



OPC UA Forge

User Manual
Version 1.3.0

Table of Contents

1. Getting Started	1
1.1. Introduction	1
1.1.1. Modules	2
1.2. Technical Details	3
1.2.1. System Requirements	3
1.2.2. Default settings	3
1.2.3. Supported Features	3
1.2.4. Supported Security Features	4
1.2.5. Supported Databases Features	4
1.3. Installing the Application	5
1.3.1. Windows	5
1.3.2. Linux	5
1.3.3. Docker	5
1.4. Running the Application	6
1.4.1. Windows	6
1.4.2. Linux	6
1.4.3. Docker	6
1.5. Web User Interface	6
1.5.1. Login	7
1.5.2. Start Modules	8
1.5.3. Navigation Bar	8
1.5.4. Logout	8
1.6. Connect to Forge OPC UA Server	9
2. Dashboard	10
2.1. Forge Server Status	10
2.2. Modules	10
2.3. Quick Access	11
3. OPC UA Server	12
3.1. Address Space	12
3.1.1. Address Space Icons	13
3.1.2. Address Space Structure	13
3.1.3. Add Node	14
3.1.4. Add Reference	16
3.1.5. Remove Node	16
3.1.6. Mapping	16
3.2. Expression	20
3.2.1. Browse Nodes	22
3.3. Namespaces	23
3.3.1. NodeSets	24
3.3.2. Namespaces Table	25
3.4. Certificates	26
3.4.1. Forge's Certificate	27
3.4.2. Manage Certificates	28
3.5. Server Settings	29
3.6. Reverse Connection	30
4. Data Sources	32
4.1. OPC UA Servers	32

4.1.1. Default Connection	32
4.1.2. Reverse Connection	33
4.1.3. Import Connections	35
4.2. OPC UA Subscribers	35
4.2.1. OPC UA over MQTT Subscribers	36
4.2.2. OPC UA over UDP Subscribers	37
4.3. S7COMM Devices	38
4.4. ADS Devices	40
4.5. EtherNet/IP Devices	41
5. Redundancy	43
6. OPC UA Publisher	44
6.1. Connections	44
6.1.1. OPC UA over MQTT	44
6.1.2. OPC UA over MQTTS	45
6.1.3. OPC UA over UDP	47
6.2. Writer Groups	48
6.2.1. Writer Groups JSON	48
6.2.2. Writer Groups UADP	49
6.3. Data Set Writer	50
6.3.1. Data Set Writer MQTT	50
6.3.2. Data Set Writer UDP	51
6.4. Variable Data Sets	52
6.5. Event Data Sets	53
7. Data Exchange	54
7.1. Import Data Exchange Groups	55
8. Event Generator	56
8.1. Event Templates	56
8.2. Generators	57
9. Event Mapper	60
10. History	62
10.1. InfluxDB	62
10.2. Storage	62
10.3. Collectors	63
10.3.1. Add Nodes to Collector	64
10.3.2. Import History Collectors	64
11. Modbus	65
11.1. Modbus Device	65
11.2. Modbus Server	65
11.3. Modbus Configuration	65
11.3.1. Modbus with TCP/IP	65
11.3.2. Modbus with Serial Port	67
11.3.3. Advanced Settings	69
11.3.4. Duplicate Modbus Device/Server	70
11.3.5. Variable Configuration	70
12. MQTT	75
12.1. Connections	75
12.2. Templates	77
12.3. Publishers	79
12.3.1. Single Node	80

12.3.2. Composite	82
12.3.3. UNS	84
12.4. Subscriptions	87
12.5. JSON Mappings	88
12.5.1. Node (value)	89
12.5.2. Node (value, JSON)	90
12.5.3. Node (value, structure)	90
12.5.4. Node (non-value)	91
12.5.5. Static Value	92
12.5.6. Method	93
12.5.7. Topic	94
13. Data Logger	95
13.1. Data Sinks	95
13.1.1. Influx	95
13.1.2. SQL	96
13.1.3. CSV	97
13.2. Logging Profiles	98
13.2.1. Influx	98
13.2.2. SQL	100
13.2.3. CSV	101
13.3. Data Loggers	102
13.4. Event Loggers	103
14. OPC UA over REST	106
14.1. Authorize	106
14.2. OPC UA API	107
15. Advanced Configurations	109
15.1. User Management	110
15.1.1. Users	110
15.1.2. User Groups	110
15.1.3. Providers	111
15.1.4. Global Permissions	113
15.1.5. Change Password	113
15.1.6. Copy API Key	114
15.2. Web Server	114
15.3. MQTT Broker	114
15.4. Transformation Dictionaries	117
15.5. Expressions	117
15.6. OpenAPI Document	118
15.6.1. Initialization API	119
15.6.2. Import API	119
15.7. License	119
15.8. Backup / Restore	119
15.9. Restart Forge	120
16. Appendix	121
16.1. Manual Port Configuration	121
16.2. File Locations	121
16.2.1. Configuration Folder	121
16.2.2. Reset Configuration	121
16.2.3. Configuring Embedded Devices	122

16.2.4. License File	122
16.2.5. Log File	122
16.2.6. Certificate Folder	122
16.2.7. Installation Folder	123
16.3. Manage Forge Service	124
16.3.1. Windows	124
16.3.2. Linux	124
16.3.3. Docker	124
16.4. Uninstalling the Application	126
16.4.1. Windows	126
16.4.2. Linux	126
16.4.3. Docker	126
16.5. Contact Us!	127
16.5.1. Support	127
16.5.2. Find more about us	127

1. Getting Started

1.1. Introduction

Prosys OPC UA Forge is modern IT/OT integration software. Prosys OPC UA Forge allows you to harmonize your data structures, improve security, and manage your system from one central point. Prosys OPC UA Forge enables you to connect with any OPC UA server and access data with any OPC UA clients or share data through MQTT or UDP protocols.

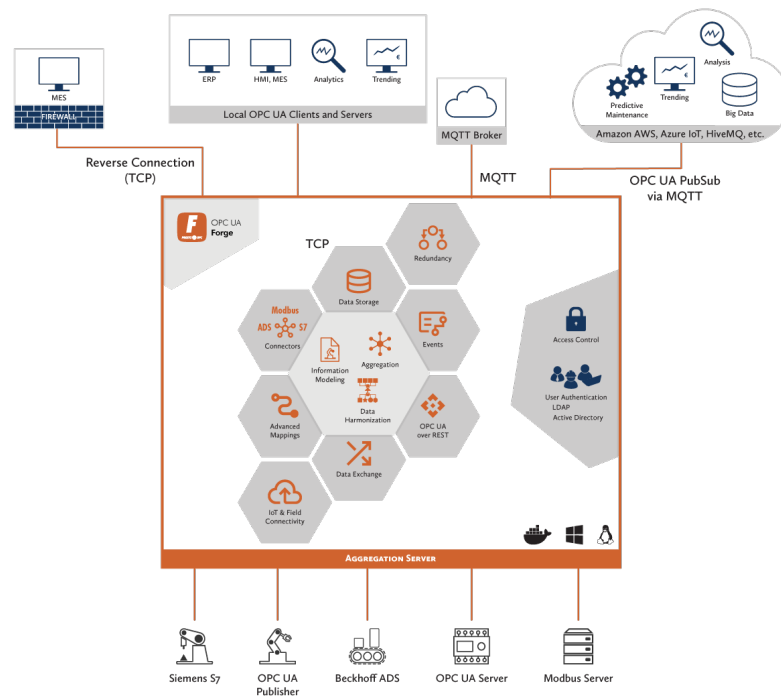


Figure 1. Forge's connections.

In addition, OPC UA Forge enables seamless communication with MQTT applications with easy configurations for MQTT publisher and subscribers. Forge also has connectors to Modbus systems, which allows you to migrate them to modern OPC UA technology. Having all your data behind a single point of access enables tighter firewall rules and better network control, simplifying your system architecture and significantly improving security. Once configured, Prosys OPC UA Forge is your secure gateway for accessing OT data of all underlying servers.



Check out our [Mastering Forge](#) series for step-by-step guides and videos on how to use the various features Forge has to offer.

1.1.1. Modules

Forge consists of Core (active always) and optional Add-On modules. Available modules in Forge depend on the inserted license keys.

Module	Description	Add-on
OPC UA Server	Browse and edit Forge's address space.	Core
Data Sources	Configure OPC UA servers as data sources.	Core
Redundancy	Configure redundancy groups for namespaces.	Core
MQTT	Configure MQTT connections.	IoT & Cloud Connectivity
MQTT Broker	Configure MQTT connections.	IoT & Cloud Connectivity
OPC UA Publisher	Configure an OPC UA Publisher.	IoT & Cloud Connectivity
OPC UA Subscribers	Configure an OPC UA Subscriber.	IoT & Cloud Connectivity
S7COMM Devices	Configure S7COMM connectors	Siemens S7 Devices
ADS Devices	Configure ADS devices as data source.	Beckhoff ADS Connector
EtherNet/IP Devices	Configure Ethernet/IP devices as data source.	Ethernet/IP Connector
Data Exchange	Configure data exchange between servers.	Data Exchange
Event Generator	Configure new events.	Event Suite
Event Mapper	Map data from Events.	Event Suite
Data Logger	Configure Data Sinks.	History Suite
History	Configure history collection.	History Suite
Modbus	Configure Modbus communication.	Modbus Connector
OPC UA over REST	Communicate with Forge over REST API.	OPC UA over Rest

1.2. Technical Details

1.2.1. System Requirements

Forge is supported in following environments:

- Windows,
- Linux,
- Docker

Web User Interface needs a modern web browser (Chrome, Edge, Firefox etc). Specific computational and memory requirements depends highly of the use case.

1.2.2. Default settings

Service	Port	Address
Forge server	56560	opc.tcp://<hostname>:56560/OPCUA/Forge
Web UI (http)	8080	<a href="http://<hostname>:8080">http://<hostname>:8080
Web UI (https)	8443	<a href="https://<hostname>:8443">https://<hostname>:8443
Rest API (swagger)		<a href="http://<hostname>:8080/swagger-ui/index.html#/">http://<hostname>:8080/swagger-ui/index.html#/
MQTT Broker	1883	

1.2.3. Supported Features

Forge supports the following OPC UA features:

- Address Space Model
- Information Model
- Historical Access
- Data Access
- Events
- Methods
- Security Model
- PubSub

1.2.4. Supported Security Features

	Supported features
Security Mode	<ul style="list-style-type: none"> • None, • Basic128Rsa15, • Basic256, • Basic256Sha256, • Aes128Sha256RsaOaep, • Aes256Sha256RsaPss
Security Policy	<ul style="list-style-type: none"> • None, • Sign, • SignAndEncrypt
User Authentication	<ul style="list-style-type: none"> • Anonymous, • User&Password • Certificate

1.2.5. Supported Databases Features

Forge enables logging to the following databases:

- CSV
- Influx
- MySQL
- MariaDB
- Microsoft SQL Server
- Posgre
- Oracle

1.3. Installing the Application

To get started with Prosystech OPC UA Forge, download the installation content for your system.

1.3.1. Windows

Run the installer and follow the instructions.

1.3.2. Linux

Run the installer and follow the instructions. To run the installer file, navigate to the directory where the installer is downloaded and run the following command:

```
sudo sh <filename>
```

1.3.3. Docker

Docker image ZIP package includes the Docker image for Forge and scripts for starting the Docker Application.

Extract the ZIP file to a location of your choice. In the next steps the path to this location will be referred to as: <installation_folder>



If you do not have a Docker Engine installed, you will need to install one first. You can follow the official Docker Installation instructions by following this link:

```
https://docs.docker.com/engine/install/
```

If you are running the Docker on Windows, you need to use Linux containers.

1.4. Running the Application

1.4.1. Windows

Follow these steps to start the Forge service in Windows:

1. Open Windows search and write "services"
2. Select the Services app
3. Find "Prosystech OPC UA Forge" from the list of services
4. Right-click "Prosystech OPC UA Forge"
 - a. **Start**: Start the service

1.4.2. Linux

Start Forge service from the terminal with command:

```
sudo service prosystech-opc-ua-forge-service start
```

1.4.3. Docker

You can use the start-up script that is available under the `<installation_folder>/bin`. Open Powershell or linux terminal in `<installation_folder>/bin` and use following commands:

```
.\forge-docker.ps1 start
```

Windows Powershell command

```
sudo forge-docker.sh start
```

Linux terminal command

1.5. Web User Interface

By default, you can access the Web UI with standard browser using the following URL

```
https://localhost:8443
```



Forge uses a self-signed SSL certificate by default. If you cannot accept it, navigate to the HTTP version at <http://localhost:8080>

See chapter [Web Server](#) for more information about respective settings.



Forge uses the following default ports: 8080, 8443, and 56560. If these ports are already in use, please refer to the [Manual Port Configuration](#) chapter for instructions on how to change the port settings manually.

1.5.1. Login

When logging in for the first time, the user is required to upload their license and create new user credentials.

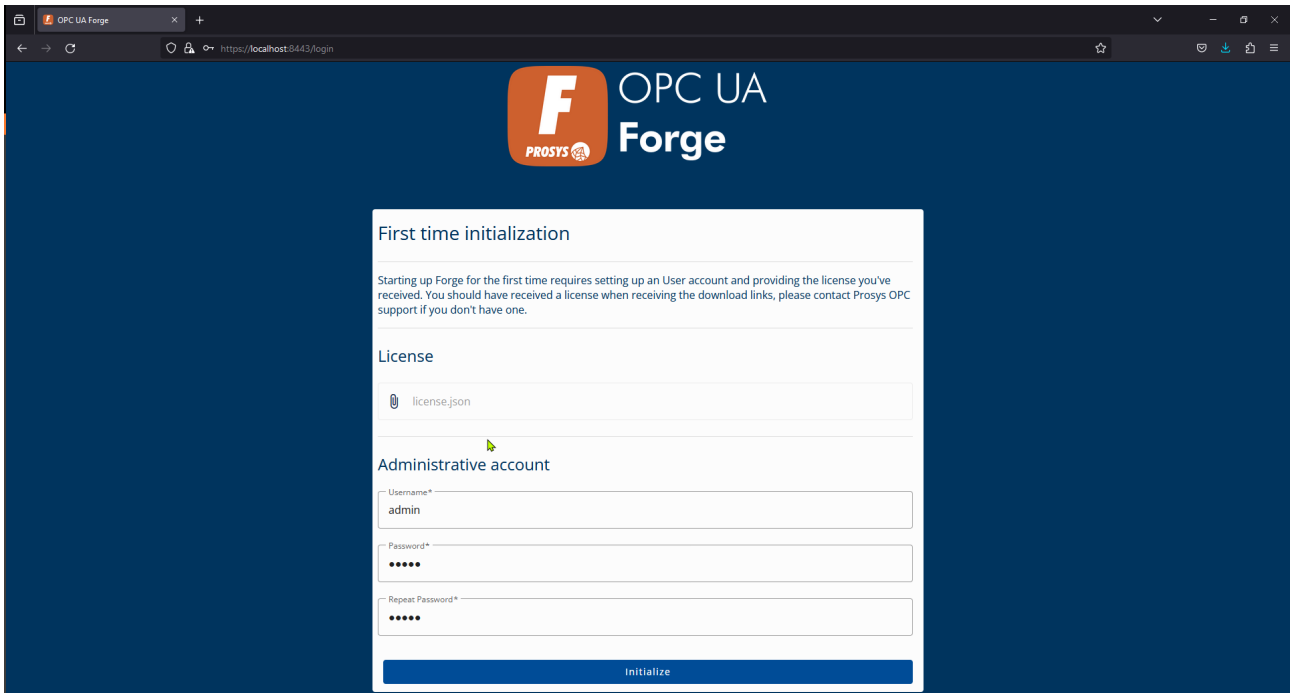


Figure 2. Forge's Initialization form.

After initialization is completed, a login form will appear. Use previously created user credentials to login.



Figure 3. Forge's login form.

1.5.2. Start Modules

After login is successfully completed, Forge will open the Dashboard view. Start the modules that you want to use, and you are ready to start configuring Forge. See chapter [Dashboard](#) for more information.

Modules

Data Logger Module	STOPPED	▶
Event Generator Module	STOPPED	▶
Event Mapper Module	STOPPED	▶
Exchange Module	STOPPED	▶
History Module	STOPPED	▶
MQTT Module	STOPPED	▶
Modbus Module	STOPPED	▶
OPC UA Publisher Module	STOPPED	▶
OPC UA Subscriber Module	STOPPED	▶

Figure 4. Core module is always enabled.

1.5.3. Navigation Bar

On the top part of the screen is the navigation bar. It can be used to navigate between modules. On the right side of the navigation bar there are three icons.



Figure 5. Navigate between modules and find additional menus.

First icon navigates you to Dashboard view. Second icon opens user menu to change password or logout. Third icon opens menu with additional views to control Forge.

1.5.4. Logout

After you have configured Forge, you may logout from the user-icon on the top right.

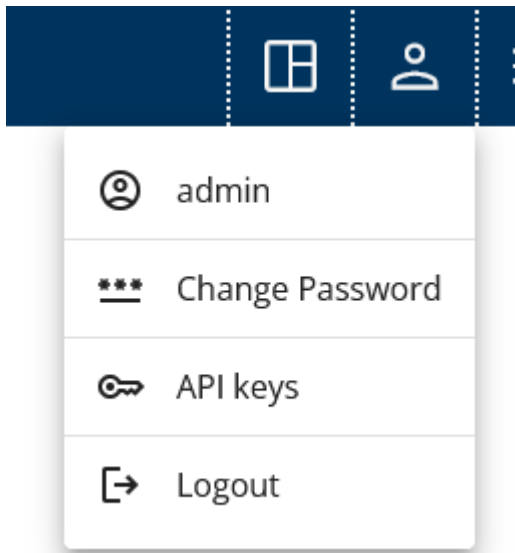


Figure 6. Logout from Forge.

1.6. Connect to Forge OPC UA Server

You can connect to Forge server with OPC UA client. You can find and copy the connection address from the [Dashboard](#) view.

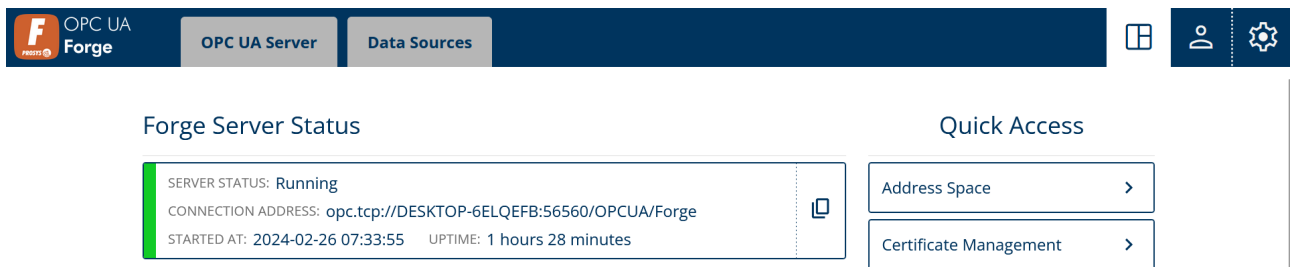


Figure 7. Copy Forge's connection address for OPC UA client.

From the Dashboard view you can find Forge server status card shown in figure above where the connection address is visible, and it can be copied to clipboard by clicking the copy-icon.



Forge's default connection address for connecting with OPC UA client is:

```
opc.tcp://<hostname>:56560/OPCUA/Forge
```

When using secure connections, client certificates can be trusted from Forge settings. See chapter [Certificates](#) for more information.

See chapter [Server Settings](#) for more information about respective settings.

2. Dashboard

Dashboard gives an overview of Forge's status and access to the most important features. You can easily control which modules are running or stopped, however respective license keys need to be in place.

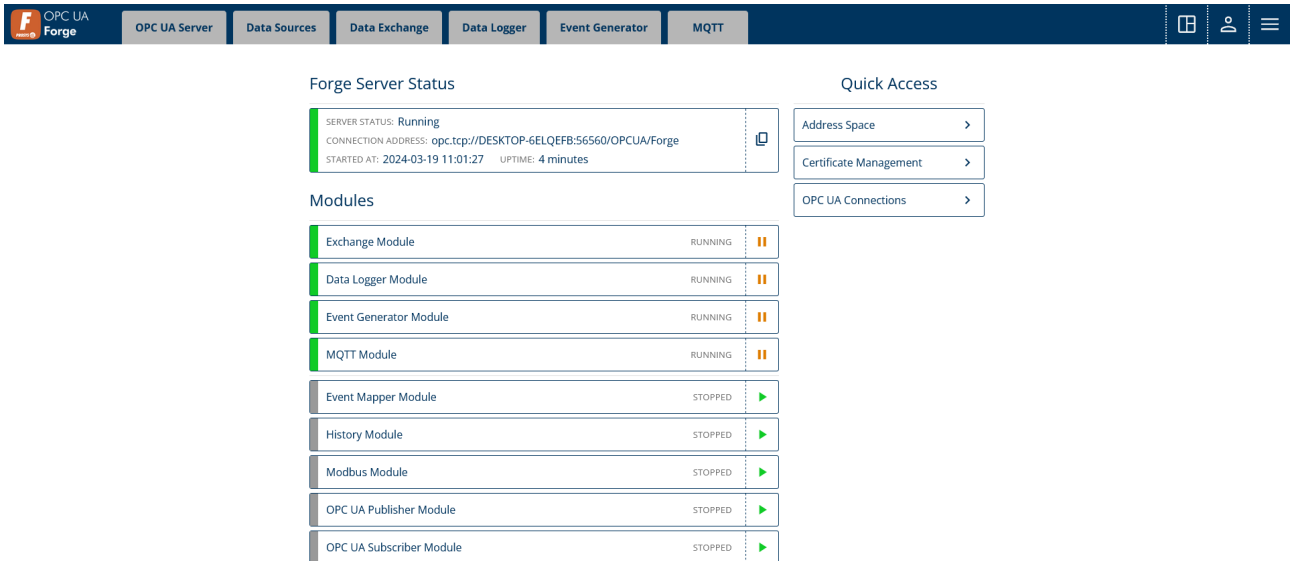


Figure 8. Dashboard is the landing page for Forge.

2.1. Forge Server Status

Forge Server Status card gives you information of Forge's server status. You find the connection address from the card and you can copy it to clipboard by clicking the right side of the card.

Forge Server Status

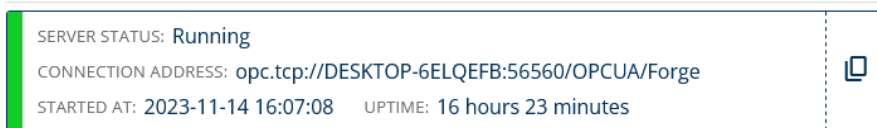


Figure 9. Quick overlook of the Forge server.

2.2. Modules

Here is a list of the modules that the current license that you have in Forge is enabling. User can activate or deactivate the enabled modules to make Forge fit the user's needs as required.

Modules

Data Logger Module	STOPPED	▶
Event Generator Module	STOPPED	▶
Event Mapper Module	STOPPED	▶
Exchange Module	STOPPED	▶
History Module	STOPPED	▶
MQTT Module	STOPPED	▶
Modbus Module	STOPPED	▶
OPC UA Publisher Module	STOPPED	▶
OPC UA Subscriber Module	STOPPED	▶

Figure 10. Manage the active modules.

