OPC Router 5.5
User Guide

inray Industriesoftware GmbH

Holstenstraße 40

25560 Schenefeld

Edition of 2025/11/06

Software Release: 5.5

© Copyright 2025 inray Industriesoftware GmbH

Inhalt

Introduction
  licensing
  notation
  program_structure
  copyright
Installation/Update
  system_requirements
  Installation Windows
  Deployment Docker
  Edge Device Management
  Update/Maintenance
Licensing
  OPC Router license portal
  license-levels
  activate-license-in-the-opc-router
  licensing-via-environment-variables
  license-overview-in-the-opc-router
  delicensing
security_guidelines
Architecture Recommendations
  basic-concepts
  architecture-for-small-use-cases
  edge-device-architecture
  scaling-to-medium-sized-solutions
  Architecture for Enterprise Solutions
Settings
  notification_group
  status-storage-options
  certificate_management
  Redundancy
  store_and_forward
  dynamic_instance_api
  dependency-storage
  User management
  Versioning
Plug-ins
  Cloud / IoT
  Topfloor
  Shopfloor
  Storage
  Advanced
  Messaging
  Printer
  Additional interfaces
  Additional functions
FAQ
  escalation-levels
  Error Diagnostics
  Connections
  How can I ...
  Best Practice
  additional-settings-for-logging-in-via-windows-user
  create-ssl-certificates
  Performance analysis
Status Indication
  service-status
  plug-in-status
  connection-status
status_indication
Network configuration
  user_account_service
  Distributed OPC Servers/DCOM
  distributed_databases
  distributed_mail_server

Introduction

Dear reader,

the OPC Router is a software platform for easy, freely configurable data exchange with a wide range of connection options in an intuitive web interface and drag-and-drop handling without additional programming knowledge.

If you want to use multiple OPC Router services, you can manage them via the central management.

The licenses for the OPC Router are unlimited in the number of configured connections and in the number and scope of data transfers. Data transfers can be triggered by various triggers.

This manual will show you the basics of OPC Router configuration. Please refer also to manuals of systems connected to OPC Router, e. g. to grant OPC Router access to these systems. If you need help setting up a plug-in or using a transfer object or trigger, you can always request documentation for it by clicking the button in the respective window. On the "Welcome Page" you will find a link to the online documentation as well as other useful links, e.g. to our "Knowledge Base".

We apologize for not providing the complete manual in English yet. Please contact us to obtain the latest translation.

inray Industriesoftware is always willing to provide training courses as well as configuring OPC Router according to your needs. Please contact us for further information and offers.

Yours team of inray Industriesoftware

Licensing models

The OPC Router can be fully tested over a period of 2 hours. For productive use, you have the choice between two license models, Subscription and Perpetual.

Subscription

The Subscription license provides three options:

- Enterprise
- Standard
- Plain

This license has a specific time duration, and upon expiration, a new license must be activated.

Perpetual

A license with this model can be regarded as a full license. After expiry of the time frame, the service continues to run, but your entitlement to support and software updates ends.

Note: Further information on licensing can be found in chapter licensing-1.

About the manual

This manual uses the following notations:

C:\Program files\inray\ log_1.txt http://localhost/	File names and paths, URL
Datasource	Names of parameters and objects' properties.
<Datasource>	Place holder to be filled during development. (Delete brackets < *and* >!)
`... Dim TextHandle TextHandle.SetCoord 10,10 ...`	Source code samples and templates
“Change”	Names of controls, tabs, buttons etc. within GUI.
File–New or File - New	Menu/Submenu entries.

Terms used

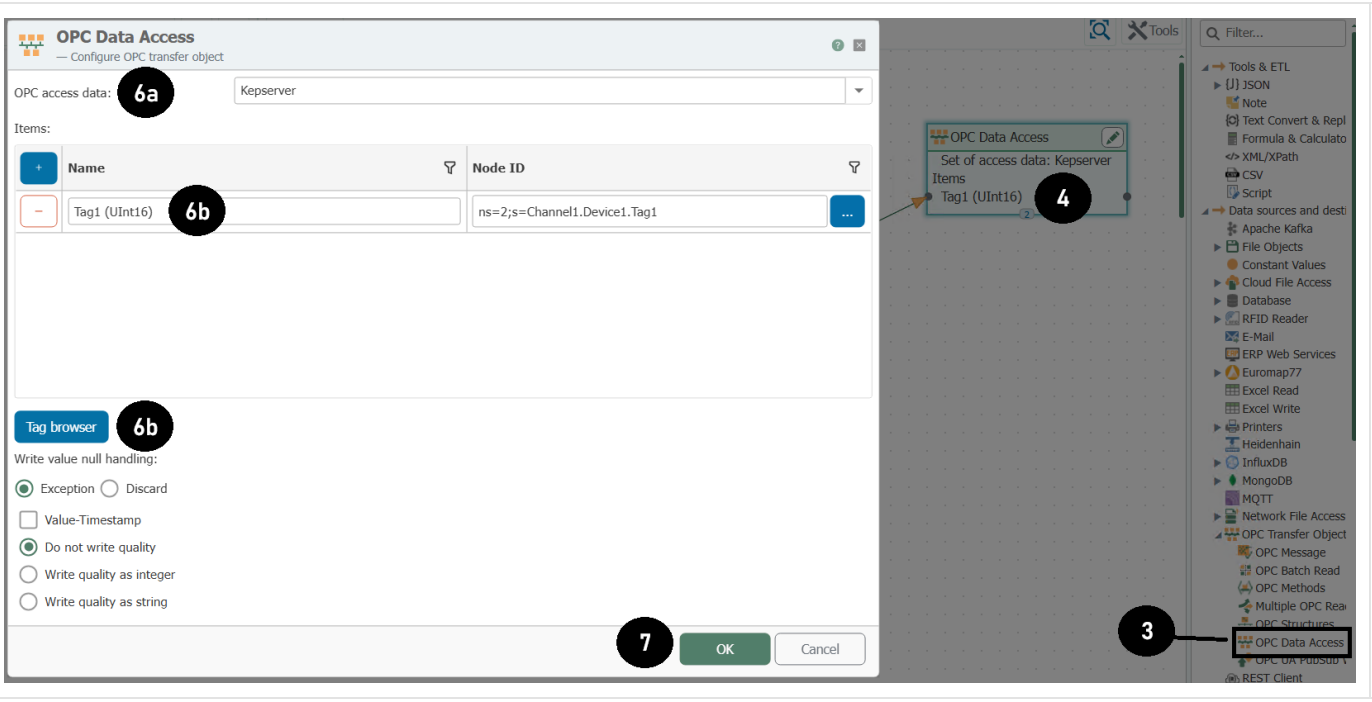
(set of) access data	Technically speaking: instance of a plug-in, e. g. the access data to one specific OPC Server, a specific database server etc. as in OPC access data, database access data etc
element (of a transfer object)	Part of a transfer object, e. g. a table's column within a database transfer object or an OPC item within an OPC transfer object or a specific variable used. Elements (sometimes also called items) are connected during development to define a transfer.
item, transfer object item	see “element”. Not to be mixed up with OPC item, so mostly element is used.
configuration	“Configuration” means (as a rule) the basic settings and the creation of access data within the OPC Router Configuration (GUI). Whereas “development” or “projection” means defining connections (also via OPC Router Configuration GUI).
plug-in	Plug-ins provide the Router's functions, e. g. connection to OPC, database or mail servers, to ERP systems, as well as use of intern variables etc. Most plug-ins have to be licensed separately. The manual uses the term for all sets of access data of a specific plug-in also.
plug-in instance	see “access data”.
development	see “projection”.
projection	The process of defining connections (projection of transfers). i. e. the process of defining sources, destinations, triggers as well as (de-)activating. Also referred to as “development” in contrast to “configuration”. However, OPC Router is both configured and projected via OPC Router Configuration (GUI).
snap-in	The dialogue left of the display
transfer	Data transfer during runtime. During projection the same as “connection”.

transfer object	Source, destination or trigger of a transfer.
variable	As a rule, an instance of the variable plug-in, i. e. a variable to be used within transfers. Other variables carry special names like template variable or placeholder.
connection	Combination of at least one data source, one destination and one trigger. A connection defines how a transfer is executed: Which data is transferred under which condition to which destination.

Images

Some of the screenshots present several numbered mouse pointers. The numbers refer to the instructions in the text. Therefore, numbers might have been left out. The following example shows steps 3 to 7 of the specific chapter.

Several pop-up dialogs might be shown side by side, even if they open one after another, just to show the results of mouse interaction:



Project planning of an OPC data source

Overview

The OPC Router 5 Management contains the projecting, the option for making changes productive and a status display at runtime.

What you will also find there:

- The router service that processes the data transfers as a runtime component

The Management is being accessible via web browser (Default address: http://localhost:5000/)

The *OPC Router Management interface* in the browser consolidates all steps for commissioning the OPC Router, including licensing:

1. *Configuration of needed plug-ins*, e. g. access data of accessible OPC and database servers, with connection details, definition of variables (Selection on the left in the snap-in directory under "Plugins", editing on the corresponding tab).
2. *Projecting of the connections:* Which data should be transferred when, from where and to where? (Selection and creation via the snap-in directory under "Connections", selection of transfer objects on the right, editing of connections in the main window)
3. Deployment of the whole project or individual connections to production.
4. Starting the service – if the service is running (default), the mandatory restart is executed automatically. (button)
5. Monitoring and error diagnosis on the left via the snap-in directory under "Status".

The following sections of the manual cover the individual steps in detail. Please note that only the licensed plug-ins are available to you.

The entire project, including the transfer protocol, is stored in local databases (MongoDB).

Remember: Project backup: If the entire computer (VM, disk backup) is not backed up, the program directory of the router where the database files are located should be backed up.

The router service or service is the runtime component that processes the data transfers in the background as a Windows service. Individual configuration is usually only necessary when used in distributed systems with Windows authentication.

Copyright

This manual and the software described are subject to licensing regulations and protected by copyright laws. Manual and software are to be used only according to the licences granted. Information within this manual are valid with reservation only and might be changed without prior information.

Microsoft, Windows, SQL Server, Excel are registered trademarks of Microsoft Group of Companies. SAP, SAP-System, ABAP, BAPI, R/3 are trademarks or registered trademarks of SAP AG. Oracle is a registered trademark of Oracle Corporation. All product and company names in this manual may be trademarks of their registered owners even if they are not listed here.

Installation/Update

OPC Router Setup will install OPC Router including management GUI, Windows Service, OPC Router Status (runtime supervision) including all plug-ins in trial mode (2 hours runtime). WebView2 is also installed with the setup. For this you need an active internet connection. After the license key has been entered, all plug-ins licensed will run unlimited while plug-ins not licensed will be deactivated.

OPC Router Setup will neither install database servers nor OPC servers to connect your PLCs. These components have to be installed separately. OPC Router's purpose is only to connect those components.

System Requirements

Windows System:

- .NET 8 (will be installed if necessary)
- **Hardware:** We strongly recommend running server applications on adequate hardware (multi-core processor, at least 8 GB RAM and SSD storage).
- **Operating system:** We recommend Microsoft Windows Server operating systems. For test purposes also client operating systems (64 bit) can be used.

  The following Windows versions are supported:

  - Windows Server 2016 x64
  - Windows Server 2019 x64
  - Windows Server 2022 x64
  - Windows Server 2025 x64
  - Windows 10 x64
  - Windows 10 IoT x64
  - Windows 11 x64

A web browser is required to display the OPC Router 5 Management. Supported web browsers are:

- Google Chrome
- Microsoft Edge

In the versions currently available at the time of the OPC Router 5 release. (At least version 100)

Please check your local firewall if a connection to the OPC Router 5 Management is possible. By default the OPC Router Management is installed on port 5000 (http://localhost:5000). If this port is already in use, the next free port is used. (e.g. 5001)

Attention: Productive operation with client operating systems: Due to technical limitations, inray explicitly recommends operation with server operating systems.

Please note: Client operating systems, such as Windows 10, may experience unwanted behavior, unplanned reboots, and incomplete records due to unforeseen forced upgrades. Other technical limitations, such as the limited number of incoming TCP / IP connections, especially in large projects, lead to downtime of connections. Please inform yourself before installation, if an operation of the OPC Router on a client operating system meets their requirements and is compatible with the license right of Microsoft.

Attention: When using a virus scanner, we highly recommand to define an exception for the OPC Router directory in the virus scanner settings. Virus scanners can impair the performance of the system and may incorrectly categorise the files of the OPC Router as threatening. Such false alarms can be triggered by updates to the virus defintions and, in the worst case, lead to an unexpected production shutdown.\ By setting an exception for the OPC Router directory, you minimise the risk of performance losses and prevent false positive detections that could disrupt operations.

Docker Systems:

- SingleCore 600MHz 64 Bit CPU, the architecture requirements may differ depending on the image:
  - Runtime Image (with integrated MongoDB)
    - x86_64 Processors with AVX instruction set.
    - ARM64 Processors from ARMb8.2-A
  - Service Image (without integrated MongoDB)
    - x86_64 processors with AVX instruction set
    - ARM/ARM64 processors from ARMv7 onwards

- 512 MB Ram
- 3 GB free disk space

The OPC Router uses an internal database to display data transfers in real time and historically.

For optimal performance, the OPC Router should be run on a system with a **multi-core CPU** and at least **4 GB RAM.**

Installation Windows

Important: Before installing, check that all system requirements are met. Familiarize yourself with the security recommendations and in particular follow the steps below.

The OPC Router Setup installs all necessary components and configures by default as follows:

- the Windows service with the start type "Automatic", i.e. the runtime component responsible for data transfers is started automatically every time the computer is started. The router service does not run immediately after installation, as no configuration has been carried out yet. After configuration (and after every update), start the service manually once using the button in the router configuration.

Note: Manual installation of certificates if there is no Internet connection

If the server on which the OPC Router is installed does not have an Internet connection, in rare cases the installation may fail due to CAB or certificate errors. In such situations, it may be necessary to manually install the necessary certificates in order to complete the installation successfully. Please contact your product support for this.

Installation

- Back up the system.
- Since the OPC Router setup may update or install Windows system components, please close all open programs before starting the setup.
- Check that all system requirements are met.
- Run the setup.exe file from your installation source.

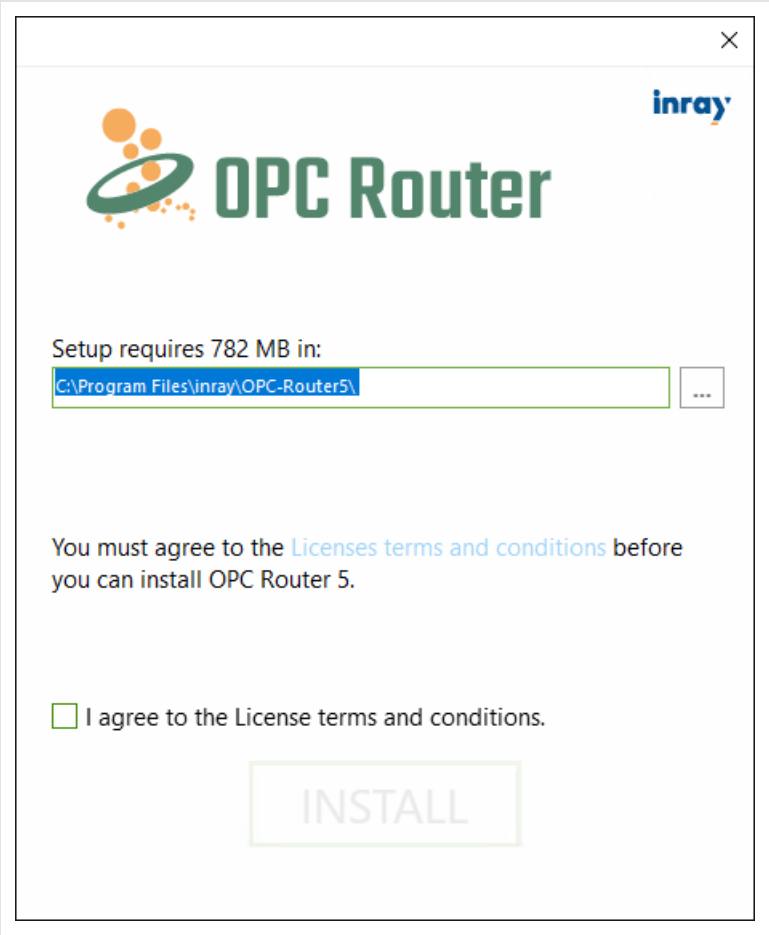
The installation file can be started by double-clicking.

Installation note: Administrator rights required

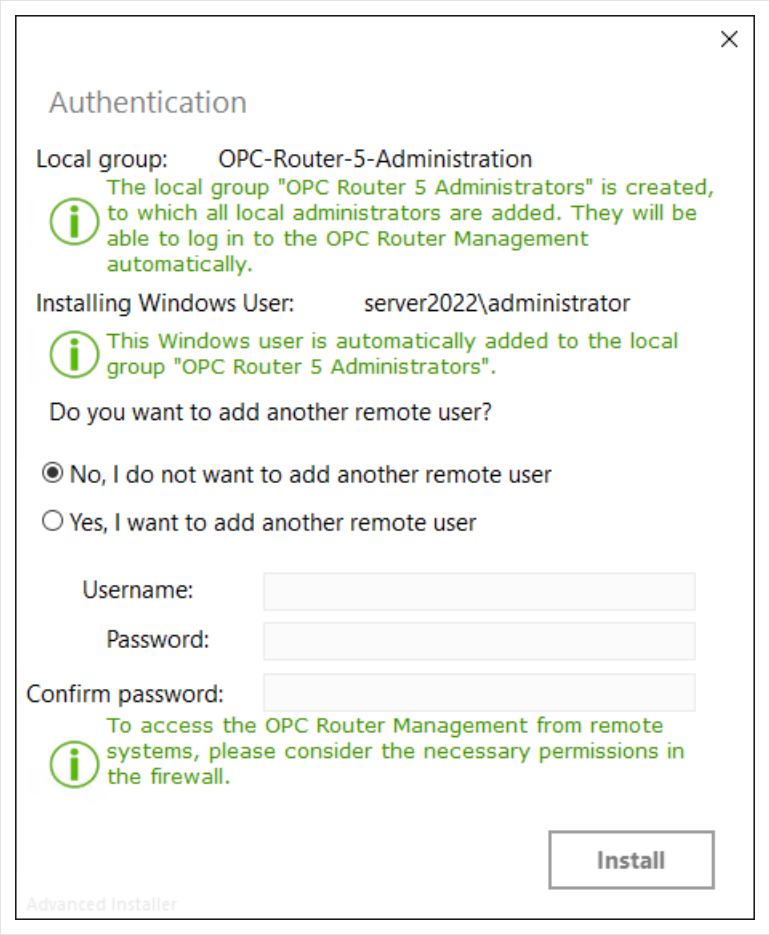
It is not necessary to be logged in as an administrator to start the installation of the OPC Router. However, please note that administrator rights are required during the installation. These rights will be requested at some point during the installation process.

Please make sure that you have access to an administrator account or can obtain the necessary permissions before proceeding with the installation. This will ensure that the installation runs smoothly and that all necessary components can be installed and configured correctly.

If you encounter access restrictions during installation, please contact your system administrator or support to obtain the necessary rights.



You can now specify the desired installation path. To install the application, you must agree to the license terms. Clicking the "Install" button takes you to another configuration screen.



Here you can create a remote user if required. A remote user is a normal user that is created in Web Management and enables

you to log in without using Windows authentication, for example to access Web Management from a remote system. Clicking the "Install" button will automatically perform the installation.

The remote user is equivalent to the initial user when installing via Docker. It is an internal user with the role *Admin* (see user roles) who is always restored after restarting the management service.

After successful installation, a shortcut should have been created on the desktop that you can use to conveniently open Web Management.

Windows users of the local group "OPC Router 5 Administrators" have the *Admin* role when logging into Web Management (see user roles).

Configuration of anti-virus programs

The database files for MongoDB are compressed. Scanning these files with a virus scanner can have a noticeable negative effect on performance. Therefore, an exception should be set up in the antivirus program for the installation directory of the OPC router so that it will not be scanned.

Create backup

You can make a backup of the OPCRouterConfiguration database using the "File - Create backup" menu. A folder called "Backup" is created in the installation directory for the backups. The name of the backup also contains the date and time: 20210130_153047_OPCRouterConfiguration.bak

Update

For software updates and maintenance, please consult the relevant section in this manual.

Update/Maintenance

Use HTTPS (Windows)

HTTPS is a protocol that encrypts and authenticates communication between a web browser and a web server.

To use HTTPS with OPC Router a certificate is required.

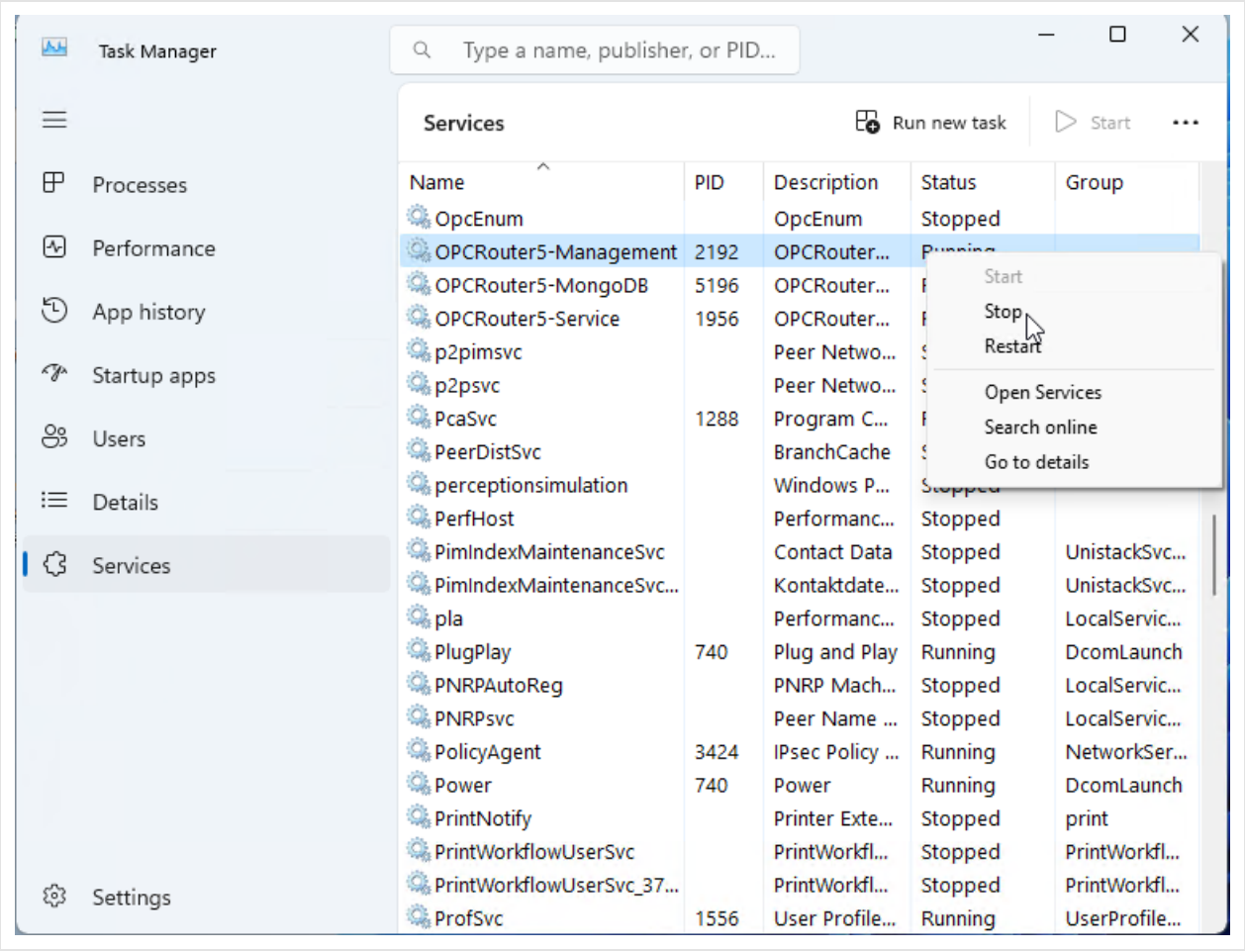
How to create SSL Certificates

Create SSL Certificates

Even with a self-signed certificate, most browsers will classify Web Management under HTTPS as "not secure." However, the connection is still encrypted, but the browser cannot verify the authenticity of the website.

Enabling HTTPS

First stop the windows services OPCRouter5-Service and OPCRouter5-Management in that order, either by the using the Windows Services Interface or the Task Manager.



Then open a new command line window as Administrator. Copy the following line into the new command line window and press `Enter` again.

