconnects a client that is inactive for too long).
- The MQTT client ignores the lack of response from the MQTT broker for a prolonged period of time and does not close the connection.
- If the connection is disrupted, no more than 10 published messages may be lost after the connection is restored and if the buffer has sufficient free space.
- MQTT Publisher may stop sending data to the MQTT broker after it is stopped and restarted. To restore the proper operation of the application, restart Kaspersky IoT Secure Gateway 1000.
- If you use the configuration where manual startup is selected for MQTT Publisher and automatic startup is selected for the OPC UA Client, then when Kaspersky IoT Secure Gateway 1000 is turned on, both applications are stopped. The reverse configuration (manual OPC UA Client startup and automatic MQTT Publisher startup) works properly.

7.4 TLS limitations

MQTT Publisher has the following limitations of the TLS protocol support:

- Only TLS protocol versions 1.2 or later are supported.
- The Kaspersky IoT Secure Gateway 1000 component, responsible for maintaining an encrypted data communication channel, does not support the use of the subjectAltName field and does not allow establishing a connection with the MQTT broker if the subjectAltName field is used in the certificate.
- The Kaspersky IoT Secure Gateway 1000 component, responsible for maintaining an encrypted data communication channel, requires that the Common name field in the certificate contain the IP address of the MQTT broker.
- Only TLS cipher suites are supported:
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
  - TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256
  - TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- Only the following digital signature algorithms are supported:
  - `ecdsa_secp521r1_sha512`
  - `ecdsa_secp384r1_sha384`
  - `ecdsa_secp256r1_sha256`
  - `ed25519`
  - `ed448`
  - `rsa_pss_pss_sha512`
  - `rsa_pss_rsae_sha512`
  - `rsa_pss_pss_sha384`

- `rsa_pss_rsae_sha384`
- `rsa_pss_pss_sha256`
- `rsa_pss_rsae_sha256`
- `rsa_pkcs1_sha384`
- `rsa_pkcs1_sha512`
- `rsa_pkcs1_sha256`

8 Other sources of information

During installation, configuration, and use of the applications, you can refer to the following additional documents:

- Kaspersky IoT Secure Gateway 1000 technical documentation.
- OPC UA protocol specification.
- MQTT protocol specification.

9 Glossary

Kaspersky Appicenter for Developers

A portal that allows interaction between application developers for KasperskyOS and their end users

Kaspersky IoT Secure Gateway 1000

A cyber-immune system based on KasperskyOS with a preconfigured set of application software. Kaspersky IoT Secure Gateway 1000 is installed on the Kraftway Rubezh-N embedded computer and serves as an Internet of Things (IoT) Secure Gateway for enterprise networks.

Kaspersky Security Center

An application designed for the centralized solution of the main tasks of managing and maintaining the organization's network protection system. The application provides the administrator with access to detailed information about the security level of an organization network and allows you to configure all the components of protection based on Kaspersky applications.

Kaspersky Security Center 14.2 Web Console

A web application designed to manage the state of the security system of enterprise networks that are protected by Kaspersky applications.

KasperskyOS

A microkernel operating system used for building secure solutions.

Message Queuing Telemetry Transport (MQTT)

A network protocol that works on top of the TCP/IP protocol stack to exchange messages between devices on the Internet of Things.

MQTT broker

A server that receives, filters, and forwards messages over the MQTT protocol.

MQTT-topic

A hierarchical path to the data source used for sending messages over the MQTT protocol.

Open Platform Communications Unified Architecture (OPC UA)

Specification defining the protocols and mechanism for data transfer in industrial networks as well as interaction between devices in these networks.

TLS

Secure protocol that uses encryption to transfer data in local networks and on the Internet. TLS is used to create secure connections between a client and a server.

Internet of Things (IoT) Secure Gateway

A system that ensures secure transmission of user traffic between sensors and an IoT platform.

Kaspersky IoT Secure Gateway 1000 web interface

Tool for working with Kaspersky IoT Secure Gateway 1000. To connect to the web interface, you need a browser installed on a computer that has access to Kaspersky IoT Secure Gateway 1000 through the internal network.

Internet of Things (IoT)

A network of interrelated electronic devices (things) that are equipped with built-in capabilities for interaction with the external environment or with each other without human involvement.

Data source

Standalone data source for exchanging messages between devices on the internet of things. For example, a data source could be an OPC UA server at the management controller of an industrial machine.

Cyberimmune information system

A system that guarantees the fulfillment of specific security objectives in all possible scenarios of system usage as stipulated by the developers.

Client

Participant of client-server interaction that sends requests to the server and receives responses to those requests.

Root certificate

Certificate of the root Certification Authority.

Root Certification Authority

Top Certification Authority that is not subordinate to any higher Certification Authority.

Encryption key

Component of a pair of encryption keys used for asymmetric encryption. Keys can be public or private.

Cipher suite

A collection of ciphers that work together to perform various cryptographic functions such as the generation of keys and authentication. Cipher suites describe the steps that the keys must perform and the order in which those steps are performed.

Unidirectional gateway (data diode)

A data gateway that is created using the software and allows only one-way data transfer. It is an effective mean of protection against confidential information leaks.

Software platform

A collection of software and tools provided to the developers for creating and running applications. Kaspersky IoT Secure Gateway 1000 acts as a software platform for OPC UA Client and MQTT Publisher.

Server

Participant of client-server interaction that processes requests from the client.

End-entity certificate

Certificate containing a public encryption key that can be used to verify or validate an end-entity, such as an MQTT client.

Certificate

Data structure with a digital signature containing a public encryption key and the ID of the client or server.

Administrator certificate

A certificate used for user authentication in the Kaspersky IoT Secure Gateway 1000 web interface.

Event

A record containing information about detecting data in the system or on the LAN that requires the attention of an employee responsible for data security in your organization. An event is stored in the memory of the Kraftway Rubezh-N embedded computer.

Data node

Structural element of an OPC UA information model containing data and metadata.

Logging level

The operating mode of the Kaspersky IoT Secure Gateway 1000 log that determines which events are logged in the application operation log, as well as the level of detail of this information.

Certificate chain

Combination of any number of intermediate certificates between the end-entity certificate and the root certificate.

Digital signature

A value calculated with an encryption algorithm and added to data in such a way that any data recipient can use the signature to verify the origin and integrity of the data.

Encryption

Conversion of data from readable format to encoded format. Encrypted data can be read or processed only after decryption.

10 Information about third-party code

Third party code information is contained in the file named legal_notices.txt.

11 Trademark notices