2.3. Quick Access

Quick Access gives you quick navigation to important features of Forge.

Quick Access

Address Space	>
Certificate Management	>
OPC UA Connections	>

Figure 11. These links will navigate you directly to the modules.

3. OPC UA Server

OPC UA Server view is to manage Forge server address space and settings. From the OPC UA Server view you can access the settings for Forge's OPC UA server, browse and edit the address space, manage Namespaces and control certificates. The main view is Address Space view, because it gathers the features from other modules under one multi-functional view. Server Settings gives you access to i.e. configure security modes and connection endpoints of Forge. OPC UA Server view gives you access to manage certificates. The certificates are needed for using secured connections. Namespaces lists all the Namespaces and enables you to import Node Sets.

3.1. Address Space

Address Space view's modeling functionality provides you with the tools and capabilities to structure and organize your data in a way that aligns with your operational needs. By leveraging the modeling capabilities within Forge, you can create a comprehensive and structured data model that accurately reflects your assets, enhancing your ability to monitor, control, and optimize your operations.

The address space can be browsed by expanding the rows and navigating further down in the hierarchy.

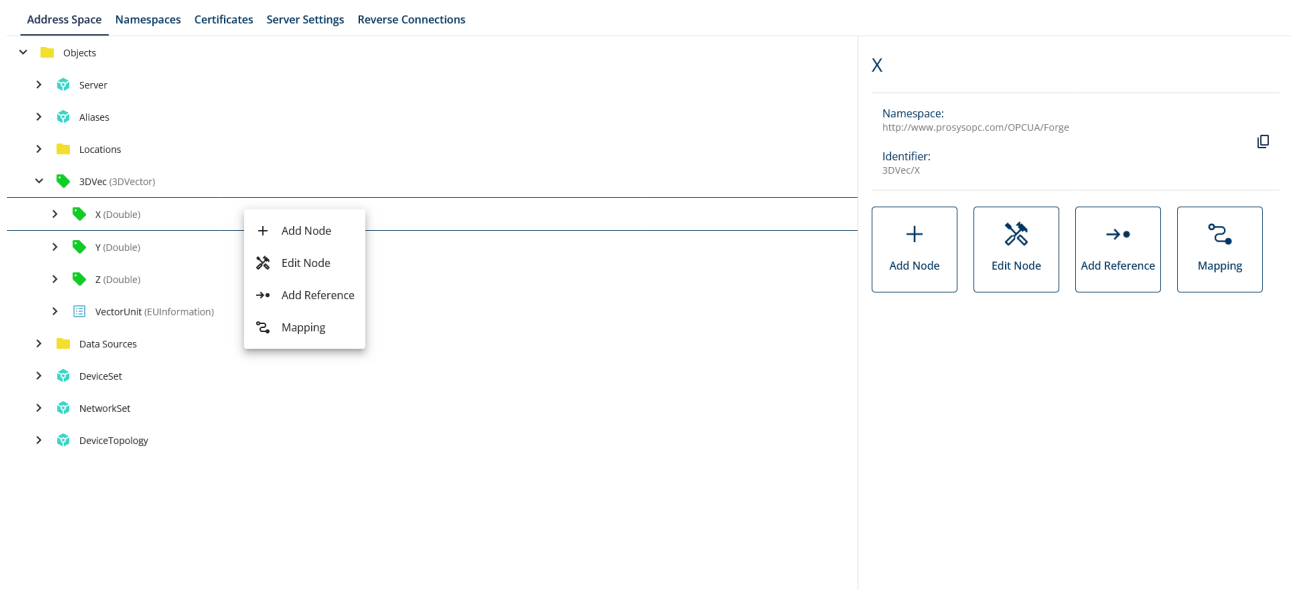


Figure 12. Overview of the Address Space view of Forge.



Hold shift to select multiple Nodes for editing.

Right click opens a context-menu from where you can access various features.

Forge's address space can be edited with the following actions:

Add Node

Creates a new Node under the selected Node.

Edit Node

Edit Node configuration.

Add Reference

Add Reference allows you to add additional references between Nodes.

Mapping

Map values from different Nodes to be presented in other Nodes.

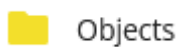
Remove Node

Removes the selected Node and everything under it.

In addition, there are features to add Nodes to Publishing, Expression, Data Exchange and History collection groups if these modules are running.

3.1.1. Address Space Icons

The following icons shows which modules are activated for certain nodes and they are listed after nodes in the address space.



Objects

Figure 13. Icon used to represent folders.



Server

Figure 14. Icon used to represent objects.



MyLevel (Double)

Figure 15. Icon used to represent variables. Data type is shown inside the brackets.



NamespaceArray (String)

Figure 16. Icon used to represent properties. Data type is shown inside the brackets.



SetSubscriptionDurable

Figure 17. Icon used to represent methods.

The following icons shows which modules are activated for certain Nodes and they are listed after Nodes in the Address Space.



Figure 18. 1.) Node is added to PubSub Data Set, 2.) Node is added to Data Exchange group, 3.) Attribute(s) of Node is mapped, 4.) Node is added to Historizing group, 5.) Node has added references.

3.1.2. Address Space Structure

Forge's Address Space has the Data Sources folder to organize the

OPC UA Servers

This folder lists all the OPC UA servers that Forge has active connection.

OPC UA Subscribers

OPC UA Subscribers folder has structure created from the subscriptions and the Variables received from messages. This folder is shown only if a Subscriber is configured in Data Sources and it is receiving messages.

Modbus Devices

This folder is shown if you have activated the Modbus module. This will list the variables which are received from Modbus server that Forge is connected to.

Modbus Servers

This folder is shown if you have activated the Modbus module. Here you can access the variables created in the internal Modbus server.

ADS Devices

Shows all connected ADS devices. This folder is visible if the ADS module is enabled.

S7COMM Devices

Shows all connected S7COMM devices. This folder is visible if the S7COMM module is enabled.

3.1.3. Add Node

The basic way to create instances into Forge is by adding a new Node in the Address Space view. To be able to add a new Node, you must select the Node where you want to add the new Node. Then you press the Add Node button and it opens you the form for configuring the new Node.

In the configuration of new Node you need to select the type for the Node. By creating the instance according to a type, the created Node will have the full type structure underneath it. The created structure is also protected from removing parts of the created instance. You can filter the model types with Model type URIs. New model types can be imported through [Node Sets](#) feature.

If the selected Model type is a variable, you need to select the Data type for the variable. The dropdown will list the possible Data types that the variable can have.

In OPC UA every Node is identified with a NodeId which needs to be unique in the context of the whole address space. The NodeId consists of Namespace, Id Type and Identifier. Namespace is used to group the address space and thus NodeIds can be separated by using different Namespace. Id Type defines only that in which format the Identifier will be given. This Identifier is value which can be user defined but it has to be unique in the context of the selected Namespace.



Default Namespace of Forge is the "http://www.prosysopc.com/OPCUA/Forge".

< New Node

Type Definition

Model type URI filter

Model type*

Data type*

Node Identity

Name*

Namespace *

Id Type *

Identifier*

Figure 19. The form for adding a new Node.

Model Type URI Filter

You can use the Namespace URI to filter the different model types.

Model Type

Offers a list from which you need to select one. The Node is created according to the model definition.

Data Type

The dropdown lists all the possible Data types for the selected Model type.

Name

The name of the Node is used to display the Node in the Forge's address space.

Namespace

Select in which Namespace the Node will be created. Uses Forge's default Namespace as default.

Id Type

Select which type of Identifier will be used.

Identifier

Auto-generated value can be changed. The Identifier needs to be unique in the address space and follow the Id Type.

3.1.4. Add Reference

Add Reference feature allows you to add additional references between Nodes. So when a client browses a Node, they receive the Node's references but also they receive an reference to additional Node, which does not exist in the original structure. You can use this to build additional reference paths that the original data sources do not provide.

Create additional reference using Add Reference feature. In the configuration of additional reference you need to select a Node from left column for which you add the new reference. Then you select a Node from right column to which the reference is pointed. In the end you going to find the right column Node under the left column Node.

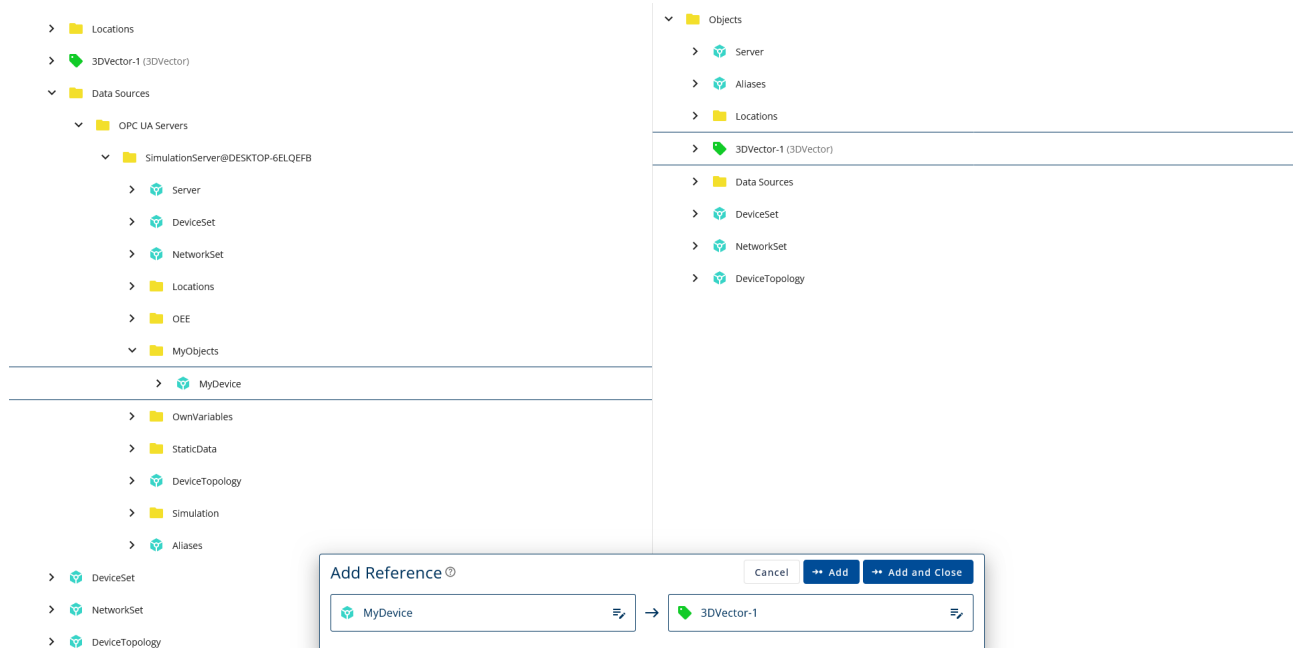


Figure 20. The form for adding a Reference.



Forge's configuration or modifications will not change anything in the underlying OPC UA servers, allowing them to remain intact and serve any third-party OPC UA clients as necessary.

3.1.5. Remove Node

You can remove a Node from the address space model by selecting the Node and clicking the Remove Node button. Then click Confirm and the Node and the structure underneath it will be removed. Some of the Nodes are protected due to Model types.

3.1.6. Mapping

In Forge, you can manipulate Nodes' values by mapping. This means that a Node can present a value, which is received from another Node. Select the Node whose value you want to map and click on Mapping button to open the mapping configuration. First, you will see the supported attributes for mapping. Use the mapping-icon on the right of an attribute to edit the mapping. There are three types of mappings to choose from:

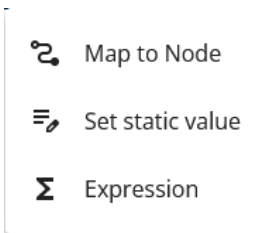


Figure 21. You can map a value from another Node, give a static value or create a complex expression.

Map to Node

Map the value from another Node.

Set static Value

Write a static value to the attribute.

Expression

Create an Expression to the attribute. Only available for value attributes. For more details, see [Expression](#).

Available attributes for mapping are shown in the figure below.

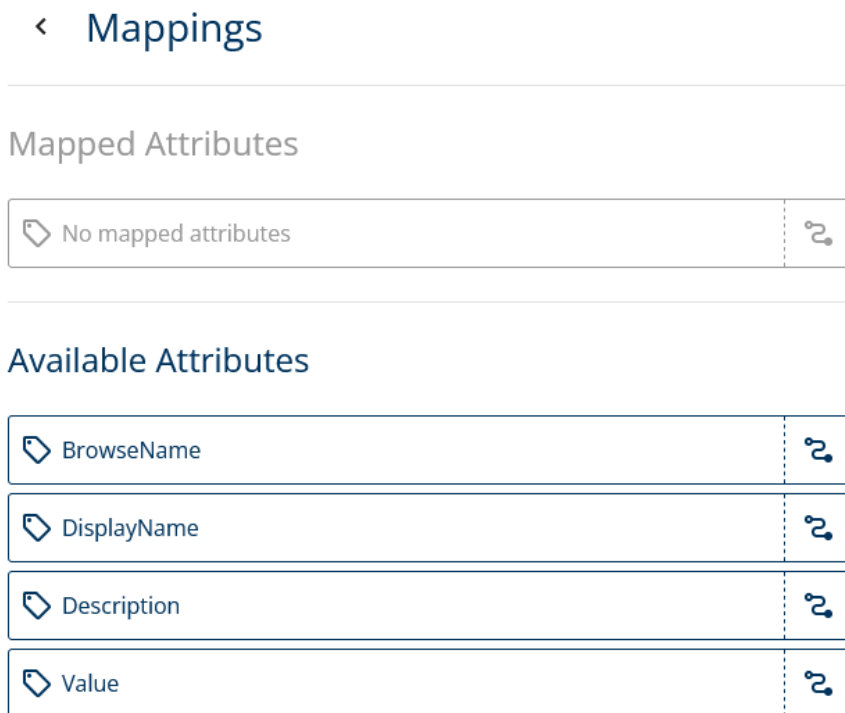


Figure 22. List of available attributes. 'No attributes to map' is shown for available attributes if mapping is not possible for the selected Node.

If there are no attributes for mapping, the list is empty and mapping is not possible for the selected Node.

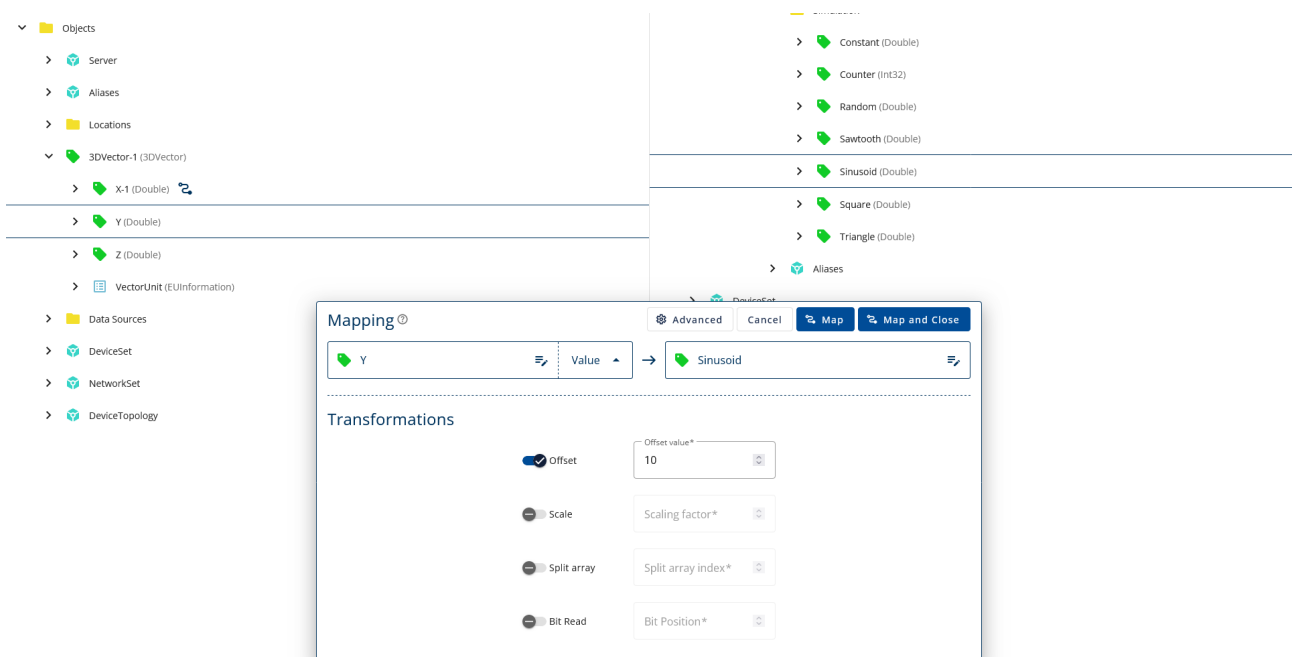


Figure 23. Map a value from another Node. With Advanced settings, you can transform the mapping value.

Mapping allows you to connect Nodes so that when a client accesses an attribute from the left column Node, they receive the value from the right column Node. The data is not synchronized actively. For active value exchange, check [Data Exchange](#).

By selecting the Advanced settings, you can transform the mapped value. Below are the available mapping transformations:

- Offset: Adds the offset value to the mapped value.
- Scale: Multiplies the mapped value by the scaling factor.
- Split Array: Selects the scalar value from the mapped array located in the given index. The indexing starts from 0.
- Bit Read: Reads if the selected bit from the mapped value is 0 or 1. The read starts from the least significant bit.
- Transformation Dictionary: Converts source values into target values according to selected [Transforms](#).



When using both Offset and Scale, the offset is added before scaling.

< Mappings

Mapped Attributes

BrowseName	STATIC MAPPING	
"X-1"		
Description	STATIC MAPPING	
"value for x-axis"		

Available Attributes

DisplayName	
Value	
Value (Double)	
Set static value*	
<input type="text" value="-4"/>	
<input type="button" value="Save"/>	<input type="button" value="Cancel"/>

Figure 24. You can also write a static value to the Node.

To manage mappings, you need to select the Node whose mapping you want to edit and then choose the Mapping from the side panel.

< Mappings

Mapped Attributes

BrowseName	STATIC MAPPING	
"X-1"		
Description	STATIC MAPPING	
"value for x-axis"		
Value	NODE MAPPING	
Sinusoid (nsu=urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer/http://www.prosysopc.com/OPCUA/SimulationNodes;/i=1005)		
OFFSET(10)		

Available Attributes

DisplayName	
-------------	--

Figure 25. You can edit and remove mappings by selecting the Mapping from the side panel.

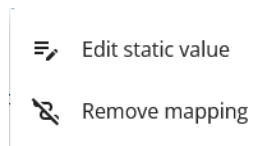


Figure 26. Remove or edit the Mapping.

3.2. Expression

The Expression feature allows users to implement complex logic using [JEXL](#) (Java Expression Language) directly within Nodes. This functionality enables the creation of reusable templates that can be applied for multiple Nodes, enhancing efficiency. By leveraging the Expression module, users can define rules to process and transform data, enabling more intelligent and responsive integration solutions.

Start by selecting a variable Node and then Expression. Expressions can be also accessed through Mapping. You can manage Expression templates from Expressions view. Access this view from the Advanced Menu with [Expressions](#).

< Expression

Templates

Template

Template
sum

Expression

Expression
a + b

Variables

a

Type*
Node USE AS TRIGGER

Node*
nsu=http://www.prosysopc.com/OPCUA/Forge;s=Pass

b

Type*
Node USE AS TRIGGER

Node*
nsu=http://www.prosysopc.com/OPCUA/Forge;s=count

Additional Triggers

+

Trigger 1

Node*
nsu=http://www.prosysopc.com/OPCUA/Forge;s=Timer

Trigger Filters

#1 Condition*
CHANGE Value

Save

Cancel

Figure 27. Create complex logic with the JEXL.

Template

Select a template if desired. Templates must be created beforehand to use this option.

Expression

Enter the JEXL expression to define the logic.

Add To Template

Check this box to add the current expression to be a template.

Template Name

Provide a name for the template if creating a new one.

Type

Select the type of variable used in the expression. Options include Node, String, Integer, Double, Float, Long, Boolean, Byte, Short, and Character. While you can write string, integer, and other values directly in the expression, specifying types allows for reusable templates with different values.

Use As Trigger

Indicate if the value should act as a trigger. The trigger condition is a change.

Node(Variable)

Specify the Node or value associated with the variable.

Node(Trigger)

If additional triggers are needed, press the plus icon next to Additional Trigger, and specify the Node to be used.

Condition Type

Define the condition under which the expression should evaluate. Options include Change, Increment, Equals, Greater Than, Less Than, High Limit, and Low Limit.

Value

Enter the value to be used in the condition evaluation.

3.2.1. Browse Nodes

The Browse Nodes dialog is used throughout the web UI to select Nodes from Forge's address space. Click the arrow-icon at the beginning of each row to browse, and the text to select a Node.

Browse nodes

◀ Previous

Root > Data Sources > OPC UA Servers > SimulationServer@DESKTOP-6ELQEFB > Simulation >

- ▼ Constant
- ▼ Counter
- ▼ **Random**
- ▼ Sawtooth
- ▼ Sinusoid
- ▼ Square
- ▼ Triangle

Selected item:

Name: Random
 Node ID: nsu=urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer/http://www.prosysopc.com/OPCUA/SimulationNodes/i=1003
 Class: Variable

Apply Cancel

Figure 28. Browse Nodes dialog is used throughout the web UI to select Nodes from the Forge's address space.

3.3. Namespaces

Namespaces is OPC UA term. Basically Namespaces are dividing the address space to groups to organize and to avoid NodeId conflicts. In Namespaces view you can find a list of the Namespaces. The main feature of the Namespaces view is the Node Set import.

OPC UA Forge | OPC UA Server | Data Sources

Address Space | **Namespaces** | Certificates | Server Settings | Reverse Connections

Imported NodeSets

[Import NodeSet](#)

Name	NamespaceUri		
Machinery Basic Building Blocks_749922466.xml	http://opcfoundation.org/UA/Machinery/	↑	🗑️
kitchen-tools.xml	http://opcfoundation.org/UA/CommercialKitchenEquipment/	↑	🗑️

Namespaces

[Add namespace](#)

Index ↑	NamespaceUri	Type	Description
2	http://www.prosysopc.com/OPCUA/Forge	Internal	🗑️
3	http://opcfoundation.org/UA/Id/	Internal	🗑️
4	http://www.prosysopc.com/OPCUA/Forge/EventManager	Internal	🗑️
5	http://www.prosysopc.com/OPCUA/Forge/Modbus	Internal	🗑️
6	http://www.prosysopc.com/OPCUA/Forge/Modbus/Data	Internal	🗑️
7	http://www.prosysopc.com/OPCUA/Forge/PubSub	Internal	🗑️
8	http://www.prosysopc.com/OPCUA/Forge/Ads	Internal	🗑️
9	http://www.prosysopc.com/OPCUA/Forge/S7Comm	Internal	🗑️
20	urn:DESKTOP-6ELQEFB.mshome.net:OPCUA:SimulationServer/http://opcfoundation.org/UA/	Remote	🗑️

Figure 29. Namespaces has two features, NodeSets and Namespaces.