If the service configuration was changed before, copy the "Path to executable" from the Property Window of the service and remove only the "--disable-https true" flag. Escape all quotation marks with the \". Paste the full line as a replacement into the value of the `binPath= ""` parameter.

```powershell title="With initial configuration" sc config OPCRouter5-Management binPath= ""C:\Program Files\inray\OPC-Router5\inray.OPCRouter.Web.Management.exe" --http-port 5000 --https-port 5001 --initial-username --initial-password"
```

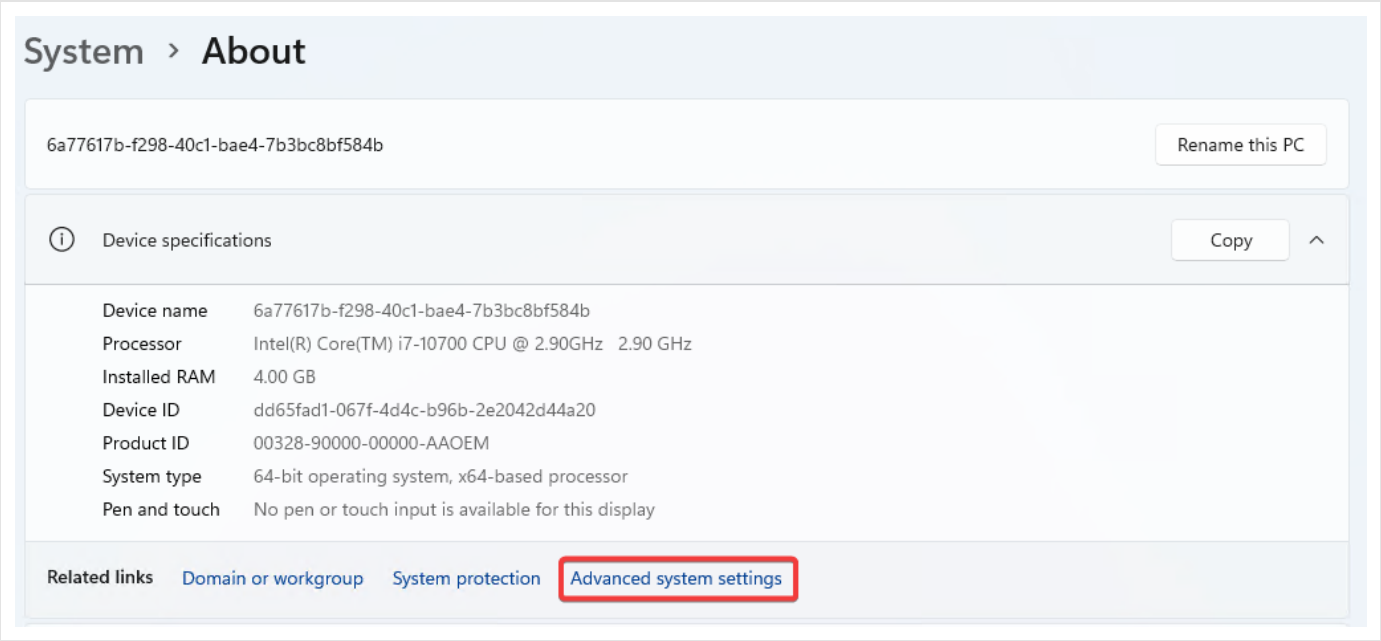
```powershell title="With changed configuration"
sc config OPCRouter5-Management binPath= "<Replace this>" --http-port 5000 --https-port 5001 --initial-username --initial-password"
```

Then add the following three system environment variables, like in the screenshot below:

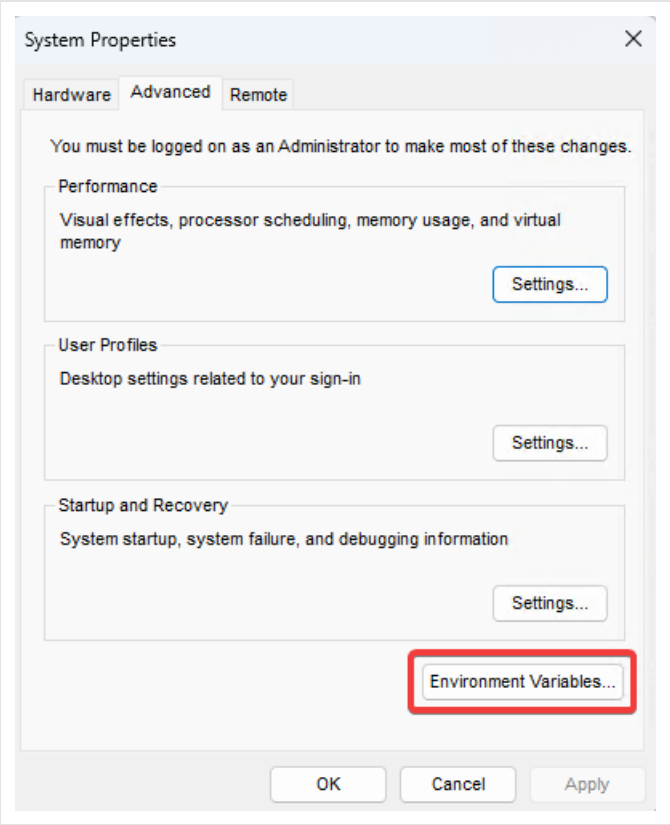
NAME	VALUE
OR_WEB_HTTPS_CERTIFICATE_PEM_FILE	<Path to the .pem certificate file>
OR_WEB_HTTPS_CERTIFICATE_KEY_PEM_FILE	<Path to the .key file>
OR_WEB_DISABLE_HTTPS	false

Add environment variables

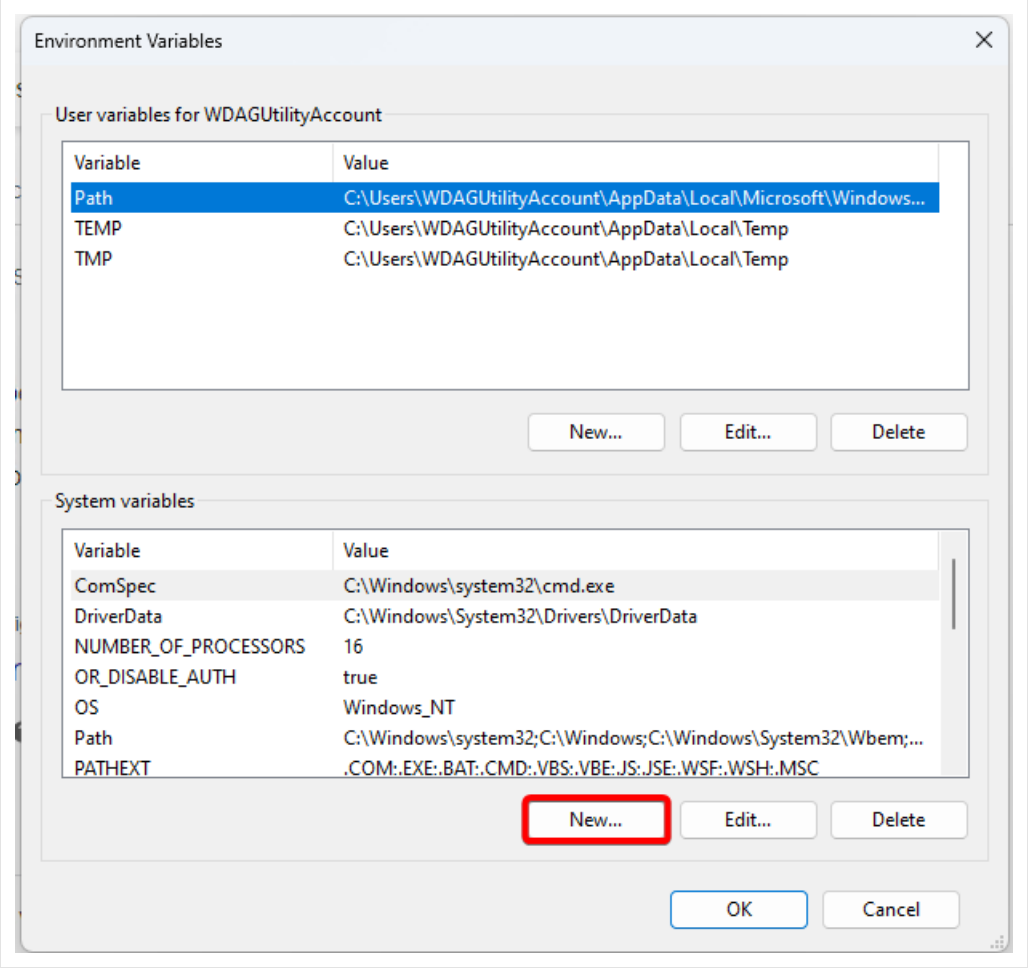
Open the about dialog for your system and click on "Advanced system settings".

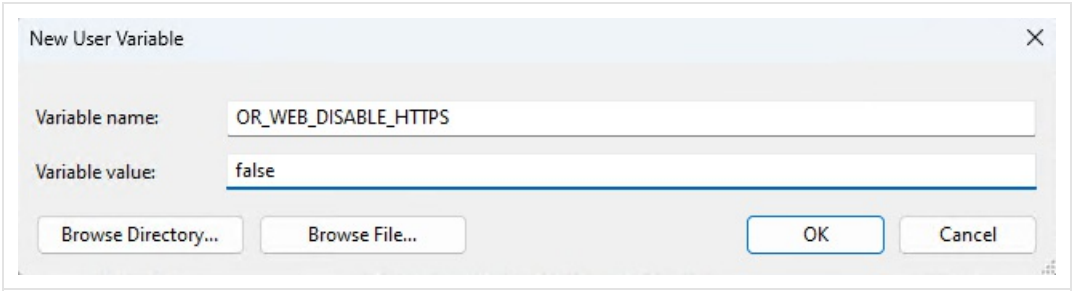


Then click on the "Environment Variables..." button.



In the environment variables dialog, click on "New..." to add another environment variable




New User Variable
Variable name:
OR_WEB_DISABLE_HTTPS
Variable value:
false
Browse Directory...
Browse File...
OK
Cancel

Deployment Docker

What is Docker?

Docker is a software platform that enables executing applications and their dependencies in isolated containers. Docker offer a certain number of advantages for the development, deployment and scaling of softwares :

- **Portability:** Docker containers can be executed on any machine on which Docker is installed, regardless of the operating system or hardware.
- **Consistency:** Docker containers ensure that application always run under the same conditions, regardless of whether they are exectued locally, in the cloud or on another server.
- **Efficiency:** Docker containers use the resources from the host machine in an optimal way, by supplying only what is necessary for the application. This reduces storage space and energy consumption.
- **Security:** Docker containers isolate the applications from one another and the host system, which reduces the risk of attacks and conflicts.

With help of Docker, the OPC Router can work with any environment, in which a Docker installation is possible.

Installing Docker

To install Docker, follow the instructions of the official Docker documentation for your platform.

To confirm that the Docker installation is a success, you can execute the following command, to your command prompt: `docker --version` If you receive a message like `Docker version 20.10.22 build 3a2c30b`, Docker is ready to be use.

Note: When run in a container (for example: Docker, Kubernetes), licensing with online synchronization is required if is operating without a TPM2 module.

Docker Container with TPM2

Fast test installation of the OPC Router with Docker

The Docker OPC Router 5 container can be installed fast for testing by executing only one command in seconds :

```
docker run -d -e OR_DISABLE_AUTH=true -e OR_I_ACCEPT_EULA=true --name opcrouter5 -p 8080:8080 opcrouter/runtime:latest
```

By executing this command, you agree to the end user licence conditions by setting the environment variable `OR_I_ACCEPT_EULA` to `true`.

This command doesn't create any persistent volumes. By deleting the container via Docker Desktop, all data of the container, for example the configured projects, will get irrevocably lost**.** The command is therefore not suitable for productive use.

By executing the command, a new container named `opcrouter5` is being created. Port 8080 is being published on the host. The Web Management should be accessible locally at http://127.0.0.1:8080.

The command `docker container rm -f opcrouter5` stops and deletes the container. The temporary volumes of the automatic created container, which contain the data and the configuration of the project, will not be automatically deleted by this command.

The Docker images of the OPC Router

The OPC Router have two different images available:

- The Runtime image (opcrouter/runtime) contains the OPC Router application, the web management, and an integrated MongoDB in which data of the OPC Router such as projects, are stored.
- Like the runtime image, the service image (opcrouter/service) contains the application and the web management, but no

integrated MongoDB. A MongoDB must be connected separately for the application to run.

Recommended runtime deployment

Instead of a test installation, a Docker Run command is described here that is suitable for a productive use. This command cannot just simply be copied and run. It must be adapted before running it.

```
docker run -d --pull always \
   --name opcrouter5 \
   -e OR_I_ACCEPT_EULA=false \
   -e INITIAL_USERNAME=***** \
   -e INITIAL_PASSWORD=***** \
   -e TZ=Europe/Berlin \
   -v opc-router-5-data:/data \
   -v opc-router5-logs:/var/log/opcrouter \
   -p 8080:8080 \
   -p 8001:8001 \
   -p 49420:49420 \
   opcrouter/runtime:latest
```

For this command to work, you must agree the end user license agreement by setting the environment variable `OR_I_ACCEPT_EULA` to `true`, and by specifying a valid username and password for the initial user in the environment variables `INITIAL_USERNAME` and `INITIAL_PASSWORD`.

Some used arguments are explained here:

- `-d`: The container is being started as a background process.
- `--pull always`: Ensures that the current version of the image is downloaded when the command is executed if it differs from an existing local version.
- `--name`: Sets a unique name for the container.
- `-e`: Sets an environment variable.
- `-v`: Sets a persistent volume.
- `-p`: Publishes an internal post on the host.

With the environment variables, it should be noted that we set an initial user with username (`INITIAL_USERNAME`) and password (`INITIAL_PASSWORD`) with which we can log into Web Management. The time zone is also being set. A more detailed explanation and list of all configurable environment variables can be found here:

Configurable environment variables

For the two available mount paths, persistent volumes are being created, so that the data on the container, like the project structure, backups of the project, log files, etc. are being kept when we delete the container and create a new one with the same volumes. A detailed explanation of the mount paths available can be found here:

Mount paths

Mount paths

From version 5.0.5011, the OPC Router Docker image has two mount paths that allow persistent volumes to be mounted.

MOUNT PATH	DESCRIPTION
/data	Data storage location for internal data such as configuration files, project file backups, licenses, and the integrated database (if available). Published and unpublished connections, plug-in configurations, and settings, as well as the transfer data are stored here.
/var/log/opcrouter	Log files for the service, web management and the integrated database (if available)

For versions older than 5.0.5011 the OPC Router docker image requires three mount paths.

MOUNT PATH	DESCRIPTION
/inray	Settings and project file backups
/var/log/opcrouter	Log files for the service, web management and the integrated database (if available)
/data/database	Data storage location for the integrated database (if available). Published and unpublished connections, plug-in configurations, and settings, as well as the transfer data are saved here.

Configurable environment variables

When creating the OPC Router containers, environment variables can be set, which are visible within the containers. These environment variables can be used to configure the OPC Router by reading them during startup.

These environment variables apply not only to the OPC Router Docker container but also to Windows installations. However, in Windows installations, the "OPCRouter5-Service" and "OPCRouter5-Management" services must be restarted for the changed environment variables to take effect.

General Environment Variables

ENVIRONMENT VARIABLE	DESCRIPTION	EXAMPLE VALUE
`OR_I_ACCEPT_EULA`	Setting this to `true` accepts the End User License Agreement. Required for application execution.	`true`
`INITIAL_USERNAME`	Sets the name of the initial user account for Web Management.	`or5`
`INITIAL_PASSWORD`	Sets the password for the initial user account for Web Management. Use either this or `INITIAL_PASSWORD_FILE`.	Don'tUseThis,ItWouldn'tBeSafe!
`INITIAL_PASSWORD_FILE`	Specifies the path to a file inside the container from which the password for the initial user account for Web Management is retrieved. Use either this or `INITIAL_PASSWORD`.	`/inray/defaultpw`

Environment Variables for Web Management

ENVIRONMENT VARIABLE	DESCRIPTION	EXAMPLE VALUE
`OR_WEB_HTTP_PORT`	Sets the port for HTTP connections. If HTTPS is available, the port refers to the HTTPS port. (Default: `8080`)	`80`
`OR_WEB_HTTPS_PORT`	Sets the port for HTTPS connections. HTTPS requires a valid SSL certificate. (Default: `8443`)	`443`
`OR_WEB_HTTPS_CERTIFICATE_PEM`	Sets PEM-encoded SSL certificate. Use either this or `OR_WEB_HTTPS_CERTIFICATE_PEM_FILE`.	[PEM Certificate]
`OR_WEB_HTTPS_CERTIFICATE_PEM_FILE`	Specifies the path to a PEM-encoded file inside the container from which the SSL certificate is retrieved. Use either this or `OR_WEB_HTTPS_CERTIFICATE_PEM`.	`/inray/https.pem`
`OR_WEB_HTTPS_CERTIFICATE_KEY_PEM`	Sets the private key of the PEM-encoded SSL certificate. Use either this or `OR_WEB_HTTPS_CERTIFICATE_KEY_PEM_FILE`.	[PEM Private Key]
`OR_WEB_HTTPS_CERTIFICATE_KEY_PEM_FILE`	Specifies the path to the private key of the PEM-encoded SSL certificate inside the container. Use either this or `OR_WEB_HTTPS_CERTIFICATE_KEY_PEM`.	`/inray/https.key`
`OR_ENABLE_HTTP_LOGGING`	Can be set to `true` to enable logging of web requests.	`true`
`OR_WEB_DISABLE_HTTPS`	Can be set to `false` to enable HTTPS. (Default: `true`)	`false`

Environment Variables for Web Management Authentication

ENVIRONMENT VARIABLE	DESCRIPTION	EXAMPLE VALUE
INITIAL_USERNAME	Sets the name of the initial user account for Web Management.	or5
INITIAL_PASSWORD	Sets the password for the initial user account for Web Management. Use either this or INITIAL_PASSWORD_FILE.	Don'tUseThis,ItWouldn'tBeSecure!
INITIAL_PASSWORD_FILE	Specifies the path to a file inside the container from which the password for the initial user account for Web Management is retrieved. Use either this or INITIAL_PASSWORD.	/inray/defaultpw
AUTH_DB_CONNECTION_STRING	Sets the connection string for the OPC Router authentication database. Use either this or AUTH_DB_CONNECTION_STRING_FILE.	Data Source=OPCRouterWebConfig.db;
AUTH_DB_CONNECTION_STRING_FILE	Specifies the path to a file inside the container from which the connection string to the OPC Router authentication database is retrieved. Use either this or AUTH_DB_CONNECTION_STRING.	/inray/authdbconn
AZURE_AD_INSTANCE	Sets the Azure Active Directory instance.	https://login.microsoftonline.com/
AZURE_AD_DOMAIN	Sets the domain of the Azure Active Directory tenant.	contoso.onmicrosoft.com
AZURE_AD_TENANT_ID	Sets the Azure Active Directory Tenant ID.	123456ab-1a2b-3c45-67de-1234ab-567cd8
AZURE_AD_CLIENT_ID	Sets the Azure Client ID.	11111111-1111-1111-1111-111111111111
AZURE_AD_CALLBACK_PATH	Sets the relative request path to which the User-Agent is redirected. (Default: /signin-oidc)	/signin-oidc
AZURE_AD_SIGNED_OUT_CALLBACK_PATH	Sets the relative request path to which the User-Agent is redirected after logging out from the Identity Provider. See http://openid.net/specs/openid-connect-session-1_0.html#RedirectionAfterLogout. (Default: /signout-callback-oidc)	/signout-callback-oidc
OR_DISABLE_AUTH	Disables user authentication in Web Management when set to true.	true
OR_DISABLE_WIN_AUTH	Disables Windows user authentication in Web Management when set to true.	true

Environment variables for licensing

ENVIRONMENT VARIABLE	DESCRIPTION
`OR_LICENSE_KEY`	The license key for the license
`OR_LICENSE_KEY_FILE` (alternative to `OR_LICENSE_KEY`)	Path to a file with the license key of the license.
`OR_LICENSE_OPCROUTERID`	The OPC Router ID of the license
`OR_INSTANCE_NAME`	Name for the licensed system
`OR_LICENSE_IGNORE_ACTIVATION_ERROR` (optional)	If this parameter is set to true, activation errors are ignored.
`OR_LICENSE_HOLDER` (optional)	License holder name
`OR_LICENSE_LOCATION` (optional)	System location
`OR_LICENSE_DESCRIPTION` (optional)	Additional description

Environment variables for redundancy

The settings are not case sensitive.

ENVIRONMENT VARIABLE	DEFAULT	DESCRIPTION
`OR_REDUNDANCY_MODE`		`0` or `disabled:` Redundancy is disabled `1` or `primary:` Primary service `2` or `secondary:` Secondary service
`OR_REDUNDANCY_SHARED_KEY`		Character sequence used for authentication of the OPC Router. This key must be the same for the primary and secondary services. An empty key is invalid and prevents the connection from being established.
`OR_REDUNDANCY_SHARED_KEY_FILE`		Alternatively, the key can also be transferred as a text file.
`OR_REDUNDANCY_ADDRESS`		Address of the primary service. This is specified in URL format (https: host name/IP address: port). The port under which the web management (the web interface of the OPC Router) can be reached must be specified as the port. **Example**: https://example.local:5000
`OR_REDUNDANCY_PRIMARY_TIMEOUT`	10	Time in seconds after which the secondary service is considered disconnected if no heartbeat has been sent.
`OR_REDUNDANCY_SECONDARY_HEARTBEAT_INTERVAL`	5	The interval in seconds at which the secondary service attempts to reach the primary service. **Note:** This value must be less than the timeout of the primary service so that the connection is not constantly considered disconnected in the primary service.
`OR_REDUNDANCY_ENABLE_PROJECT_SYNC`	true	Activates project synchronisation.

ENVIRONMENT VARIABLE	DEFAULT	DESCRIPTION
OR_REDUNDANCY_PROJECT_SYNC_ALLOW_EXPERT	true	Transferring expert settings during project transfer
OR_REDUNDANCY_PROJECT_SYNC_PING_INTERVAL	5	Ping interval (in seconds) for project synchronisation.
OR_REDUNDANCY_LOCAL_MANAGEMENT_ADDRESS		Address at which the local management can be reached by the other OPC router.

Database Environment Variables

ENVIRONMENT VARIABLE	DESCRIPTION	EXAMPLE VALUE
OR_DATABASE_CONNECTION_STRING	Sets the connection string to establish a connection to an external MongoDB. Overrides the connection to the internal database, if present. Use either this or OR_DATABASE_CONNECTION_STRING_FILE.	mongodb://127.0.0.1:27017
OR_DATABASE_CONNECTION_STRING_FILE	Specifies the path to a file inside the container from which the connection string to establish a connection to an external MongoDB is retrieved. Overrides the connection to the internal database, if present. Use either this or OR_DATABASE_CONNECTION_STRING.	/inray/dbconnection
OR_DATABASE_USERNAME	Sets the MongoDB user to be used for authentication.	root
OR_DATABASE_PASSWORD	Sets the password of the MongoDB user to be used for authentication. Use either this or OR_DATABASE_PASSWORD_FILE.	AlsoDon'tUseThis!
OR_DATABASE_PASSWORD_FILE	Specifies the path to a file inside the container from which the password for the MongoDB user used for authentication is retrieved. Use either this or OR_DATABASE_PASSWORD.	/inray/dbpw
OR_DATABASE_NAME_PREFIX	Sets a name prefix for the Config, Runtime, and Status databases.	OR5
OR_DATABASE_CERTIFICATE_FILE	Specifies the path to a file inside the container from which the x.509 certificate for authentication with MongoDB is retrieved.	/inray/db.pem

Environment Variables for Importing Project Files

ENVIRONMENT VARIABLE	DESCRIPTION	EXAMPLE VALUE
OR_IMPORT_SOURCE	Specifies the path to an OPC Router project file inside the container that should be loaded directly into the runtime during startup.	/inray/project.rpe

ENVIRONMENT VARIABLE	DESCRIPTION	EXAMPLE VALUE
`OR_IMPORT_CLEAR_CONFIG`	If set to `true`, the data in the runtime and config databases will be cleared before importing projects with `OR_IMPORT_SOURCE`. This will result in the loss of existing connections and plug-in configurations.	`true`
`OR_IMPORT_DONT_PUBLISH_CONFIG`	If set to `true`, doesn't publish the entire config database after importing with `OR_IMPORT_SOURCE`.	`true`
`OR_IMPORT_OVERWRITE_EXISTING`	If set to `true`, allows the import with `OR_IMPORT_SOURCE` to overwrite existing connections, templates and plugins, otherwise existing values are kept on conflict.	`true`
`OR_IMPORT_RUNTIME_CONFIGURATION_FILE`	Specifies a path to a yaml configuration file describing overrides of values for plugins and settings in the runtime database to be applied after service start.	`/inray/runtime_config.yaml`

Relevant ASP.NET Environment Variables

ENVIRONMENT VARIABLE	DESCRIPTION	EXAMPLE VALUE
`ASPNETCORE_FORWARDEDHEADERS_ENABLED`	If set to `true`, application header information is forwarded. This is necessary for connecting with reverse proxies. (Default: `false`)	`true`

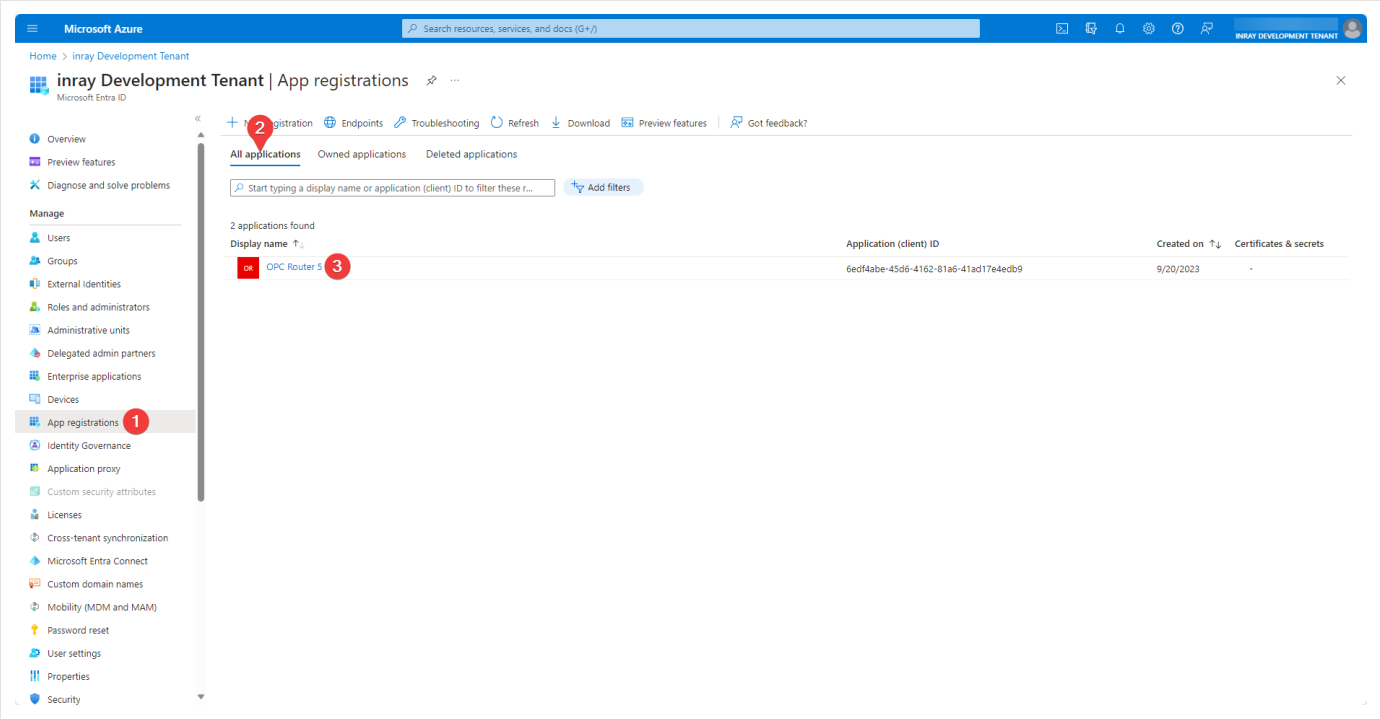
Connect Microsoft Entra ID/Azure AD

Microsoft Entra ID, formerly Azure Active Directory, is a cloud-based service for identity and access management of applications hosted in Microsoft Azure as well as applications running in other cloud or on-premises environments. Entra ID offers features such as Single Sign-On, Multi-Factor Authentication, Self-Service Password Reset, Conditional Access Policies, and Identity Protection. With Entra ID, organizations can centrally manage their users, groups, and devices and control access to their resources.

The OPC Router 5 Web Management allows user management and authentication through Entra ID by integrating Entra ID.

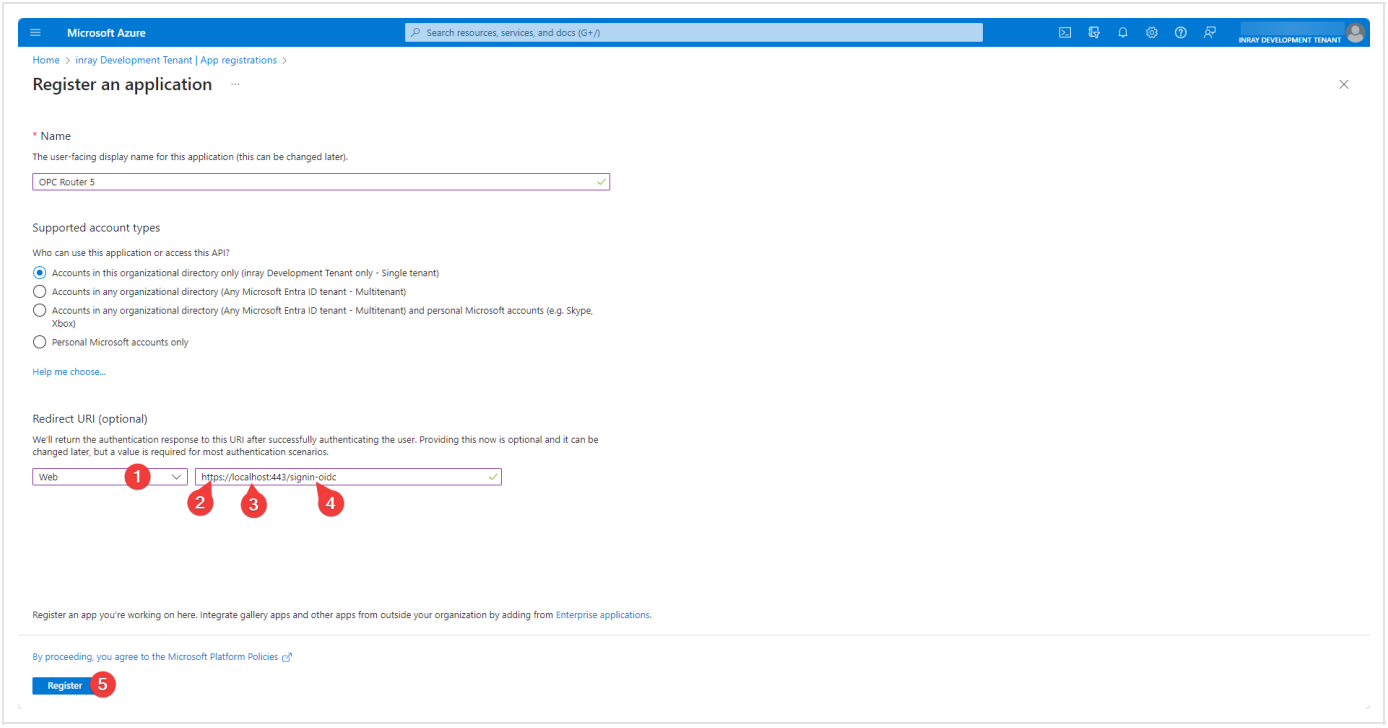
This guide assumes the presence of an existing Microsoft Entra ID tenant.

For OPC Router to use Entra ID, a new registration must be added to the application registrations of your Microsoft Entra ID tenant.



The appearance and navigation of the Microsoft Entra ID interface may vary.

On the **App registrations (1)** tab of your Microsoft Entra ID tenant, you can create a new application by clicking on **New registration (2)**.



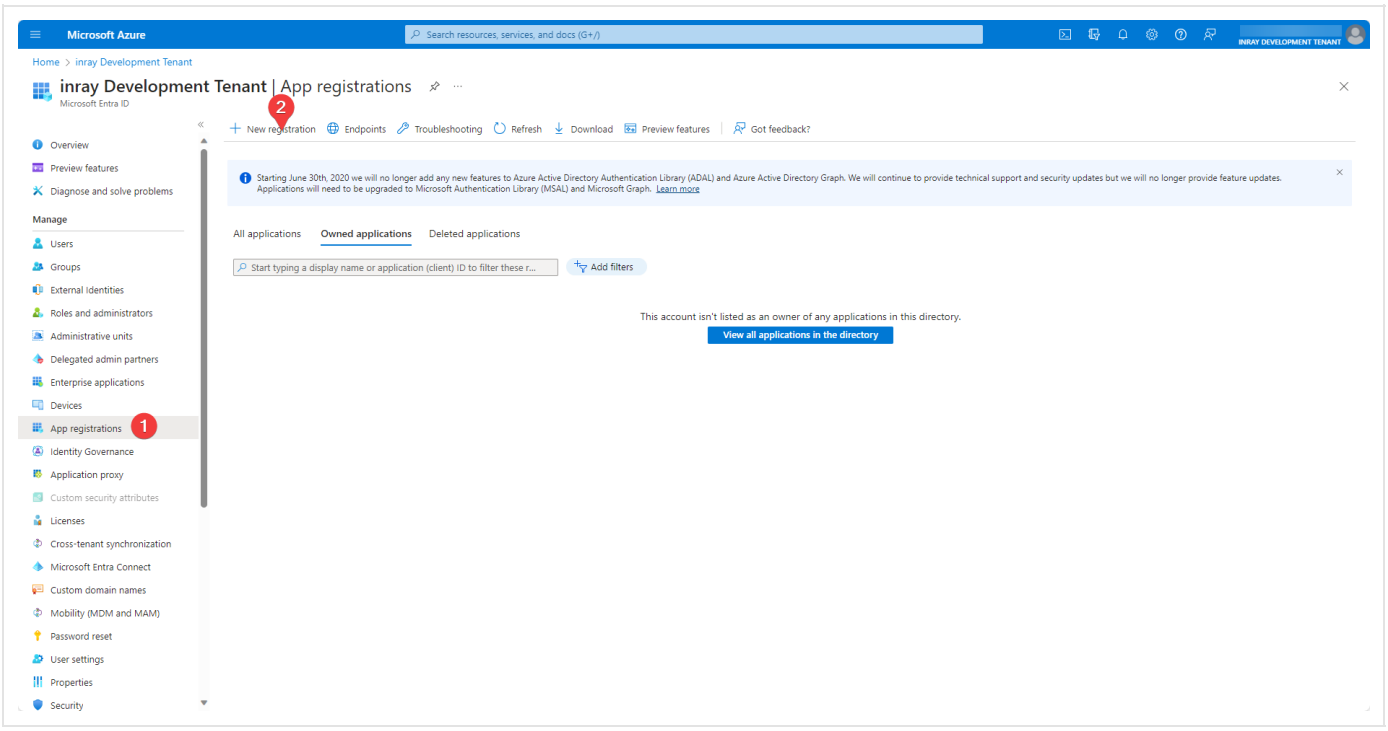
The appearance and navigation of the Microsoft Entra ID interface may vary.

It is important to customize the application's Redirect URI. In the **dropdown list (1)**, select `Web`. In the **Address (3)** field, configure the address where the Web Management is accessible. This must be a valid `https` **(2)** address of the Web Management; otherwise, it will not be allowed. The SSL certificate does not need to be issued by a certification authority; self-signed certificates are also permissible. Learn how to set up HTTPS for your OPC Router Web Management here:

use-https.md

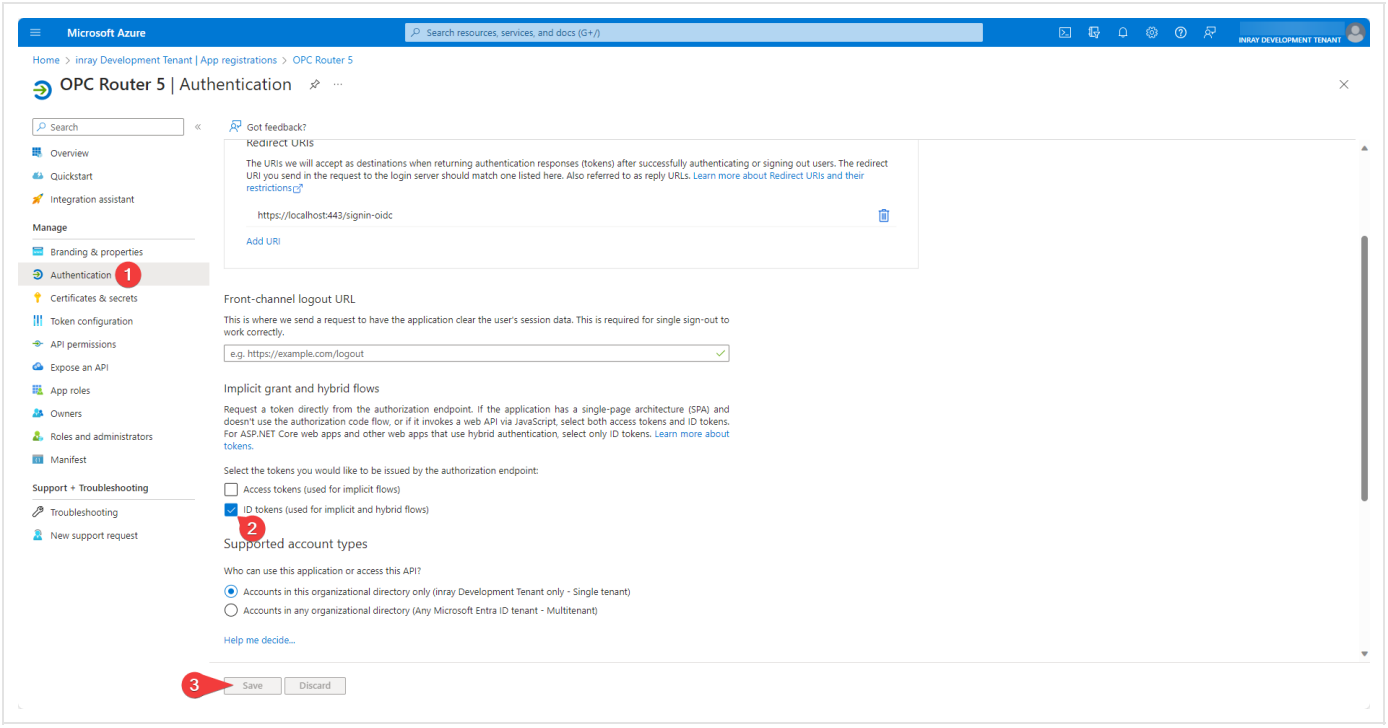
Additionally, the **Endpoint (4)** must pass the configured login request path. This is `/signin-oidc` by default but can be overridden by the `AZURE_AD_CALLBACK_PATH` environment variable if needed.

The newly created application needs further configuration to enable successful authentication via Microsoft Entra ID.



The appearance and navigation of the Microsoft Entra ID interface may vary.

To do this, on the **App registrations (1)** tab, under the **All applications (2)** group, select your newly created **application (3)**.



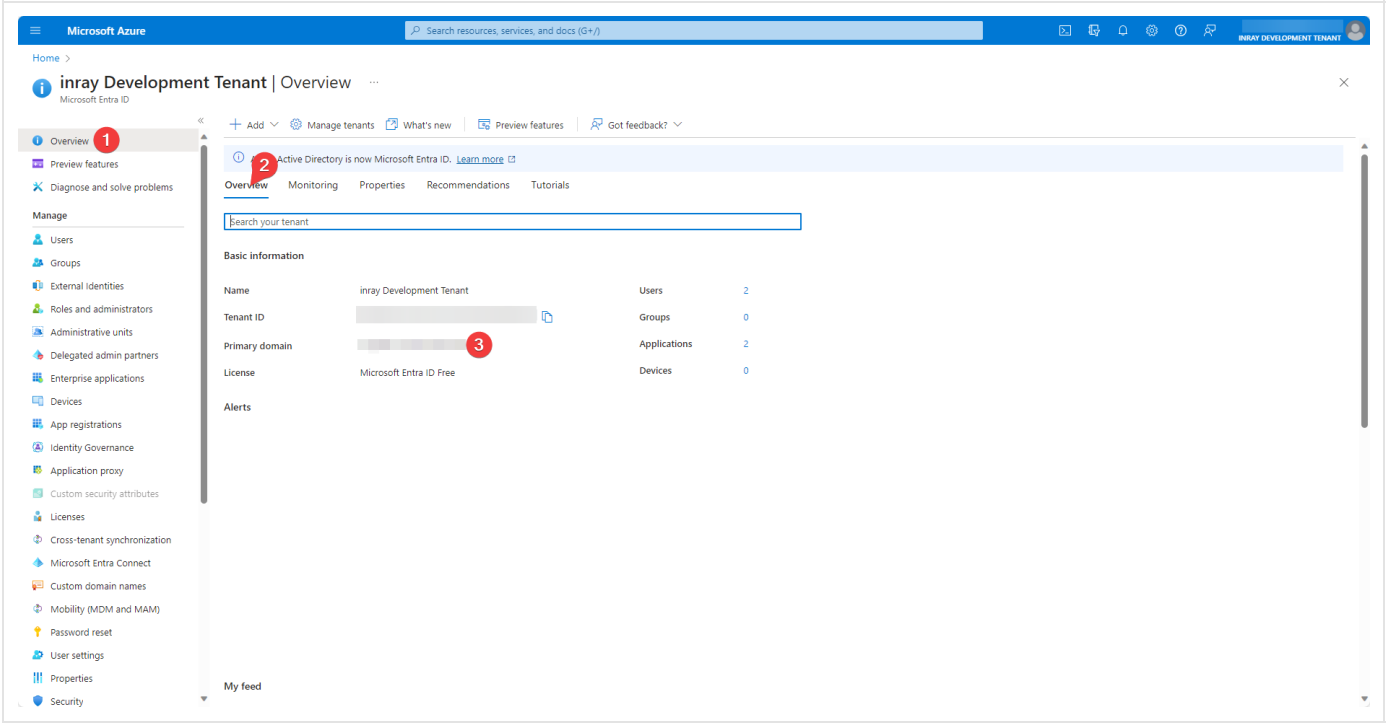
The appearance and navigation of the Microsoft Entra ID interface may vary.

On the **Authentication (1)** tab, there is a checkbox for **ID token (2)**, which must be enabled for the Web Management user management to work with Microsoft Entra ID. **Save (3)** this configuration change.

In your Entra ID tenant, you can now navigate to your configured application under Enterprise applications and use the Users and groups tab for user management.

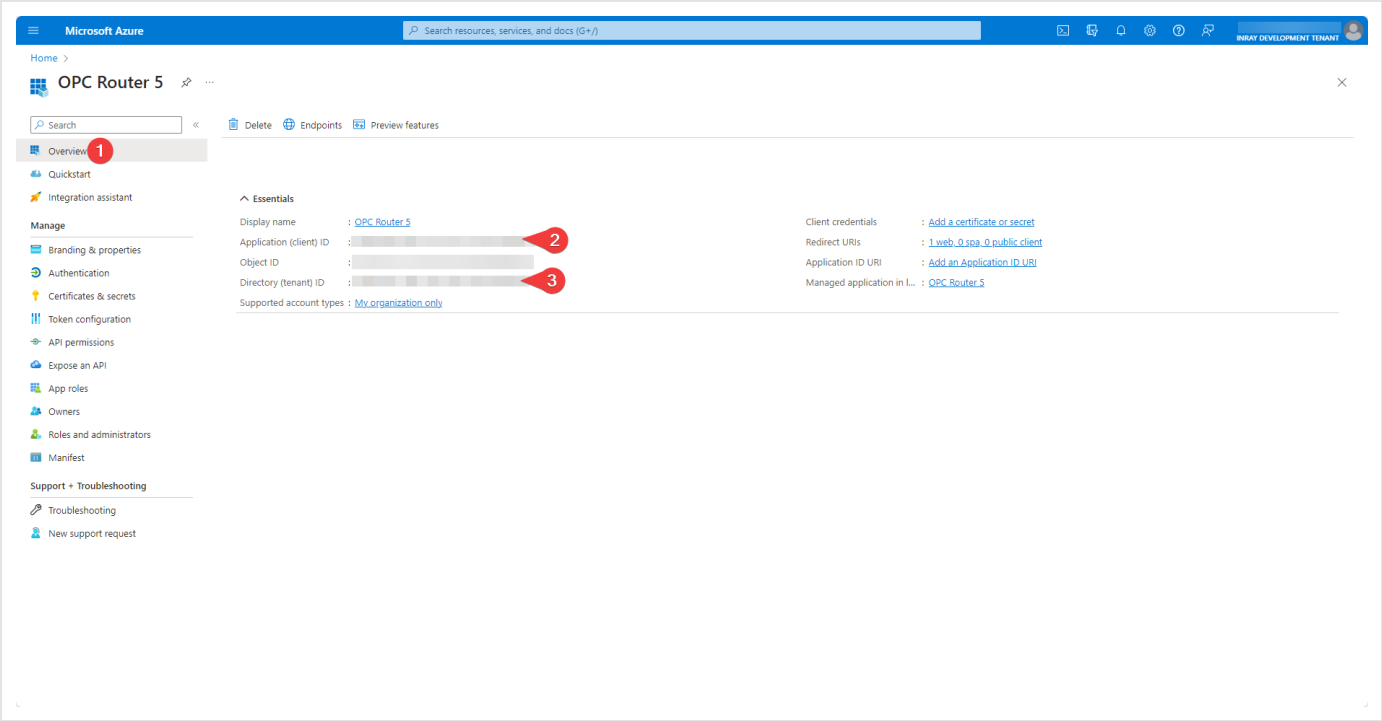
Creating OPC Router 5 Containers with Entra ID Integration

To connect OPC Router with Entra ID, three pieces of information from the tenant must be available: the Application ID, the Directory ID, and a domain under which the Entra ID configuration is accessible.



The appearance and navigation of the Microsoft Entra ID interface may vary.

The **Primary domain (3)** can be found in the **Overview (2)** tab of your Entra ID tenant.



The appearance and navigation of the Microsoft Entra ID interface may vary.

The **Application ID (2)** and the **Directory ID (3)** are displayed on the **Overview (1)** tab of the created application registration.

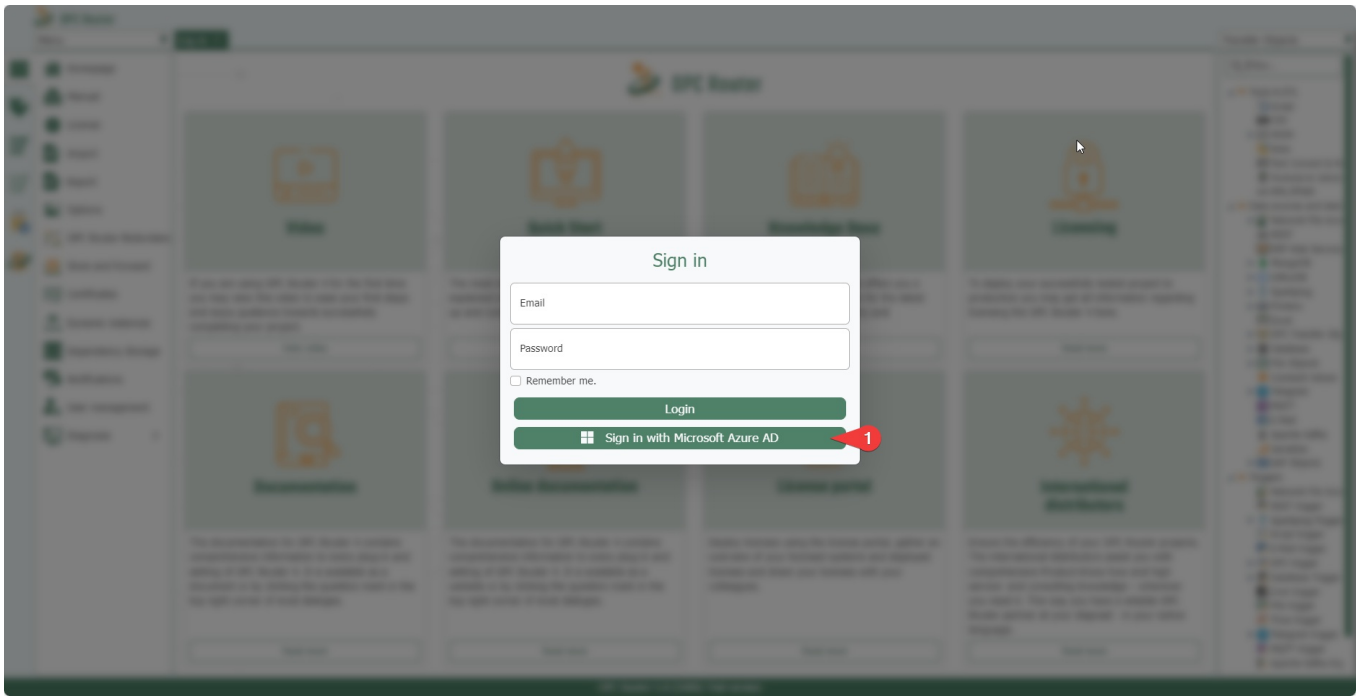
With this information, you can now execute a Docker run command that creates an OPC Router container with a connection to Entra ID:

```
docker run -d \
   -e OR_I_ACCEPT_EULA=true \
   -e AZURE_AD_DOMAIN=domain.onmicrosoft.com \
   -e AZURE_AD_TENNANT_ID=1111111-1111-1111-1111-1111111 \
   -e AZURE_AD_CLIENT_ID=1111111-1111-1111-1111-1111111 \
   --name opcrouterentra \
   opcrouter/runtime
```

By running the command and setting the environment variable `OR_I_ACCEPT_EULA` to `true`, you agree to the End User License Terms.

Enter your (primary) domain into the `AZURE_AD_DOMAIN` environment variable, your directory ID into the `AZURE_AD_TENNANT` environment variable, and your application ID into the `AZURE_AD_CLIENT_ID` environment variable.

For Entra ID authentication to work, HTTPS must be set up on the container. The necessary settings have been omitted here for demonstration purposes. The unaltered command is therefore not executable.



After successful setup, a button for logging in via **Azure AD/Entra ID (1)** should appear below the login button. Users added through the Entra ID tenant can use this button to log in.

External database

Instead of the internal database of the opcrouter/runtime image, an external database (MongoDB) can also be connected to store project data and transmitted values. For this purpose, it is recommended to use the opcrouter/service image, as it does not have an internal database and is therefore compatible with 32-bit systems.

Connecting an external database to the opcrouter/runtime image is possible, but it does not deactivate or automatically delete the internal database. The internal database continues to run within the container.

When is Using an External Database Recommended?

Using an external database is particularly recommended when deploying the OPC Router through orchestration software like Kubernetes or Docker Compose. This enables easy distribution of applications across multiple systems. For example, an opcrouter/service container can run on a 32-bit ARM system, while an opcrouter/runtime container can only run on 64-bit systems due to the integrated database.

MongoDB is not compatible with 32-bit systems. Therefore, an external database must also be operated on a 64-bit system.

Setting Up an External Database

For this example, we will create a MongoDB Docker container as an external database. We will use the official `mongo` image with the tag `6-jammy`, which provides the latest MongoDB 6 version. Other MongoDB instances can also be integrated.

> MongoDB is not compatible with 32-bit systems. Therefore, an external database must also be operated on a 64-bit system. Setting Up an External Database For this example, we will create a MongoDB Docker container as an external database. We will use the official mongo image with the tag 6-jammy, which provides the latest MongoDB 6 version. Other MongoDB instances can also be integrated.

This MongoDB deployment does not use authentication and does not persist data on named volumes. Therefore, it is suitable for demonstration purposes only and not for production use.

Please note that MongoDB must be configured as a Replica Set for use with OPC Router 5. This is ensured here by the `--replSet rs0` argument. An existing MongoDB database can also be configured as a Replica Set using the `rs.initiate()` command.

To make MongoDB accessible from our OPC Router container, we ensure that the `externaldb` container is in the same bridge network. Although this should not be necessary in most Docker environments, it is done here for demonstration purposes. Additionally, no external port needs to be opened since we only need to access the database within the network.

We determine the IP address of the created `externaldb` container within the `bridge` network so that we can use it in our connection string:

```
docker container inspect
-f "{{range .NetworkSettings.Networks}}{{println .IPAddress}}{{end}}"
externaldb
```

Here, a formatting option is used in the `-f` argument to display only the necessary IP address. The command should return a single IP address; for example, we will use the IP address `172.17.0.2` for further steps. This results in the connection string `mongodb://172.17.0.2:27017`.

Creating OPC Router Containers with External Database Connection

Now, you can easily create a container from the `opcrouter/service:latest` image. Here, we also explicitly specify the network as `bridge`, and we expose port 8080 externally so that we can access the OPC Router at localhost:8080 upon successful deployment.

The connection string passed to the `OR_DATABASE_CONNECTION_STRING` environment variable needs to be adjusted to your connection string.

```
docker run -d
    --name service
    --network bridge
    -e OR_I_ACCEPT_EULA=true
    -e OR_DISABLE_AUTH=true
    -e OR_DATABASE_CONNECTION_STRING="mongodb://172.17.0.2:27017"
    -p 8080:8080
    opcrouter/service:latest
```

By running the command and setting the `OR_I_ACCEPT_EULA` environment variable to `true`, you agree to the End User License Agreement.

For demonstration purposes, the authentication of the web management is disabled using the `OR_DISABLE_AUTH` environment variable, although this is not recommended.

The crucial aspect here is setting the connection string to the `OR_DATABASE_CONNECTION_STRING` environment variable. The connection string can also include the username and password if needed to access authenticated databases. Further information about connection strings can be found in the official MongoDB documentation.

For the MongoDB connection to work, MongoDB must be configured as a Replica Set. Otherwise, the OPC Router's connection attempt will fail and log a `MongoDB.Driver.MongoCommandException`. You can learn more about Replica Sets in the official MongoDB documentation.

Use HTTPS

HTTPS is a protocol that encrypts and authenticates communication between a web browser and a web server.

To use HTTPS with OPC Router a certificate is required.

How to create SSL Certificates

Create SSL Certificates

Even with a self-signed certificate, most browsers will classify Web Management under HTTPS as "not secure." However, the connection is still encrypted, but the browser cannot verify the authenticity of the website.

Creating OPC Router Containers with HTTPS

```
docker run -d \
   -e OR_I_ACCEPT_EULA=true \
   -v $pwd/https.pem:/inray/https.pem \
   -v $pwd/https.key:/inray/https.key \
   -e OR_WEB_HTTPS_CERTIFICATE_PEM_FILE=/inray/https.pem \
   -e OR_WEB_HTTPS_CERTIFICATE_KEY_PEM_FILE=/inray/https.key \
   -e OR_WEB_DISABLE_HTTPS=false \
   -p 443:8443 \
   --name opcrouterhttps \
   opcrouter/runtime
```

By running the command and setting the `OR_I_ACCEPT_EULA` environment variable to `true`, you agree to the End User License Agreement.

The added volumes mount the local certificate and private key on the host, allowing the container to access them. Since these are mounts of local files and not copies, any changes or deletions of the files may affect the container. Alternatively, you can transfer the certificate and private key to the container using the `docker cp` command.

The environment variables `OR_WEB_HTTPS_CERTIFICATE_PEM_FILE` and `OR_WEB_HTTPS_CERTIFICATE_KEY_PEM_FILE` specify the paths to the certificate and private key. The `OR_WEB_DISABLE_HTTPS` environment variable is set to `false` to enable HTTPS. The OPC Router's default HTTPS port (`8443`) is exposed as `443` externally. The Web Management should now be accessible at https://localhost:443.

Reverse-Proxy (Traefik, nginx, etc.)

See also:

http-s-failover-concepts-for-rest-opc-ua-server-and-webmanagement-in-the-opc-router

To operate the OPC Router in Docker behind a reverse proxy like Traefik or nginx, certain settings need to be configured. It's important to note that the OPC Router uses both HTTP and WebSocket connections, which must be correctly forwarded by the reverse proxy.

This section describes only the necessary steps for web management. Additional adjustments may be required for the REST server or other plugins.

To optimize network performance, we recommend enabling SSL offloading in the reverse proxy rather than in the OPC Router, especially when network traffic between the reverse proxy and the OPC Router can be considered secure.

Path Prefix

To access the OPC Router through a reverse proxy with a path prefix, two environment variables must be set. The `WEB_BASE_PATH` environment variable specifies the path prefix used by the reverse proxy. The `ASPNETCORE_FORWARDEDHEADERS_ENABLED` environment variable must be set to `true` to allow the OPC Router to process header information forwarded by the reverse proxy.

Example Traefik configuration:

```
docker run ... \
 -e "WEB_BASE_PATH=/opc-router"
 -e ASPNETCORE_FORWARDEDHEADERS_ENABLED=true
 --label "traefik.http.routers.reverse-proxy.rule=PathPrefix(/opc-router)"
```

Hostname

To make the OPC Router accessible under a specific hostname, you can set up a reverse proxy. Ensure that the reverse proxy host itself is reachable at the specified DNS address. Additionally, the `ASPNETCORE_FORWARDEDHEADERS_ENABLED` environment variable must be set to `true` to enable the OPC Router to correctly process header information from the reverse proxy.