3.3.1. NodeSets

Forge enables you to load standard and custom NodeSets, allowing you to define and incorporate specific data structures and types into your system. This feature ensures that your data model accurately represents your industrial assets.

This feature allows you to import OPC UA models and model instances by loading NodeSet files into Forge. Imported NodeSets will be listed in the Imported NodeSets table. You can add new ones by clicking the Import NodeSet button, and remove them by clicking the remove icon at the end of the corresponding row.

Import NodeSet

Select a NodeSet2 file to import

kitchen-tools.xml
📎

Import mode

Instances only

Types and instances

Both types and instances are loaded from the NodeSet. **Node instances imported in this mode have their editing limited to mapping features.** New instances created from the types can be modified freely.

Description*

Kitchen Tools

Load file
X Clear

Figure 30. NodeSets are an efficient way to bring hierarchical components to Forge.

Instances only

Only Node instances are imported and converted to Forge Nodes. The Nodes imported in this mode can be freely edited after importing. Only NodeSet files having instance declarations can be loaded.

Types and instances

Both types and instances are loaded from the NodeSet file. Node instances imported in this mode have their editing limited to mapping features. New instances created from the types can be modified freely.

Description

User-defined short description of the NodeSet.

Imported NodeSets
📄 Import NodeSet

Name	NamespaceUris		
Machinery Basic Building Blocks_749922466.xml	http://opcfoundation.org/UA/Machinery/	↑	🗑️
kitchen-tools.xml	http://opcfoundation.org/UA/CommercialKitchenEquipment/	↑	🗑️

Figure 31. Imported NodeSets are shown in the list.

You can update the NodeSets in Forge by clicking the upload icon and uploading the updated version of

the NodeSet file.

Update NodeSet

Machinery Basic Building Blocks_749922466.xml
📎

Update Name

Name
 Machinery Basic Building Blocks_749922466.xml

i
Updating NodeSets requires a restart.

Update
Cancel

Figure 32. Update your NodeSet file without the need to fully reconfigure the address space.



Removing a NodeSet will remove all the Nodes which are instances from NodeSet or Nodes that are using the respective NodeSet types.

3.3.2. Namespaces Table

Forge provides the ability to add and manage Namespaces. This ensures you an ability to organize Nodes, preventing naming conflicts with Nodelds and enhancing clarity in your data model.

This feature displays all Namespaces in Forge’s address space and lets you add new ones. The table shown in [Figure 33](#) shows the index and NamespaceUri of each namespace, and you can remove them by clicking the remove-icon at the end of the corresponding row. Notice, some of the Namespaces are locked and can’t be removed. The type can be Internal for Namespaces created in Forge, Remote for Namespaces coming from Data Sources and NodeSet for Namespaces imported from Node Sets.

New namespace

NamespaceUri*

Description

Add
Cancel

Figure 33. Add new Namespaces to Forge’s server.

NamespaceUri

Identifies this component. Give input in valid URI format.

Description

Describes the namespace.

Namespaces

+ Add namespace

Index ↑	NamespaceUri	Type	Description
2	http://www.prosysopc.com/OPCUA/Forge	Internal	
3	http://opcfoundation.org/UA/DI/	NodeSet	
4	http://www.prosysopc.com/OPCUA/Forge/EventManager	Internal	
5	http://www.prosysopc.com/OPCUA/Forge/Modbus	Internal	
6	http://www.prosysopc.com/OPCUA/Forge/Modbus/Data	Internal	
7	http://www.prosysopc.com/OPCUA/Forge/PubSub	Internal	
20	urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer/http://opcfoundation.org/UA/	Remote	
21	urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer/urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer	Remote	
22	urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer/http://www.prosysopc.com/OPCUA/SimulationServer/	Remote	
23	urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer/http://www.prosysopc.com/OPCUA/SimulationNodes/	Remote	
24	urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer/http://www.prosysopc.com/OPCUA/SimulationNodes/SimulationConfiguration/	Remote	
25	urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer/http://www.prosysopc.com/OPCUA/StaticNodes	Remote	
26	urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer/http://www.prosysopc.com/OPCUA/SampleAddressSpace	Remote	
27	urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer/http://prosysopc.com/oeef/	Remote	
28	urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer/http://opcfoundation.org/UA/	Remote	

Figure 34. All Namespaces are listed here with additional information.

3.4. Certificates

This view is where you manage certificates. The top of the view displays Forge’s own certificate, while other certificates are listed under Trusted and Rejected. To move a certificate from one list to another, simply click the trust icon or reject icon at the end of the corresponding row. It is also allowed to add new certificates. Additionally, you can download and upload the own certificate and add new certificates manually from the web UI.

OPC UA Certificates

+ Add Certificate

Forge Certificate
VALID UNTIL: 2034-02-18
^

NAME:	Forge@DESKTOP-6ELQEFB
APPLICATION URI:	urn:DESKTOP-6ELQEFB:OPCUA:Forge
VALID FROM:	2024-02-21 16:22
VALID TO:	2034-02-18 17:22
SUBJECT:	DC=DESKTOP-6ELQEFB,O=Prosys OPC,CN=Forge@DESKTOP-6ELQEFB
ISSUER:	DC=DESKTOP-6ELQEFB,O=Prosys OPC,CN=Forge@DESKTOP-6ELQEFB
SIGNED BY:	Self Signed
SIGNATURE ALGORITHM:	SHA256withRSA
SERIAL NUMBER:	18dcc4343f1
THUMBPRINT:	0xdb4d8a9661b374ae8d4f32c1550c4ec965c9a958

Download Certificate
Upload Certificate

Trusted Certificates

NAME	VALID TO	APPLICATION URI	REJECT
UaBrowser@DESKTOP-6ELQEFB	2032-02-05 16:44	urn:DESKTOP-6ELQEFB:ProsysOPC:UaBrowser	✖
SimulationServer@DESKTOP-6ELQEFB	2034-01-27 14:18	urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer	✖
KEPServerEX/UA Server	2026-08-27 09:52	urn:DESKTOP-6ELQEFB:Kepware.KEPServerEX.V6:UA%20Server	✖

Rejected Certificates

NAME	VALID TO	APPLICATION URI	TRUST
ModbusServer@DESKTOP-6ELQEFB	2033-05-02 10:15	urn:DESKTOP-6ELQEFB:ProsysOPC:ModbusServer	🔒

Figure 35. See all the certificates here.

3.4.1. Forge's Certificate

Forge's own certificate can be seen from the Certificates view. The user has possibility to download the certificate or upload a new certificate for Forge.

Forge Certificate
VALID UNTIL: 2034-02-18
^

NAME:	Forge@DESKTOP-6ELQEFB
APPLICATION URI:	urn:DESKTOP-6ELQEFB:OPCUA:Forge
VALID FROM:	2024-02-21 16:22
VALID TO:	2034-02-18 17:22
SUBJECT:	DC=DESKTOP-6ELQEFB,O=Prosys OPC,CN=Forge@DESKTOP-6ELQEFB
ISSUER:	DC=DESKTOP-6ELQEFB,O=Prosys OPC,CN=Forge@DESKTOP-6ELQEFB
SIGNED BY:	Self Signed
SIGNATURE ALGORITHM:	SHA256withRSA
SERIAL NUMBER:	18dcc4343f1
THUMBPRINT:	0xdb4d8a9661b374ae8d4f32c1550c4ec965c9a958

Download Certificate
Upload Certificate

Figure 36. See all the details of Forge's certificate.

You will need the certificate file and private key file to upload a new certificate.

Set Own Certificate

Certificate File
Forge@DESKTOP-6ELQEFB_2048.der

Private Key File
Forge@DESKTOP-6ELQEFB_2048.pem

Save

Figure 37. Upload a new certificate for Forge.

Certificate File

Specify the file containing the SSL/TLS certificate to be uploaded for secure communication.

Private Key File

Provide the file containing the private key corresponding to the SSL/TLS certificate for secure communication.

3.4.2. Manage Certificates

When OPC UA client tries to connect to Forge using security mode, it will send its certificate to Forge. The certificate will go to Rejected list, and before the connection is possible to establish, the certificate needs to be trusted.

Trusted Certificates			
NAME	VALID TO	APPLICATION URI	REJECT
UaBrowser@DESKTOP-6ELQEFB	2/5/32, 4:44 PM	urn:DESKTOP-6ELQEFB:ProsysOPC:UaBrowser	
SimulationServer@DESKTOP-6ELQEFB	10/2/33, 1:50 PM	urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer	
KEPServerEX/UA Server	8/27/26, 9:52 AM	urn:DESKTOP-6ELQEFB:Kepware.KEPServerEX.V6:UA%20Server	
ModbusServer@DESKTOP-6ELQEFB	5/2/33, 10:15 AM	urn:DESKTOP-6ELQEFB:ProsysOPC:ModbusServer	


Figure 38. Trusted certificates can be rejected by clicking the reject icon on the right of each row.

Rejected Certificates			
NAME	VALID TO	APPLICATION URI	TRUST
UaExpert@DESKTOP-6ELQEFB	5/1/28, 4:25 PM	urn:DESKTOP-6ELQEFB:UnifiedAutomation:UaExpert	

Figure 39. Rejected certificates can be trusted by clicking the trust icon on the right of each row.

Add certificates manually from the web UI. Click the Add Certificate button to upload certificate to Forge.

Add Certificate

Certificate Validation*
AcceptPermanently

Save

Figure 40. You can add certificates from the web UI.

Select File

Choose the file containing the certificate that you wish to import into the system. This file typically has a .cer, .crt, or .pem extension.

Certificate Validation

Specify the validation action to be taken for the imported certificate.

- AcceptOnce: Accept the certificate for this session only, prompting for validation again in subsequent sessions.
- AcceptPermanently: Accept the certificate permanently, storing it for future sessions without requiring validation again.
- Reject: Reject the certificate, preventing it from being used for authentication or secure communication.

3.5. Server Settings

These configurations allow you to change the settings of Forge server. Manage the available connection modes for OPC UA clients to fine-tune the security of your system. Enable Reverse Connection for advanced connections in high security environments. Note that Forge must be restarted for these changes to take effect.

OPC UA Server Configuration

Endpoint Configuration

Server address

Port* Path

Security Modes

None Sign Sign&Encrypt

Security Policies

Basic128Rsa15 Basic256 Basic256Sha256 Aes128Sha256RsaOaep Aes256Sha256RsaPss

User Authentication

Anonymous Username and Password Certificate

Bind Addresses

IPv6

Discovery Server

Register at Local Discovery Server

Figure 41. Manage Forge's server settings.

Server Port

In which port the Forge server is running.

Security Modes

Select which security modes are supported. These options will affect which security modes can be used when the client connects to the Forge server.

Security Policies

Select which security policies are supported. These options will affect how the authentication is done when the client tries to connect to the Forge server.

User Authentication

Choose the method of user authentication supported by the OPC UA server.

Bind Addresses

Select whether IPv6 is supported.

Register to

By enabling Local Discovery Server, you allow clients to find Forge server from the discovery server.



Note that some of the Security Policies (Basic128Rsa15 and Basic256) have already been deprecated in OPC UA version 1.04, but to enable interoperability with all client applications, it may be necessary to keep them enabled.

3.6. Reverse Connection

Reverse Connection allows you to manage all the reverse connection addresses. With reverse connection, you can establish the connection from the server. This makes it possible to have the connection with a OPC UA client without opening firewalls since the connection is activated by the

server.

OPC UA Reverse Connections + Add Connection

New Reverse connection

Name*
Client-1

Endpoint URL*
opc.tcp://localhost:4840

Save Cancel

Figure 42. Manage the reverse connections.

Name

User-defined name for the reverse connection.

Endpoint URL

Connection endpoint that the connecting client is using. For example, `opc.tcp://<client_ip>:<port>`, where `client_ip` is the IP address of the client application. Ensure that the selected port is available and not already in use by other applications on the client machine.

Reverse connections are listed and connection status can be seen (Green: connection ok, Red: connection lost, Grey: connection status unknown.):

OPC UA Reverse Connections + Add Connection

NAME: Raspi_Forge ENDPOINT: opc.tcp://10.50.100.228:55555	✖
NAME: ReverseConnection ENDPOINT: opc.tcp://DESKTOP-6ELQEFB:55556	✖

Figure 43. Green: connection ok, Red: connection lost, Grey: connection status unknown.

4. Data Sources

Data Sources view allows users configure connections to access data from different sources. Users can make connections to OPC UA servers and configure OPC UA subscribers to receive data.

4.1. OPC UA Servers

Forge can connect to numerous OPC UA servers as an OPC UA client, enabling server aggregation and efficient data exchange across systems.

You can use this view to manage and monitor connections to underlying servers. To add a new connection, click the Add Connection button. To remove a connection, click the remove icon at the end of the corresponding row.

4.1.1. Default Connection

The default connection occurs when Forge connects to another OPC UA server as a client.

OPC UA Connections ↕ Import + Add Reverse Connection + Add Connection

New OPC UA Connection

Endpoint Discovery

Discovery URI
 🔍

Available Endpoints
 Aes256Sha256RsaPss SignAndEncrypt ✔

Endpoint Configuration

Name* Connection Mode

Endpoint URI*

Security Policy* ✔ Security Mode* Authentication Type*

Certificate Validation Overrides

Disable ApplicationUri Check Disable CertificateTime Check Disable KeyUsage Check
 Disable CaHasRevocationList Check Disable CertificateRevocation Check Accept All Certificates

Namespace Aggregation Configuration

Namespace Prefix Strategy*

Test connection ☁ Add Cancel

Figure 44. You must test the connection successfully before adding it.

Discovery URI

Write the connection address from which you want to fetch the endpoints. Click the Search icon to search for endpoints. (for example, `opc.tcp://localhost:53530/OPCUA/SimulationServer`)

Select Endpoint

You can choose the combination of an endpoint and Security Mode. Select one option by clicking the row. The most secure option will be selected automatically.

Endpoint Name

Give a name for the connection.

Connection Mode

Choose the mode for the connections. Options include:

- Enabled: The connection is active, and communication with the server is established.
- Disabled: The connection is not established, but the configurations are saved for future use.
- Simulated: Attribute mappings are not directed to a server with Simulated Connection Mode but they are read from the node having the mapping.

Endpoint URI

Full URI for the selected endpoint.

Authentication

Enable or disable authentication with the connection. If the authentication is enabled, fill the credentials as required.

Test Connection

Shows whether the connection is OK.

Namespace Prefix Strategy

Decide which strategy is used to generate the name for the aggregated namespaces. Namespace is constructed from the <prefix> combined with the namespace name in the server that you are connecting to (e.g. urn:<hostname>:OPCUA:SimulationServer/http://opcfoundation.org/UA/Machinery/). Note that the names of the namespaces shall be different.

- ApplicationUri: Use the applicationUri as the prefix value. This is the default option.
- EndpointName: Use the name of the endpoint as the prefix value.
- Custom: Give your custom prefix value.

4.1.2. Reverse Connection

Forge can make reverse connections to other OPC UA servers. To add a new reverse connection, click the Add Reverse Connection button. Check the details for the fields above.

To initiate the reverse connection with Forge to an OPC UA server (Target server), you need to have configured a reverse connection endpoint in the Target server. This means that you have configured the Target server to listen to a certain port for reverse connections. After the reverse connection endpoint is configured in the Target server, you can configure Forge to try a reverse connection to that endpoint with the following form.

OPC UA Connections

⬇ Import
+ Add Reverse Connection
+ Add Connection

New OPC UA Reverse Connection

Endpoint Configuration

Name* Connection Mode

Endpoint prefix* Hostname* Port*

Security Policy* Security Mode* Authentication Type*

Certificate Validation Overrides

Disable ApplicationUri Check
 Disable CertificateTime Check
 Disable KeyUsage Check
 Disable CaHasRevocationList Check
 Disable CertificateRevocation Check
 Accept All Certificates

Namespace Aggregation Configuration

Namespace Prefix Strategy*

Figure 45. Reverse connection can bypass problems with firewalls.

Endpoint Name

Enter a descriptive name for the reverse endpoint connection.

Connection Mode

Choose the mode for the connections. Options include:

- Enabled: The connection is active, and communication with the server is established.
- Disabled: The connection is not established, but the configurations are saved for future use.
- Simulated: Attribute mappings are not directed to a server with Simulated Connection Mode but they are read from the node having the mapping.

Endpoint Prefix

Select a suitable prefix for your case.

Hostname

Enter the hostname or IP address of the Forge server.

Port

Use the port configured for the Target server to listen. Ensure the port is available on the Forge computer.

Security Policy

Choose a security policy supported by the Target server.

Security Mode

Select a security mode supported by the Target server.

Authentication

Specify the authentication method for the connection, if applicable.

Once you have added connections, you can see a list of the connections and monitor the status of the connections.

OPC UA Connections Import Add Connection

NAME: KEPServerEX/UA@DESKTOP-6ELQEFB ENDPOINT URI: opc.tcp://127.0.0.1:49320 SECURITY POLICY: Basic256Sha256 SECURITY MODE: SignAndEncrypt AUTHENTICATION: Anonymous	
NAME: ModbusServer@DESKTOP-6ELQEFB ENDPOINT URI: opc.tcp://DESKTOP-6ELQEFB:53510/ModbusServer SECURITY POLICY: Aes256Sha256RsaPss SECURITY MODE: SignAndEncrypt AUTHENTICATION: Anonymous	
NAME: SimulationServer@DESKTOP-6ELQEFB ENDPOINT URI: opc.tcp://DESKTOP-6ELQEFB:53530/OPCUA/SimulationServer SECURITY POLICY: Aes256Sha256RsaPss SECURITY MODE: SignAndEncrypt AUTHENTICATION: Anonymous	

Figure 46. See the connection status from the list. Green: connection ok, Red: connection lost, Grey: connection status unknown.

4.1.3. Import Connections

You can import connection configurations with a CSV file. Click the Import button and download the template for the CSV file. Fill the connection details into the template file and the upload it to Forge from the same dialog.

Import Connection

To import a connection, the uploaded .csv file with the connection data needs to be formatted according to the template file, which can be downloaded below.

Template

Import .csv

No file selected.

Import Connection
Cancel

Figure 47. Import multiple connection configurations seamlessly using a CSV file.

4.2. OPC UA Subscribers

Receive data with OPC UA subscriber. Connect Forge to a network or to MQTT broker as subscriber and access the data flowing through. Forge will generate Nodes from the data and then you can connect those values for further use for example in Mapping or Data Exchange.

To add new a Subscriber, press the Add Subscriber button, which opens a form for adding the new Subscriber. The form will depend on the used protocol.

4.2.1. OPC UA over MQTT Subscribers

Configure OPC UA over MQTT subscribers.

OPC UA Subscribers + Add Subscriber

New Subscriber connection

Name*
MQTT subscriber

Protocol*
mqtt://

Hostname / IP Address*
localhost

Port*
1883

Subscriber Client ID
ClientID*
urn:DESKTOP-6ELQEFB:OPCUA:Forge

Authentication
Username Password

Save Cancel

Figure 48. Receive data to Forge with OPC UA over MQTT subscriber.

Name

User-defined name for the subscriber.

Protocol

Select mqtt:// or secure mode mqtt://.

Hostname/IP Address

The address of the MQTT broker you are connecting to.

Port

Write the port number in which the MQTT broker is running.

Client ID

The default value is given. For other client ID change the value.

Username

For authentication you may fill the username for the MQTT broker.

Password

For authentication you may fill the password for the MQTT broker.

When using the secured MQTT protocol, MQTTS, the form will have two additional fields.

OPC UA Subscribers

+ Add Subscriber

New Subscriber connection

Name*
MQTT subscriber

Protocol*
mqtt://

Hostname / IP Address*
localhost

Port*
1883

Subscriber Client ID
Clientid*
urn:DESKTOP-6ELQEFB:OPCUA:Forge

Authentication
Username Password

Certificates
Client Certificate Certificate Authority Chain

Save Cancel

Figure 49. Add security to OPC UA over MQTT subscriber by using MQTTS protocol.

Client Certificate

Forge's Client Certificate for more secure communication.

Certificate Authority Chain

Fill the Certificate Authority Chain.

4.2.2. OPC UA over UDP Subscribers

For OPC UA over UDP subscribers the following data needs to be filled.

OPC UA Subscribers

+ Add Subscriber

New Subscriber connection

Name*
UDP Subscriber

Protocol*
opc.udp://

Hostname / IP Address*
224.0.5.1

Port*
4840

Network Interface
Network Interface*
Bluetooth Device (Personal Area Network)

Save Cancel

Figure 50. Fetch data to Forge with OPC UA over UDP Subscriber.

Name

User-defined name for the subscriber.

Protocol

Select opc.udp://.

Hostname/IP Address

The address of the UDP network.

Port

Which port is used in the communication.

Network Interface

Select which network interface is used in the communication.

4.3. S7COMM Devices

S7COMM (S7 Communication) is a Siemens proprietary protocol that runs between programmable logic controllers (PLCs) of the Siemens S7COMM family. Configure a connection to receive data from these devices.

S7COMM Devices
+ Add Device

New S7COMM Device

General settings ENABLED

Name* Controller Type

TCP settings

Address Port

Main CPU settings

Rack Slot TSAP

Secondary CPU settings

Rack Slot

Client settings

Rack Slot TSAP

Transport settings Ping

PDU Size Read Timeout Retry Time Ping Interval

Advanced settings

Figure 51. Connect Forge to S7COMM devices.

Enabled

Toggle to enable or disable the S7COMM configuration.

Name

Enter a descriptive name for the S7COMM configuration.

Controller Type

You may select the controller type to avoid problems during the remote device identification process when connecting to the remote device. The options are S7-300, S7-400, S7-1200 or S7-1500.

TCP Address

Enter the physical IP address/hostname of the PLC or CP's.

TCP Port

Specify the TCP port number used for communication with the S7COMM controller.

Main CPU Rack

Enter the rack value for the main CPU (PLC).

Main CPU Slot

Enter the slot value for the main CPU (PLC).

Main CPU TSAP

Enter the TSAP (Transport Service Access Point) of the main CPU.

Secondary CPU Rack

Enter the rack value for the secondary CPU (PLC).

Secondary CPU Slot

Enter the slot value for the secondary CPU (PLC).

Client Rack

Enter the rack value for the client.

Client Slot

Enter the slot value for the client.

Client TSAP

Enter the TSAP (Transport Service Access Point) of the client.

Ping

Toggle to activate or deactivate the ping functionality for the connection. Enabling this option ensures that the TCP channel remains open, particularly when the application necessitates sampling times exceeding the configured "read-timeout" duration, preventing unnecessary closure of the channel.

PDU Size

This parameter determines the maximum size of data packets transmitted to and received from the PLC. During the connection setup, both parties negotiate a mutually acceptable maximum size. If the negotiated size exceeds the specified value, the driver automatically divides large requests to ensure they stay within this limit.

Read Timeout

This sets the maximum waiting time for reading on the TCP channel. If no traffic is detected within this duration, it's assumed that the connection with the counterpart is lost and needs to be reestablished. If the channel is closed, a failover is initiated if a secondary channel exists, or it's awaited for automatic restoration, typically every 4 seconds.