Docker Container with TPM2

Advantages:

- Allows multiple online or offline licensed Docker containers on the same Linux host.
- Docker containers **without TPM 2 require regular internet access**, or they will stop functioning. (Windows is not affected by this.)
- TPM is only used for **storing a license** and is not related to online or offline licensing.
- Supported from **OPC Router 4.27.5018.76** Hotfix 6.

Prerequisites:

1. **A Linux host with TPM**
    1. A **physical host** with a TPM 2 chip.
    2. For **Hyper-V VMs**, it can be enabled through VM settings:

    3. For **Wago Edge devices**, it must be enabled in the BIOS:

        ```
        BIOS => Advanced => Trusted Computing => Security Device Support => Enable
        ```

    4. Verify with:

        ```
        ls /dev | grep tpm
        ```

        It's successful if `tpm` is listed:

Starting Docker with TPM

Example to start an OPC Router Docker container with TPM:

```
docker run -d \
   -p 27021:27017 \
   --name opc-router1 \
   --hostname opc-router1 \
   -v /data/db1:/data/db \
   -v /data-keys:/data/key-storage/ \
   --device /dev/tpmrm0:/dev/tpmrm0 \
   -e "INRAY_RSA_STORAGE=TPM" \
   -e "OR_I_ACCEPT_EULA=true" \
   -e "DOTNET_RUNNING_IN_CONTAINER=true" \
   opcrouter/runtime
```

Relevant Arguments: --hostname opc-router1

Used by the container as its identity on the host. The hostname must remain the same for a container; otherwise, it loses its identity. Each container should have a unique hostname.

```
-v /data/db1:/data/db -v /data-keys:/data/key-storage/
```

Specifies where on the host the TPM key files are stored. A separate folder is created for each --hostname. If a folder is deleted or cannot be reached by the container, the license associated with it is also lost.

Technical Explanation: TPM storage space is extremely limited. Therefore, as much as possible must be offloaded to the hard drive. The files can only be read by the physical TPM that created them, so no additional file protection measures are required.

```
--device /dev/tpmrm0:/dev/tpmrm0
```

This allows the container to use the host's TPM resource manager.

-v /dev/tpmrm0:/dev/tpmrm0 may appear to work, but it doesn't. The TPMRM must always be passed with --device.

```
-e "INRAY_RSA_STORAGE=TPM"
```

This forces the container to use TPM for licensing.

Verification:

When the container is running, it should be licensable both online and offline. The container should no longer require an internet connection. A container or host restart should still preserve the license. In the database, a TrustLevel of 2 should be recorded

Edge Device Management

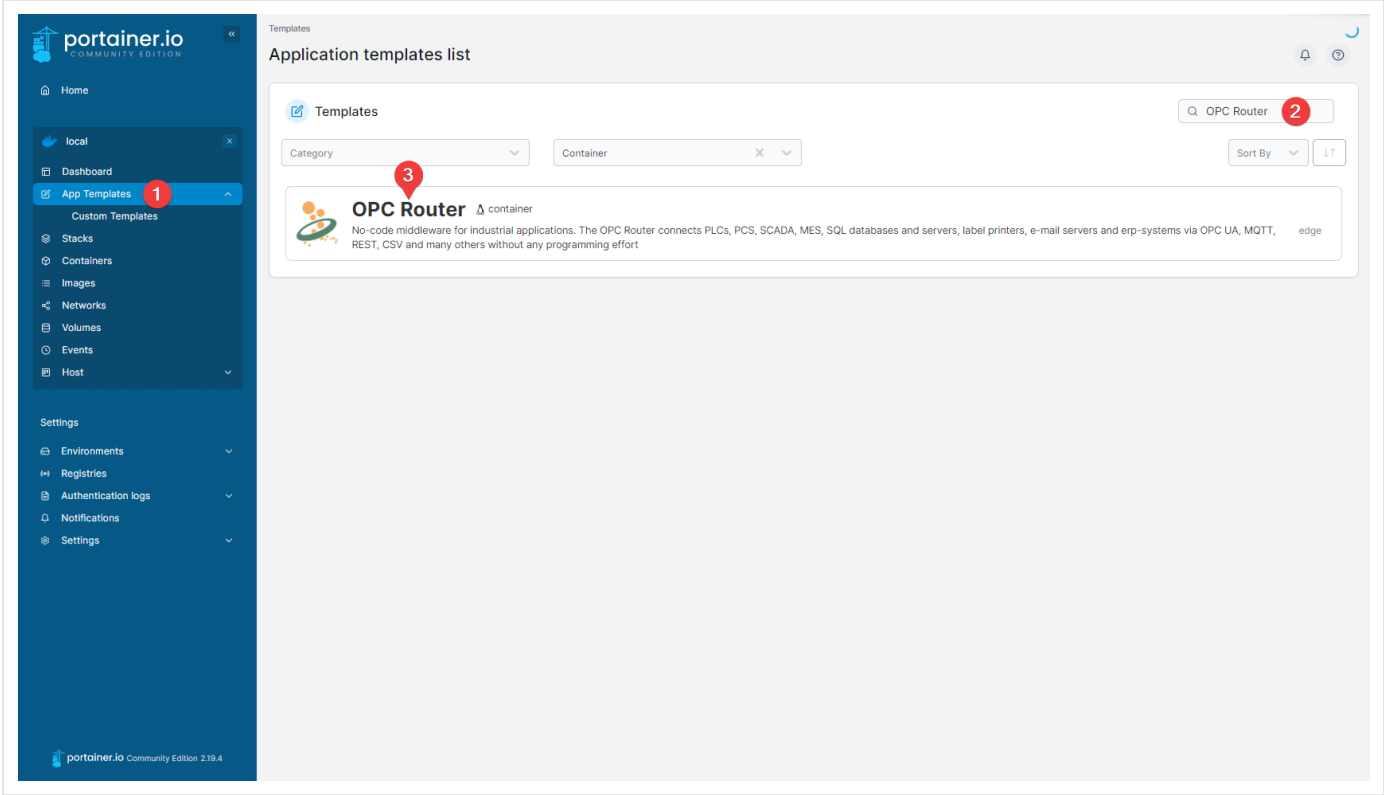
Installing the OPC Router via edge device management software such as Portainer, Siemens Industrial Edge or Azure IoT enables simplified and centralised deployment on edge devices. These tools provide an intuitive user interface for managing containers and applications, making it easier to deploy and update the OPC Router on a wide range of devices. By using these edge management platforms, companies can efficiently scale and adapt their automation infrastructure to ensure optimal performance and availability of their systems.

Portainer

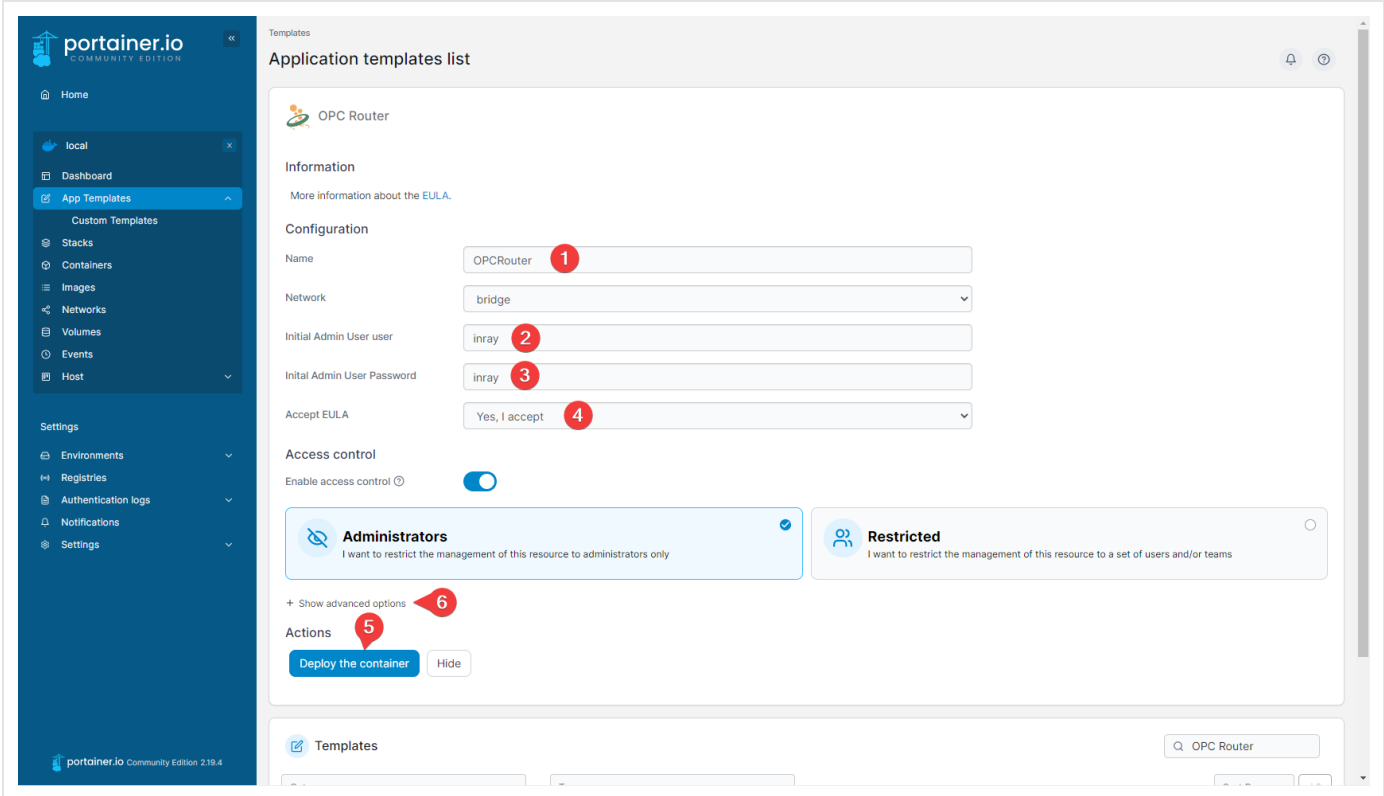
Portainer is a software of container management and aim to deploy, manage and survey the software containers. Portainer is offering a great variety of software that can be deployed in only one click via the easy-to-use user interface on an edge device.\ \ The OPC Router is provided by Portainer in the form of an App Template and an Edge Template. The App Template allows to easily install the OPC Router in the local Docker environment, while the Edge Template allows to deploy and configure multiple OPC Routers in external clusters.

OPC Router Portainer App Template deployment

In order to Install the OPC Router Portainer App template, a Portainer environment is required. The installation of Portainer is described in the Portainer documentation.



Navigate to the **App Templates (1)** and search for "OPC Router" **(2)**. The result of your research should be that OPC Router Template **(3)**. Open the configuration view by clicking on the template.

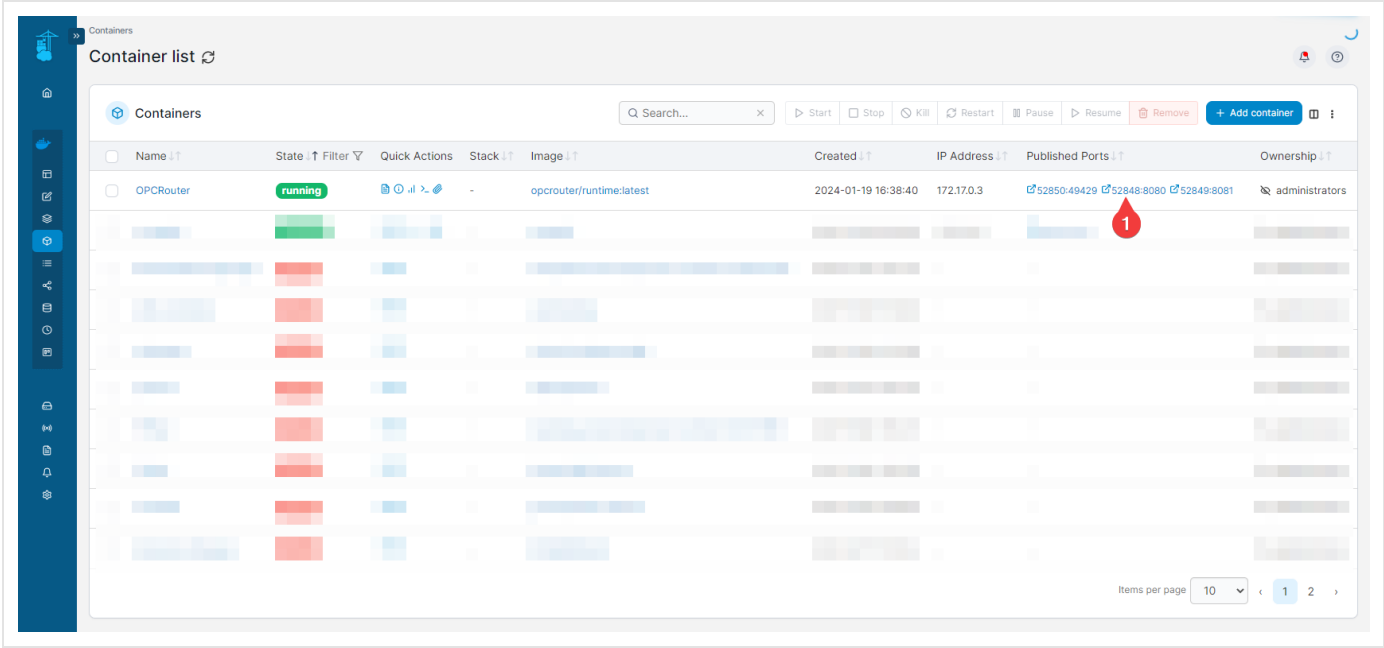


Enter now a name for your OPC Router instance **(1)** and define the username **(2)** and the password **(3)** for the initial user.

By clicking on **Deploy the container (5)**, you agree to the end user licence conditions by setting Accept EULA to **"Yes, I accept"**

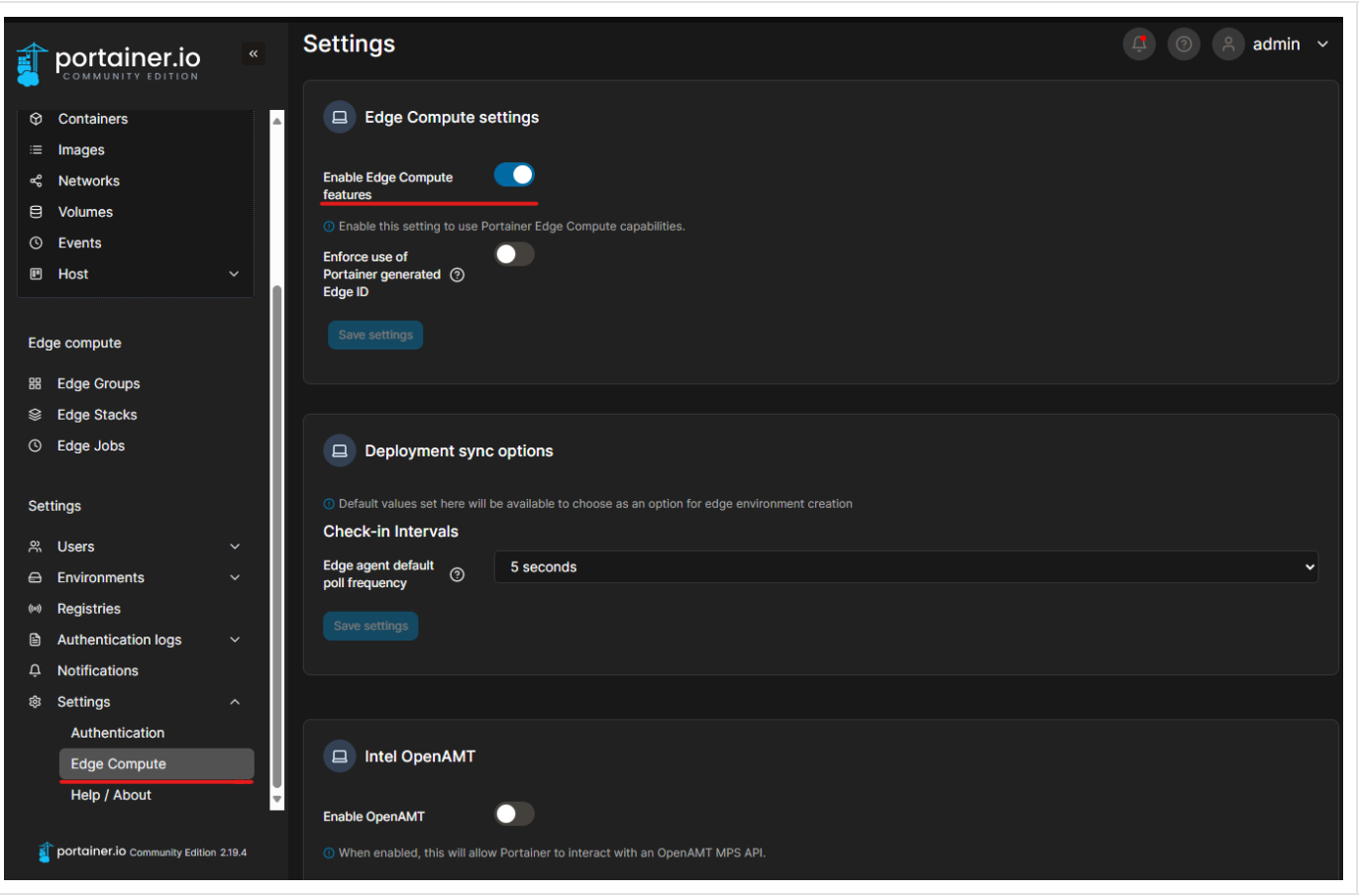
(4).

An OPC Router Runtime will be now created and started by clicking on **Deploy the container (5)**. The ports via which it can be reached from outside are randomly assigned.\ \ It is alternatively possible to choose freely the ports by opening the **Advanced Options (6).**

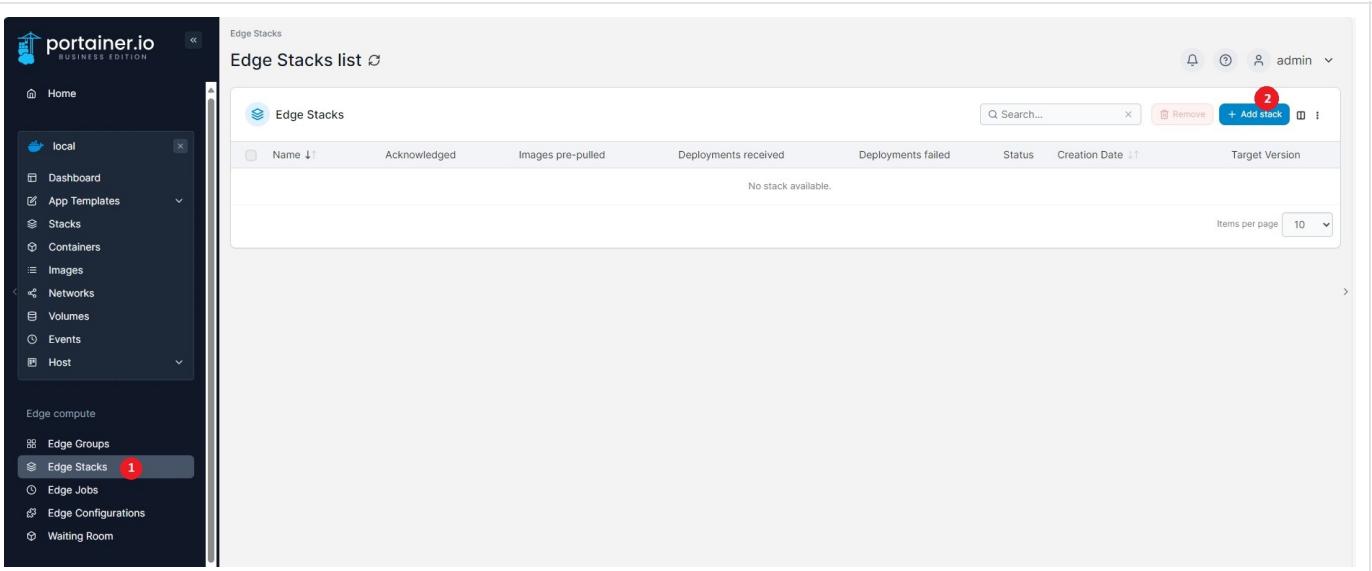


The OPC Router should now be shown on the Container list. Its web management can be accessed via the port to which the internal port 8080 is mapped **(1)**.

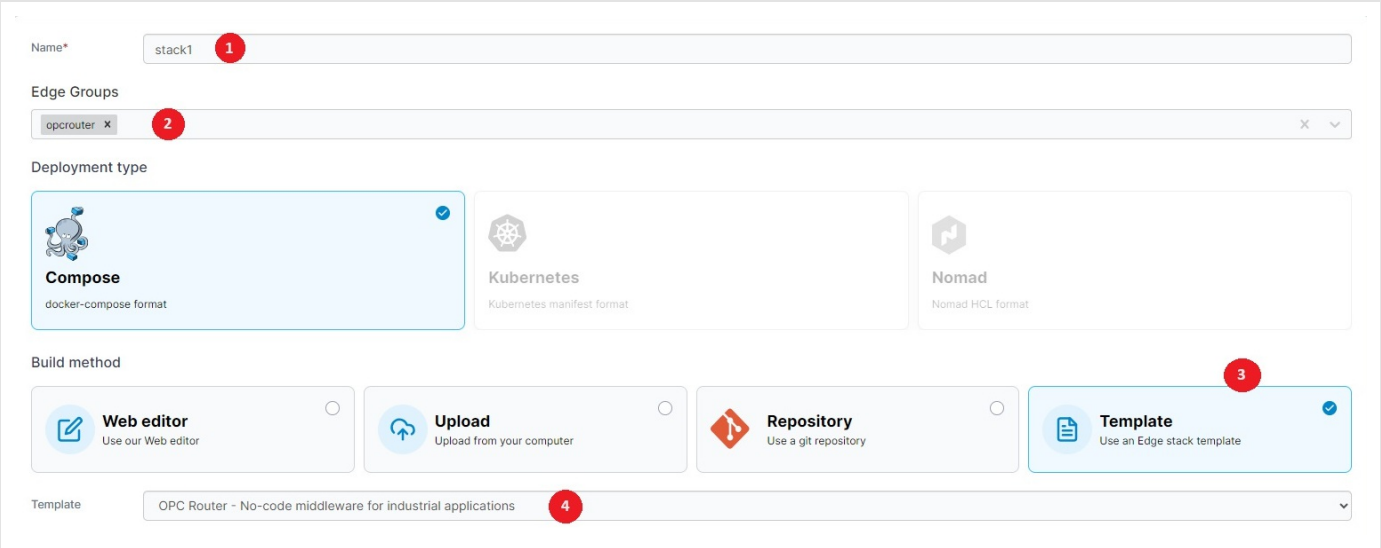
OPC Router Portainer Edge Template deployment



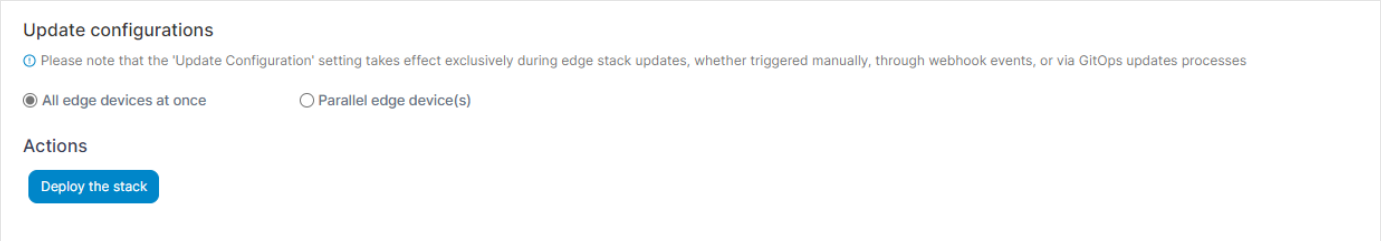
In order that the following steps works, the "Edge Compute" function must be enabled, as shown in the image above.\ \ You must also have previously created at least one Edge Environment and added it to an Edge group. Tags can be used to assign Edge Environments to Edge Groups.\ \ How to an Add Edge environment : Add an Edge environment\ Documentation on Edge groups : Edge Groups



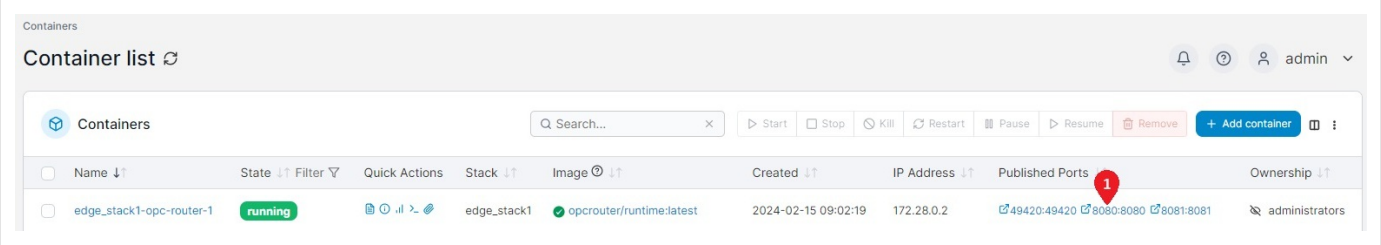
Navigate to the **Edge Stacks (1)** and click on **Add stack (2)**.



On the new page, enter the name that you want into the **Name Field (1)**. Choose a group in the **Edge Groups (2)** Choose **Template (3)** as the Build method and the template **"OPC Router - No-code middleware for industrial applications" (4)**\



And finally, click on "Deploy the stack".



The OPC Router should now be shown on the Container list. Its web management can be accessed via the port to which the internal port 8080 is mapped **(1)**.