Retry Time

This parameter defines the time interval between retries for failed communication attempts. If the channel remains inactive, a safe stop of the EventLoop ensures no additional tasks are generated, preserving system resources.

Ping Interval (s)

Specify the interval, in seconds, for sending ping requests. This duration is typically set based on developer experience but is generally recommended to be half the read-timeout duration to ensure timely responsiveness and connection stability.

4.4. ADS Devices

The Automation Device Specification (ADS) is the communication protocol of TwinCAT. It enables the data exchange and the control of TwinCAT systems. Configure connections to TwinCAT3 devices and enable communication with these devices.

ADS devices + Add Device

New ADS device

General settings ENABLED

Name*

TCP settings

Address* Port*

ADS settings

Source AMS Net Id* Source AMS Port*

Target AMS Net Id* Target AMS Port*

Figure 52. Connect Forge to ADS devices.

Enabled

Toggle to enable or disable the device Specification (ADS) configuration.

Name

Enter a descriptive name for the ADS configuration.

Address

Specify the IP address or hostname of the automation device.

Port

Specify the port number used for communication with the automation device.

Source AMS Net Id

Enter the Net Id of the source Automation Management System (AMS). The Net Id, or Network Identifier, is a unique identifier assigned to each device on a network.

Source AMS Port

Specify the port used by the source AMS.

Target AMS Net Id

Enter the Net Id of the target AMS. The Net Id, or Network Identifier, is a unique identifier assigned to each device on a network.

Target AMS Port

Specify the port used by the target AMS.

4.5. EtherNet/IP Devices

The EtherNet/IP Device Connection configuration allows you to establish communication with EtherNet/IP devices within your network, enabling data exchange with connected devices.

EtherNet/IP Devices + Add Device

New EtherNet/IP Device

General settings ENABLED

Name* Controller Type

TCP settings

Address* Port*

EtherNet/IP settings

Communication Path

Backplane Slot USE BIG-ENDIAN UNCONNECTED OPERATIONS

Figure 53. Connect Forge to EtherNet/ip devices.

Enabled

Toggle to enable or disable the configuration.

Name

Enter a descriptive name for the configuration.

Controller Type

Specify the type of controller for the connection. Options include Generic and Logix.

- Generic: Used for devices that adhere to the EtherNet/IP protocol but are not part of the Allen-Bradley Logix family of controllers.
- Logix: Designed specifically for Allen-Bradley Logix controllers, such as CompactLogix and ControlLogix.

Address

Enter the IP address or hostname of the device.

Port

Define the port number used for communication. Typically, the default is 44818 for EtherNet/IP devices.

Communication Path

Specify the communication path within the device network for addressing sub-devices or modules (only Logix).

Backplane

Indicate the backplane identifier if required by the device's architecture.

Slot

Define the slot number where the controller is located.

Use Big Endian

Enable this option if the device uses big-endian byte order for data encoding.

Unconnected Operations

Toggle to force the driver to use unconnected requests.

5. Redundancy

Forge's Redundancy Module allows you to set up backup data sources, like OPC UA servers or other devices, to ensure system reliability. When you configure these data sources with redundant namespaces, Forge will automatically switch to a backup source if the primary one becomes unavailable. This ensures continuous access to your data without interruptions, even if one of the data sources fails.

Since Forge handles the redundant OPC UA Servers, you no longer need to handle redundancy with your own OPC UA client. You only need your client to connect to Forge which ensures an established connection to an available data source. Make sure that the redundant namespaces have the same Identifiers in the NodeIds of corresponding nodes.

Namespace Redundancy Groups + Add Group

New Namespace Redundancy Group

General settings

Name*

Namespaces

Namespace +

#1	urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer/http://www.prosysopc.com/OPCUA/SimulationNodes/	✕
#2	urn:UADEMO.prosysopc.com:OPCUA:SimulationServer/http://www.prosysopc.com/OPCUA/SimulationNodes/	✕

Figure 54. Configure namespaces to be redundant.

Name

Enter a descriptive name for the namespace redundancy group configuration.

Namespace

Select the namespace from the list that you want to add to the group. Then press the + sign to add the selected namespace to the list. The minimum number of namespaces needed is two (2).

6. OPC UA Publisher

The OPC UA publisher configuration empowers users to establish and customize the publishing side of OPC UA Pub Sub communication. It provides a streamlined interface for defining parameters such as connection details, data sets, and message formats, enabling efficient and secure transmission of variable and event data from OPC UA servers to subscribers.

Pub Sub can utilize communication through local network protocols such as UDP and Ethernet-APL. Alternatively, communication can occur through broker-based protocols such as MQTT.

Pub Sub messages can be formatted as UADP (Unified Architecture Datagram Protocol) or JSON, which allows for interoperability between systems that use different formats.

6.1. Connections

OPC UA Publisher Connections configuration allows users to define the communication settings for Pub Sub communication. With three available options—MQTT, MQTTS, and UDP—users can select the most suitable protocol for their specific use case.

6.1.1. OPC UA over MQTT

Configure OPC UA publisher connection over MQTT.

OPC UA Publisher Connections
+ Add Connection

New OPC UA Publisher Connection

General settings ENABLED

Name
Internal Broker

Publisher ID
String:urn:DESKTOP-6ELQEFB:OPCUA:Forge ↶

Connection address settings

Protocol* mqtt:// Hostname / IP* localhost Port* 1883

MQTT settings

Client ID*
urn:DESKTOP-6ELQEFB:OPCUA:Forge:Publisher ↶

Username admin Password •••••

MQTT payload format

Payload Format
JSON

Add
Cancel

Figure 55. Forge has internal MQTT broker which can be used for OPC UA publisher.

Enabled

Toggle to enable or disable the connection.

Name

Provide a descriptive name for the configuration to identify it within the application.

Publisher ID

Specify the identifier for the publisher associated with the connection. Default Publisher ID is recommended.

Protocol

Select MQTT as the communication protocol for the connection. Other options include MQTT, MQTTS, and UDP.

Hostname / IP

Enter the hostname or IP address of the MQTT broker.

Port

Specify the port number used for communication with the MQTT broker. The default port for MQTT is often 1883.

Client ID

Provide the client identifier for the connection. Default Client ID is recommended.

Username

Enter the username required for authentication (if applicable).

Password

Provide the password required for authentication (if applicable).

Payload Format

Define the format for the payload of the messages exchanged over the connection.

- JSON: (JavaScript Object Notation) is a lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate.
- UADP: (User Datagram Protocol for Adaptive Data Publication) format, which is binary encoded. UADP is optimized for efficient transmission of large volumes of data over the network, making it suitable for high-performance applications.

6.1.2. OPC UA over MQTTS

MQTTS ensures secure communication over MQTT by adding TLS encryption, enhancing data integrity and confidentiality in industrial and IoT applications.

OPC UA Publisher Connections

+ Add Connection

New OPC UA Publisher Connection

General settings ENABLED

Name
Internal Broker

Publisher ID
String:urn:DESKTOP-6ELQEFB:OPCUA:Forge

Connection address settings

Protocol*
mqtt://

Hostname / IP*
localhost

Port*
8883

MQTT settings

Client ID*
urn:DESKTOP-6ELQEFB:OPCUA:Forge:Publisher

Username
admin

Password
•••••

SSL Certificate settings

Select Certificate

Select PrivateKey

Select CA Certificates

MQTT payload format

Payload Format
JSON

Figure 56. MQTTS is more secure than MQTT.

Enabled

Toggle to enable or disable the connection.

Name

Provide a descriptive name for the configuration to identify it within the application.

Publisher ID

Specify the identifier for the publisher associated with the connection. Default Publisher ID is recommended.

Protocol

Select MQTTS as the communication protocol for the connection. Other options include MQTT, MQTTS, and UDP.

Hostname / IP

Enter the hostname or IP address of the MQTT broker.

Port

Specify the port number used for communication with the MQTT broker. Default port for MQTTS is often 8883.

Client ID

Provide the client identifier for the connection. Default Client ID is recommended.

Username

Enter the username required for authentication (if applicable).

Password

Provide the password required for authentication (if applicable).

Select Certificate

Choose the appropriate certificate for secure communication.

Select PrivateKey

Provide the private key corresponding to the selected certificate.

Select CA Certificates

Choose the CA certificates for certificate chain validation.

Payload Format

Define the format for the payload of the messages exchanged over the connection.

- JSON: (JavaScript Object Notation) is a lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate.
- UADP: (User Datagram Protocol for Adaptive Data Publication) format, which is binary encoded. UADP is optimized for efficient transmission of large volumes of data over the network, making it suitable for high-performance applications.

6.1.3. OPC UA over UDP

Configure OPC UA connection over UDP.

OPC UA Publisher Connections + Add Connection

New OPC UA Publisher Connection

General settings ENABLED

Name
UDP connection

Publisher ID
String:urn:DESKTOP-6ELQEFB:OPCUA:Forge

Connection address settings

Protocol* Hostname / IP* Port*

UDP settings

Network Interface*

Figure 57. UDP protocol supports only UADP payload format.

Enabled

Toggle to enable or disable the connection.

Name

Provide a descriptive name for the configuration to identify it within the application.

Publisher ID

Specify the identifier for the publisher associated with the connection. Default Publisher ID is recommended.

Protocol

Select UDP as the communication protocol for the connection. Other options include MQTT, MQTTS, and UDP.

Hostname / IP

Enter the hostname or IP address of the destination endpoint.

Port

Specify the port number used for communication with the destination endpoint. Default port for UDP is often 4840.

Network Interface

Choose the network interface through which the UDP packets will be sent.

- lo (loopback): The loopback interface allows communication between processes running on the same device. This option is typically used for local testing or communication within the same machine.
- eth8: The Ethernet interface with the specified identifier (eth8). Ethernet interfaces are commonly used for wired network connections.
- wlan3: This option will use the wireless LAN interface with the specified identifier (wlan3). WLAN interfaces, also known as Wi-Fi interfaces, are used for wireless network connections.

6.2. Writer Groups

Writer Group configuration is different between connections with JSON payload format and UADP payload format.

Select Connection

Choose the OPC UA connection to associate with the writer group. If there exist only one connection, this input is hidden.

6.2.1. Writer Groups JSON

Writer Group for JSON format has following configuration form.

OPC UA Writer Groups Select Connection
Internal Broker + Add Writer Group

New OPC UA Writer Group

General settings ENABLED

Name* Publish Interval (ms)* Id*

JSON Network message content mask

Figure 58. Advanced button allows you to modify the message content by selecting the message headers.

Enabled

Toggle to enable or disable the writer group.

Name

Provide a descriptive name for the writer group configuration.

Publish Interval (ms)

Specify the interval, in milliseconds, at which data will be published by the writer group.

Id

Assign a unique identifier within the connection to the writer group.

JSON Network Message Content Mask

Define the content mask for JSON network messages. This parameter specifies which fields will be included in the JSON message payload.

6.2.2. Writer Groups UADP

Writer Group for UADP format has following configuration form.

OPC UA Writer Groups Select Connection
UDP conn-1 + Add Writer Group

New OPC UA Writer Group

General settings ENABLED

Name* Publish Interval (ms)* Id*

UADP Network message content mask

⚙️ Advanced **Save** **Cancel**

Figure 59. Advanced button allows you to modify the message content by selecting the message headers.

Enabled

Toggle to enable or disable the writer group.

Name

Provide a descriptive name for the writer group configuration.

Publish Interval (ms)

Specify the interval, in milliseconds, at which data will be published by the writer group.

Id

Assign a unique identifier within the connection to the writer group.

UADP Network Message Content Mask

Define the content mask for UADP network messages. This parameter specifies which fields will be included in the UADP message payload.

6.3. Data Set Writer

Data set writers are grouped into writer groups. The configuration form depends whether the connection is MQTT or UDP.

6.3.1. Data Set Writer MQTT

Data set writer with MQTT connection has topic configurations.

[Edit Data Set Writer](#)

Data Set settings

Data Set*

Writer settings ENABLED

Name* Id*

MQTT Topic settings DEFAULT

Metadata topic*

Data topic*

Advanced settings

Data Set field content mask* Key Frame Count*

JSON Data Set message content mask*

Figure 60. Configure the message content from Advanced button.

Data Set

Select the data set to associate with the writer configuration.

Enabled

Toggle to enable or disable the data set writer.

Name

Enter a descriptive name for the data set writer configuration.

Default

Toggle to enable or disable the default MQTT topic configuration for the data set.

Metadata Topic

Specify the MQTT topic for publishing metadata related to the data set.

Data Topic

Specify the MQTT topic for publishing data related to the data set.

Data Set Field Content Mask

Define the content mask for the data set fields. This parameter specifies which fields will be

included in the data set.

Key Frame Count

Set greater or equal to 1. With value 1, every message is a Key Frame (every value in the data set is sent). A delta frame (not a key frame) is a message which contains only data which has been changed.

JSON/UADP Data Set Message Content Mask

Define the content mask for JSON or UADP data set messages. This parameter specifies which fields will be included in the JSON or UADP message payload.

6.3.2. Data Set Writer UDP

Data set writer configuration form for UDP connection.

Edit Data Set Writer

Data Set settings

Data Set*

Writer settings ENABLED

Name* Id*

Advanced settings

Data Set field content mask* Key Frame Count*

UADP Data Set message content mask*

Figure 61. Configure the message content from Advanced button.

Data Set

Select the specific data set to associate with the writer configuration.

Enabled

Toggle to enable or disable the data set writer.

Name

Provide a descriptive name for the data set writer configuration.

Data Set Field Content Mask

Define the content mask for the data set fields. This parameter specifies which fields will be included in the data set.

Key Frame Count

Set greater or equal to 1. With value 1, every message is a Key Frame (every value in the data set is sent). A delta frame (not a key frame) is a message which contains only data which has been changed.

UADP Data Set Message Content Mask

Define the content mask for UADP data set messages. This parameter specifies which fields will be included in the UADP message payload.



If you have not configured any data sets in Variable Data Sets the Data Set dropdown list is empty. You are required to first define a data set to be able to create a Data Set Writer.

6.4. Variable Data Sets

Variable data set is used to publish OPC UA variable data. You can configure different attributes from a variable to be published. Value attribute is the default.

Variable DataSets

+ Add DataSet

New DataSet

General settings

Name*
MyVariableDataSet

Default Field Name*
BrowsePath

Browse Root
nsu=http://www.prosysopc.com/OPCUA/Forge;s=SimulationServer@DESKTOP-6ELQEFB

Figure 62. Variables are added to variable data set after the form is saved.

Name

Enter a descriptive name for the variable data set configuration.

Default Field Name

Specify the default field name to be used for the data set.

- **BrowsePath:** Use the browse path of the node as the default field name for the data set. The browse path represents the hierarchical path to the node within the OPC UA address space.
- **BrowseName:** Use the browse name of the node as the default field name for the data set. The browse name is a human-readable name assigned to the node within the OPC UA address space.
- **ExpandedNodeId:** This option will use the expanded node ID of the node as the default field name for the data set. The expanded node ID uniquely identifies the node within the OPC UA server.

Browse Root

Define the root Node from which to browse for variables to include in the data set. This specifies the starting point for selecting OPC UA variables to be included in the data set.



After a Variable Data Set is created, variables can be added to the Variable Data Set from [Address Space](#) view.

Add and edit variables in variable data set.

6.5. Event Data Sets

Event data set allows you to have data from events in Pub Sub messaging.

Event DataSets

+ Add DataSet

New DataSet

General settings

Name*
ServerEventDataSet

Event Node*
i=2253

SELECTED EVENT FIELDS +

Message	<input checked="" type="checkbox"/>
Time	<input checked="" type="checkbox"/>
SourceNode	<input checked="" type="checkbox"/>
SourceName	<input checked="" type="checkbox"/>

Figure 63. Remember to select the event fields which you want to include in the event data set.

Name

Enter a descriptive name for the event data set configuration.

Event Node

Select an object node containing the event in the OPC UA server.

Selected Event Fields

Choose the specific fields of the event Node to be included in the data set. This allows users to select which event fields should be included in the data set for publishing.

7. Data Exchange

The Data Exchange module within Forge allows you to transfer data between Nodes, ensuring efficient synchronization of variables across your system. You can configure multiple Groups with different modes for data transfer to optimize your system. You can for example exchange important process values between two servers to improve the interoperability in your production process.

Data Exchangers

New Data Exchanger

General settings ENABLED

Name*

Collecting Mode* Interval* Exchange Mode*

Figure 64. Create Data Exchange Group.

Enabled

Toggle to enable or disable the Data Exchange Group.

Name

Provide a descriptive name for the group configuration to identify it within the application.

Collecting Mode

Choose the collecting mode for gathering data:

- Subscription: Exchange data based on subscription to OPC UA nodes.
- Polling: Exchange data periodically by polling OPC UA nodes.
- PollingAndSubscription: Combines both approaches. Data is exchanged at least at the specified interval through polling, while also receiving updates more quickly when changes occur.

Interval (ms)

Specify the interval, in milliseconds, at which data is exchanged.

Exchange Mode

Select the exchange mode for exchanged data. Options include All and Delta. All exchanges all data according to the Interval, while Delta exchanges the data only if there have been changes in the value.

After a Group is created, the user can add items into the Group. All these items inside the Group will have the same data transfer settings.

MyDataExchange - New Exchange Item

Item settings

Source Node*
nsu=http://www.prosysopc.com/OPCUA/Forge;s=3DVec/X

Target Node*
nsu=urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer/http://www.prosysopc.com/OPCUA/SampleAddressSpace;s=MyLevel

INCLUDE TIMESTAMPS

Figure 65. You can have multiple items in one Group.

Source Node

The Node from where the data is sent.

Target Node

The Node which receives the value.

Include Timestamps

Whether the source timestamp is wanted to be included in the data exchange.

7.1. Import Data Exchange Groups

You can import Data Exchange Group configurations with CSV file. Click the Import button and download the template for the CSV file. Fill the Data Exchange group details into the template file and then upload it to Forge from the same dialog.

Import Collecting Items

To import a collecting items, the uploaded .csv file with the collecting item data needs to be formatted according to the template file, which can be downloaded below.

Import .csv

DataExchangeCollectingItemsCsvHeaderFile.csv

Figure 66. Importing is a fast way to bring configurations.

8. Event Generator

The Event Generator module within Forge allows users to create and manage new events, facilitating a dynamic and responsive operational environment. Event Generator empowers users to generate new events with custom event templates. Users can define these templates by creating custom structures for event properties, incorporating static values, and embedding place-holders for variables. You can utilize the same template across multiple event generators, ensuring harmonized event messages throughout the system.

8.1. Event Templates

The first step is to configure an event template. There you first give a name for the template which you will be using to identify the templates from each other later in the process. One event template can have multiple property templates. Property template is defining the text which will be shown in the selected property of the event. The text can have static text and values but also place-holders for variables which will be mapped later in the process.

Figure 67. Create event templates to be used in Generators.

Name

Provide a descriptive name for the template configuration to identify it within the application.

Event Type

Only BaseEventType is supported at the moment.

Event Property

In which property of a basic event the template is shown.

Template

Property templates define how the event generator creates content for the selected event property. The template can contain both static values and mapped variables. Mapped variables are defined by a variable surrounded with curly brackets: e.g. {Variable1}. Full example:

```
MY-EVENT@{VariableA}@{VariableB}@1000
```

Detected Variables

Shows the detected variables from the syntax above. (Static: MY-EVENT@, @, @1000. Mapped: VariableA, VariableB). Mapped variables mean that user can map data from Nodes to those variables in the event generator.



Event Templates can be used for multiple event generators.

8.2. Generators

In OPC UA, only Object Nodes can have events. Clients can monitor the events from these objects and get notified of the events. This feature allows you to create new events. You select the Node which will have the event and then you use the templates that you created as a content for the event. Finally, you need to configure a trigger for the event which will activate the event generation.

Event Generators

+ Add generator

New Event Generator

General settings

Name
MyEvent

Source NodeId
nsu=http://www.prosysopc.com/OPCUA/Forge;s=MyObject

Template
MyTemplate

Template Variables

#1 level-meas
Source NodeId
nsu=http://www.prosysopc.com/OPCUA/Forge;s=3DVec/X

#2 limit-value
Source NodeId
nsu=http://www.prosysopc.com/OPCUA/Forge;s=3DVec/Z

Generation Triggers

#1 Trigger NodeId*
nsu=http://www.prosysopc.com/OPCUA/Forge;s=3DVec/Y

Monitoring Mode*
Subscription

Interval*
500

Trigger conditions

Condition type*
Greater Than

Comparison value*
1000

Add Cancel

Figure 68. Event Generator configuration.

Name

Provide a descriptive name for the Generator configuration to identify it within the application.

Source NodeId

This NodeId refers to the Node for which the event is added. Source Node must be an Object Node.

Template

Select a template to be used in the event. These templates can be managed in [Event Templates](#) view.

Template Variables

If the template has defined variables, user can map data to these variables. Data can be mapped from any variable found in Forge. Set Source NodeId for variables.

Trigger NodeId

The Node whose value will be monitored to trigger the Event. This can be any variable found from Forge. Trigger is the rule which will activate the event.

Monitoring Mode

Choose the mode for collecting data. Options include Subscription and Polling, depending on

whether data is collected through continuous subscriptions or periodic polling.

Interval (ms)

Specify the interval, in milliseconds, at which the trigger condition is checked.

Condition Type

Select the comparator how you want to check the value. Options include Change, Increment, Equals, Greater Than, Less Than, High Limit, and Low Limit.

Comparison Value

Give a value to compare to in the condition.

9. Event Mapper

The Event Mapper enables integration and mapping of events between different systems or applications within the Forge. It allows event-driven architectures, by mapping data from events to address space. With Event Mapper, users can configure how events are collected, processed, and exchanged, facilitating real-time data flow and synchronization. Whether it's monitoring changes in industrial processes, tracking user interactions, or responding to system events, Event Mapper provides a flexible and efficient solution for event-driven workflows within Forge.

Map data from events to address space. Event Mapper will create event object under a selected node. This new object will have the mapped data from source event.

Event Mappers + Add Mapper

New Event Mapper

General settings ENABLED

Name*

Collecting Mode* Interval* Exchange Mode*

Figure 69. Event Mapper configuration form.

Enabled

Toggle to enable or disable the Event Mapper.

Name

Provide a descriptive name for the Event Mapper configuration to identify it within the application.

Collecting Mode

Choose the collecting mode for gathering data:

- Subscription: Collect data based on subscription to OPC UA nodes.
- Polling: Collect data periodically by polling OPC UA nodes.
- PollingAndSubscription: Combines both approaches. Data is collected at least at the specified interval through polling, while also receiving updates more quickly when changes occur.

Interval (ms)

Specify the interval, in milliseconds, at which data is collected.

Exchange Mode

Select the storage mode for logged data. Options include All and Delta. All stores all data collected, while Delta stores only the changes or differences in data since the last logging.

Add items to Event Mapper group

EventManager - New Mapper Item

Item settings

Event Node*
 nsu=urn:DESKTOP-6ELQEFB:OPCUA:SimulationServer/http://www.prosysopc.com/OPCUA/SampleAddressSpace;s=MyDevice

Mapping Root Node*
 nsu=http://www.prosysopc.com/OPCUA/Forge;s=MyObject

SELECTED EVENT FIELDS	
SourceNode	<input checked="" type="checkbox"/>
SourceName	<input checked="" type="checkbox"/>
Time	<input checked="" type="checkbox"/>
Message	<input checked="" type="checkbox"/>
Severity	<input checked="" type="checkbox"/>

Add Cancel

Figure 70. Map event properties to address space.