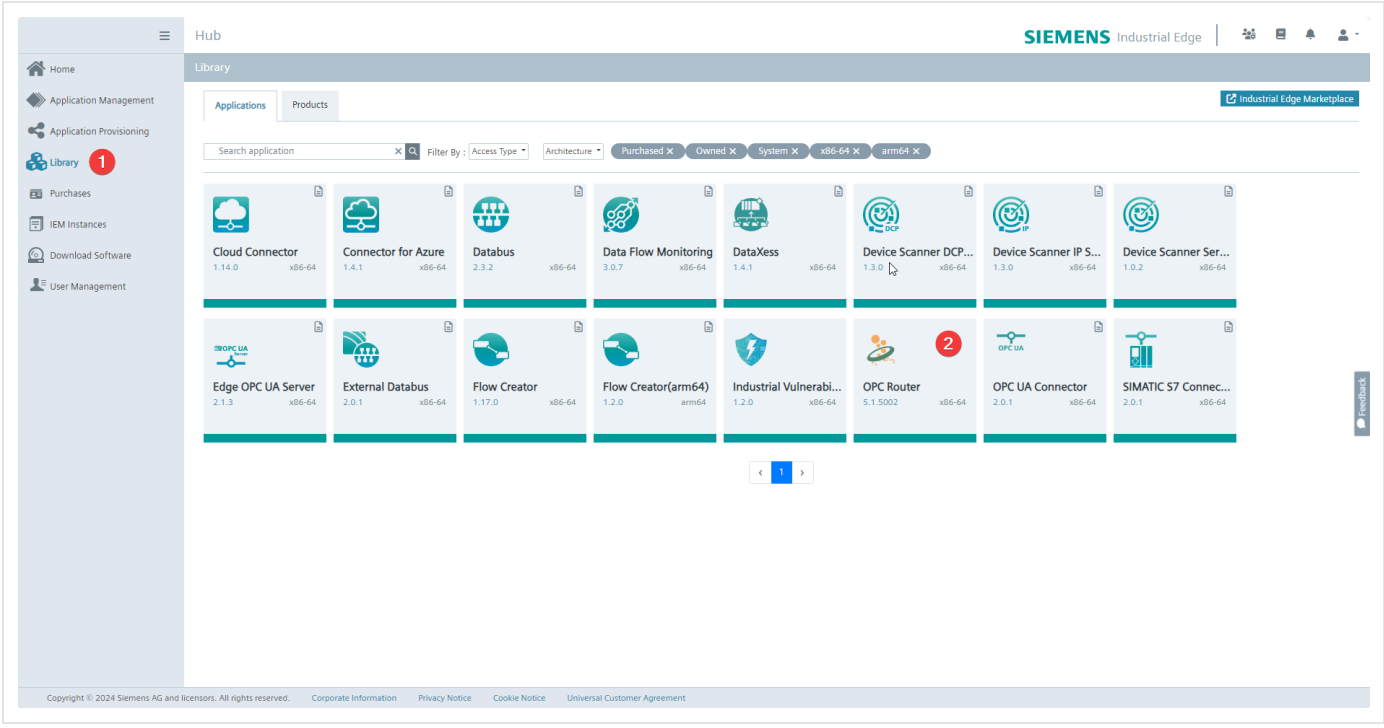
Siemens Industrial Edge

Siemens Industrial Edge offers an open and flexible platform for integrating IT applications directly into industrial automation systems. For OPC Router customers, this means a simplified way of collecting, processing and securely exchanging data directly at the source - in other words at the machine or plant. The platform enables the native integration of IT into automation without having to interrupt the production process. By supporting industrial protocols such as PROFINET, OPC UA and Ethernet/IP, OPC Router customers can collect and process their data efficiently and in real time, which is particularly useful when it comes to implementing AI services.\ \ This edge computing solution expands the possibilities of industrial automation by providing a secure runtime environment for edge applications based on high-level languages and the Docker IT standard. This enables powerful local data processing and analysis directly at the automation level, resulting in optimised operating processes, reduced costs and increased plant availability.\ \ Siemens and its partners offer a wide range of hardware components that comply with industry standards and are specially designed for use in industrial environments. These hardware solutions include SIMATIC controllers, HMI devices with integrated edge functionality, industrial PCs and virtualised systems. In addition, certified edge devices are available from partners in the Siemens Industrial Edge Ecosystem.\ \ For OPC Router customers, this offers the advantage that the OPC Router can be installed and managed via the central management platform from Siemens Industrial Edge. This allows the OPC Router to be easily deployed and managed on the individual systems, resulting in more efficient utilisation of resources and improved scalability. Centralised management also makes it easier to update and maintain the OPC Router installations, as software changes can be made centrally and distributed to the various systems.\

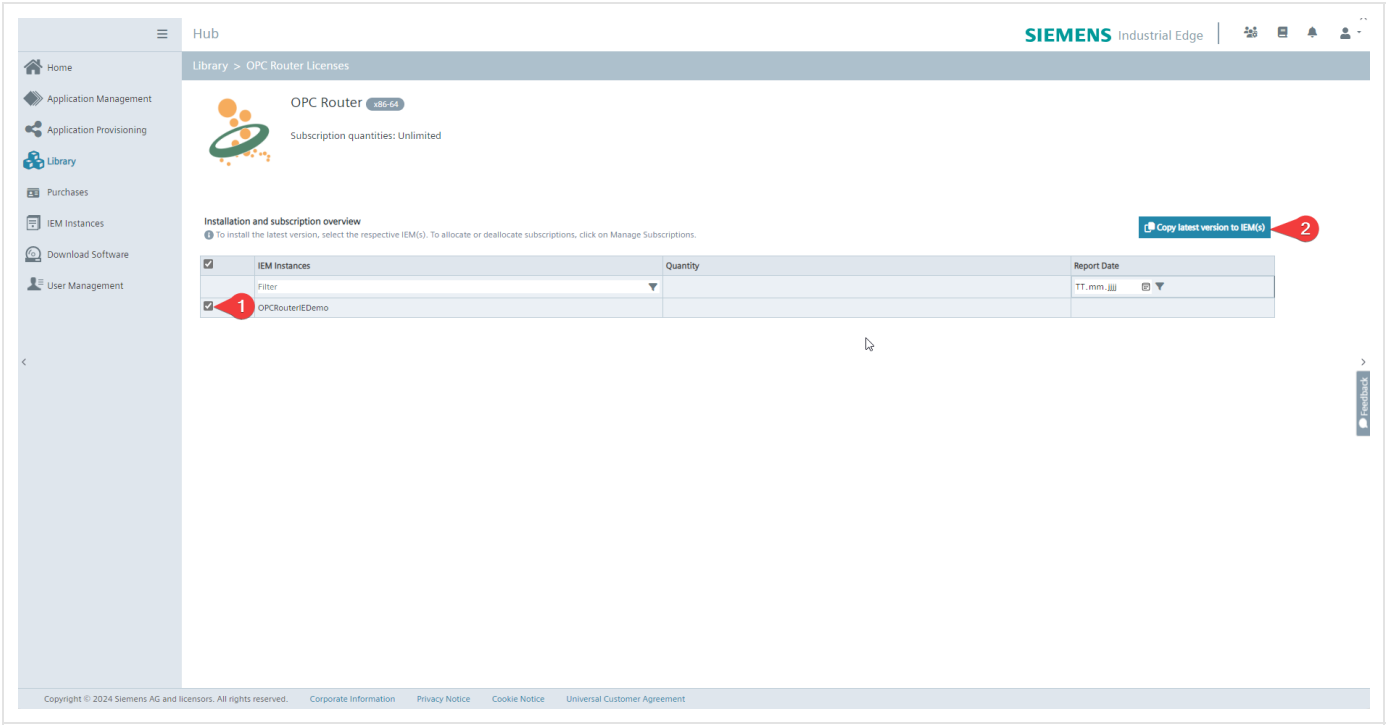
Preparation and requirements

To Install OPC Router with Siemens Industrial Edge, an access to the Hub Siemens Industrial Edge and also an instance of the Industrial Edge management and atleast one device (x84_64) assigned to it are required. \ \ Actually, the OPC Router Application is only available for the devices with x86_64 architecture.\ \ In order to install OPC Router via Siemens Industrial Edge, you must also buy the demo version or a licence from the Siemens Industrial Edge Marketplace. The OPC Router application should then be displayed under your application library in the Industrial Edge Hub.

Installation of the OPC Router on Siemens Industrial Edge

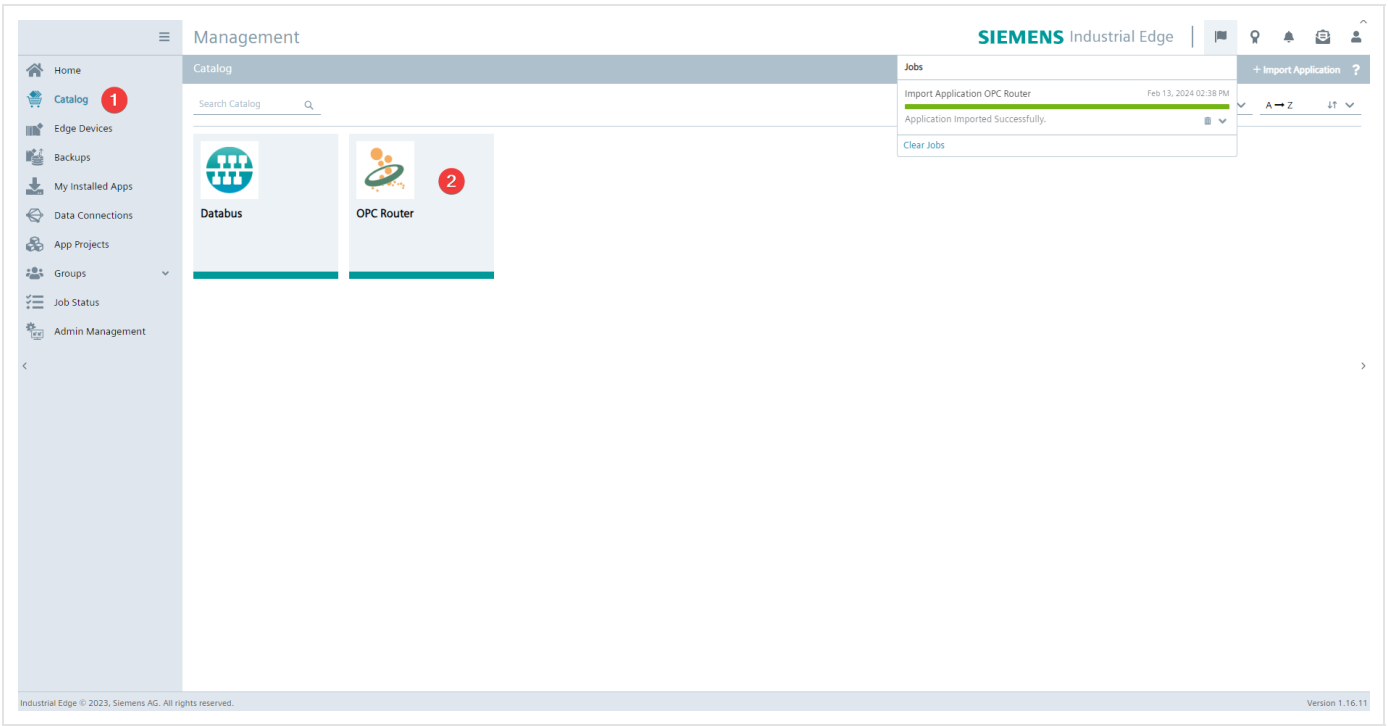


In the Industrial Edge Hub, navigate to the **Library (1)** and select **OPC Router (2)**.

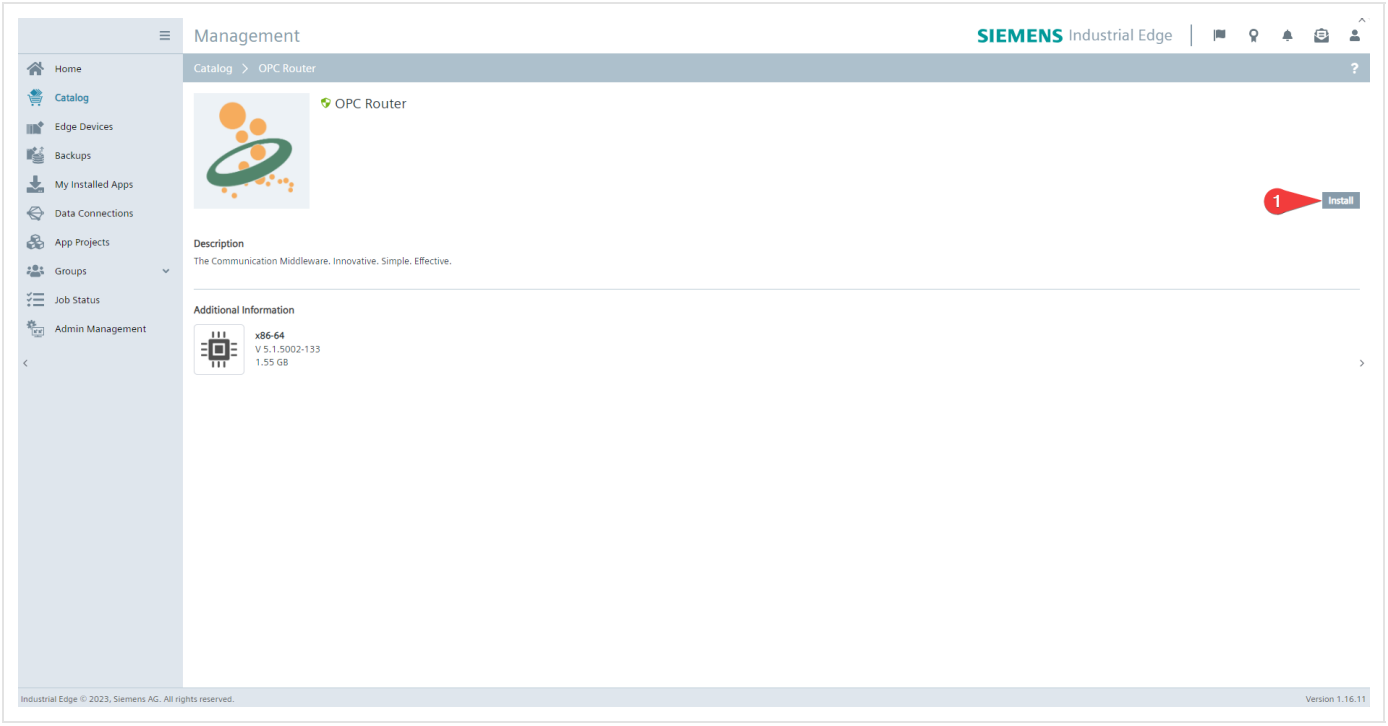


Select the Siemens Industrial Edge Management instances **(1)** on which you want to make the OPC Router application available. By pressing the **"Copy latest version to IEM(s)" (2)**, the managements instances are instructed to copy the OPC Router application from the Industrial Edge Hub, this process can take a few minutes (15-30 minutes).\

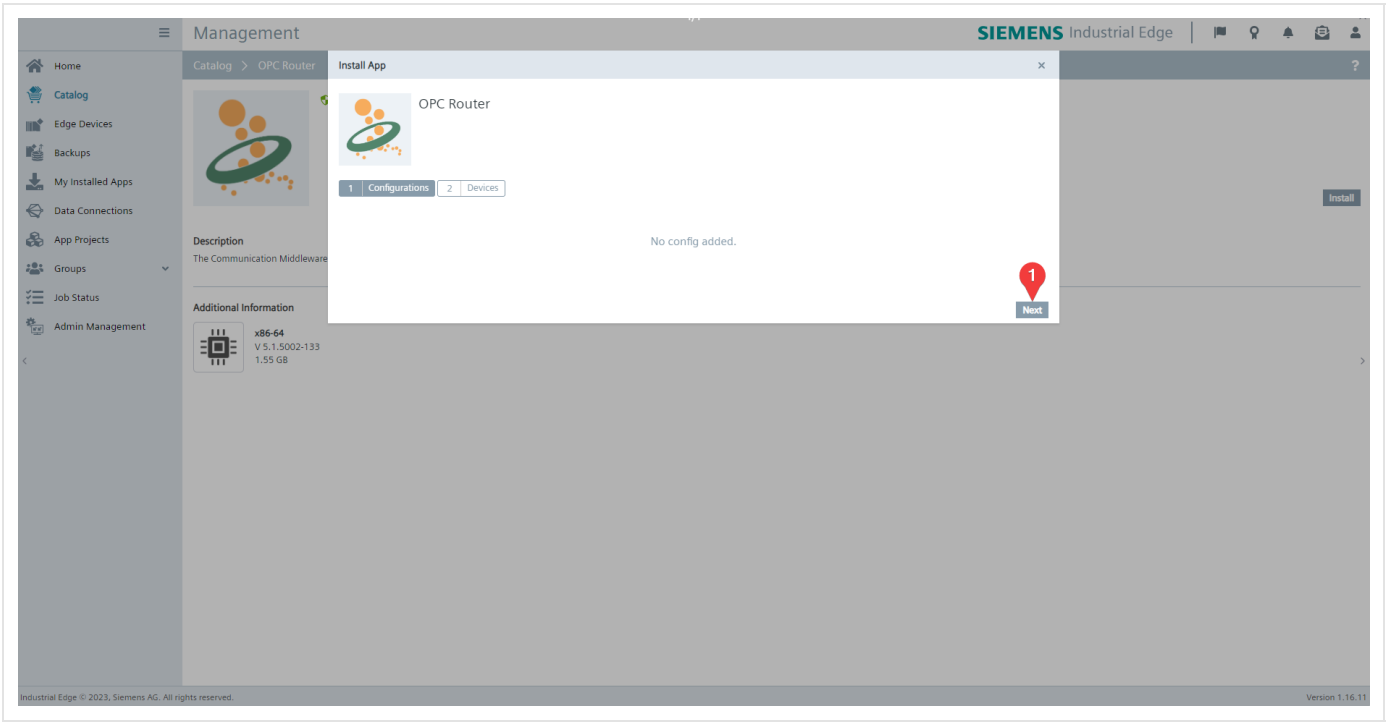
The next step requires an user with rights to install applications on devices for the Industrial Edge Management.



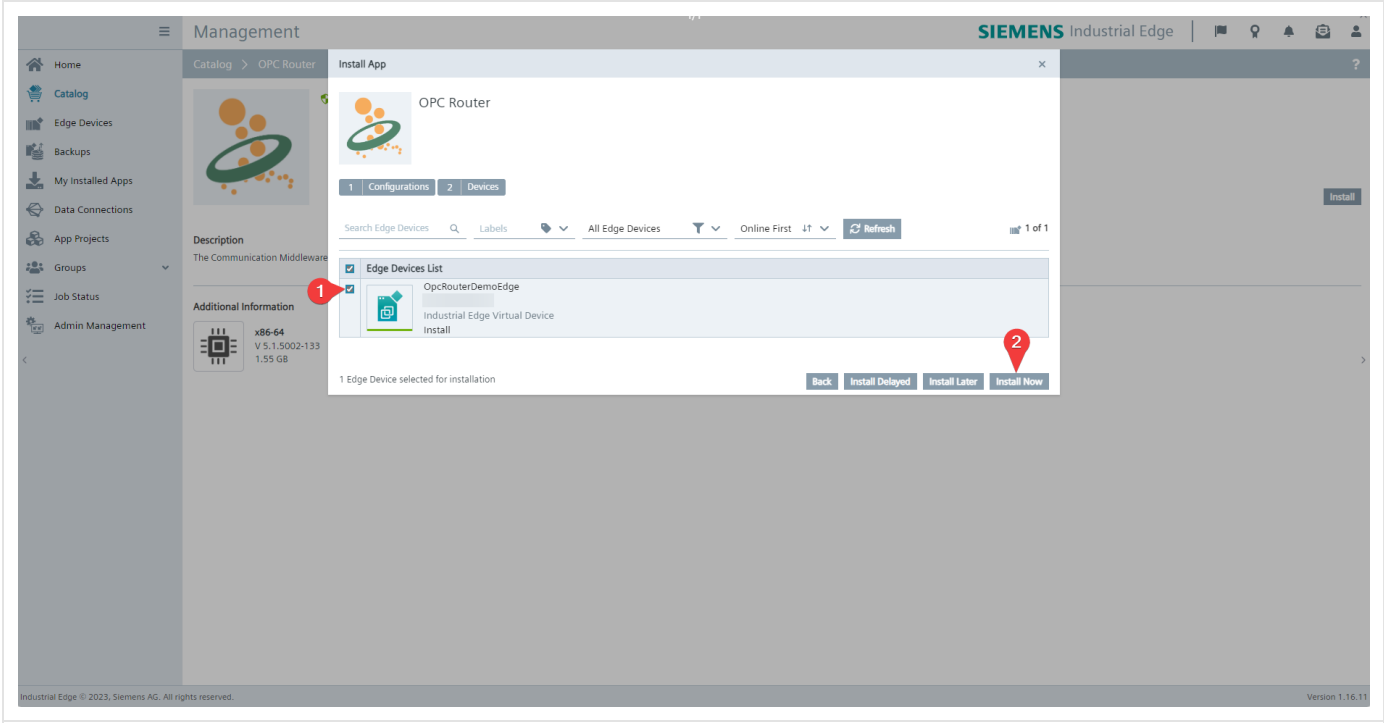
Go now to your Siemens Industrial Edge Management instance. When the importation process in your actual Industrial Edge Management is finished, you can find the OPC Router under the catalogue and select it to install it on a device connected to the instance.\



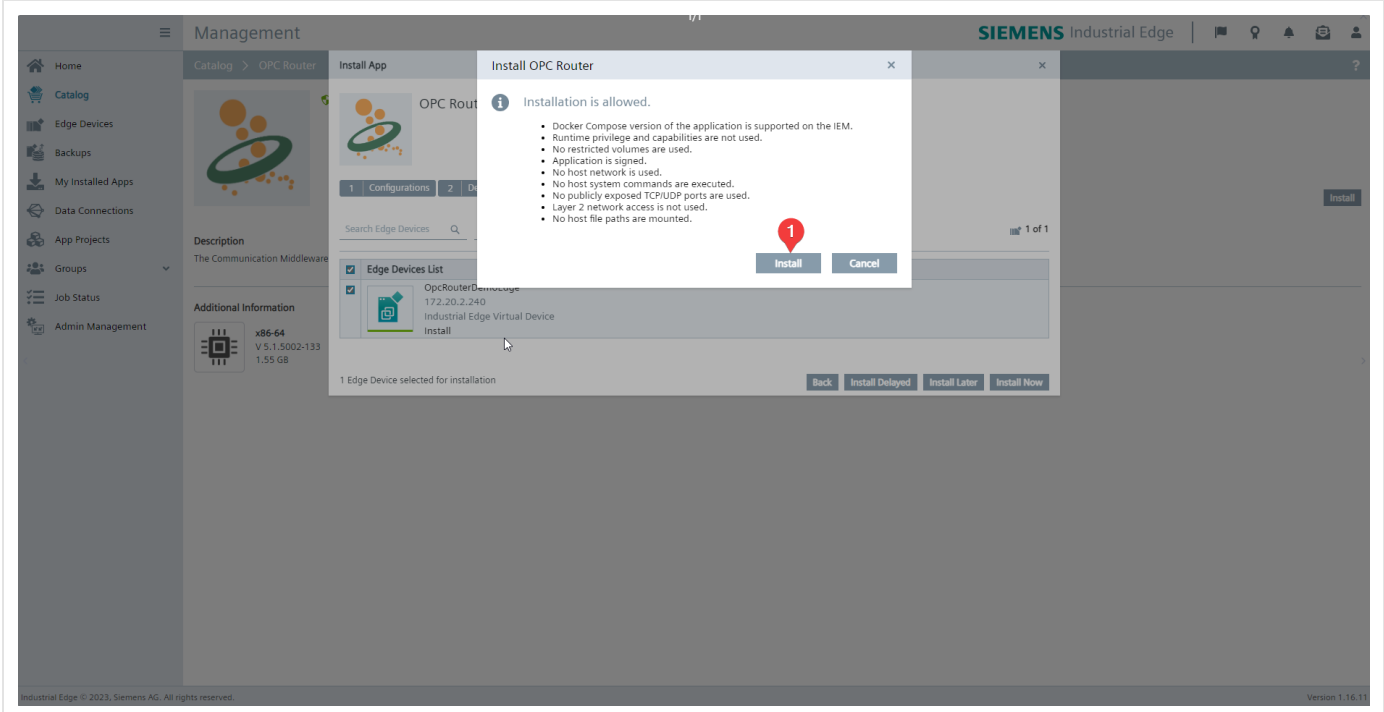
Click on **Install (1)** to install the OPC Router on a device.



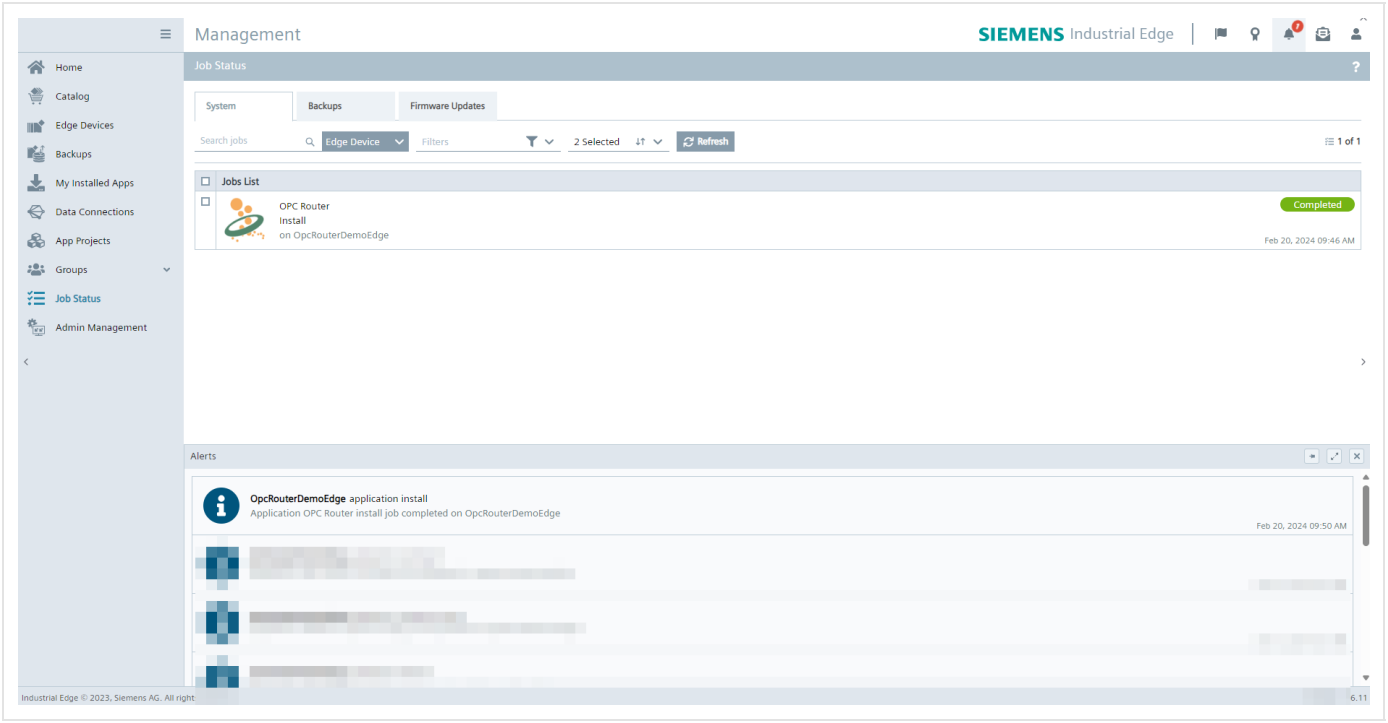
The OPC Router application for Siemens Industrial Edge has no alternative configurations, click **Next (1).**



Select now the device/devices on which the OPC Router must be installed **(1)**. Click then on **Install Now (2)** or on one of the other installation options.