Event Node

Specify the Node representing the event in the source system.

Mapping Root Node

Define the root Node in the target system where the event data will be mapped.

Selected Event Fields

Choose the specific fields or attributes of the event Node to be mapped to the target system.

This will create object to address space.

The screenshot shows the Event Mapper interface. On the left, a tree view displays the object hierarchy under 'MyObject'. The 'BaseEventType' object is selected, and its details are shown in the right-hand panel. The details panel includes the following information:

- Namespace:** http://www.prosysopc.com/OPCUA/Forge/EventManager
- Identifier:** 5a5e47c9-e4b9-49b2-actf-9493f019056d
- Add Reference:** A button with a right-pointing arrow and a plus sign, used to add a reference to the selected object.

Figure 71. Event mapper creates BaseEventType object to address space to the selected location.

10. History

The History module collects and stores data into an InfluxDB database, and then provides an OPC UA Historical Access interface to access the collected data. This means that the current data and historical data can be accessed from same Node. It provides robust Forge-InfluxDB communication, allowing users to create new or use existing InfluxDB buckets as storage destinations. Customized collectors with different data collection settings enable efficient storage of OPC UA Nodes, and enable history read operations for OPC UA clients. This way both current and history data can be accessed from same location. This feature provides a reliable method for buffering data, ensuring a continuous data stream even during connection losses.



History module is optimized for Forge's internal usage to enable OPC UA Historical Access. If you want to log data for your own usage, go to [Data Logger](#).

10.1. InfluxDB

InfluxDB has to be installed and configured to be able to use this feature. Details of setting up the InfluxDB can be found from their own web site, <https://docs.influxdata.com/influxdb/v2/install/>.



You can use any 2.x version of the InfluxDB, but the newest is recommended.

Follow the instructions provided at the download page to complete the InfluxDB setup.

10.2. Storage

In History module you need to first configure the Storage settings. The form needs values from the InfluxDB application. Fill in the information and click Save button to be able to continue to configure the Collectors.

History Database Connection Configuration

Before configuring History Collectors, you need to configure a valid [InfluxDB](#) connection. You can either create the bucket manually, or supply Forge with the required permissions and allow it to create the bucket itself.

InfluxDB Connection Configuration	
Organization information	
Organization Name	ProsysOPC
Organization Id	3894b3d4ca9fb2fb
Bucket configuration	
Bucket Name	forge-bucket
ALLOW FORGE TO CREATE BUCKET: <input checked="" type="checkbox"/>	
API access	
API Url	http://localhost:8086
API Token	ghSBgx1Pl64q0tKhicEPHr9PTt0w5URiIE_v-DnqYhBt-_aUoZuzZouHAjBEEs08kn1LlI0dd45JSMMPiirQ==
<input type="button" value="Save"/> <input type="button" value="Revert"/>	

Figure 72. You need the information from InfluxDB to fill in the connection settings.

Organization Name

After setting up the InfluxDB, this value can be found from the InfluxDB UI.

Organization Id

After setting up the InfluxDB, this value can be found from the InfluxDB UI.

Allow Forge to Create Bucket

Whether user allow new bucket to be created according to the settings below or you need to fill the following information to match an existing bucket in your InfluxDB.

Bucket Name

If creating a new bucket is allowed this is the user defined name to identify the bucket. Else, this needs to be the name of a bucket that is already existing in your InfluxDB.

API Url

The Url in which the InfluxDB is running. (default: <http://localhost:8086>)

API Token

This is the user token given in the setup of InfluxDB.

10.3. Collectors

Collectors groups the Nodes and allows to use different collecting parameters.

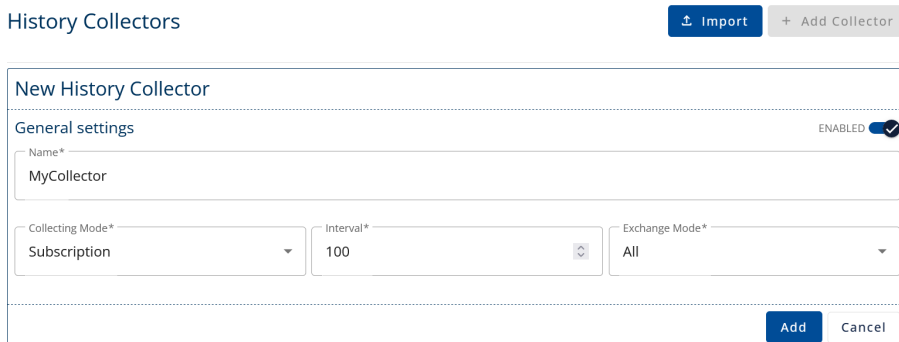


Figure 73. Create History Collector Group.

Name

Provide a descriptive name for the collector configuration to identify it within the application.

Collecting Mode

Choose the collecting mode for gathering data:

- Subscription: Collect data based on subscription to OPC UA nodes.
- Polling: Collect data periodically by polling OPC UA nodes.
- History: If the node has HistoryRead enabled and the connection to the data source is disrupted, the missed values will be read from history and published as an array after the connection has been re-established.
- PollingAndSubscription: Combines both approaches. Data is collected at least at the specified interval through polling, while also receiving updates more quickly when changes occur.

Interval (ms)

Specify the interval, in milliseconds, at which data is collected.

Exchange Mode

Select the storage mode for logged data. Options include All and Delta. All stores all data collected, while Delta stores only the changes or differences in data since the last logging.

10.3.1. Add Nodes to Collector

After the Collector is created, you can add Nodes to the collector and the data from those Nodes will be stored into InfluxDB after which the data can be accessed with history reads from the Nodes.

New History Item

Item settings

Node*
nsu=http://www.prosysopc.com/OPCUA/Forge;s=3DVec/Y

Add Cancel

Figure 74. Add Nodes to the Collector.

Node

This NodeId refers to the Node from where the value is written to InfluxDB.

10.3.2. Import History Collectors

You can import History Collector configurations with CSV file. Click the Import button and download the template for the CSV file. Fill the History Collector details into the template file and the upload it to Forge from the same dialog.

Import Collecting Items

To import a collecting items, the uploaded .csv file with the collecting item data needs to be formatted according to the template file, which can be downloaded below.

Template

Import .csv

HistoryCollectingItemsCsvHeaderFile.csv

Import Collecting Items Cancel

Figure 75. Import is a fast way to bring configurations.

11. Modbus

This module offers communication with Modbus protocol, streamlining the integration of Modbus clients and servers into the OPC UA ecosystem. It allows users to configure Modbus Device connections in Forge to fetch data from Modbus Servers, which is then converted into OPC UA format for standardized data representation. Additionally, users can configure the Modbus Servers to wrap OPC UA data into Modbus format, ensuring compatibility and efficient data exchange between Forge and Modbus connections. This solution simplifies the process of bridging the gap between OPC UA and Modbus communication standard, enabling interoperability and efficient data integration.

11.1. Modbus Device

Modbus Devices are created in Forge in order to be able to connect and fetch data from external Modbus Servers. You configure a Modbus Device which connects to the external Modbus Server and can read the registers. These read values are converted into Forge's address space under *Data Sources/Modbus Devices* folder.

11.2. Modbus Server

Forge's Modbus Server is configured to publish data from Forge for external Modbus Clients. Workflow is following:

1. Create Modbus Server with suitable configuration and protocol
2. Configure variables for the created Modbus Server
 - a. This creates Nodes to Forge's address space under *Data Sources/Modbus Servers*
3. Add data to those variables with [Data Exchange](#)

After making configuration, you have an internal Modbus Server to which you can connect with external Modbus Clients and receive the data from Forge.

11.3. Modbus Configuration

The forms to configure Modbus Device/Server are similar.

11.3.1. Modbus with TCP/IP

These configurations are used with TCP/IP protocol.

Figure 76. TCP/IP communication specific configuration.

Device Enabled

You can enable or disable communication to the device temporarily without removing it from the configuration. Only devices that are enabled will show up in the OPC UA Server address space.

Name

User defined name to identify the Modbus device.

Description

User-defined short description of the Modbus device.

Protocol Variant

Modbus protocol variant used for the Modbus device: Modbus TCP and Modbus RTUTCP are used for TCP/IP communication.

Swap Bytes

Defines that the order of the two bytes inside a register should be swapped from the default Modbus interpretation (only applies to data types with a length of 16 bits or more). Default mode interprets the first byte as the most significant byte (MSB first). Setting Swap Bytes to true will change the mode to interpret the first byte as the least significant byte (LSB first).

Swap Words

Defines that the order of two words in two subsequent registers should be swapped from the default Modbus interpretation (only applies to data types with a length of 32 bits or more). Default mode interprets the first word as the most significant (high).

Disable Pooling

Modbus connections are shared between devices by default. If you need separate connections, disabling pooling will force separate connections for this device.

IP Host Address

For Modbus Device configuration this value is the IP address or hostname of the Modbus Server that you are connecting to. For the Modbus Server configuration this is the IP or hostname where the Modbus Server will be hosted.

TCP Port

The TCP port number of the Modbus device. The standard port number for Modbus protocol is 502,

so that will be used by default.

Unit ID

Unit ID of the Modbus device. This is usually 0, but for example if you are using Modbus RTUTCP, the Unit ID is the RTU Node address of the target device. Also in some rare cases, devices use a different Unit ID with Modbus TCP, in which case you will need to configure the correct ID here.

11.3.2. Modbus with Serial Port

These configurations are used with serial port communication.

Modbus Client connections

+ Add Device

New Modbus device

General settings DEVICE ENABLED

Name:

Description:

Protocol Variant:

DEFAULT BYTE ORDER
 Swap Bytes Swap Words

Serial connection settings

Serial Port: Unit ID: Baud Rate:

Parity: Flow Control In: Flow Control Out:

Data Bits: Stop Bits: Echo:

No configured devices.

Figure 77. Serial port communication specific configuration.

Device Enabled

You can enable or disable communication to the device temporarily without removing it from the configuration. Only devices that are enabled will show up in the OPC UA Server address space.

Name

User defined name to identify the Modbus Device.

Description

User-defined short description of the Modbus Device.

Protocol Variant

Modbus protocol variant used for the Modbus device: Modbus RTU and Modbus ASCII are used for serial port communication.

Swap Bytes

Defines that the order of the two bytes inside a register should be swapped from the default Modbus interpretation (only applies to data types with a length of 16 bits or more). Default mode interprets the first byte as the most significant byte (MSB first). Setting Swap Bytes to true will

change the mode to interpret the first byte as the least significant byte (LSB first).

Swap Words

Defines that the order of two words in two subsequent registers should be swapped from the default Modbus interpretation (only applies to data types with a length of 32 bits or more). Default mode interprets the first word as the most significant (high).

Serial Port

Name of the serial port that the Modbus device is connected to.

Unit ID

Unit ID of the Modbus device. This is usually 0, but for example if you are using Modbus RTUTCP, the Unit ID is the RTU Node address of the target device. Also in some rare cases, Devices use a different Unit ID with Modbus TCP, in which case you will need to configure the correct ID here.

Baud Rate

Serial port speed, i.e., bit rate (bits/s).

Parity

Parity bit mode used for error-checking with every character. The choice between None, Odd, Even, Mark, or Space parity depends on the specific requirements of the communication system and the devices involved. Odd and even parity are the most commonly used options for error checking in serial communication. The selection of a particular parity setting should be consistent across all communicating devices to ensure proper data integrity.

- NONE: With none parity, there is no additional parity bit added to the data. This means that each character is transmitted without any additional checking for errors.
- EVEN: In even parity, the number of set bits (1s) in each byte, excluding the parity bit, is configured to be an even number. The parity bit is then set to make the total number of bits (including the parity bit) even. If the data has an odd number of set bits, the parity bit is set to 1; if it has an even number of set bits, the parity bit is set to 0.
- ODD: In odd parity, the number of set bits (1s) in each byte, excluding the parity bit, is configured to be an odd number. The parity bit is then set to make the total number of bits (including the parity bit) odd. If the data has an even number of set bits, the parity bit is set to 1; if it has an odd number of set bits, the parity bit is set to 0.
- MARK: Mark parity always sets the parity bit to 1. This means that the total number of bits (including the parity bit) will always be an odd number. This setting is rarely used in modern applications.
- SPACE: Space parity always sets the parity bit to 0. This means that the total number of bits (including the parity bit) will always be an even number. Similar to mark parity, space parity is not commonly used in modern applications.

Flow Control In

Inward data flow control mode.

- OFF: In OFF flow control, there is no explicit mechanism for controlling the flow of data. Devices simply send data as quickly as they can, assuming that the receiving device can handle the incoming data at the same rate. While this may work well in certain situations, it can lead to data loss or corruption if the receiving device is unable to keep up with the transmission speed.
- DSR: The DTR (Data Terminal Ready) signal from one device is connected to the DSR (Data Set Ready) input of the other device, and vice versa. When both devices are ready to communicate,

they assert their respective DTR and DSR signals.

- CTS: CTS uses additional control lines, typically RTS (Request to Send) and CTS (Clear to Send), to manage the flow of data. When the sending device has data to transmit, it asserts the RTS signal, indicating its intention to send data. The receiving device responds with the CTS signal, indicating that it is ready to receive. This way, the sender is informed when it's appropriate to send data, preventing buffer overflows on the receiver's end.
- XONXOFF_OUT: Software flow control involves sending special control characters (XON and XOFF) within the data stream itself to manage flow. When the receiving device's buffer is nearing full capacity, it sends an XOFF character to the sender, instructing it to pause transmission temporarily. When the buffer has sufficient space, the receiver sends an XON character, indicating that the sender can resume transmission. This method is purely software-based and doesn't require additional control lines.

Flow Control Out

Outward data flow control mode. Same options as in Flow Control In.

Data Bits

The number of data bits in each character. It can be set to 7 or 8 bits. The most common setting is 8 bits.

Stop Bits

The number of stop bits sent at the end of every character. This can be increased to 1.5 or 2 stop bits for added reliability in noisy environments.

Echo

Enable/disable echoing. Echo is a feature that, when enabled, causes the device to send back the received data to the sender. This can be useful for diagnostic purposes to verify that the data is being transmitted correctly.

11.3.3. Advanced Settings

These settings apply for both TCP and serial port devices. These advanced settings are crucial for optimizing the performance, reliability, and efficiency of Modbus communication. The specific values for these settings should be chosen based on the characteristics of the network, the capabilities of the devices, and the requirements of the overall system. It's essential to find a balance between responsiveness and resource usage to ensure the effective operation of the Modbus communication network. The advanced settings can be accessed with the Advanced Settings button below the Device settings form.

Advanced settings

Modbus Timeout (ms) 5000	Reconnect Interval (ms) 10000
Max Bits Per Modbus Read 2000	Minimum Sampling Rate (ms) 1000

Figure 78. Advanced settings for Modbus device.

Modbus Timeout (ms)

The Modbus Timeout is the maximum time, in milliseconds, that a Modbus device will wait for a response from another device before considering the communication attempt unsuccessful. If a response is not received within the specified timeout period, the Modbus device may trigger an

error or attempt a retransmission, depending on the implementation.

Reconnect Interval (ms)

The Reconnect Interval is the time, in milliseconds, that a Modbus device waits before attempting to reconnect to a Modbus network or device after a connection has been lost. This setting helps control the frequency of reconnection attempts, preventing continuous rapid attempts that may overload the network or cause unnecessary resource consumption.

Max Bits Per Modbus Read

This setting defines the maximum number of bits that a Modbus device is allowed to read in a single Modbus read request. Modbus uses terms like "coils" to represent individual bits in the case of discrete inputs or outputs. Setting a maximum limit helps control the amount of data transferred in a single read request and can be useful for managing bandwidth and optimizing communication efficiency.

Minimum Sampling Rate (ms)

The Minimum Sampling Rate specifies the minimum time, in milliseconds, that a Modbus device should wait between successive sampling or polling cycles. In Modbus communication, devices often poll or sample data from other devices at regular intervals. Setting a minimum sampling rate ensures that the device doesn't poll too frequently, preventing unnecessary network traffic and resource usage.

11.3.4. Duplicate Modbus Device/Server

You can easily copy Device/Server with Copy button. This will create an identical copy but with a modified name. Screenshots shown below are for Devices, but functions are identical to Servers also.



Figure 79. Copied devices will be Disabled by default.

Device is Disabled by default. You can change the settings and Enable the copied devices using the Edit button.

11.3.5. Variable Configuration

Inside the Modbus Device/Server you need to configure the registers where the data is located in the Modbus system. These configurations can be accessed for each Modbus Device/Server with Edit button from the respective device card. Note that the variable forms are similar for both Modbus

Devices/Servers.

Modbus Device connections

+ Add Device

modbus-device-1 <small>OPTIONAL DESCRIPTION FOR THE DEVICE</small> <small>PROTOCOL: TCP ADDRESS: 127.0.0.1:502 UNITID: 1</small>	
modbus-device-2 <small>PROTOCOL: TCP ADDRESS: 127.0.0.1:502 UNITID: 2</small>	
modbus-device-3 <small>PROTOCOL: TCP ADDRESS: 0.0.0.0:502 UNITID: 1</small>	

Figure 80. Example of the list of configured Modbus Devices (client connections).

Modbus Device/Server has different categories (InputRegisters, HoldingRegisters, DigitalInputs, DigitalOutputs) to fetch data from. Expand category to see configured variables.

modbus-device-1 <small>OPTIONAL DESCRIPTION FOR THE DEVICE</small> <small>PROTOCOL: TCP ADDRESS: 127.0.0.1:502 UNITID: 1</small>	
Device settings	
Digital Inputs	
Digital Outputs	
Input Registers	
Holding Registers	

Figure 81. Pressing the Arrow expands the categories.

Pressing the Add button will open the following form, which is used to configure variables into the Modbus Device/Server.

New Input Register

Address configuration

Address*

Name*

Description

description here

Register configuration

Data Type*

Bit Offset

Array Length*

BYTE ORDER

Swap Bytes Swap Words

SCALING

Scaling Factor*

Scaling Offset*

Figure 82. Variable configuration.

Address

The starting address of the variable inside the Modbus table. The address range starts from 0 for each table and ends in address 65535 (in contrast to the Modbus Coil/Register numbering scheme that starts from 1 for Digital Outputs, 10001 for Digital Inputs, etc.)

Name

User defined name to identify the variable.

Description

Additional description to be used for the variable.

Array length

Defines the number of sequential Modbus data items of the chosen data type that will be presented as an array of the given length (value of 1 means that the variable is a scalar, i.e., not an array). Maximum size for the data in an array is 2000 bits, which means that the maximum length of an array depends on the length of the data type. For string data types, the array length defines the maximum length of bytes (CHAR) or words (WCHAR) that the value can hold (this can be different from the actual number of characters in the string).

The following fields are only available for Input and Holding Registers forms.

Data Type

Data type used to interpret the value of the register.

Bit Offset

Defines the bit offset (between 0 and 15) inside the specified register for the starting point of the variable (only applies to data types with a length of 8 bits or less). For example, a bit offset value of 3 means that the variable value starts from the 4th bit of the register.

Swap Bytes

Defines that the order of the 2 bytes in a register should be swapped from the default Modbus interpretation (only applies to data types with a length of 16 bits or more). The default mode interprets the first byte as the most significant byte (MSB first). Setting Swap Bytes to true will change the mode to interpret the first byte of a word as the least significant byte (LSB first).

Swap Words

Defines that the order of the 2 words in the 2 subsequent registers should be swapped from the default Modbus interpretation (only applies to data types with a length of 32 bits or more). Default mode interprets the first word as the most significant (high).

Scaling

Select the scaling method from the available options, which currently include:

- None: No scaling applied.
- Linear Scaling: Apply linear scaling to the data using the formula: $\text{Factor} * \text{Value} + \text{Offset}$. For arrays, the scaling applies to each element individually.

Scaling Factor

Set the scaling factor value to be used in the linear scaling formula. This factor adjusts the magnitude of the scaled values.

Scaling Offset

Specify the offset value to be applied in the linear scaling formula. This offset adjusts the baseline of the scaled values.

Configured Modbus variable.



Input Registers			+	^
ADDRESS	NAME	DATA TYPE		
0	inputRegister0	SINT		

Figure 83. Configured variable.

The configured Modbus Devices/Servers and variables can be seen in the address space.

The Name value of a variable is also used for the BrowseName and DisplayName, but not for the NodeId. The NodeId will comprise of the Modbus Device's name, the category (InputRegisters, HoldingRegisters, DigitalInputs, DigitalOutputs), and the bit offset inside the address, for example 'modbus-device-1:InputRegisters:0:0'.

Address Space Namespaces Certificates Server Settings Reverse Connections

- Objects
 - Server
 - Aliases
 - Locations
 - Data Sources
 - OPC UA Servers
 - Modbus Devices
 - modbus-device-1
 - DigitalInputs
 - DigitalOutputs
 - HoldingRegisters
 - InputRegisters
 - InputRegister0 (Int16)
- ParameterSet
- HardwareRevision (String)
- DefaultSwapWords (Boolean)
- Port (UInt16)

InputRegister0

Namespace:
http://www.prosysopc.com/OPCUA/Forge/Modbus/Data

Identifier:
modbus-device-1.InputRegisters.0.0

→ •
Add Reference

Figure 84. Configured Modbus variable is visible at the address space.

12. MQTT

Configuring MQTT connections in Forge to send and receive custom MQTT messages significantly boosts Forge's functionality. This enhancement enables precise customization of MQTT setups to meet specific project needs. By transmitting and receiving tailored MQTT messages, Forge becomes a versatile integration software for efficient data exchange and communication across diverse networks and systems. This advancement not only amplifies Forge's utility but also unlocks new avenues for innovation and collaboration in industrial automation.

[Connections](#) [Templates](#) [Publishers](#) [Subscriptions](#)

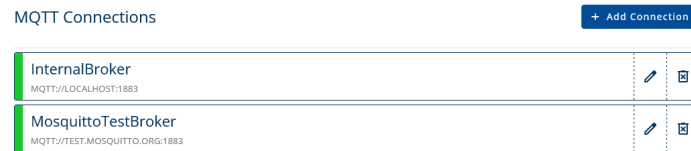


Figure 85. Navigate to different configurations from the sub-navigation bar.

12.1. Connections

When setting up an MQTT connection in Forge, the following parameters need to be filled in:

MQTT Connections

+ Add Connection

New MQTT connection

General settings DEVICE ENABLED

Name*
LocalBroker

Connection parameters

Protocol: mqtt:// Hostname*: localhost Port: 1883

ClientId: forge-server

Automatic Reconnect Clean Session

Connection Timeout*: 60 Keep-alive Interval*: 10

Authentication AUTHENTICATION ENABLED

Username: admin Password: ●●●●●●

Encryption

Disable Certificate Validation Disable Hostname Verification

Add Cancel

Figure 86. To fill the MQTT connections form you need the information of the MQTT broker that you are connecting to.

Device Enabled

Enable or disable the MQTT connection.

Name

Provide a name for the MQTT connection to identify it within the application.

Protocol

Select the protocol for the MQTT connection. Options include:

- mqtt:// for MQTT over TCP
- mqtts:// for MQTT over SSL/TLS

Hostname

Specify the hostname or IP address of the MQTT broker/server.