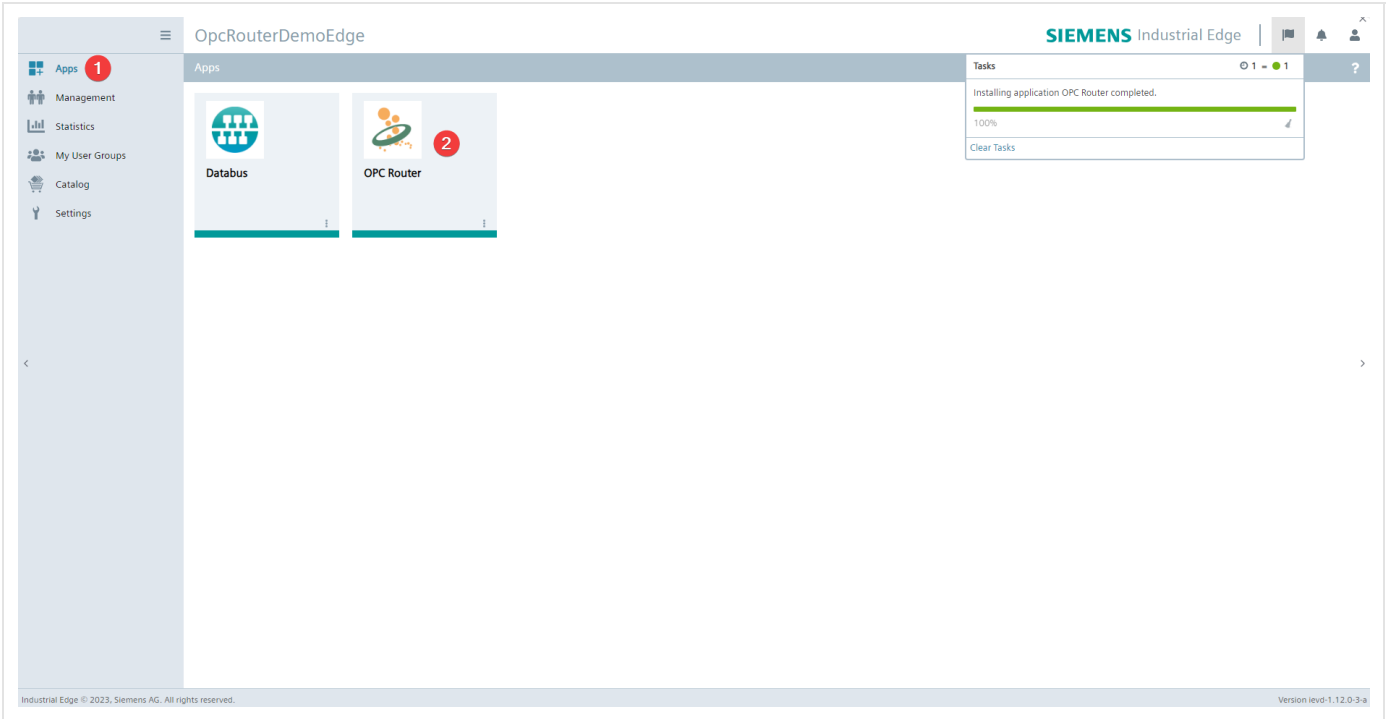


If the device fulfill the required conditions, you can now start the installation by clicking on **Install (1).**



The job list display the actuel state of the installation process. When the process is finished, a notifcation will also appears.

If the OPC Router is uninstalled from a device, all datas stored there in volumes and as well as the project configuration and the licensing are lost. However, the datas are retained if the OPC Router version is updated.



Connect now to the web interface on the devices on which you have installed the OPC Router. The **OPC Router (2)** should now display under the **Apps (1)**, a click on it will open the management of the OPC Router. From here, you can now use the OPC Router as usual and, for example, assign it a license.

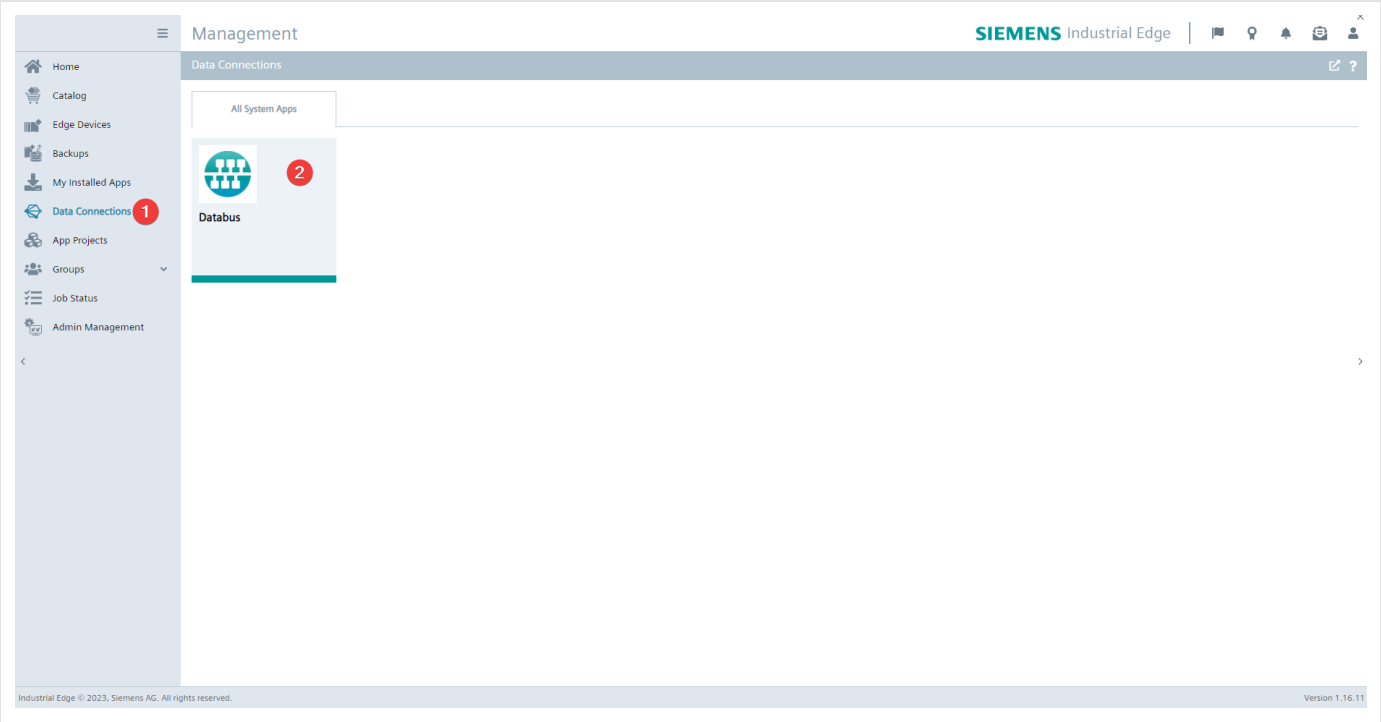
Connection of the Industrial Edge Databus

The Industrial Edge Databus is a standard application developed by Siemens for the exchange of datas between different applications inside the Industrial Edge Ecosystem. Industrial Edge Databus can connect easily to the OPC Router as an MQTT Plug-In instance, which enables mutual data exchange.

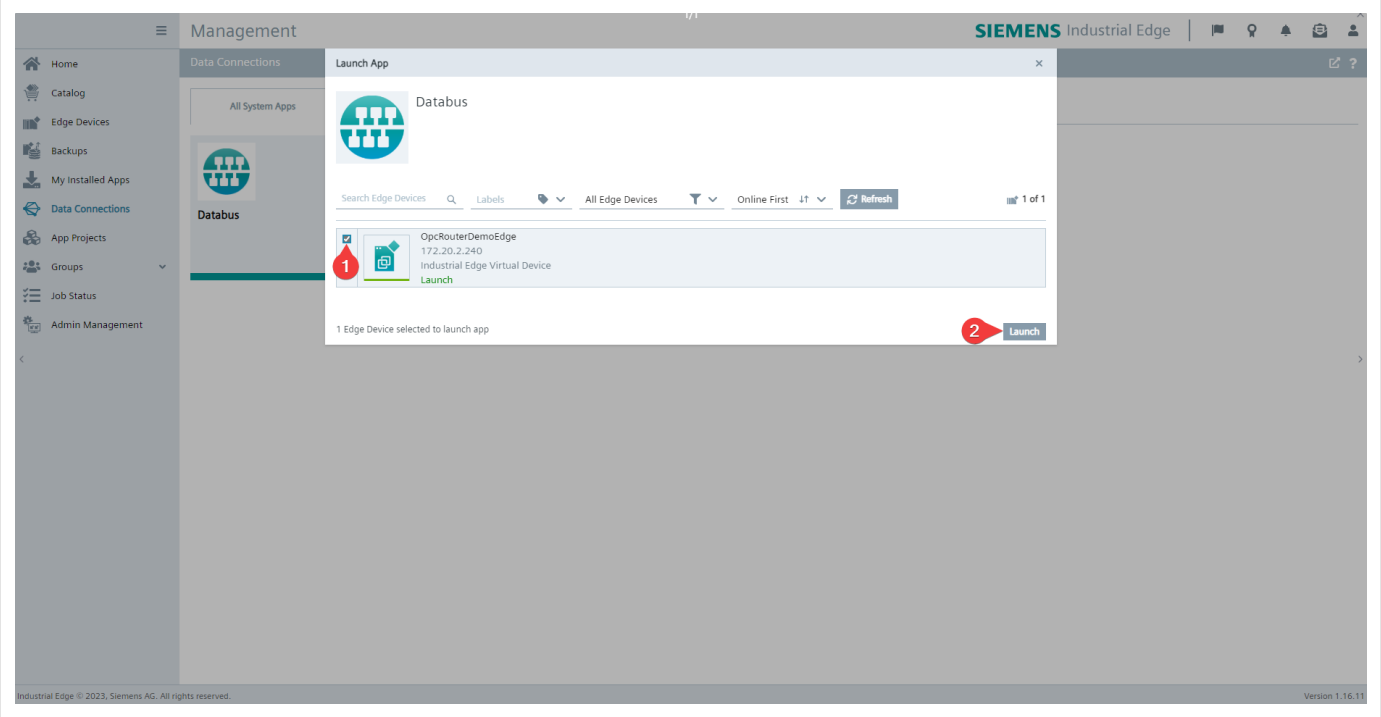
For the following section, it will be supposed that an Industrial Edge Databus is installed on the same device, and also the Industrial Edge Databus Configurator in the same management environmenet as the OPC Router.

Initial configuration of the Industrial Edge Databus

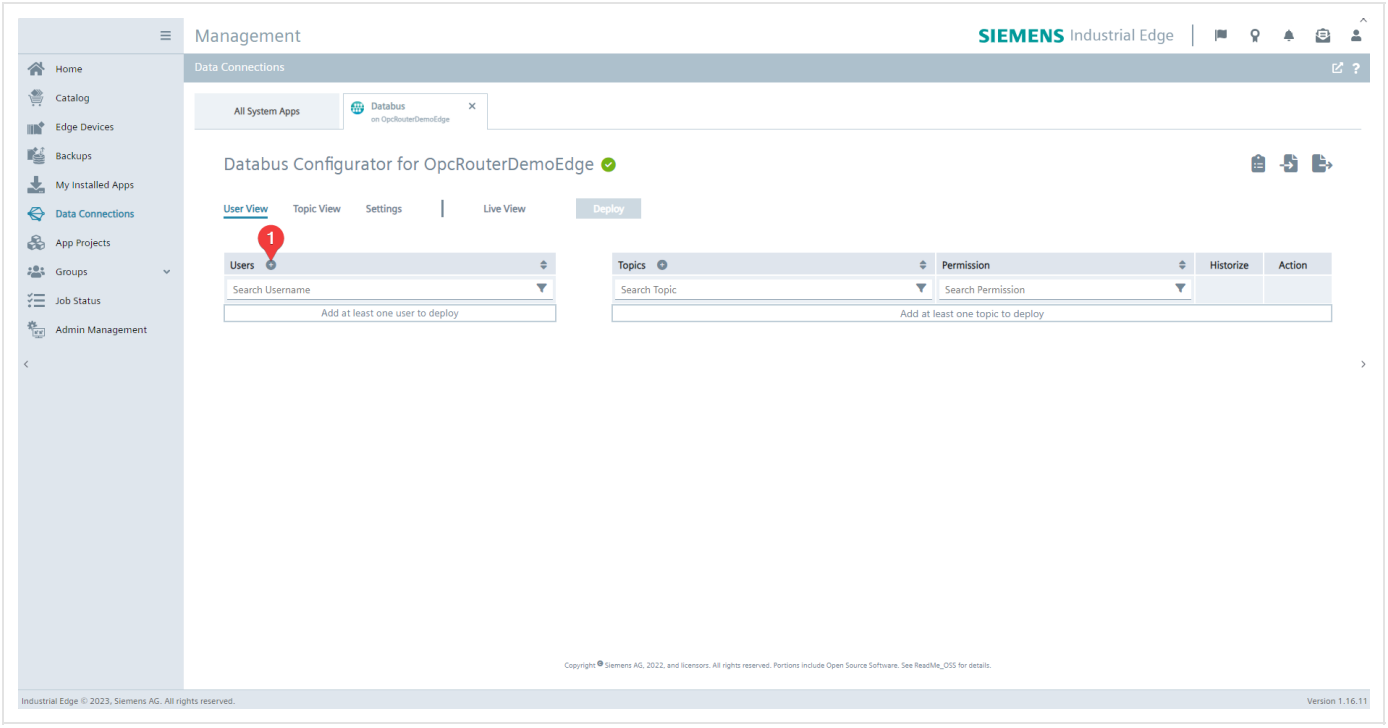
This chapter can be skipped if the Databus already have a topic and associated users with publish and subscribe rights to which the OPC Router can be connected.



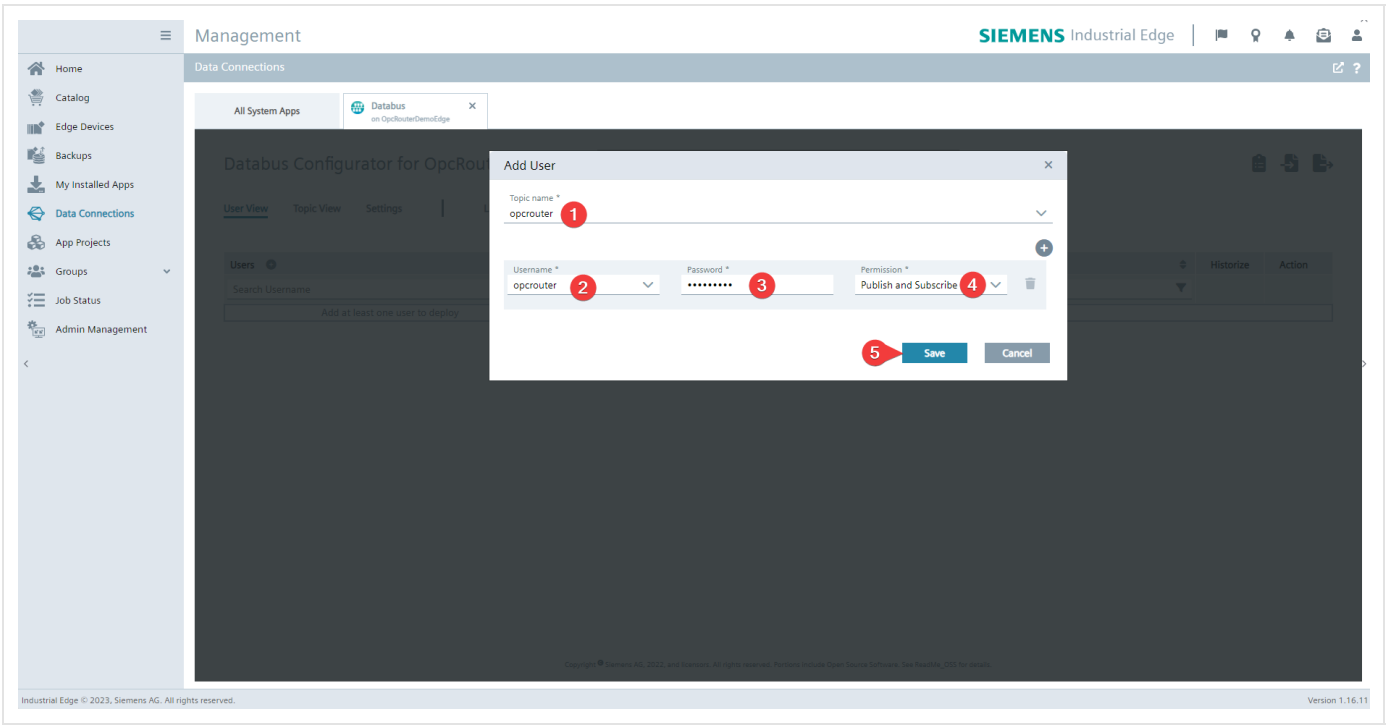
Select the **Databus (2)** under the menu item **Data connections (1)**.



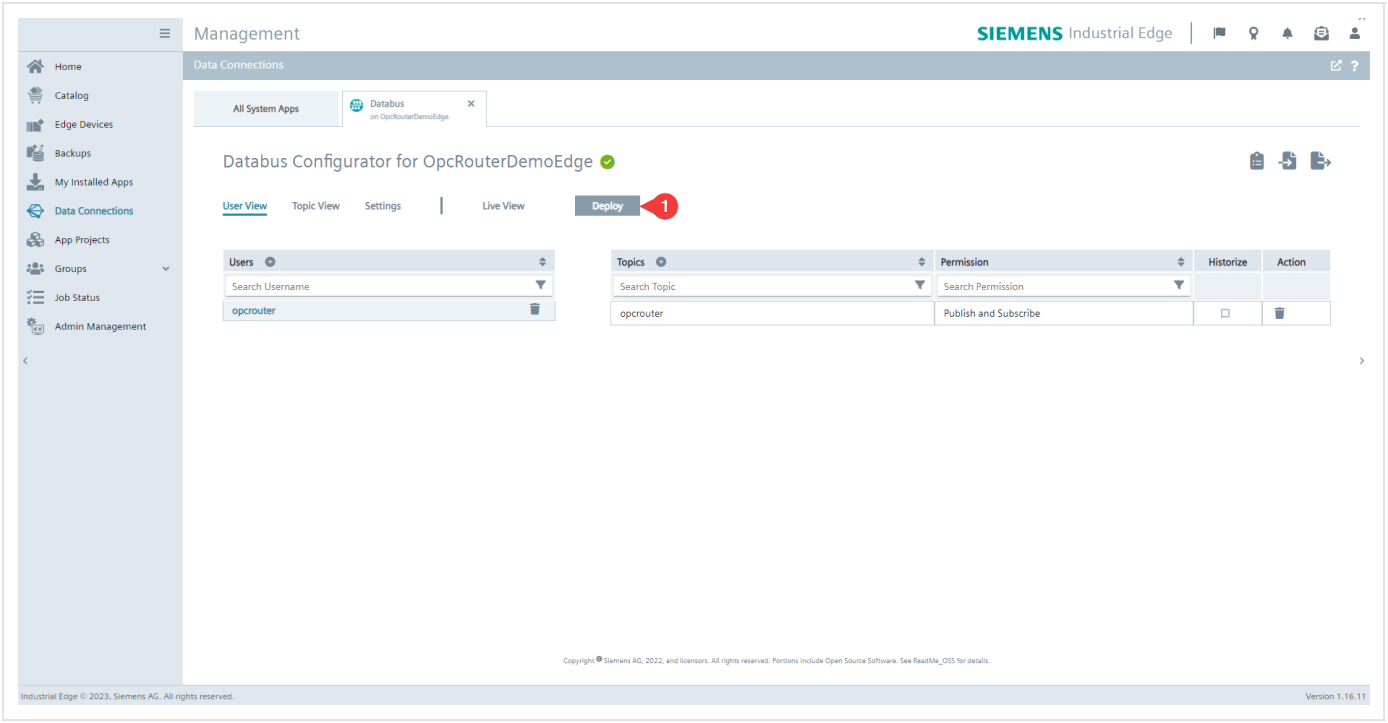
Select the device **(1)** on which is the OPC Router and click on **Launch (2)** to launch the configurator for the Industrial Edge Databus.



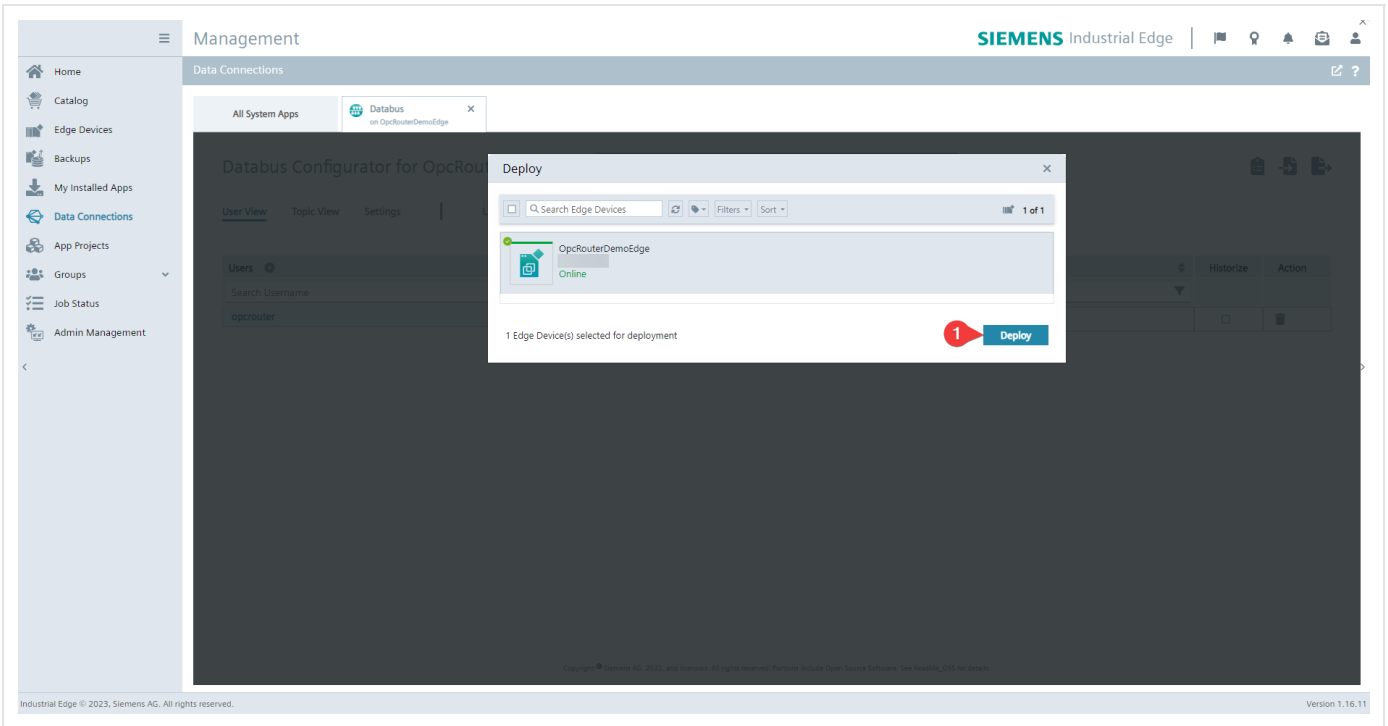
Create a new user by clicking on the **"+"/plus symbol (1).**



Then, enter the **Topic name (1),** the **Username (2)** and the **Password (3).** For the following examples, opcrouter is used as a topic and username only for demo use. For the permission, select **Publish and Subscribe (4).** Confirm now your entries by clicking on **Save (5).**

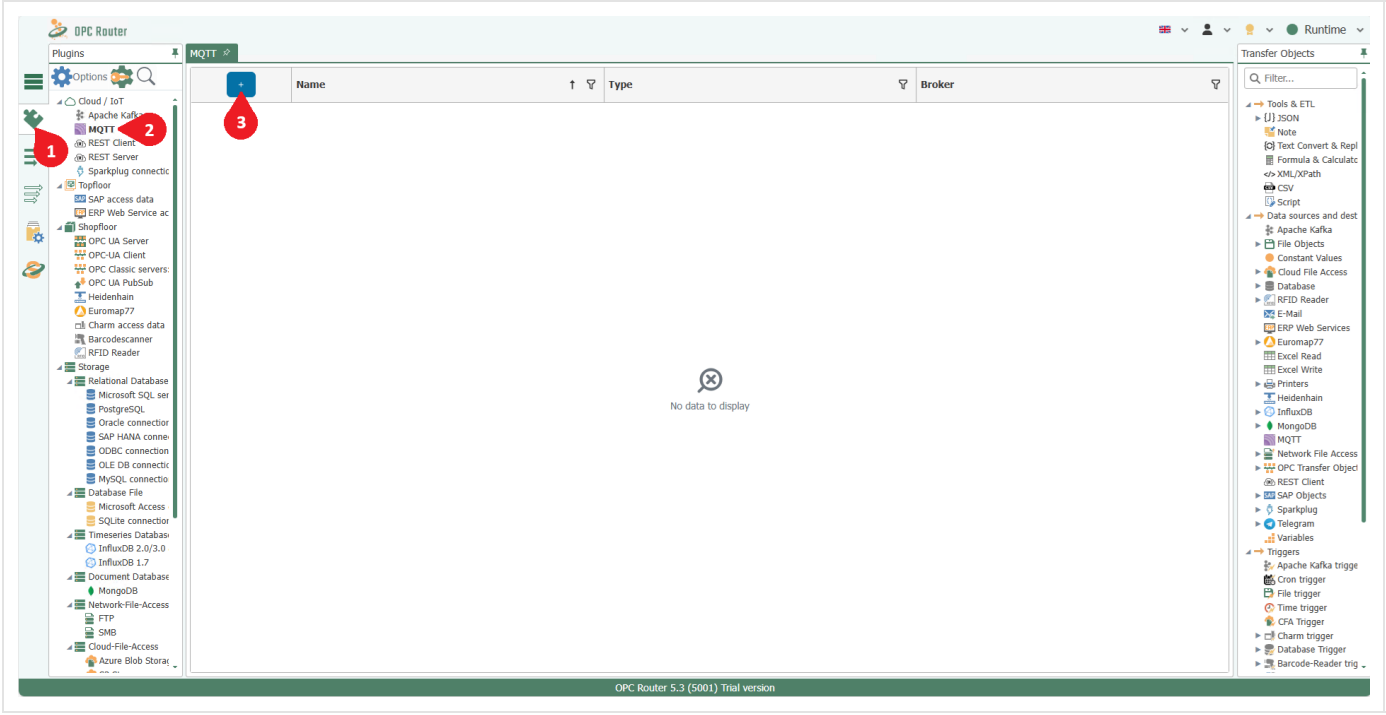


You can now apply your configuration by clicking on **Deploy (1).**

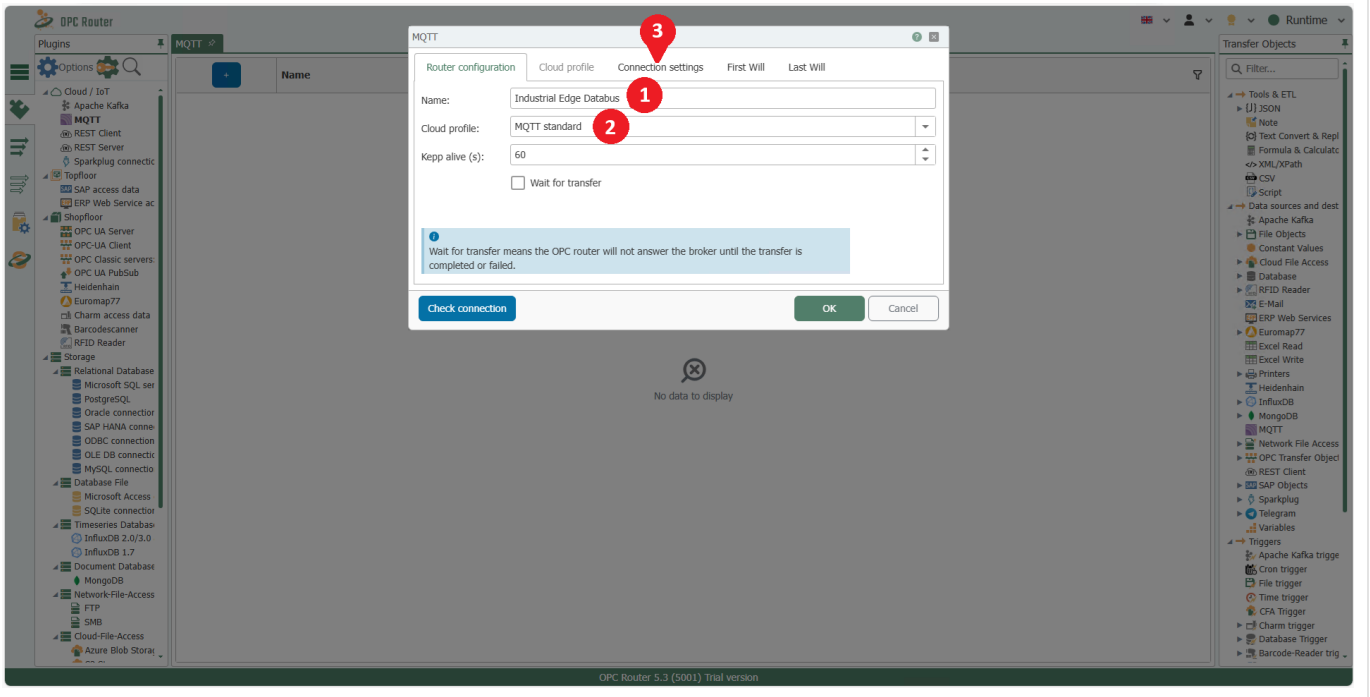


Confirm the device on which the configuration must be applied by clicking on **Deploy (1)**, and wait that the process is finished.

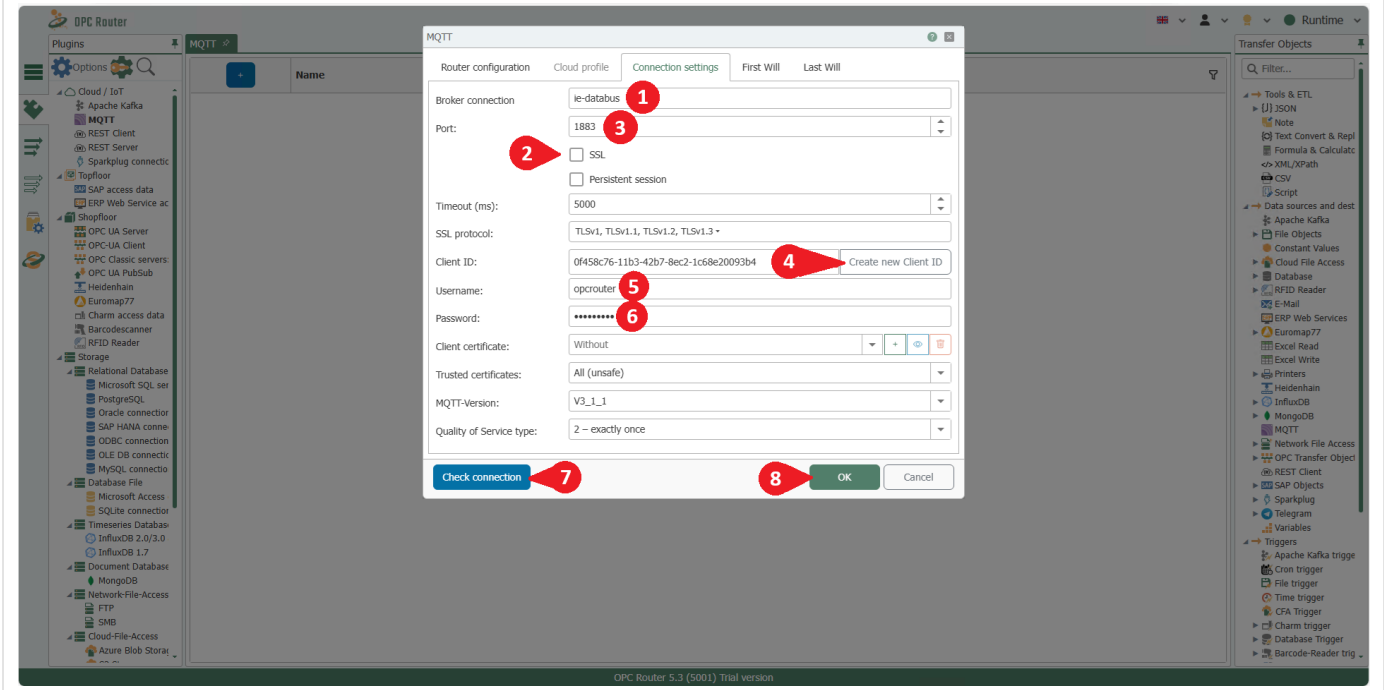
Connect the OPC Router to Databus



Select the **MQTT plug-in (2)** in the OPC Router Management of the device under the menu item Plug-ins **(1)** and create a new instance of this plug-in **(3).**



Give the instance a name of your choice **(1)** and make sure that **MQTT standard (2)** is selected as the cloud profile. Then navigate to the **Connection settings tab (3)**.



Enter **ie-databus (1)** as the Broker connection and switch off **SSL (2)**. This should automatically set the correct port **1883 (3)** - make sure that this is the case. Create a new Client ID by clicking on **Create new client ID (4)** and enter the **Username (5)** and the **Password (6)**. If the **Check connection (7)** is succesful, confrim the entries by clicking **OK (8),** otherwise check your entries for deviations from the settings selected above.\ \ You can now use the MQTT transfer object to send data to topics to which the user has authorisations and use the MQTT trigger to receive data on topics.

Update/Maintenance

Important Update Notice: Secure Your Project and System

Before proceeding with an update of the OPC Router, it is crucial to ensure the security of your project data and system configurations. We strongly recommend adhering to the following steps:

1. **Project Backup**: Backup all of your OPC Router projects. Export the configurations and save them in a secure location. This step ensures that in case of complications during the update process, you can revert to the original settings and configurations.
2. **System Backup**: In addition to project backup, it is advisable to backup the entire system. Utilize standardized IT mechanisms such as VM snapshots or other system backup solutions available in your infrastructure. A complete system backup enables you to perform a full restoration to the state before the update in case of emergencies, ensuring operational continuity and data integrity.

By taking these precautions, you minimize the risk of data loss and system failures that may occur during or after the update process. Your diligence in this phase supports a smooth transition to the new features and enhancements of the OPC Router and protects your investments in your automation solutions.

Tip: How to Create a Backup of Your OPC Router Configuration

Use the backup function of the OPC Router to create a backup copy of your configuration database. Follow these steps:

1. Open the menu via "File" and select "Create Backup".
2. The system automatically saves a backup file in the "Backup" folder located in the installation directory of the OPC Router.
3. The filename of the backup is appended with the date and time to facilitate easy identification and organization. For example: 20210130_153047_OPCRouterConfiguration.bak.

Update to OPC Router 5

We recommend performing each update of the OPC Router first on a test system and testing the corresponding project there.

Depending on the installed OPC Router version, different steps are required to update to OPC Router 5. Here are the steps for various scenarios:

Update to OPC Router 5.4 and greater

In addition to the steps listed below, we recommend an intermediate step to a version 5.1, 5.2 or 5.3 when updating from OPC Router 4 to OPC Router 5 (from version 5.4) due to the changed data storage in YAML files.

Update from OPC Router 4.25

From OPC Router version 4.25 onwards, export your entire OPC Router project and disable the OPC Router service. Now, install OPC Router 5 and import your project. Uninstalling OPC Router 4.28 is not necessary.

Update up to OPC Router 4.18

For OPC Router versions older than 4.18, an update to version 4.18 is initially required. To do this, uninstall the old version and install the latest maintenance version of version 4.18. In OPC Router version 4.18, activate your entire OPC Router project via the OPC Router Management before uninstalling this version. Then install version 4.28 and perform the project migration using the automatic process in OPC Router Management 4.28.

Update from OPC Router 3

For OPC Router versions older than OPC Router 4.0, refer to the OPC Router 4 manual on how to perform the update to OPC Router 4.18, and then proceed with the update from OPC Router 4.18 to OPC Router 5.

With your OPC Router 4 license and a valid maintenance contract, you can license OPC Router 5. If OPC Router 4 is intended to remain in production, an additional license will be required.

Update Process for Windows

Updating the OPC Router from one version in the 5 series to the next higher version involves uninstalling the current version and then installing the new version. It's important to note that your project files and configuration settings will not be affected by this process.

Note: To ensure compatibility of your current configuration with the new version of the OPC Router, we strongly recommend conducting a pre-update test on the production server. This can be done, for example, on a virtual machine. Generally, the OPC Router is designed to automatically adjust existing projects after an update, if necessary.

1. **Compatibility Check**
   - Before updating on the production server, ensure that your current configuration is compatible with the new OPC Router version. This can be done through a pre-update test on a virtual machine. In most cases, the OPC Router automatically adjusts existing projects after an update, if necessary.

2. **System Backup**
   - Perform a full backup of your system to ensure that you can revert to the previous configuration in case of any issues.

3. **Preparation**
   - As the OPC Router setup may update or install Windows system components, close all open programs before starting the setup.
   - Make a note of the user account under which the OPC Router service is running, as you will need this information later.

4. **Uninstalling the Old Version**
   - Uninstall the existing OPC Router 5.x version via the Control Panel. Your project configurations and files will remain intact.

5. **Checking System Requirements**
   - Ensure that all system requirements for the new version of the OPC Router are met.

6. **Installing the New Version**
   - Run the setup.exe file from your installation source to install the new version of the OPC Router.
   - Follow the instructions of the installation routine carefully.

7. **Configuring the User Context**
   - Configure the user context of the OPC Router service according to the notes from step 3.

8. **Project Activation**
   - After completing the installation, open the configuration environment of the OPC Router.
   - Switch your project to productive mode.

9. **Restarting the OPC Router Service**
   - Restart the OPC Router service to apply the changes and complete the update.

Update Process for Docker

Updating the OPC Router from a version in the 5 series to the next higher version involves removing the container of the current version and subsequently redeploying a container of a new version. It is important to note that your project files and settings could be affected by this process depending on the container configuration unless data persistence is ensured beforehand.

1. **Compatibility Check**
   - Before updating on the production server, ensure that your current project setup is compatible with the new OPC Router version. This can be done through a preliminary test on a virtual machine. In most cases, the OPC Router automatically adjusts existing projects after an update if necessary.
   - Ensure that all system requirements for the new version of the OPC Router are met.

2. **Preparation**
   - Export your project to ensure that you can revert to the previous configuration in case of any issues.
   - Ensure that data is persisted in relevant paths to avoid losing project data, settings, license activations, or log files.

If no mappings were set up during the creation of the existing containers, all data will be lost when deleting the containers. To manually back up data from the container, you can use the following command for all relevant paths:

```
docker cp containername:/path/in/container /path/in/dockerhost
```

1. **Removing the container with the old version**
   - The existing container can be removed using the following command:\ `docker rm containername`\ If the container was created using Compose, it is recommended to remove the container again using `docker compose down` to ensure that no unused components remain.

This step cannot be undone. Make sure to back up all data you intend to use in the new container beforehand.

1. **Creating the container with the new version**\ Download the desired new image using `docker pull image:tag`. Available images and tags can be found on our Docker Hub page.
   - **Compose:**
     - Adjust the value of the `image` field in the docker-compose.yml file to create a container with the desired version.
     - Modify environment variables, port mappings, and volume mappings according to the documentation if necessary.
     - Create and start the container using `docker compose up`.
   - **Docker CLI:**
     - Create and start the container using `docker run <options> image:tag`. The options to use depend on your specific use case and can be found in the documentation.

2. **Finalization**
   - If you have transferred data from your previous container, the container is now ready for use, and no further actions are required. However, if you set up the container without transferring data, you can now use the demo, apply a license, create a new project, or import an existing project, and then put everything into production.

Licensing

The OPC Router offers flexible licensing models and can be operated without a license in 2-hour demo mode. In this mode, all functions are available, but the runtime environment terminates the data transfers after two hours, which can be reactivated for two hours by restarting manually. The two license models Perpetual and Subscription are explained below.

Licensing options

Perpetual

- One-off payment for unlimited use.
- Contains basic functions such as scripts, calculators, data transfers and more.
- Expandable with additional plug-ins, add-ons and ETL tools.
- **The maintenance and support entitlement** must be purchased additionally at the time of purchase and is valid for **one year**. The entitlement can be extended annually in order to continue to receive support and updates.

Subscription

- Annually adjustable rental license with different levels.
- **Includes maintenance and support** for the duration of the subscription.
- Divided into three levels: **Plain**, **Standard** and **Enterprise.**
- Each level offers a basic set of functions with the option of expansion.

A subscription license is valid for a limited period of time. If the license has expired, a new license must be installed, an extension must be installed in order to continue using the OPC Router productively. The automatic license synchronization can do this for you in the background.

Attention: For subscription licenses, a new license must be installed before the subscription expires in order to maintain full operation, otherwise the system will revert to 2-hour demo mode.

A detailed overview of the license levels can be found in the License Levels chapter.

Further information on the available licenses and models can be found on our website: License models for the OPC Router.

Further licenses

In addition, inray offers demo and education licenses. These behave in the same way as the subscription license.

If you have any questions about these licenses, please contact the sales department.

License activation

Note: Licenses must be activated on the system. A license portal is available for this purpose. This portal makes it easy to manage and activate licenses to ensure smooth use of the OPC Router.

- **Online activation:**
    - Possible with or without synchronization.
    - Without synchronization, only the exchange of activation information takes place automatically.
    - With synchronization, extensions or renewals are automatically synchronized.\
- **Offline activation:**
    - Requires the exchange of an activation file via a system with Internet access.

Note: In container operation (e.g. Docker, Kubernetes), online synchronization is required if this is operated without a TPM2 module.

Docker Container with TPM2

- **Licensing via environment variables**
  - For licensing via environment variables, the system must have internet access.

OPC Router license portal

The OPC Router License Portal serves as a central point of contact for managing and monitoring your OPC Router licenses. It enables easy access to license information and the management of licensed systems.

Access to the portal

The portal can be reached at: https://license.opc-router.com/

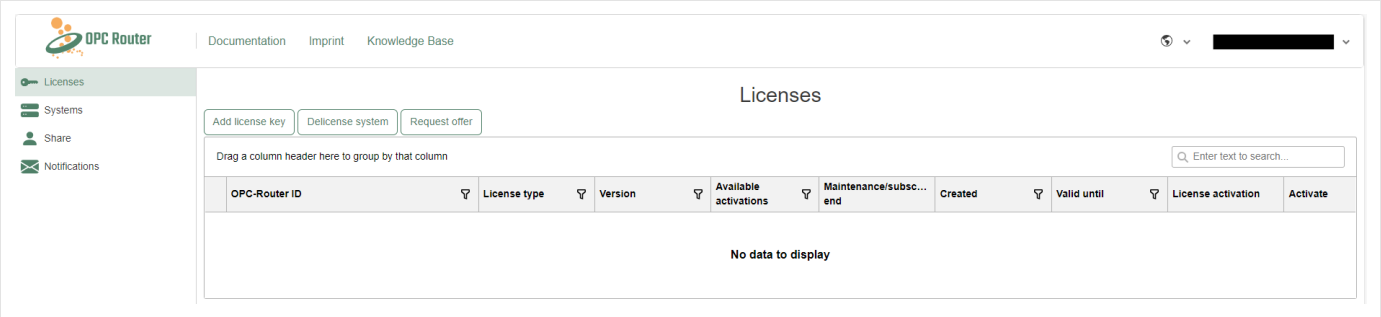
Account creation and registration

- **New users** can create an account by entering the required information.
- **Existing users** can log in with their login data or use the login via services such as Microsoft.

Note: If a license is transferred from one account to another, both account holders will receive an e-mail notification about the license transfer.

License overview

The license portal provides a comprehensive view of all licenses, their types, versions and activation status. This overview facilitates the management of your OPC Router installations and license inventories.



License overview

Explanation of the license overview

COLUMN	EXPLANATION
OPC-Router ID	Each license is uniquely identified by an OPC Router ID.
License type	The license type indicates whether it is a subscription, perpetual, educational or another type of license.
Version	The Version column shows the OPC Router version from which the license is valid.
Available activations	Shows how many activations are still available. In exceptional cases, more activations than have been purchased are permitted with a corresponding warning and notification to the distributor. Overprovisioning is indicated by a negative value.
Maintenance/subscription end	In the case of a perpetual license, the number of days for which support is still available is displayed. A subscription license shows how long the license is valid for.
Created	Creation date of the license
Valid until	Date on which the license expires
License activation	An exclamation mark indicates that a manual license activation is required. This is the case for offline systems or if license information has changed in the portal, for example due to an extension of maintenance or the purchase of additional plug-ins.

Further information

The lines can be expanded to display further information. After expanding the license overview, the licensed modules and the systems on which the license is active are displayed.

License details

Activate license (offline licensing)

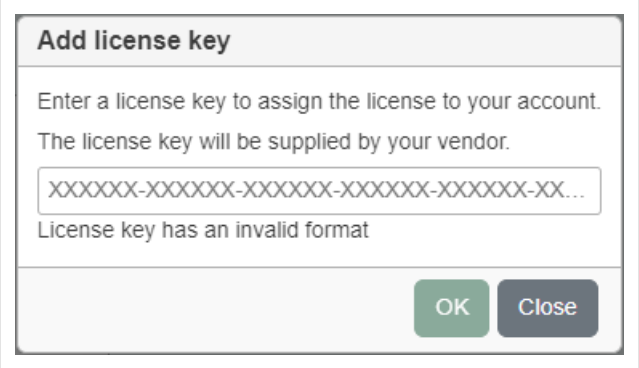
For systems without a direct internet connection, the license portal offers an option for offline licensing. This is made possible by the key button in the "Activate" column. One click initiates the necessary steps to activate a license without online access. The process is described in more detail on the Activate license in the OPC Router page. This procedure is particularly useful for secure network environments that do not allow direct Internet access for security reasons.

The lines can be expanded to display further information about the license. After expanding, you will see an overview of the licensed modules and the systems on which the license has been activated.

Add license to portal

You can add a license to the license portal if you have received a license key from your supplier. To add a license in the license portal, please follow the steps below.

1. Click on the "Add license key" button.
2. Enter or paste in your license key and click on "OK".



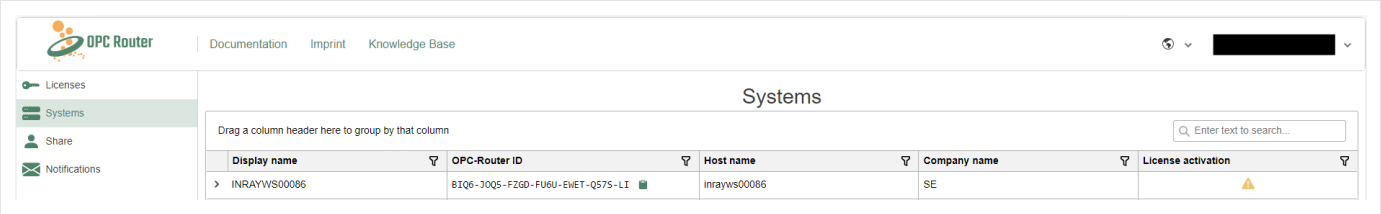
Entering the license key

Once added, the license is displayed in the license overview.

Remove license from a system

In the license portal, you can delicense a system independently using the "Delicense system" button. How exactly this works is described on page Delicensing.

System overview



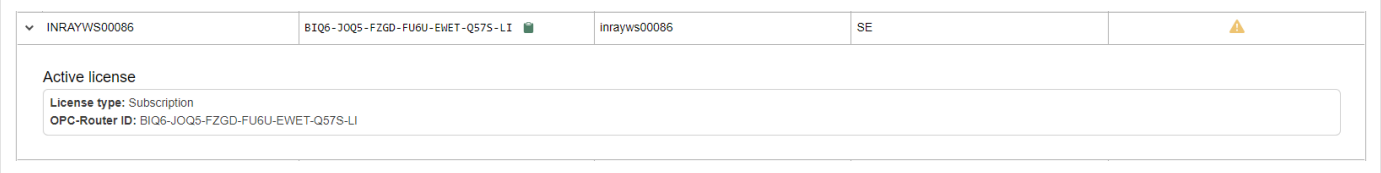
System overview

Explanation of the system overview

COLUMN	EXPLANATION
OPC-Router ID	ID of the license that is active on the system.
Host name	The system name that was specified in the OPC Router
Company name	The licensee specified in the OPC Router.
License activation	An exclamation mark indicates that a manual license activation is required. This is the case for offline systems or if license information has changed in the portal, for example due to an extension of maintenance or the purchase of additional plug-ins.

Further information

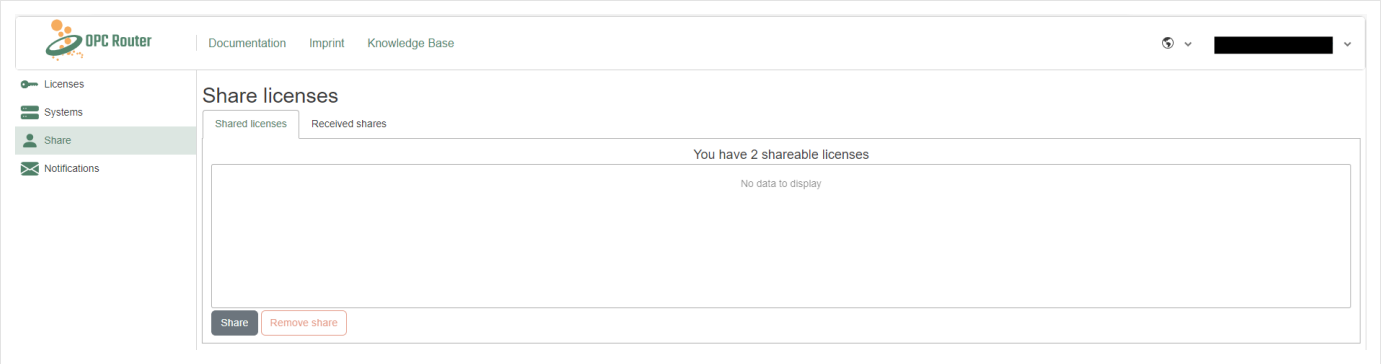
The lines can be expanded to display further information. The license type is displayed in the system overview after expanding.



System details

Sharing and transferring licenses

Sharing licenses



Share licenses

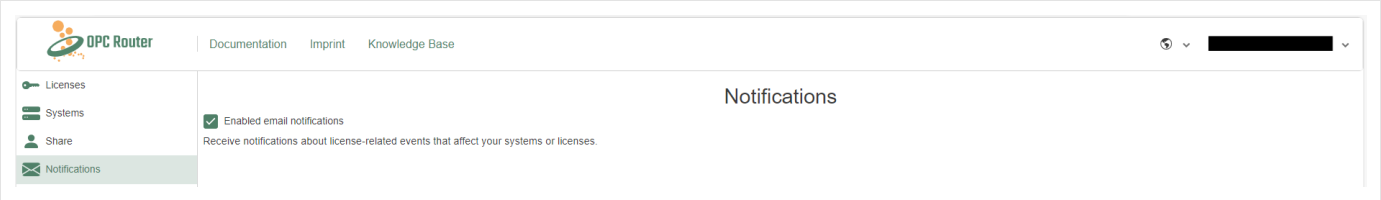
The portal allows you to share licenses with colleagues to facilitate collaboration. In the "Share" section, you can share your licenses with colleagues and receive approvals from colleagues. All you have to do is enter the email address of the account you want to share the license with. Shared licenses can be used and applied in the same way as personal licenses. Of course, shares can be removed at any time.

Transferring licenses

It is easy to transfer licenses to other accounts in the OPC Router license portal. To transfer a license, simply add the license key to the account to which the license is to be transferred. How to add a license key is described here. This automatically transfers the administration authorization to the other account. This function facilitates the reorganization of licenses within teams or companies.

Note: When a license is transferred from one account to another, both account holders receive an e-mail notification of the license transfer.

Email notifications



Notifications

If the notifications are activated, you will receive emails when changes are made to licensed systems, when a license is activated and for other license-related events that affect your systems or licenses.

If a license has been shared with you, you will also receive notifications for this license.

License levels

The plain license offers you complete freedom in the configuration of the required plug-ins and add-ons.

The following functions are already included:

- Scripts
- Calculator
- XML/ XPath transfer
- JSON transfer
- CSV transfer
- Time trigger / Cron trigger
- Variables
- Constants
- Text replace
- Local file transfer

The following functions and plug-ins are included in the standard license:

- OPC UA / Classic Client
- OPC UA Server
- OPC UA Pub/Sub
- MQTT
- Sparkplug
- RESTful
- DB (InfluxDB, SQL Server…)
- Telegram Messenger
- Email
- RFID-AutoID (OPC UA)
- Printer (Domino, Videojet etc.)
- Network File Access
- Excel
- CHARM
- Barcode Scanner
- All functions of the plain license

The following plug-ins and add-ons are included in the Enterprise license:

- All available functions
- All available plug-ins
- All available add-ons
- Redundancy
- Add-on Enterprise+

- Comprehensive innovation guarantee

The Enterprise+ add-on includes the functions "Dynamic template instances", "Redundant connection of sub-systems" and "Extended notification groups".

Activate license in the OPC Router

You can test all standard modules extensively in demo mode. After licensing, the licensed modules are productive, the others remain in demo mode.

If you require a longer demo license, this can be issued by our sales department. E.g. for a 30-day demo license. Please contact our sales department for this.

Licensing is carried out via the OPC Router license portal. Licenses can be imported online or offline.

The following domains must be accessible for online licensing:

https://opcrouter.eu.auth0.com\
https://login.opc-router.cloud\
https://license.opc-router.com\
https://crl.inray.cloud\
http://inray.de\
https://inray.de

Licensing

Licensing can be done via environment variables. You can find more information on this on the page licensing-via-environment-variables.md.

Proceed as follows for licensing in the OPC Router.

1. Navigate to the menu by clicking on .
2. Select the menu item "License".

Now you can perform Offline Licensing or Online Licensing. You will find the respective steps in the corresponding paragraph.

Offline licensing

1. Click the “Offline Licensing” button.
2. Enter your data in the “Enter system data” window and confirm by clicking “Next”. If the system has already been licensed, this window will not appear.
3. A request file will now be automatically created and downloaded. This file is needed on a device with internet access to activate the license. To create the file on a device with internet access, you can save the file to an internal network drive or USB stick, for example.
4. Use a device with internet access to open the OPC Router License Portal.
5. Log in to the OPC Router license portal. If you do not yet have a user account, you can create a new user account or log in with your existing Microsoft or LinkedIn account.
6. After successful login, you will see an overview of your licenses. If the desired license is not listed, you can add the license using the “Add license key” button.
7. Click on the key symbol in the “Activate” column of the license you want to use.
8. Follow the instructions in the pop-up and upload the request file. You can start at step 2, because you have already created the request file.
9. Confirm the dialog with “Activate”. If there is overprovisioning, a reason must be given.
10. An activation file is now automatically created and downloaded. Copy this file to the server on which the OPC Router is installed.
11. Click on the “Import license file” button in the “Licenses” menu.
12. Now select the activation file in the file manager and confirm to complete the activation.

Online licensing

1. Click on the “Online Licensing” button.
2. A popup will now appear with a link and a code. Click the link to continue the process in a new tab.

3. Please now compare the code from the OPC Router with the code displayed in the newly opened tab. If the codes match, click on “Confirm”. If the codes do not match, please click on “Cancel” and try again.
4. Click on “Continue with License Portal”.
5. Log in to the OPC Router license portal. If you do not yet have a user account, you can create a new user account or sign in with your existing Microsoft or LinkedIn account.
6. After successful registration, a message will appear telling you that you are finished. You can now close the tab and continue the process in the OPC Router.
7. Enter your data in the “Enter system data” window and confirm by clicking “Next”. If the system has already been licensed, this window will not appear.
8. Another window will now open in which you can select a license. You can also enter a license key here, if you have one, to add a new license.
9. Click Next to confirm your selection.

The licenses are applied immediately and the OPC Router service does not need to be restarted. The validity period for each licensed plug-in and add-on is displayed in the overview at the bottom of the page. Further information can be found on the License overview in the OPC router page.