Port

Specify the port number on which the MQTT broker/server is listening.

ClientId

Provide a unique client identifier for this MQTT connection.

Automatic Reconnect

Enable or disable automatic reconnection to the MQTT broker in case of disconnection.

Clean Session

Enable or disable the clean session flag for the MQTT connection.

Connection Timeout

Set the duration in seconds after which the connection attempt to the MQTT broker will time out if

not successful.

Keep-alive Interval

Set the interval in seconds at which the MQTT client sends keep-alive pings to the broker to maintain the connection.

Authentication

Enable or disable authentication for the MQTT connection.

Username

If authentication is enabled, provide the username required for authentication to the MQTT broker.

Password

If authentication is enabled, provide the password required for authentication to the MQTT broker.

Disable Certificate Validation

Enable or disable certificate validation for SSL/TLS connections. When checked, the application will not validate the SSL/TLS certificates presented by the MQTT broker.

Disable Hostname Verification

Enable or disable hostname verification for SSL/TLS connections. When checked, the application will not verify whether the hostname in the certificate matches the hostname of the MQTT broker.

MQTT Connections

+ Add Connection













 localbroker MQTT://LOCALHOST:1883		
 mosquittoBroker MQTT://LOCALHOST:1884		
 publicAuthTest MQTT://TEST.MOSQUITTO.ORG:1884		
 publicTest MQTT://TEST.MOSQUITTO.ORG:1883		

Figure 87. Configured MQTT connections status can be seen from the status color. (Red: Not connected, Green: Connected, Grey: Unknown)

12.2. Templates

The Template feature in Forge simplifies MQTT configuration by offering feature to design re-usable templates for publishers and subscribers, ensuring standardized and efficient communication setups with minimal user effort.

New MQTT template

General settings

Name*

Topic*

Quality of Service (QoS)*
 Retain Message

JSON Mappings +

#1

Type*

Value Path*

Timestamp Path*

Timestamp Format*

Preview

```

{
  "sensor": {
    "measurements": {
      "value": "value"
    }
  },
  "timestamp": "yyyy-MM-ddTHH:mm:ss.SSSZ"
}

```

Figure 88. These re-usable templates help to create new publishers and subscriptions.

Name

This field is for entering the name of your new MQTT template. It's important to choose a name that clearly identifies the purpose of the template.

Topic

Specify the MQTT topic to which messages will be published or from which they will be subscribed. The topic is a UTF-8 string that the broker uses to filter messages for each connected client.

Quality of Service (QoS)

This dropdown menu allows you to select the level of assurance for message delivery.

- 0 - At Most Once: message is delivered at most once or not at all.
- 1 - At Least Once: message is delivered at least once. Duplicates can occur.
- 2 - Exactly Once: message is delivered exactly once.

Retain Message

Select whether the published message should be retained by the MQTT broker. The broker will keep a copy of the message even after it has been delivered to all current subscribers. This is useful if you want new subscribers to receive the most recent message on a topic.

JSON Mappings

Specify the type of message mapping. More detailed instructions to fill the JSON Mappings can be found from [JSON Mappings](#).

- **Node (value)**: Maps a value from a specific Node to the MQTT message payload. (Publishers/Subscriptions)

- **Node (value JSON):** Maps a structured value from a specific Node to the MQTT message payload. (Publishers)
- **Node (value structure):** Maps a structured value from a specific Node to the MQTT message payload. (Publishers/Subscriptions)
- **Node (non-value):** Maps non-value attributes from a specific Node to the MQTT message payload. (Publishers/Subscriptions)
- **Static Value:** Publishes a static value as the MQTT message payload. (Publishers)
- **Method:** Publishes method arguments. (Publishers)
- **Topic:** Publishes own topic as part of the payload. (Publisher)



See more detailed instructions to JSON Mappings from [JSON Mappings](#)

MQTT Templates

+ Add Template

BasicSubTemplate MYTOPIC/TEST/VALUE		
BasicPubTemplate SUB/TORECEIVE/VALUE		
AlertTemplate ENTERPRISE/SITE/AREA/PRODUCTIONLINE/<DEVICE>/<ALERT>		
SmartSensorTemplate SERVICE/SENSOR/		

Figure 89. Manage the created templates.

12.3. Publishers

An MQTT publisher is used for publishing messages to one or more MQTT subscribers. It plays a crucial role in distributed systems where real-time data sharing or event notification is essential. By publishing data to MQTT topics, publishers enable subscribers to receive relevant information promptly, facilitating efficient communication, monitoring, and control across various connected devices or systems. This asynchronous messaging paradigm allows for scalable and decoupled communication, making MQTT publishers widely adopted in IoT, industrial automation, and messaging applications.

MQTT publisher has three types of configurations: [Single Node](#), [Composite](#) and [UNS](#). Start configuring an MQTT publisher by selecting the connection where you want to add publisher and then type of the publisher:

Select MQTT Connection

Choose the MQTT connection to be used by the publisher. If there exist only one connection, this input is hidden.

Add Publisher

- **Single Node:** Ideal for adding multiple Nodes to individual messages, while maintaining a consistent message structure across all messages.
- **Composite:** Useful for creating more complex message structures that combine multiple values from different Nodes within the same message.
- **UNS:** Easily publish big structures of your address space with single configuration in UNS topic hierarchy.

12.3.1. Single Node

To configure Single Node publisher, you need to create the message configuration. After that, you can add Nodes which you want to send with that message configuration. Every Node will be sent as its own message.

MQTT Publishers Import Add Publisher

New MQTT Publisher ENABLED

General settings

Name*
MyMeasurement

Collecting Mode* Subscription Sampling Interval (ms)* 1000 Storage Mode* All

RE-PUBLISH ON RECONNECT
 RE-PUBLISH ON INTERVAL

Re-publish Interval (s)
3600

Template

Template

Topic settings

Topic Prefix
MyMeasurement/

Quality of Service (QoS)* 0 - At Most Once RETAIN MESSAGES

JSON Mappings +

Figure 90. Create single Node configuration and add a Node to that configuration afterwards.

Name

Enter a name for the new MQTT publisher to identify it within the application.

Enable

You can enable or disable the publisher temporarily without removing it from the configuration.

Collecting Mode

Choose the collecting mode for gathering data:

- Subscription: Collect data based on subscription to OPC UA Nodes.
- Polling: Collect data periodically by polling OPC UA Nodes.
- History: If the node has HistoryRead enabled and the connection to the data source is disrupted, the missed values will be read from history and published as an array after the connection has been re-established.
- PollingAndSubscription: Combines both approaches. Data is collected at least at the specified interval through polling, while also receiving updates more quickly when changes occur.

Sampling Interval (ms)

Set the sampling interval in milliseconds for data collection.

Storage Mode

Select the storage mode for published data. Options include All, which publishes all data collected and Delta, which publishes only the data that has changed since the last publish.

Re-Publish On Reconnect

Re-publish the previous data after reconnecting.

Re-Publish On Interval

Re-publish the previous data always after a certain interval.

Re-Publish Interval

The interval at which the previous data is always re-published.

Template

Select a predefined template for configuring the publisher settings, or leave it blank for manual configuration.

Topic

Specify the MQTT topic to which messages will be published by the publisher.

Quality of Service (QoS)

Select the desired Quality of Service level for message delivery:

- 0 - At Most Once: message is delivered at most once or not at all.
- 1 - At Least Once: message is delivered at least once. Duplicates can occur.
- 2 - Exactly Once: message is delivered exactly once.

Retain Messages

Enable or disable message retention by the MQTT broker. When enabled, the last published message on the topic is retained and delivered to new subscribers.

JSON Mappings

Define how data from OPC UA Nodes is mapped to JSON format for publication by selecting suitable type:

- [Node \(value\)](#)
- [structure](#)
- [Node \(non-value\)](#)
- [Static Value](#)
- [Topic](#)

After the single Node configuration is ready, you can start adding Nodes to the configuration. Press the list icon on the single Node configuration and fill the following form.

New Published Item

Item settings

NodId* 🔍

Topic*

Quality of Service (QoS)*

Retain*

Figure 91. Item can either inherit settings or have unique configuration.

NodId

Select the Node that you want to publish.

Topic

Specify the MQTT topic to which the Node's data will be published. This topic will be added to the end of the prefix defined in the single Node configuration.

Quality of Service (QoS)

Select the desired Quality of Service level for message delivery:

- Inherit: QoS level is inherited from the MQTT publisher.
- 0 - At Most Once: message is delivered at most once or not at all.
- 1 - At Least Once: message is delivered at least once. Duplicates can occur.
- 2 - Exactly Once: message is delivered exactly once.

Retain

Indicate whether the message should be retained on the broker. Options include True (retain the message), False (do not retain the message) and Inherit (inherited from the publisher).

12.3.2. Composite

Create structured messages, which can have values from different Nodes included in the same message.

New MQTT Publisher

General settings ENABLED

Name*
MyPublisher

Publishing Mode* On-Collect |
 Collecting Mode* Subscription |
 Sampling Interval (ms)* 1000 |
 Storage Mode* All |
 Partial Messages Not allowed

RE-PUBLISH ON RECONNECT |
 Re-publish Interval (s) 3600 |
 RE-PUBLISH ON INTERVAL

Template

Template

Topic settings

Topic*
my/topic/test

Quality of Service (QoS)* 0 - At Most Once |
 RETAIN MESSAGES

JSON Mappings +

#	Type*	
#1	Node (value)	Transformation Dictionary

Figure 92. Publisher allows you to send data out in MQTT protocol.

Name

Enter a name for the new MQTT publisher to identify it within the application.

Enabled

You can enable or disable the publisher temporarily without removing it from the configuration. UNS defaults to disable so after the configuration you can fine tune the settings individually before publishing.

Publishing Mode

Select the preferred publishing mode:

- On-Collect: Publish data when it is collected from the source.
- On-Method: Publish data from method. Use this if JSON mapping type method is used.

Collecting Mode

Choose the collecting mode for gathering data:

- Subscription: Collect data based on subscription to OPC UA Nodes.
- Polling: Collect data periodically by polling OPC UA Nodes.
- History: If the node has HistoryRead enabled and the connection to the data source is disrupted, the missed values will be read from history and published as an array after the connection has been re-established.
- PollingAndSubscription: Combines both approaches. Data is collected at least at the specified interval through polling, while also receiving updates more quickly when changes occur.

Sampling Interval (ms)

Set the sampling interval in milliseconds for data collection.

Storage Mode

Select the storage mode for published data. Options include All which publishes all data collected

and Delta which publishes only the data that is changed since the last publish.

Partial Messages

Select Allowed if you want to allow messages that have missing Nodes. Select Not Allowed if you want messages be sent only if every Node in the message exists.

Template

Select a predefined template for configuring the publisher settings, or leave it blank for manual configuration.

Topic

Specify the MQTT topic to which messages will be published by the publisher.

Quality of Service (QoS)

Select the desired Quality of Service level for message delivery:

- 0 - At Most Once: message is delivered at most once or not at all.
- 1 - At Least Once: message is delivered at least once. Duplicates can occur.
- 2 - Exactly Once: message is delivered exactly once.

Retain Message

Enable or disable message retention by the MQTT broker. When enabled, the last published message on the topic is retained and delivered to new subscribers.

JSON Mappings

Define how data from OPC UA Nodes is mapped to JSON format for publication by selecting suitable type:

- [Node \(value\)](#)
- [structure](#)
- [JSON](#)
- [Node \(non-value\)](#)
- [Static Value](#)
- [Method](#)
- [Topic](#)

12.3.3. UNS

This publisher allows you to easily create publisher that structures UNS topic hierarchy. Every Node will be sent in own message. Select a root Node and every variable underneath the root Node will be published. You can fine-tune the settings for each message separately after you first save the configuration.

New MQTT Publisher

General settings ENABLED

Name*
MySimulationServer

Sampling Interval (ms)*: 1000 Storage Mode*: All

RE-PUBLISH ON RECONNECT Re-publish Interval (s): 3600
 RE-PUBLISH ON INTERVAL

UNS settings INCLUDE ROOT IN UNS PATH

Root Node*
nsu=http://www.prosysopc.com/OPCUA/Forge;s=SimulationServer@WIN-SEDJP8C11B2

Template

Template

Topic settings

Topic Prefix
SimServer

Quality of Service (QoS)*: 0 - At Most Once RETAIN MESSAGES

JSON Mappings +

Figure 93. Publish large structure with one configuration. Publisher is disabled by default.

Name

Enter a name for the new MQTT publisher to identify it within the application.

Enabled

You can enable or disable the publisher temporarily without removing it from the configuration. UNS defaults to disable so after the configuration you can fine tune the settings individually before publishing.

Sampling Interval (ms)

Set the sampling interval in milliseconds for data collection.

Storage Mode

Select the storage mode for published data. Options include All, which publishes all data collected and Delta, which publishes only the data that is changed since the last publish.

Root Node

Select the Node from which the UNS topic will be created. Every variable underneath the selected Root Node will be published in own topics and messages.

Template

Select a predefined template for configuring the publisher settings, or leave it blank for manual configuration.

Topic Prefix

You can give a prefix for the topic structure. This can contain multiple topic levels. For example, finland/espoo/prosysopc.

Quality of Service (QoS)

Select the desired Quality of Service level for message delivery:

- 0 - At Most Once: message is delivered at most once or not at all.
- 1 - At Least Once: message is delivered at least once. Duplicates can occur.
- 2 - Exactly Once: message is delivered exactly once.

Retain Messages

Enable or disable message retention by the MQTT broker. When enabled, the last published message on the topic is retained and delivered to new subscribers.

JSON Mappings

Define how data from OPC UA Nodes is mapped to JSON format for publication by selecting suitable type:

- [Node \(value\)](#)
- [structure](#)
- [Node \(non-value\)](#)
- [Static Value](#)
- [Topic](#)

After saving the configuration, you can edit the messages individually by clicking the list icon. Update the list of items by pressing the update-icon.

NODE	TOPIC	QOS	RETAIN		
MyLevel	SimulationServer@WIN-SEDJP8C11B2/MyObjects/MyDevice/MyLevel	0	<input type="radio"/>		
AckedState	SimulationServer@WIN-SEDJP8C11B2/MyObjects/MyDevice/MyLevelAlarm/AckedState	0	<input type="radio"/>		
ActiveState	SimulationServer@WIN-SEDJP8C11B2/MyObjects/MyDevice/MyLevelAlarm/ActiveState	0	<input type="radio"/>		
Comment	SimulationServer@WIN-SEDJP8C11B2/MyObjects/MyDevice/MyLevelAlarm/Comment	0	<input type="radio"/>		
EnabledState	SimulationServer@WIN-SEDJP8C11B2/MyObjects/MyDevice/MyLevelAlarm/EnabledState	0	<input type="radio"/>		
LastSeverity	SimulationServer@WIN-SEDJP8C11B2/MyObjects/MyDevice/MyLevelAlarm/LastSeverity	0	<input type="radio"/>		
CurrentState	SimulationServer@WIN-SEDJP8C11B2/MyObjects/MyDevice/MyLevelAlarm/LimitState/CurrentState	0	<input type="radio"/>		
Quality	SimulationServer@WIN-SEDJP8C11B2/MyObjects/MyDevice/MyLevelAlarm/Quality	0	<input type="radio"/>		
MySwitch	SimulationServer@WIN-SEDJP8C11B2/MyObjects/MyDevice/MySwitch	0	<input type="radio"/>		
StateDisabledByMethod	SimulationServer@WIN-SEDJP8C11B2/Server/PublishSubscribe/Diagnostics/Counters/StateDisabledByMethod	0	<input type="radio"/>		

Items per page: 10 | 1 - 10 of 462

Figure 94. Edit the created individual items with this view.



UNS configuration is disabled by default. To start publishing, click the edit icon, enable the configuration, and then press Save.

MQTT Topic Publishers

+ Add Publisher

Select MQTT connection

InternalBroker ENDPOINT: mqtt://localhost:1883

MyPublisher ✎ ✕
MY/TOPIC/TEST

Measurement ✎ ✕
DEVICE/MEASUREMENTS/TEMPERATURE

Figure 95. Publishers are effective way to share information to multiple receivers.

12.4. Subscriptions

MQTT subscriptions enable Forge to receive messages from MQTT publishers, forming a vital component in IoT communication. Given MQTT's widespread use in smart devices, this feature is essential for accessing data from these devices effortlessly.

MQTT Topic Subscriptions

+ Add Subscription

Select MQTT connection

MosquittoTestBroker ENDPOINT: mqtt://test.mosquitto.org:1883

New MQTT subscription

Template
 Template

General settings

Name*
 TestSubscriber

Topic*
 test/subscription/topic/value

Quality of Service (QoS)*
 0 - At Most Once

JSON Mappings +

#1 Type*
 Node (value)

Figure 96. To configure the subscription, you need to know the structure of the MQTT message you are subscribing. You also need the Node to which you are going to map the value.

Select MQTT connection

Choose the MQTT connection to be used by the subscriber. If there exist only one connection, this input is hidden.

Template

Select a template if available for configuring the subscriber settings, or leave it blank for manual configuration.

Name

Enter a name for the new MQTT subscription to identify it within the application.

Topic

Specify the MQTT topic to which the subscription will listen for incoming messages.

Quality of Service (QoS)

Choose the desired Quality of Service level for message delivery:

- 0 - At Most Once: message is delivered at most once or not at all.
- 1 - At Least Once: message is delivered at least once. Duplicates can occur.
- 2 - Exactly Once: message is delivered exactly once.

JSON Mappings

Define how incoming JSON-format messages from MQTT are mapped to OPC UA Nodes by selecting suitable type.

- [Node \(value\)](#)
- [Node \(value structure\)](#)
- [Node \(non-value\)](#)

MQTT Topic Subscriptions

+ Add Subscription

Select MQTT connection

MosquittoTestBroker

ENDPOINT: mqtt://test.mosquitto.org:1883

TestSubscriber

TEST/SUBSCRIPTION/TOPIC/VALUE

✎
✖

Figure 97. All the subscriptions can be managed from the list.

12.5. JSON Mappings

This chapter provides detailed explanations of various types of JSON Mappings. With the + icon on the top right corner of the JSON Mappings form, you can add more mappings to the same configuration. You can also copy the JSON Mapping for faster configuration from the Copy icon on the left panel in JSON Mappings form.

JSON Mappings +

#1

Type*
Node (value)

Value Path*
object.property.value

Timestamp Path
object.property.timestamp

Timestamp Format*
yyyy-MM-ddTHH:mm:ssZ

NodeId*
i=2256

#2

Type*
Node (non-value)

Path*
object.property.node

NodeId*
i=2256

Attribute*
DisplayName

Preview

```

{
  "object": {
    "property": {
      "node": "2256[DisplayName]",
      "timestamp": "yyyy-MM-ddTHH:mm:ssZ",
      "value": "2256[Value]"
    }
  }
}

```

Add Cancel

Figure 98. JSON Mapping example.

12.5.1. Node (value)

This mapping type is used for both publishers and subscriptions. It maps a single value from a specific OPC UA Node to the MQTT message payload. For publishers, this means the value from the OPC UA Node is included in the message payload being sent to MQTT topics. For subscriptions, this indicates that the MQTT message payload received will contain the value for the specified OPC UA Node.

JSON Mappings +

#1

Type*
Node (value)

Value Path*
object.property.value

Timestamp Path
object.timestamp

Timestamp Format*
yyyy-MM-ddTHH:mm:ssZ

NodeId
i=2256

Preview

```

{
  "object": {
    "property": {
      "value": "2256[Value]"
    },
    "timestamp": "yyyy-MM-ddTHH:mm:ssZ"
  }
}

```

Figure 99. Node (value) is quick selection to map value attribute of the Node.

Value Path

Provide the path to the value in the MQTT message. The paths should be entered in dot notation (e.g. object.property).

Timestamp Path

Provide the path to the timestamp in the MQTT message. If left empty, no timestamp is included in the MQTT message.

Timestamp Format

In this field, you can specify the format for timestamps in your messages.

NodeId

Provide the NodeId of the OPC UA Node from/to which data will be mapped to/from MQTT messages.

12.5.2. Node (value, JSON)

This mapping type is used specifically for publishers. It maps a structured value from a specific OPC UA Node to the MQTT message payload in JSON format. This allows for more complex data structures to be included in the MQTT messages.

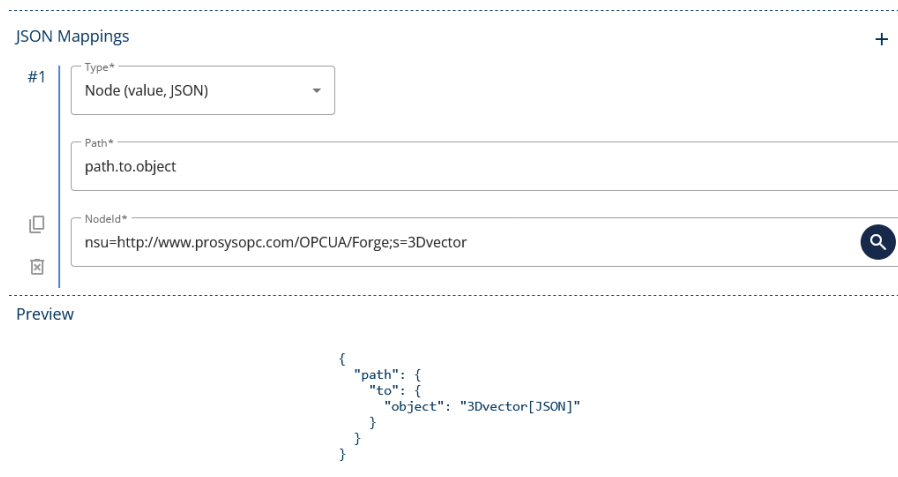


Figure 100. Publish structure variable.

Path

Provide the path to the value in the MQTT message. The paths should be entered in dot notation (e.g. object.property).

NodeId

Provide the NodeId of the OPC UA Node from which data will be mapped to MQTT messages.

12.5.3. Node (value, structure)

This mapping type is used for publishers and subscriptions. It maps a structured value from a specific OPC UA Node to the MQTT message payload. Similar to the JSON mapping, this enables the inclusion of structured data in MQTT messages.

JSON Mappings +

#1

Type*
Node (value, structure) ▼

Timestamp Path
object.timestamp

Timestamp Format*
yyyy-MM-ddTHH:mm:ssZ

NodId
i=2256 🔍

Value Paths +

From Path*
from.this.path

To Path*
to.this.path

From Path*
then.from.this.path

To Path*
then.to.this.path

Preview

```

{
  "from": {
    "this": {
      "path": "2256[to.this.path]"
    }
  },
  "object": {
    "timestamp": "yyyy-MM-ddTHH:mm:ssZ"
  },
  "then": {
    "from": {
      "this": {
        "path": "2256[then.to.this.path]"
      }
    }
  }
}

```

Figure 101. Map structure value.

Timestamp Path

Provide the path to the timestamp in the MQTT message. If left empty, no timestamp is included in the MQTT message.

Timestamp Format

In this field, you can specify the format for timestamps in your messages.

NodId

Provide the NodId of the OPC UA Node to which data will be mapped from MQTT messages.

From Path

Provide the path to the value in the MQTT message. The paths should be entered in dot notation (e.g. object.property).

To Path

Provide the path to the value in the MQTT message. The paths should be entered in dot notation (e.g. object.property).



Add more rows to Value Paths with + icon on the top right corner.

12.5.4. Node (non-value)

This mapping type is applicable to both publishers and subscriptions. It maps non-value attributes from a specific OPC UA Node to the MQTT message payload. This can include attributes such as BrowseName, DisplayName, Description, etc. This mapping allows for additional contextual information to be included in MQTT messages.

JSON Mappings +

#1

Type*
Node (non-value)

Path*
path.to.attribute

NodeId
i=2256

Attribute*
DisplayName

Preview

```

{
  "path": {
    "to": {
      "attribute": "2256[DisplayName]"
    }
  }
}

```

Figure 102. Map non-value attributes from a specific OPC UA Node to the MQTT message payload.

Path

Provide the path to the value in the MQTT message. The paths should be entered in dot notation (e.g. object.property).

NodeId

Provide the NodeId of the OPC UA Node from/to which data will be mapped to/from MQTT messages.

Attribute

Specify the attribute of the OPC UA Node that you want to map to the MQTT message. This could include attributes such as BrowseName, DisplayName, Description, etc.

12.5.5. Static Value

This mapping type is used only for publishers. It allows users to publish a static, predefined value as the MQTT message payload. This is useful for scenarios where the message content does not vary and can be predetermined.

JSON Mappings +

#1

Type*
Static value

Path*
object.property.engineering-unit

Value
kg

DataType
String

Preview

```

{
  "object": {
    "property": {
      "engineering-unit": "kg"
    }
  }
}

```

Figure 103. Add static value to the MQTT message such as engineering unit.

Path

Provide the path to the value in the MQTT message. The paths should be entered in dot notation (e.g. object.property).

Value

Enter the static value that you want to publish as the MQTT message payload.

DataType

Specify the data type of the static value.

12.5.6. Method

This mapping type is used for publishers. On-Method publishing mode shall be used. This JSON mapping enables the publication of method arguments as part of the MQTT message payload. This allows for the invocation of methods on the OPC UA server or another system through MQTT messaging.

JSON Mappings +

#1

Type*
Method

Method Objectid
i=2253 🔍

Methodid
i=12749 🔍

Input Argument Paths

[0] SubscriptionId
argument.path.in.json.message

[0] SubscriptionId
other.argument

Preview

```

{
  "argument": {
    "path": {
      "in": {
        "json": {
          "message": "12749[SubscriptionId]"
        }
      }
    }
  },
  "other": {
    "argument": "12749[LifetimeInHours]"
  }
}

```

Figure 104. Map information from method into MQTT message.

Method Objectid

Provide the NodeId of the OPC UA Object Node which has the method that is wanted to be included in the MQTT messages.

Methodid

Provide the NodeId of the OPC UA method Node which is wanted to be mapped to the MQTT messages.

Input Argument Paths

Provide the path to the value in the MQTT message. The paths should be entered in dot notation (e.g. object.property).

12.5.7. Topic

Include topic in the payload.

JSON Mappings +

#1

Type*
Topic

Path
Message.Topic

INCLUDE PREFIX

Preview

```
{  
  "Message": {  
    "Topic": "Topic"  
  }  
}
```

Figure 105. Include topic in the payload.

Path

Provide the path to the value in the MQTT message. The paths should be entered in dot notation (e.g. object.property).

Include Prefix

Enable if you want to include the prefix in the payload.



To access list items in JSON Mappings, use square brackets (e.g. list[index]). Indexing starts from zero.

13. Data Logger

Data Logger enables storage of data from various sources for future analysis and reference. Data Logger supports Influx, SQL and CSV logging. Configure Data Sinks for various databases and then use Logging Profiles to make logging coherent. Finally, use Data Loggers to start logging values from Forge address space.

13.1. Data Sinks

Data Sinks represent the destinations where logged data is stored. These sinks can include databases, file systems, or other storage repositories.

13.1.1. Influx

Configure InfluxDB connection.

Data Sinks
+ Add Sink

New Data Sink

General settings

Name*

Sink Type*

Influx

Organization settings

Organization Name*

Organization ID*

Bucket settings

Bucket Name*

ALLOW FORGE TO CREATE BUCKET

API Access settings

API Url*

Authorization Token*

ENABLE GZIP

Add
Cancel

Figure 106. Configure Data Sink to InfluxDB.

Name

Enter a name for the new Influx Data Sink configuration to identify it within the application.

Sink Type

Specify the type of Data Sink being configured, which in this case is InfluxDB.

Organization Name

Provide the name of the organization associated with the InfluxDB configuration. Value can be found from the InfluxDB UI.

Organization ID

Enter the unique identifier for the organization. Value can be found from the InfluxDB UI.

Allow Forge to Create Bucket

Specify whether Forge is permitted to create a new bucket (storage container) within InfluxDB.

Bucket Name

If allowing Forge to create a bucket, specify the desired name for the bucket. If not, enter the name of the existing bucket where data will be stored.

Enable Gzip

Choose whether to enable Gzip compression for data transmission to InfluxDB. This option can help optimize network bandwidth usage.

API Url

Provide the URL for accessing the InfluxDB API.

Authorization Token

Enter the authorization token or credentials necessary for Forge to access and interact with InfluxDB.

13.1.2. SQL

Configure SQL connection.

Data Sinks + Add Sink

New Data Sink

General settings

Name* Sink Type*

Database settings

SQL Database Type* Hostname / IP* Port*

Database name*

Authentication settings

Username* Password*

Figure 107. Configure Data Sink to various SQL databases.

Name

Enter a name for the new SQL Data Sink configuration to identify it within the application.

Sink Type

Specify the type of Data Sink being configured, which in this case is a SQL database.

SQL Database Type

Choose the type of SQL database being used for data storage. Options include MySQL, MariaDB,

Microsoft SQL Server, PostgreSQL and Oracle.

Hostname / IP

Provide the hostname or IP address of the server where the SQL database is hosted.

Port

Enter the port number through which the SQL database server can be accessed.

Database Name

Specify the name of the database within the SQL server where the data will be stored.

Username

Enter the username used to authenticate and access the SQL database.

Password

Provide the password associated with the specified username for accessing the SQL database.



Remember to enable TCP communication from the Microsoft SQL server when creating the connection with Forge.

13.1.3. CSV

Configure Data Sink for CSV file.

Data Sinks + Add Sink

New Data Sink

General settings

Name* Sink Type* CSV

Directory settings

Directory Path*

Archive Directory Path*

File size limit settings

File Size Limit* File Size Limit Unit* MB Total File Size Limit* Total File Size Limit Uni GB Max Archive Age

COMPRESS ARCHIVES

Add Cancel

Figure 108. CSV Data Sink configuration form.

Name

Enter a name for the new CSV Data Sink configuration to identify it within the application.

Sink Type

Specify the type of Data Sink being configured, which in this case is CSV.

Directory Path

Provide the directory path where the CSV files will be stored.

Archive Directory Path

Specify the directory path where archived CSV files will be stored.

Compress Archives

Choose whether to enable compression for archived CSV files.

File Size Limit

Set the maximum size limit for individual CSV files. If file size limit is exceeded, a new file will be created. A new file is also created when the date changes.

File Size Limit Unit

Specify the unit for the file size limit, such as kilobytes (KB), megabytes (MB) or gigabytes (GB).

Total File Size Limit

Define the total size limit for all CSV files combined. When this limit is exceeded, the oldest file will be deleted.

Total File Size Limit Unit

Specify the unit for the total file size limit, such as kilobytes (KB), megabytes (MB) or gigabytes (GB).

Max Archive Age

Specify the maximum number of archived CSV files before the oldest one is deleted.

13.2. Logging Profiles

Logging Profiles define the configuration settings for data logging operations. These profiles determine how data is logged, specifying parameters such as field names and data format.

Select Data Sink

Choose the destination where the logged data will be stored. Options include the Data Sinks configured in Forge. If there exist only one Data Sink, this input is hidden.

13.2.1. Influx

Influx Logging Profile form.

Logging Profiles select data sink
MyInfluxDB + Add Profile

New Logging Profile

General settings

Name*
InfluxProfile

Influx settings

Measurement*
forge-measurement

Influx field mappings +

#1 **Timestamp**

Source*
Source Timestamp

#2 **Field / Tag**

Type* Tag Tag name * ID Source* Logging Id

#3 **Field / Tag**

Type* Field Field name * Value Source* Value

Add
Cancel

Figure 109. Profiles allow user to configure the format for data storing.

Name

Enter a name for the new Influx Logging Profile configuration to identify it within the application.

Profile Type

Choose the type of the profile:

- Data: Log data of a variable.
- Event: Log data of an OPC UA event.

Measurement

Specify the measurement associated with the logged data. This typically represents the entity or object being measured, such as a sensor, device, or system component. The existing measurements in the Influx bucket are listed.

Source

Choose the source of the timestamp associated with the logged data.

- Source Timestamp: Use the timestamp provided by the data source.
- Server Timestamp: Use the timestamp generated by the server when the data is received.
- Client Timestamp: Use the timestamp generated by the client when the data is sent.

Type(Field/Tag)

Select the type of data being logged.

- Tag: Represents metadata or descriptive information associated with the logged data.
- Field: Represents the actual data value being logged.

Tag Name(Field/Tag)

Specify the name of the tag associated with the logged data. This could include metadata such as sensor type, location, or any other relevant information.

Source(Field/Tag)

Choose the source of the data to be logged.

- Logging Id: Use the unique identifier assigned to the logged data.
- Value: Log the actual data value.
- Status Code: Log the status code associated with the data (e.g., success, failure).
- Custom: Specify a custom value for the data to be logged.

Multi-type Fields for Value allows to add different types of values.

13.2.2. SQL

SQL Logging Profile form. The table with columns must be created for the database with its own tools, before configuring the profile in Forge. Forge uses Datetime format for timestamps, which you should consider when creating columns.

Logging Profiles

select data sink
MySQLserver

+ Add Profile

New Logging Profile

General settings

Name*

SQL settings

Table name*

SQL column mappings +

#1	<input type="checkbox"/>	Column name* <input type="text" value="Timestamp"/>	Source* <input type="text" value="Source Timestamp"/>
#2	<input type="checkbox"/>	Column name* <input type="text" value="ID"/>	Source* <input type="text" value="Logging Id"/>
#3	<input type="checkbox"/>	Column name* <input type="text" value="Value"/>	Source* <input type="text" value="Value"/>

Add
Cancel

Figure 110. Profiles allow user to configure the format for data storing.

Name

Enter a name for the new SQL Logging Profile configuration to identify it within the application.

Profile Type

Choose the type of the profile:

- Data: Log data of a variable.
- Event: Log data of an OPC UA event.

Table Name

Specify the name of the SQL table where the logged data will be stored.

Column Name

Select the columns to include in the SQL table. This could include metadata such as sensor type, location, or any other relevant information.

Source

Choose the source of the data to be logged.

- Logging Id: Use the unique identifier assigned to the logged data.
- Source Timestamp: Use the timestamp provided by the data source.
- Server Timestamp: Use the timestamp generated by the server when the data is received.
- Client Timestamp: Use the timestamp generated by the client when the data is sent.
- Value: Log the actual data value.
- Status Code: Log the status code associated with the data (e.g., success, failure).
- Custom: Specify a custom value for the data to be logged.

13.2.3. CSV

CSV Logging Profile form.

Logging Profiles select data sink MyCsv + Add Profile

New Logging Profile

General settings

Name*
CSVProfile

CSV settings

File name*
forge-data

Separator*
;

CSV column mappings +

#1	Column name* Timestamp	Source* Source Timestamp
#2	Column name* ID	Source* Logging Id
#3	Column name* Value	Source* Value

Add Cancel

Figure 111. Profiles allow user to configure the format for data storing.

Name

Enter a name for the new CSV Logging Profile configuration to identify it within the application.

Profile Type

Choose the type of the profile:

- Data: Log data of a variable.
- Event: Log data of an OPC UA event.

File Name

Specify the name format for the CSV files generated by the Logging Profile. Filename will be formatted like name-<date>-<N>, where the <date> is the current date and N is number that increases every time new file is created.

Separator

Choose the character used as the separator between columns in the CSV files.

Column Name

Select the columns to be included in the CSV files. This could include metadata such as sensor type, location, or any other relevant information.

Source

Choose the source of the data to be logged.

- Logging Id: Use the unique identifier assigned to the logged data.
- Source Timestamp: Use the timestamp provided by the data source.
- Server Timestamp: Use the timestamp generated by the server when the data is received.
- Client Timestamp: Use the timestamp generated by the client when the data is sent.
- Value: Log the actual data value.
- Status Code: Log the status code associated with the data (e.g., success, failure).
- Custom: Specify a custom value for the data to be logged.

13.3. Data Loggers

Data Loggers are components responsible for capturing and recording data from the nodes of the address space. They facilitate the collection of data points. Use the Logging Profiles with type Data and configure collecting parameters for the Loggers.

Data Loggers
+ Add Logger

New Data Logger ENABLED

General settings

Name*

Collecting settings

Collecting mode* Interval (ms)* Storage mode*

Logging Profile

Profile*

Figure 112. Data Logger configuration form.

Enabled

Toggle to enable or disable the Data Logger.

Name

Enter a descriptive name for the Data Logger configuration to identify it within the application.

Collecting Mode

Choose the mode for collecting data.

- Subscription: Collect data based on subscription to OPC UA nodes.
- Polling: Collect data periodically by polling OPC UA nodes.
- History: If the logged node has HistoryRead enabled and the connection is disrupted, the missed values will be read and stored using the history read once the connection is re-established.
- PollingAndSubscription: Combines both approaches. Data is collected at least at the specified interval through polling, while also receiving updates more quickly when changes occur.

Interval (ms)

Specify the interval, in milliseconds, at which data is collected.

Storage Mode

Select the storage mode for logged data. Options include All, which stores all data collected and Delta, which stores only the data that is changed since the last logging.

Profile

Choose the Logging Profile configuration to be applied for logging data. This determines how data is formatted and where it is stored. Use profiles with Profile Type Data.

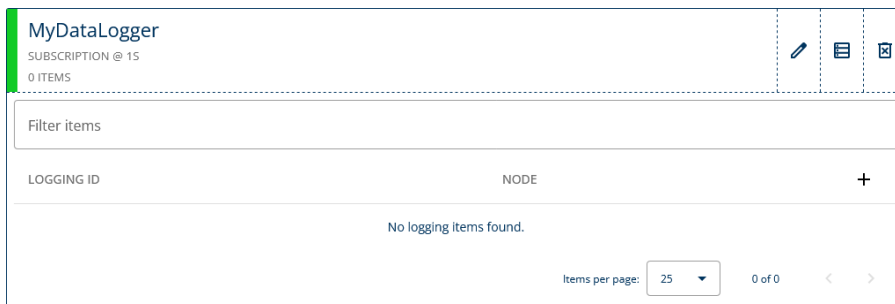


Figure 113. Add Nodes to the logger to start logging the values.

13.4. Event Loggers

Event Loggers are components responsible for capturing and recording data from the OPC UA events. Event loggers listens events of the configured objects. Use the Logging Profiles with type Event and configure collecting parameters for the Loggers.

Event Loggers

+ Add Logger

New Event Logger

General settings ENABLED

Name*

Collecting settings

Collecting mode* Interval (ms)* Storage mode*

Logging Profile

Profile*

Figure 114. Event Logger configuration form.

Enabled

Toggle to enable or disable the Event Logger.

Name

Enter a descriptive name for the Event Logger configuration to identify it within the application.

Collecting Mode

Choose the mode for collecting data.

- Subscription: Collect data based on subscription to OPC UA nodes.
- Polling: Collect data periodically by polling OPC UA nodes.

Interval (ms)

Specify the interval, in milliseconds, at which events are collected.

Storage Mode

Select the storage mode for logged event. Options include All, which stores all events collected and Delta, which stores only the events that is changed since the last logging.

Profile

Choose the Logging Profile configuration to be applied for logging data. This determines how data is formatted and where it is stored. Use profiles with Profile Type Event.

EventLogger

SUBSCRIPTION @ 15

0 ITEMS

Filter items

LOGGING ID	NODE	+
No logging items found.		


Items per page: 0 of 0

Figure 115. Add Nodes to the logger to start logging the values.

Next, add Nodes to the created Data Logger by pressing the list-icon and the the plus-icon. A new dialog will open with the following configurations.

MyDataLogger - New Logging Item

Item settings

NodeId*
nsu=http://www.prosysopc.com/OPCUA/Forge/EventManager;s=5a5e47c9 

Logging Id*
Severity

Figure 116. Add nodes to be stored with the logger.

NodeId

Enter the unique identifier of the Node to be added to the Data Logger configuration.

Logging Id

Provide a unique identifier or name for the logging instance associated with the Node. This identifier helps in tracking and organizing logged data within the Data Logger configuration.

14. OPC UA over REST

Forge offers RESTful openAPI with SwaggerUI. The API document can be accessed from:

<http://localhost:8080/swagger-ui/index.html#/>

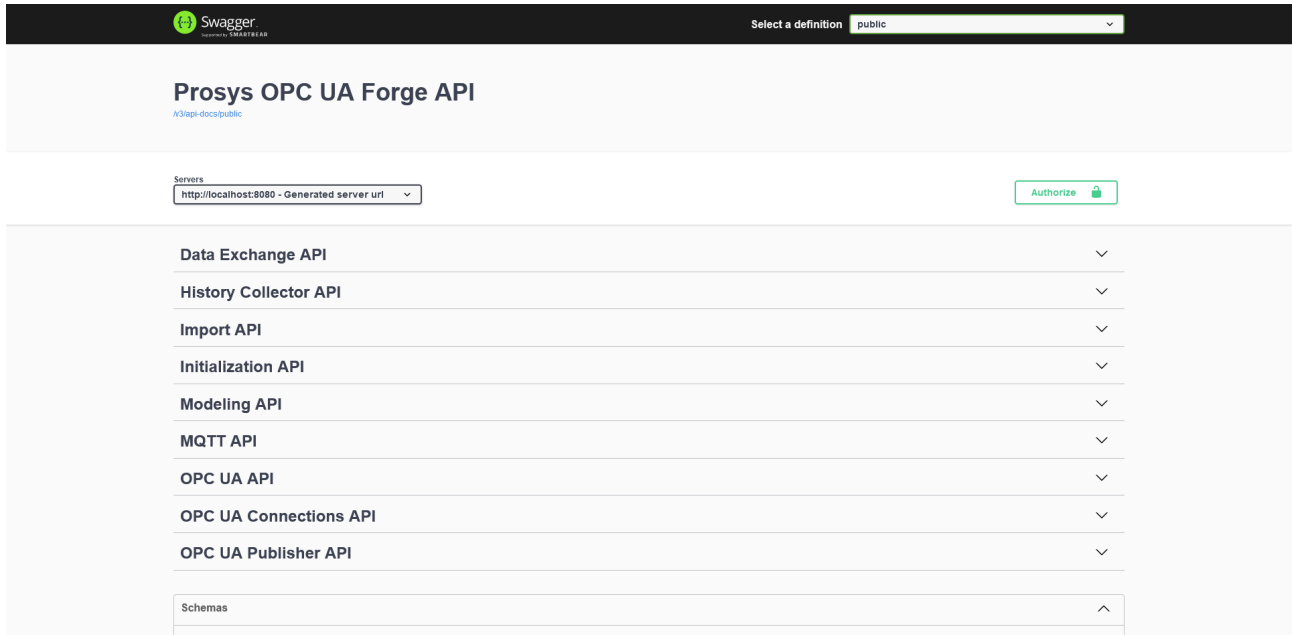


Figure 117. Users can interact with OPC UA services through a RESTful API.

Forge provides support for OPC UA over REST, allowing users to interact with OPC UA services through a RESTful API. The RESTful API allows users to perform various OPC UA operations, including reading and writing data, monitoring Nodes, and managing subscriptions. The manual will provide detailed instructions on how to structure requests and handle responses for these operations. This API is designed to be user-friendly, providing a standardized and accessible means of communication. The API is documented using Swagger. This documentation provides a comprehensive guide to the available endpoints, methods, parameters, and response formats.

14.1. Authorize

To authorize the usage of swagger, you need to give the user credentials. You can either use username and password or API key to authorize yourself. The username and password are the same as used to login to Forge. To give the user credentials, you must click the Authorize button.



Figure 118. Access the user credentials dialog with this button. User credentials are needed for testing the endpoints.

This will open a dialog, where you can fill out the information and click Authorize.

Available authorizations ✕

Authentication (http, Basic)

Username:

Password:

Figure 119. Username and password are the same that you use to login to Forge.

Now you are ready to test out the endpoints through the Swagger UI.

To authenticate with API key you need to first create API key from the UI. Click the user icon from the top right corner and select API key. Create new API key and copy it to the Authentication dialog in SwaggerUI.

es

🏠
👤

API keys + Add API key

New API key

API key settings

API key
6c7cp+2GXWY+KWPXmmkAy13dx85XjxWwH6n5aNI5pleQG+3qz73t64rKrqv7//ENI2Ag/4V/UEJbWRF/VzCA==

Description*
Admin-token

⚠ NOTE! The API key is available only on creation, copy it to a safe place!

- 👤 admin
- 🔑 Change Password
- 🔑 API keys
- 🚪 Logout

Figure 120. Create and copy API key from Forge.

Available authorizations ✕

Password:

ApiKey (apiKey)

Name: X-API-Key
In: header
Value:

Figure 121. Paste the API key to Swagger. Using API key to authorize can reduce the load of the authentication service.

14.2. OPC UA API

OPC UA API provides the users with an option to access Forge's address space Nodes using an API. This feature allows you to read and write variable values, access history data and call methods. With OPC UA API, it is straightforward to exchange Node data between Forge and your own programs.



The timestamp format used is: YYYY-MM-DDTHH:MM:SS.SSSZ, for example: 2024-09-20T08:10:50.885Z.

15. Advanced Configurations

The Configuration page is opened from the menu icon on the top right corner. The configuration module consists of five different sub-views, which can be navigated through the sub-navigation bar.

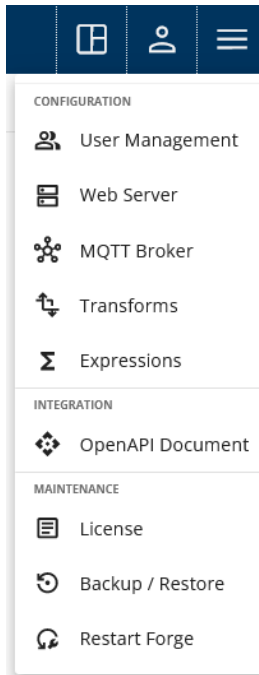


Figure 122. Open this menu from the menu icon from top right corner.

User Management

Manage user access and permissions

Web Server

Configure the Web Server binding address.

MQTT Broker

Configure the integrated MQTT broker configurations.

Transforms

Configure transformation dictionaries to be used in Forge.

Expressions

Configure expression templates to be used in address space.

OpenAPI Document

[Initialization API](#) Swagger API to initialize Forge. [Import API](#) Swagger API to import configurations to Forge.

License

Show and update your Forge license.

Backup / Restore

Create backup of the configuration or restore configuration.

Restart Forge

Restart Forge.

15.1. User Management

The User Management provides administrators with essential tools to manage user access and permissions within Forge. From individual user accounts to group management and global permissions

15.1.1. Users

Users sub-view within the User Management allows administrators to create, modify, and manage individual user accounts within Forge. By assigning specific roles and permissions to each user, administrators can tailor access levels to match organizational requirements, promoting collaboration and accountability across the platform.

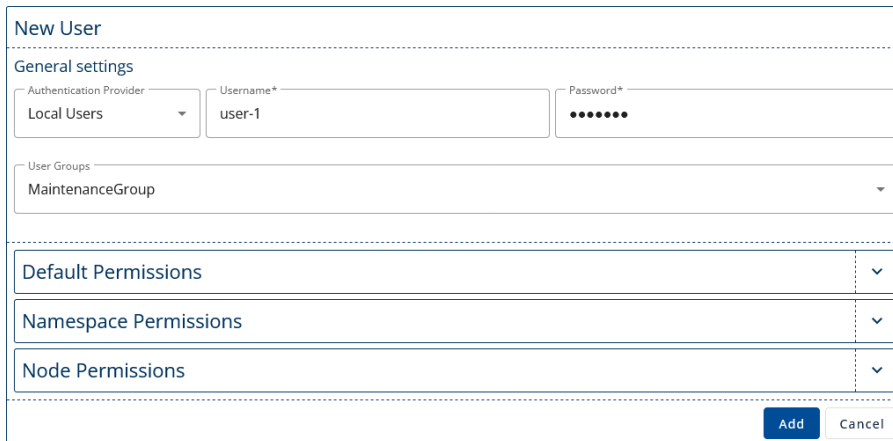


Figure 123. Add new users. Set user to group or give unique permissions.

Authentication Provider

Select Local User or another configured authentication provider.

Username

Enter the username for the user.

Password

Set the password for the user.

User Groups

Assign the user to one or more user groups to inherit the permissions.

Default Permissions

Define the default permissions for the user.

Namespace Permissions

Specify permissions for namespaces.

Node Permissions

Set permissions for individual Nodes.

15.1.2. User Groups

In User Groups, administrators can organize users into logical groups based on roles, departments, or other criteria. By grouping users together, administrators can streamline permission management,

making it easier to grant or restrict access to resources and functionalities across Forge.

Figure 124. Create group to handle permissions for specific roles.

Group Name

Enter the name for the user group.

Default Permissions

Define the default permissions for the group.

Namespace Permissions

Specify permissions for namespaces.

Node Permissions

Set permissions for individual Nodes.

15.1.3. Providers

The Providers section enables administrators to integrate external identity providers, such as LDAP and Active Directory, with Forge, facilitating centralized authentication and user management. By leveraging these external identity providers, organizations can enhance security, simplify user on-boarding processes, and ensure consistency across authentication mechanisms.

Figure 125. Configure LDAP as an authentication provider.

Provider Type

Select LDAP.

Name

Provide a name for the LDAP authentication configuration.

URL

Enter the URL of the LDAP server.

System Username

Specify the username for accessing the LDAP server.

System Password

Provide the password for accessing the LDAP server.

User DN Template

Define the template for constructing the user's distinguished name (DN).

New Authentication Provider

General settings

Provider type* Name*

Active Directory settings

URL*

System username System password

Principal Suffix*

Search Base*

Search Filter

Figure 126. Configure LDAP as an authentication provider.

Provider Type

Select Active Directory.

Name

Provide a name for the Active Directory authentication configuration.

URL

Enter the URL of the Active Directory server.

System Username

Specify the username for accessing the Active Directory server (if applicable).

System Password

Provide the password for accessing the Active Directory server (if applicable).

Principal Suffix

Enter the principal suffix for the Active Directory server.

Search Base

Define the search base for Active Directory queries.

Search Filter

Specify the search filter for Active Directory queries.

15.1.4. Global Permissions

Global Permissions govern universal access rights across Forge, enabling administrators to establish rules for user interactions. These rules includes Forge permissions, OPC UA permissions, namespace permissions, and Node permissions, ensuring comprehensive control over user access and actions.

Default Permissions			
Forge Permissions		OPC UA Permissions	
PERMISSION	ENABLED	PERMISSION	ENABLED
Administrate	<input checked="" type="checkbox"/>	Browse	<input checked="" type="checkbox"/>
Edit	<input checked="" type="checkbox"/>	Call	<input checked="" type="checkbox"/>
View	<input checked="" type="checkbox"/>	Delete History	<input checked="" type="checkbox"/>
		Insert History	<input checked="" type="checkbox"/>
		Modify History	<input checked="" type="checkbox"/>
		Read	<input checked="" type="checkbox"/>
		Read History	<input checked="" type="checkbox"/>
		Read Role Permissions	<input checked="" type="checkbox"/>
		Receive Events	<input checked="" type="checkbox"/>
		Write	<input checked="" type="checkbox"/>
		Write Attribute	<input checked="" type="checkbox"/>
		Write Historizing	<input checked="" type="checkbox"/>
		Write Role Permissions	<input checked="" type="checkbox"/>

Figure 127. Customize the permissions to enhance Forge usage.

Namespace Permissions

NAMESPACE +

<http://opcfoundation.org/UA/DI/> ✎ ✕

Items per page: 25 1 - 1 of 1 < >

Figure 128. Restrict the usage of namespaces using these permissions.

Node Permissions

NODE +

[Data Sources \(nsu=http://www.prosysopc.com/OPCUA/Forge;s=Data Sources\)](http://www.prosysopc.com/OPCUA/Forge;s=Data Sources) ✎ ✕

Items per page: 25 0 of 0 < >

Figure 129. Complete the permissions, including permissions for a single Node.

15.1.5. Change Password

By clicking the user-icon next to the menu icon in the top-right corner, a menu will open where you can select Change password.

15.1.6. Copy API Key

By clicking the user-icon next to the menu icon in the top-right corner, a menu will open where you can select API keys. Here, you can create and manage API keys for the current user.

15.2. Web Server

Manage the Web UI binding address. Note that Forge must be restarted for these changes to take effect.

Web Server Configuration

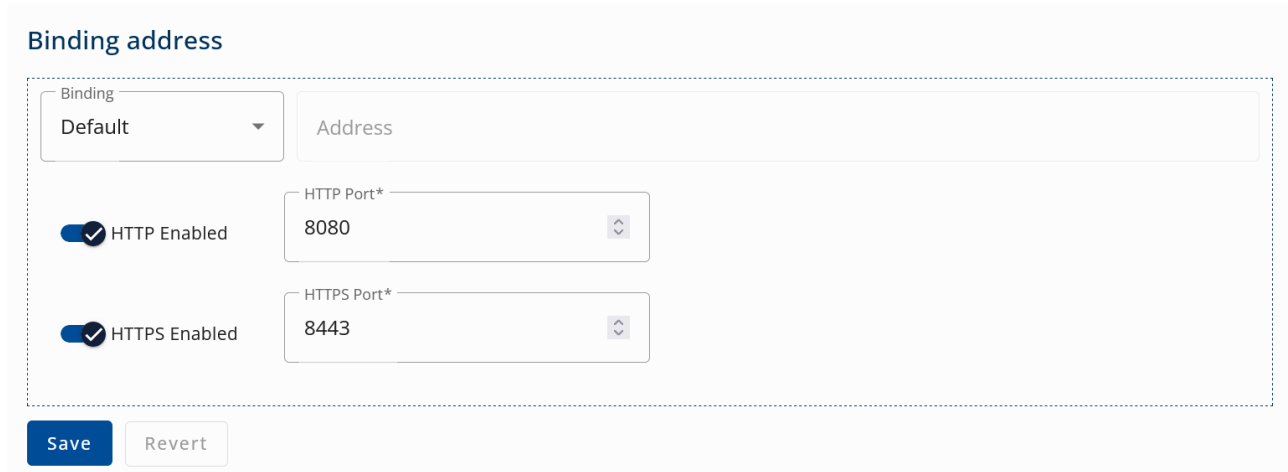


Figure 130. Define the Web UI binding address.

Binding

Gives different preset values for address.

- Default:
- Any: 0.0.0.0
- Localhost: 127.0.0.1
- Custom: Write the IP address.

Address

Can have preset values or custom value.

HTTP Port

The port which the Web UI will use in HTTP. Enable/disable the HTTP with the slide toggle.

HTTPS Port

The port which the Web UI will use in HTTPS. Enable/disable the HTTPS with the slide toggle.

15.3. MQTT Broker

The configuration of the MQTT broker within Forge involves specifying a variety of parameters that control its behavior and capabilities. Below is a detailed explanation of each parameter.

MQTT Broker Configuration

MQTT Broker
Enabled

Broker Address

Port*

Service Configuration

Enable In-Memory Persistence
 Enable Retained Messages

Maximum Quality of Service*

Queued Messages Discard Strategy*

Subscription Configuration

Enable Subscription Identifiers
 Enable Wildcard Subscriptions
 Enable Shared Subscriptions

Per Client Configuration

Queued Messages*

Server Receive Maximum*

Max Topic Aliases Per Client*

Per Packet Configuration

Max Packet Size (bytes)*

Max Keep Alive*

Allow Unlimited Keep Alive
 Enable Topic Aliases

Time to Live Configuration

Message Expiry Max Interval (s)*

Session Expiry Max Interval (s)*

Security Configuration

Allow Empty Client Id
 Payload UTF-8 Validation
 Topic Name and Client ID UTF-8 Validation

Allow Requesting Problem Information

Figure 131. Define the MQTT broker.

Enabled

Start or stop the MQTT broker using this toggle.

Port

The port which the MQTT broker will be using.

Enable In-Memory Persistence

By default persistence is done to disk, so they are not lost even if broker restarts. In-Memory for memory based persistence. Persistent clients don't lose their granted subscriptions, even if they are offline and reconnect.

Enable Retained Messages

Determines whether the broker will store the last message sent to a topic and deliver it to new subscribers immediately upon subscription.

Maximum Quality of Service

This selects which QoS level is the highest. The choice of QoS level impacts the trade-off between message delivery assurance, protocol overhead, and network efficiency. QoS 2 has the highest protocol overhead and QoS 0 minimal overhead.

- Exactly Once (QoS 2), message is delivered exactly once.

- At Least Once (QoS 1), message is delivered at least once. Duplicates can occur.
- At Most Once (QoS 0), message is delivered at most once or not at all.

Queued Messages Discard Strategy

The discard strategy when the maximum amount of queued messages is reached. Discard Newest for discarding new messages. Discard Oldest for discarding the oldest queued message when a new message arrived (higher performance impact).

Enable Subscription Identifiers

Determines whether subscription identifiers are used, which can enhance subscription management and message flow analysis.

Enable Wildcard Subscriptions

Determines whether a wildcard subscription is allowed. A wildcard subscription is a subscription with a topic filter that contains wildcard characters (# and +)

Enable Shared Subscriptions

Enables multiple clients to share a subscription, allowing them to distribute the message load among themselves.

Max Packet Size (bytes)

Specifies the maximum size of any MQTT packet in bytes that will be accepted by the broker. Adjust according to network capacity and expected payload sizes.

Max Keep Alive (s)

Specifies the maximum time in seconds that the broker allows between communications with a connected client. If the client does not send a message within this period, the broker closes the connection.

Allow Unlimited Keep Alive

Determines whether the broker accepts connections from clients that send a CONNECT packet with a keepAlive=0 setting, disabling the keep-alive mechanism.

Enable Topic Aliases

Determines whether topics can be substituted with an alias to reduce packet size. Topic aliases must be numbers between 1 and 65535.

Message Expiry Max Interval (s)

The maximum duration (in seconds) that a message will be stored by the broker before it expires.

Session Expiry Max Interval (s)

The maximum duration (in seconds) that the broker stores session state after a client disconnects. Adjust based on the expected frequency of client reconnections.

Allow Empty Client Id

Determines whether the broker allows clients to connect without specifying a client ID.

Payload UTF-8 Validation

Enables or disables validation that the payload of published messages is valid UTF-8. Useful for applications that require strict adherence to UTF-8 encoding.

Topic Name and Client ID UTF-8 Validation

Determines whether validation is performed to ensure that topic names and client IDs are valid

UTF-8 strings.

Allow Requesting Problem Information

Determines whether clients are permitted to request reason strings and user property values from the broker. Useful for debugging and detailed error reporting.

15.4. Transformation Dictionaries

The Transformation Dictionary feature enables the value conversion. By defining specific mappings within the dictionary, users can easily translate data from one format to another, facilitating interoperability and compatibility across different systems and protocols. Utilize Transformation Dictionaries in attribute [Mapping](#), [Data Exchange](#), and [MQTT](#) modules, ensuring smooth data exchange and accurate representation across various applications and environments.

Begin by creating new Transformation Dictionary by clicking the +Add Dictionary.

Figure 132. Add new Transformation Dictionaries.

Name

Enter a descriptive name for the Transformation Dictionary.

After that, you can add items to the Transformation Dictionary.

Figure 133. Source value is transformed to target value.

Source value

The value that is obtained or retrieved.

Target value

The value to which the received value is converted.

Apply these Transformations Dictionaries in other modules.

15.5. Expressions

Manage all your Expression templates from this view, enabling you to create, edit, and remove templates with ease. This centralized interface simplifies the management of your templates, ensuring

your expressions are organized and up-to-date. Usage of the Expression templates is instructed in [Expression](#).

Expression Templates + Add Template

New Expression Template

General settings

Name
complex

Expression

Expression
let a = node
a * b - a

Variables

node b

Add Cancel

logical ✎ ✕

IF (A && B) TRUE ELSE IF (B && C) TRUE ELSE FALSE

sum ✎ ✕

A + B

Figure 134. Create complex templates and utilize them in address space.

Name

Enter a descriptive name for the Expression template.

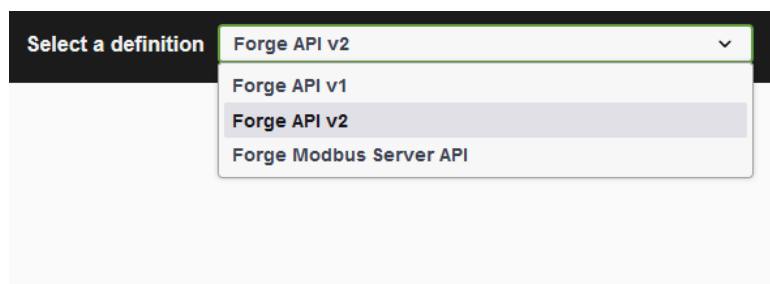
Expression

Enter the JEXL expression to define the logic.

15.6. OpenAPI Document

A new tab will open in your browser, displaying the OpenAPI documentation.

Forge offers full visibility to its API, allowing you to manage your Forge server completely with your own REST methods. The API document is grouped into the modules. Forge has three API documents which you can find from the top right corner of the page.



Authorize

Figure 135. Change between the different APIs.

15.6.1. Initialization API

Initialization is required at the beginning of the first time usage of Forge. The initialization method loads the license into correct place and user credentials are created.

15.6.2. Import API

Forge simplifies the configuration process by allowing users to import multiple configurations in a single operation. This not only saves time but also ensures consistency and accuracy in the configuration process. The first step in the import process is to load the import template. Once the template is loaded, users can fill it out with the relevant configuration details. After filling out the template, users can upload it to initiate the import process. The import files can either be in JSON or CSV format.

Import API is especially useful when you want to operate with a large number of Nodes, where configuring everything by hand would be very time-consuming. CSV files can easily be edited to contain even thousands of rows each corresponding to a single import operation. This way creating numerous Nodes and using them all for example in Data Logger, MQTT Publisher or Data Exchanger modules becomes fast and convenient.



API can be accessed from, <http://<hostname>:8080/swagger-ui/index.html#/>

15.7. License

The License view provides users with a detailed overview of their current license content, enabling them to verify and manage their licensing information. Additionally, this view allows users to upload new license. Press the Load button to upload new license.

Forge License

E-MAIL	jiamu.ni@prosysopc.com
ORGANIZATION	Eval_August
LICENSE TYPE	Evaluation
SERIAL NUMBER	e001007
START DATE	May 28, 2024
EXPIRY DATE	Aug 31, 2024
LICENSE KEY	<input type="button" value="Show Key"/>

Figure 136. See all the details of the current Forge license.

15.8. Backup / Restore

The Backup / Restore view enables you to create backups of your current configurations, providing a safeguard against data loss and configuration errors. In this view, you can also restore previous configurations from existing backups, allowing for quick recovery and rollback to a known good state. This ensures the stability and continuity of system operations by maintaining a reliable configuration management process.

Configuration Backup / Restore

Download a complete backup of the Forge configuration.

Backup

Restore a backup of the Forge configuration. A restart is required after restoration.

Restore

Close

Figure 137. Select to backup or restore the Forge configuration.

15.9. Restart Forge

Clicking the **Restart** button will initiate a server restart, which may take some time since all the settings need to be applied and connections to be re-established. Once the restart is complete, the application should function normally. A restart is required to apply changes to the configuration, for example.

Restart Forge

Are you sure you want to restart Forge?

NOTE: Restarting might take up to 1 minute, reload the UI after a moment.

Confirm

Cancel

Figure 138. Confirm restart from the dialog.

16. Appendix

16.1. Manual Port Configuration

Forge's default port settings are stored in two configuration files: `web-server-configuration.json` and `server-config.json`. These files are generated during Forge's initial start-up. To modify the port numbers, follow these steps:

1. Start Forge: Run Forge once to generate the necessary configuration files.
2. Open a Text Editor with Administrator Privileges: Ensure you have the necessary permissions to modify system files.
3. Locate and Open the Configuration Files: Navigate to Forge's [Configuration Folder](#), and open `web-server-configuration.json` and `server-config.json` in the text editor.
4. Modify the Port Numbers: Change the port numbers as needed within the configuration files.
5. Save the Changes and Start Forge: After saving the modifications, start Forge to apply the new port settings.

16.2. File Locations

This section explains important file locations.

16.2.1. Configuration Folder

The configuration files for the Prosys OPC UA Forge are stored in different locations, depending on the way it was installed. Default locations are:

	Configuration folder
Windows	C:\ProgramData\ProsysOPC\Forge
Linux	/etc/ProsysOPC/Forge
Docker	<installation_folder>/data



<installation_folder> refers to the location where the docker image ZIP was extracted.

16.2.2. Reset Configuration

To reset Forge to its default settings, simply delete the Configuration Folder mentioned above. Upon the next startup, Forge will automatically recreate the folder with default settings, restoring the application to its original configuration state. This streamlined process ensures quick and hassle-free restoration of default settings when needed.

16.2.3. Configuring Embedded Devices

As the configuration is based on JSON files, you can always modify the configuration files directly on an embedded device, even when it does not have a graphical user interface. However, the suggested way to initialize the configuration of a Docker Installation in an embedded device (a computer without graphical user interface), is to create the configuration in a Standard Installation and copy it from there to the Docker Installation (as mentioned above).

16.2.4. License File

Forge's license file is stored in the [Configuration Folder](#) with name *license.lic*. The license contains license keys for the features and only the features which are included in the license are enabled. License can be bound to specific hostname and it can have an expiration date.

16.2.5. Log File

Log file is often helpful for debugging issues. Log is written daily in the same file if the file size limit is not exceeded. The logging level can be changed from INFO to WARN or ERROR by modifying the file `logback.xml` in the Forge folder. This is done by changing the level parameter in the following line: `<logger name="com.prosysopc.ua.app.forge" level="INFO" additivity="false">`.

	prosys-opc-ua-forge.log location
Windows	C:\ProgramData\ProsysOPC\Forge\log
Linux	/var/log/ProsysOPC/Forge
Docker	<installation folder>\data\log

16.2.6. Certificate Folder

Certificate folder is located in Configurations folder.

	Location
Windows	C:\ProgramData\ProsysOPC\Forge\PKI\CA
Linux	/var/log/ProsysOPC/Forge/PKI/CA
Docker	<installation folder>\data\Forge\PKI\CA

The CA folder has the following sub-folders:

certs

For the trusted certificates

crl

For certificate revocation lists

private

For own certificate

rejected

For the rejected certificates

16.2.7. Installation Folder

By default, the application is installed to the folder:

	Installation folder location
Windows	C:\Program Files\Prosystech\Prosystech OPC UA Forge
Linux	opt/prosystech-opc-ua-forge
Docker	<installation_folder>

16.3. Manage Forge Service

16.3.1. Windows

Follow these steps to start the Forge service in Windows:

1. Open Windows search and write "services"
2. Select the Services app
3. Find "Prosystech OPC UA Forge" from the list of services
4. Right-click "Prosystech OPC UA Forge"
 - a. **Start**: Start the service
 - b. **Stop**: Stop the service

16.3.2. Linux

Manage Forge service from the terminal with commands:

```
sudo service prosystech-opc-ua-forge-service start
```

Start the Forge service.

```
sudo service prosystech-opc-ua-forge-service status
```

Check the status of the Forge service.

```
sudo service prosystech-opc-ua-forge-service stop
```

Stop the Forge service.

16.3.3. Docker

You can use the start-up script that is available under the `<installation_folder>/bin`.

The script can perform five different actions:

start

Start Forge in the Docker container. Loads the image first if it is not loaded yet.

stop

Stop and remove the Docker container.

load

Load the Docker image.

unload

Unload the Docker image. Stops the container first.

status

Current status of the Docker image, Docker container and OPC UA server.

Use `.\forge-docker.ps1 <parameter>` or `forge-docker.sh <parameter>` to perform the action in the Docker Application in Windows or Unix-based systems, respectively.



The Docker requires Administrator privileges in Linux, so you have to run the commands with `sudo`.

The Docker package contains *readme.txt* which has more instructions for docker usage.

16.4. Uninstalling the Application

16.4.1. Windows

On Windows the application can be uninstalled through the Control Panel or the Apps & features menu, or optionally with the uninstaller that is located in the installation folder.

16.4.2. Linux

On Linux, open the terminal and navigate to the installation folder (default folder is `opt/prosys-opc-ua-forge`) and use the command `sudo ./uninstall`.

16.4.3. Docker

On Linux distributions, please follow the official Docker Uninstallation instructions.

CentOS: <https://docs.docker.com/engine/install/centos/#uninstall-docker-engine>

Debian: <https://docs.docker.com/engine/install/debian/#uninstall-docker-engine>

Fedora: <https://docs.docker.com/engine/install/fedora/#uninstall-docker-engine>

Ubuntu: <https://docs.docker.com/engine/install/ubuntu/#uninstall-docker-engine>

On Windows, you can uninstall the Docker Desktop through the Control Panel.

16.5. Contact Us!

16.5.1. Support

If you need support or have any questions, please contact us using the following options.

Evaluation License Users

Include your name and email address, and if available also your organization's name, to help us get started. Expected response time is one to three business days.

Commercial License Users

Include the serial number of your license (or the actual license key) to ensure the fastest possible service. Expected response time is one business day.

- forge-support@prosysopc.com
- sales@prosysopc.com
- +358 9 420 9007 (CET 08-16, Monday to Friday)

16.5.2. Find more about us

Check out our website and social media accounts.

- www.prosysopc.com
- LinkedIn: Prosys OPC Ltd
- Youtube: @prosysopc

Find our other OPC UA native products from our website (<https://prosysopc.com/products/>).