Future system extensions will be licensed for your specific customer name and OPC Router ID. License keys will be credited to your account and do not need to be entered separately.

Licensing via environment variables

The OPC Router can be licensed using environment variables. This method is particularly suitable for use cases where Docker or Docker Compose is used. It allows many systems to be licensed with little effort.

An overview of the required and optional environment variables can be found here: Environment variables for licensing

A deployment token is required for environment variable licensing. You can generate this in the Licensing Portal on the Deployment Automation page.

Examples

Docker run command

Linux

```
docker run -d --pull always \
    --name opcrouter5 \
    -e INITIAL_USERNAME="******" \
    -e INITIAL_PASSWORD="******" \
    -e OR_I_ACCEPT_EULA=true \
    -e OR_LICENSE_KEY="YourLicense" \
    -e OR_LICENSE_OPCROUTERID="YourRouterID" \
    -e OR_LICENSE_DEPLOYMENTTOKEN="YourToken" \
    -e OR_INSTANCE_NAME="YourInstance" \
    -v opc-router-5-data:/data \
    -v opc-router-5-logs:/var/log/opcrouter \
    -p 8080:8080 \
    -p 8001:8001 \
    -p 49420:49420 \
    opcrouter/runtime:latest
```

Windows

```
docker run -d --pull always `
    --name opcrouter5 `
    -e INITIAL_USERNAME="******" `
    -e INITIAL_PASSWORD="******" `
    -e OR_I_ACCEPT_EULA=true `
    -e OR_LICENSE_KEY="YourLicense" `
    -e OR_LICENSE_OPCROUTERID="YourRouterID" `
    -e OR_LICENSE_DEPLOYMENTTOKEN="YourToken" `
    -e OR_INSTANCE_NAME="YourInstance" `
    -v opc-router-5-data:/data `
    -v opc-router-5-logs:/var/log/opcrouter `
    -p 8080:8080 `
    -p 8001:8001 `
    -p 49420:49420 `
    opcrouter/runtime:latest
```

Docker Compose

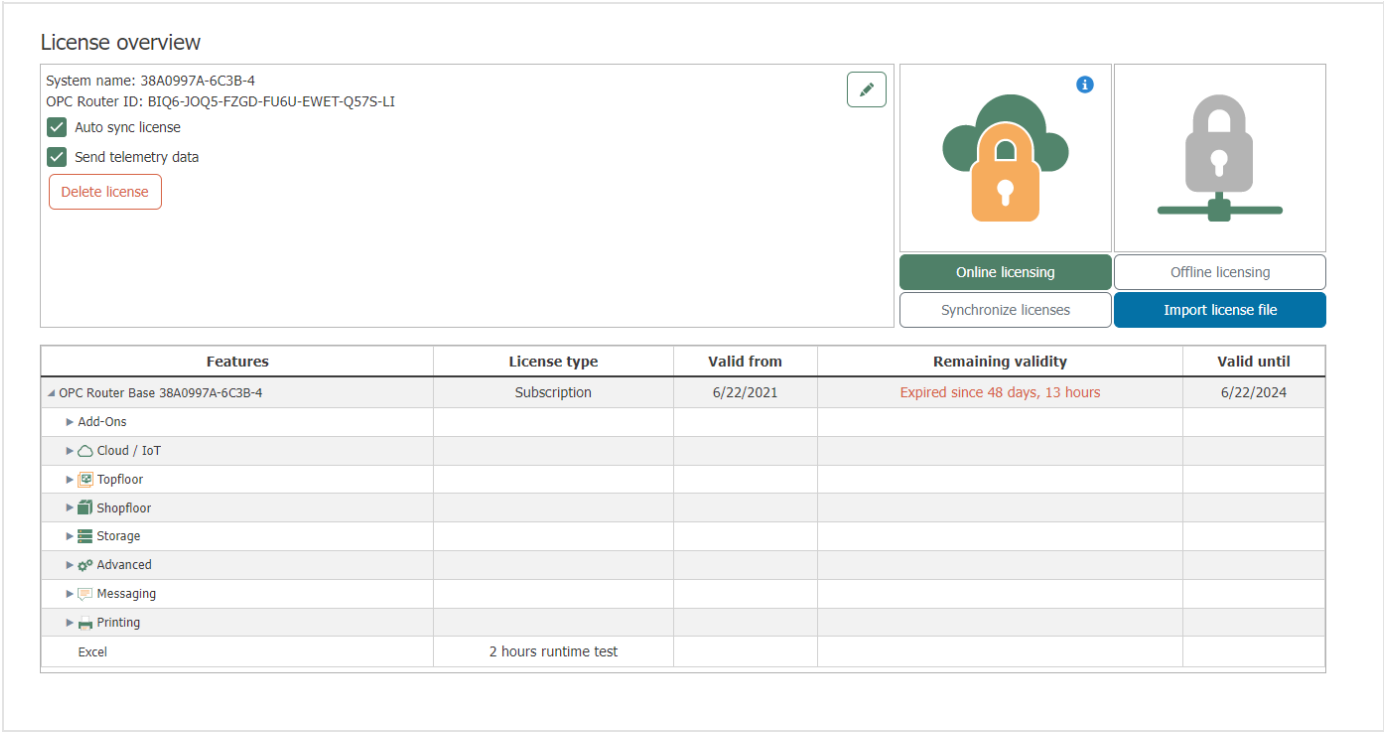
```yaml title="docker-compose.yaml" services: opc_router: image: opcrouter/runtime:latest ports:

- "8080:8080"
- "8001:8001"
- "49420:49420" environment:
- INITIAL_USERNAME=*****
- INITIAL_PASSWORD=*****
- OR_I_ACCEPT_EULA=true
- OR_LICENSE_KEY=YourLicense
```

- OR_LICENSE_OPCROUTERID=YourRouterID
- OR_LICENSE_DEPLOYMENTTOKEN=YourToken
- OR_INSTANCE_NAME=YourInstance volumes:
- opc-router-5-data:/data
- opc-router-5-logs:/var/log/opcrouter volumes: opc-router-5-data: opc-router-5-logs: ```

License overview in the OPC Router

The license overview can be found in the OPC Router menu under "License". In addition to the license overview, you will also find the option of licensing via "Online licensing" and "Offline licensing".



COLUMN	EXPLANATION
Features	All the features offered by the OPC Router are displayed here. The add-ons are also displayed here.
License type	The type of license is displayed here. **Possible types of licenses:** • 2 hours runtime test, if no license is available • Subscription • Perpetual • Educational • Demo If the support and update entitlement has expired, this is also displayed for the license type. It is also displayed if the license for the version of the router used is invalid.
Valid from	Date from which the license is valid.
Remaining validity	Displays the remaining running time in days and hours. If the remaining running time is low, the remaining time is displayed in red.
Valid until	Shows until when the license is valid. If the license has expired, this shows how long it has been expired.

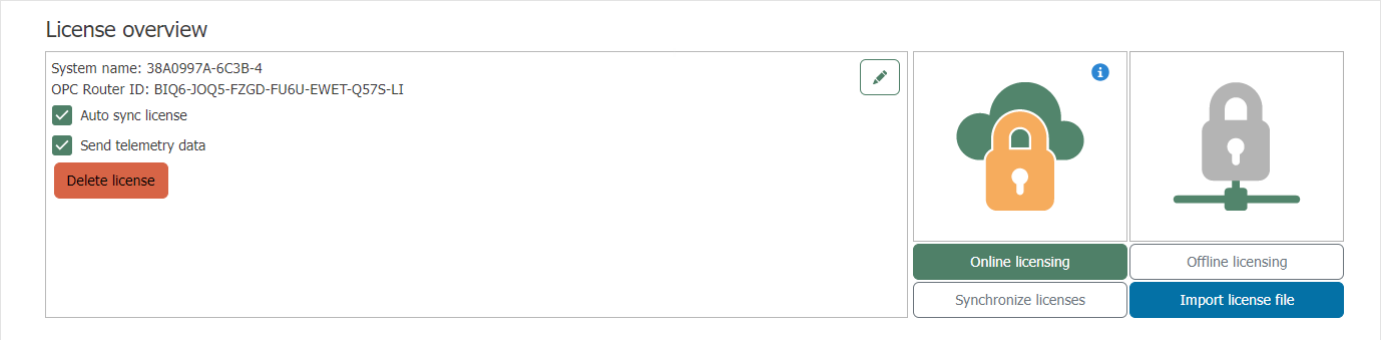
Delicensing

The delicensing allows you to decouple the system from the current OPC Router license in order to use it on another system.

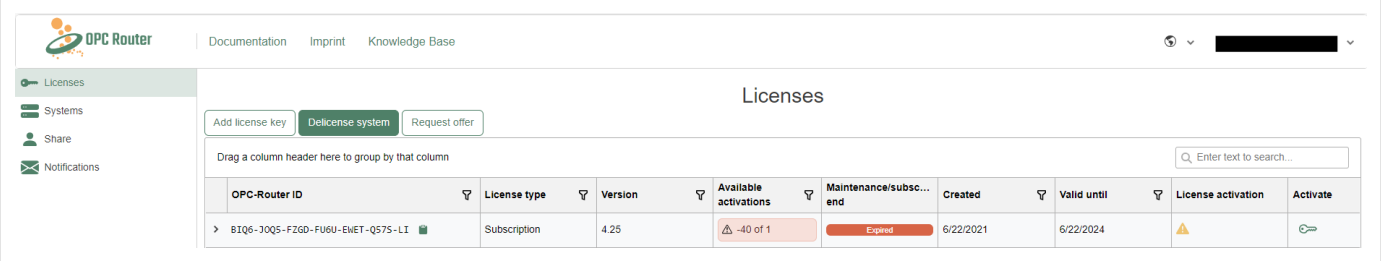
The delicensing of a system cannot be reversed! However, it can be licensed again.

To perform a delicensing, please follow the steps below:

1. Open the license overview in your OPC Router.
2. Now click on the "Delete license" button.



1. Confirm the process.
2. A license file will now be downloaded automatically. Please save it securely.
3. Now go to the license portal website and log in.
4. Click the "Delicense system" button in the license overview.



1. Upload the delicensing file and confirm with the "Delicense system" button.

After successful delicensing, you can see in the license overview that one more activation is available.

Security Guidelines

OPC Router is being installed via a simple setup program. The GUI makes configuration and development (of Router projects) simple as can be. Nonetheless it is very important to read and follow these guidelines carefully to avoid data loss and resulting damages and ensure data and system integrity.

Important: Please consult your system administrator before installing or updating and follow your company's own policy or code of conduct concerning software. We strongly recommend that access to the computer with OPC Router installed is limited and for authorized staff only. Keep in mind that access data to connected systems have to be stored on this computer.

Knowledge assumed

Installation, configurations and administration (as well as understanding the manual) assumes the following knowledge: Administration of Windows operating systems, IT security, knowledge about 3rd party systems connected (e. g. database administration, OPC Servers, MES, ERP systems used in this context).

Installation and Update

1. We recommend to backup your production system before installing or updating.
2. OPC Router Setup might update or install Windows components. Please exit all running programs before installing.
3. We strongly recommend to ensure that existing configuration is compatible with any update before updating your production system (e. g. update test within quality control environment)
4. If you change the installation path proposed by the setup routine, please remember your settings to simplify any support by inray.

Development

Router projects can be developed within your production system. However, we recommend to backup your system before installing new plug-ins or activate new transfers. If you are testing new transfers within your production system, please remember to backup your database and ensure data consistency. We recommend to test all Router projects thoroughly before going productive.

DANGER

Testing your Router projects within production environment might, dependent on your transfer configuration, result in changing your production facilities' configuration (via programmable logic controllers, PLCs) and result in loss of material or severe damage of machinery. Please be sure that only authorized and qualified staff activate such projects. inray takes no liability for such self-inflicted damages (see below).

Limitations of Liability and Warranty

The OPC Router and its manual are provided by inray Industriesoftware GmbH “as is” and any express or implied warranties, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose and non-infringement, are disclaimed. In no event shall inray or their suppliers be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, savings or profits; or business interruption) however caused and on any theory of liability whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

inray Industriesoftware advises you to double check Router projects carefully before going productive. inray does not guarantee that Router projects developed according to this manual are compatible with future versions. Limitations of liability and warranty included in the end user license agreement EULA and/or other agreements between inray and the licensee remain unaffected.

Errors Excepted

This manual is adapted to the Router's development. However, inray Industriesoftware does not guarantee that the information in

this manual is always consistent with the current version of OPC Router. The manual is therefore subject to errors, especially where 3rd party systems are concerned (Windows components, OPC configuration). Please check manuals of 3rd party systems regularly.

Architecture recommendations

In the age of Industry 4.0 and the associated exponential increase in data volumes, the scalability of IT systems is one of the biggest challenges. The OPC Router is the central platform for enabling and optimizing communication between different industrial systems. This chapter of the manual is dedicated to the architecture recommendations for using the OPC Router to operate both small and large and complex systems efficiently and reliably.

Objective and scope

This chapter aims to provide clear and structured guidance on how to implement and scale the OPC Router in different sized organizations. It is intended to help both IT administrators and business decision makers to make informed decisions about the architecture of their systems.

Relevance for different company sizes

- **Small businesses**: For small businesses, it is crucial to implement cost-effective yet powerful solutions that can grow with the company. The OPC Router offers a flexible platform that is easy to use and expand.
- **Medium-sized companies**: These companies often face the challenge of making their systems scalable and future-proof in order to be able to respond to growing data volumes and more complex production processes. The OPC Router enables seamless integration and scaling.
- **Large companies and corporations**: In large companies, the reliability and high availability of systems is of the utmost importance. The OPC Router offers robust solutions for load distribution, redundancy and performance optimization.

With these architecture recommendations, we want to help you find and implement the best possible solution for your specific requirements. Whether you are just starting with small use cases or already operating complex enterprise solutions, this guide offers valuable insights and practical advice for companies of all sizes.

Basic Concepts

This chapter explains the basic concepts that are required to understand and implement the architecture recommendations for the OPC Router. It provides an introduction to the key topics that are discussed in detail later in the manual.

The OPC Router as a central platform

The OPC Router is a central platform that enables communication between different industrial systems and devices. Its ability to seamlessly transfer and process data makes it a crucial part of the modern Industry 4.0 landscape. The platform supports various protocols and standards, ensuring a high level of flexibility and interoperability.

Key features of the OPC Router

- **Protocol support**: Support for OPC UA, MQTT, REST, databases, SAP and many other protocols.
- **Data integration**: Integration of data from various sources such as PLCs, databases, IoT devices and cloud services.
- **Visualization and monitoring**: Real-time monitoring and visualization of data streams and connections.
- **Automation**: Automation of processes through rule- and event-based workflows.

Fundamentals of Scalability: Vertical and Horizontal

In the context of operating the OPC Router, both vertical and horizontal scaling are crucial to meeting the requirements of different company sizes. Here is a more detailed look at both scaling approaches in the specific context of OPC Router operation.

Vertical scaling

Definition: Vertical scaling (scale-up) refers to increasing the capacity of a single server or instance by adding more resources such as CPU, RAM or storage.

Use cases: Vertical scaling is particularly suitable for smaller installations or scenarios in which higher performance per instance is required. This could be the case if a single OPC Router processes a large number of data points or performs complex data processing tasks.

Advantages of vertical scaling

- **Easy to implement**: Does not require any changes to the existing application architecture.
- **Low complexity**: Since only one instance needs to be managed, administration and configuration are easier.
- **Immediate performance improvement**: Adding resources can immediately improve performance.

Challenges of vertical scaling

- **Limited scalability**: The capacity of a single server has physical limits.
- **Single point of failure**: If the only instance fails, this can lead to downtime.

Example: A medium-sized company operates a production hall with an OPC Router that collects data from all machines. By adding more RAM and CPU capacity, the processing speed and the amount of data that can be processed can be increased without the need for an additional instance.

Horizontal scaling

Definition: Scale-out means adding additional servers or instances to distribute the load and increase capacity.

Use cases: Ideal for larger installations or scenarios where high availability and load balancing are required. This is particularly useful in large production environments or distributed locations.

Advantages of horizontal scaling

- **Virtually unlimited scalability**: Capacity can be expanded almost indefinitely by adding more instances.
- **Improved fault tolerance**: Redundancy allows individual instances to fail without affecting the overall system.
- **Flexibility**: New instances can be easily added or removed to respond to changing requirements.

Challenges of horizontal scaling

- **Increased administrative effort**: Multiple instances must be managed and monitored.

Example: A large company operates several production halls and locations. By setting up multiple OPC Router instances for each production hall or location, data processing can be performed locally, thus distributing the load and increasing system reliability. If necessary, additional instances can be added to expand capacity.

Practical implementation

Vertical scaling

1. **Resource analysis**: Determine current resource requirements and identify bottlenecks.
2. **Upgrade**: Add additional CPU, RAM or storage to the existing OPC Router instance.
3. **Monitoring**: Monitor performance after the upgrade to ensure that the expected improvements are realized.

Horizontal scaling

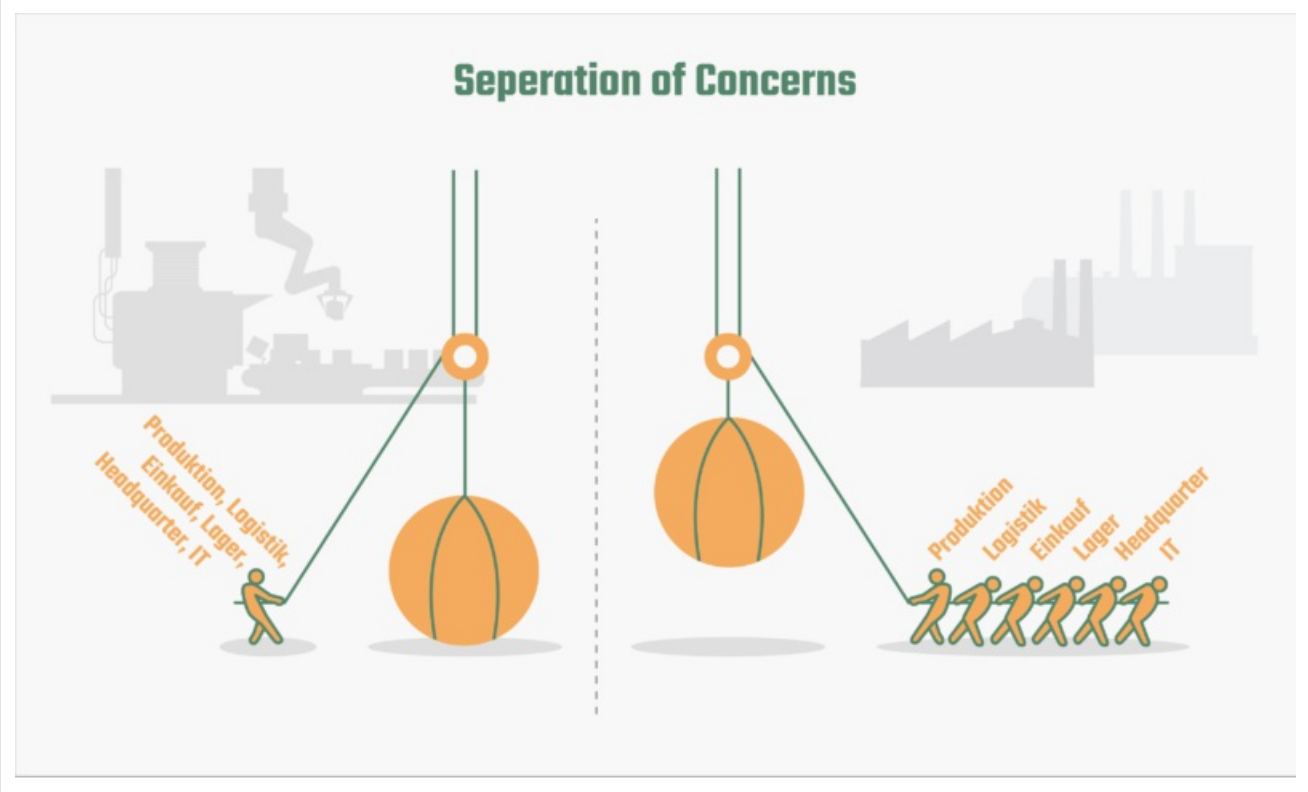
1. **Needs analysis**: Analyze the load and identify areas that would benefit from an additional instance.
2. **Instance deployment**: Set up additional OPC Router instances, configure them and connect them to the existing network.
3. **Load balancing**: Export the project components from the existing OPC Router instance that are to be transferred to the additional instance.
4. **Commissioning**: Import the exported project components into the new instance, deactivate the connections in the original instance. Switch the new instance to productive mode and verify that the new instance has been successfully integrated.
5. **Monitoring and maintenance**: Monitor the performance of all instances and perform regular maintenance.

Separation of Concerns in the OPC Router

Definition and significance of SoC

Separation of Concerns (SoC) is an architectural principle that aims to clearly separate different responsibilities and tasks within a system. This leads to a clearer structure and more efficient operation of the systems.

In the context of the OPC Router, this means that the various “concerns” or responsibilities within an OPC Router instance are grouped and separated so that there are no dependencies. This makes extensions and scaling much easier.



How SoC is applied in the OPC Router

Modularization of the “concerns”

- **Individual instances for different responsibilities**: In an OPC Router architecture, different responsibilities, such as the connection of different production areas or plants, should be combined in separate instances. This means that each instance of the OPC Router is responsible for a specific task or area.
- **Example**: One instance could be responsible for communicating with the machines in hall A, while another instance processes the data from hall B.

Benefits of separating concerns

- **Easier extensions**: When a new plant, hall or location is introduced, another OPC Router instance with the same or similar configuration can simply be added. This reduces the effort and complexity of extending the system.
- **Scalability**: The clear separation of responsibilities means that the individual instances can be scaled independently of each other. If requirements in one area increase, only the corresponding instance needs to be expanded.
- **Independent updates**: Security, software and configuration updates can be carried out more easily and in a more targeted manner. Since the instances are separate from each other, updates can be carried out without affecting multiple production departments or processes at the same time.
- **Isolation of errors**: Problems in one instance (e.g. hardware errors or software crashes) do not affect the other instances. This increases the overall stability and reliability of the system.

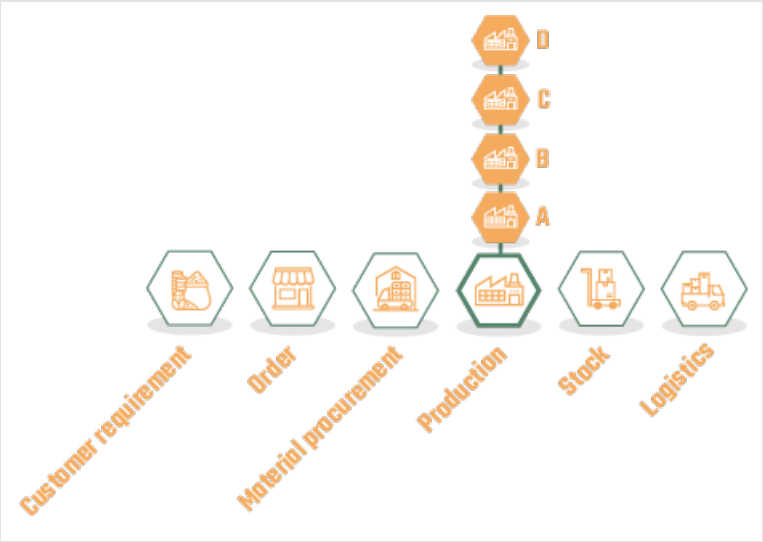
Practical implementation

Setting up separate instances

- **Planning**: Start by planning which areas or tasks should be separated. Identify the main responsibilities and distribute them to separate instances.
- **Implementation**: Set up the instances of the OPC Router according to the planned distribution. Make sure that each instance has the necessary connections and configurations to fulfill its specific tasks.
- **Maintenance and monitoring**: Monitor the instances regularly and perform any necessary maintenance work. Use the separation of the instances to make targeted updates and improvements.

Example of an architecture

- **Instance 1**: Responsible for machine control and data collection in hall A.
- **Instance 2**: Responsible for quality control and inventory management in hall B.
- **Instance 3**: Responsible for energy monitoring and environmental controls throughout the plant.



Summary

Understanding the basic concepts of vertical and horizontal scaling, as well as the principle of separation of concerns, is crucial for the successful implementation and use of the OPC Router in different company sizes. While vertical scaling offers a simple solution for smaller use cases, horizontal scaling enables almost unlimited expansion and higher reliability. Separation of concerns helps to reduce complexity and increase system efficiency. The following chapters will explore these concepts in more detail and present

practical examples of their application.

Architecture for small use cases

This chapter describes the architecture recommendations for using the OPC Router in small use cases. These recommendations are specifically intended for companies that operate a manageable number of machines or devices and require a cost-effective solution for data integration and processing. It also discusses how small use cases can serve as a basis for future growth and expansion.

The OPC Router is the ideal platform for digitization and offers companies the opportunity to start small and grow step by step. With the OPC Router, you can start with a small use case and expand the solution as needed. This supports a sustainable digitization strategy in which the platform grows with the company's ideas and requirements.

Definition and characteristics of small use cases

Small use cases are characterized by a limited number of machines or devices that need to communicate with each other. Typically, these are individual production lines, smaller workshops or locations with a low data volume.

Typical characteristics of small use cases

- A small number of machines or devices
- Limited data volume
- Simple network infrastructure
- Basic requirements for data integration and processing

Typical examples of small use cases

Example 1: Small production line A small production line consisting of several machines that need to communicate with each other to optimize the production process.

Example 2: Automated warehouse management A small warehouse in which various automated systems (such as conveyor belts, robots and sensors) are networked with each other to make warehouse operations more efficient.

Minimum hardware and software requirements

Hardware requirements

- Standard server or high-performance PC
- Sufficient CPU capacity (at least 4 cores)
- At least 8 GB RAM
- SSD with sufficient capacity for data processing and storage

Software requirements

- Operating system: Windows or Linux
- OPC Router software with the required plug-ins for communication with the existing machines and devices
- Network infrastructure: LAN or WLAN with sufficient bandwidth and stability

Virtualized environments

- The OPC Router can also be operated in virtualized environments such as VMWare or HyperV.
- Advantages of virtualization:
    - **Flexibility**: Easy scalability of resources as needed.
    - **Efficiency**: Better resource utilization by consolidating multiple virtual machines on a single physical server.
    - **Backup and restore**: Easy creation of snapshots and backups of the entire virtual machine.
- Requirements for virtualization:
    - Ensure that the host server provides sufficient resources (CPU, RAM, storage) for the virtual machines to operate.
    - Configure the network connection of the virtual machines carefully to ensure optimal performance and security.

Containerized environments

- The OPC Router can also be operated in containerized environments such as Docker.
- Advantages of containerization:
  - **Portability**: Containers can be easily moved between different environments without the need for customization.
  - **Isolation**: Containers provide an isolated environment for applications, resulting in improved security and stability.
  - **Efficiency**: Containers require fewer resources than full virtual machines and can be started and stopped more quickly.
  - **Easy updates**: Container updates are easy to perform by updating the image tag and restarting the container.
- Requirements for containerization:
  - Ensure that the host system supports Docker and provides sufficient resources for the containers to run. Use Docker Compose to orchestrate and manage complex multi-container applications.

Example architecture: Small production site

Network setup

- **Central OPC Router instance**: A single instance of the OPC Router is installed on a server or high-performance PC. This instance manages the communication between the machines and devices in the production line.
- **Machine and device data**: The machines and devices are connected to the OPC Router instance via a local area network (LAN). The data is collected and processed in real time.
- **Data integration**: The OPC Router integrates data from various sources, such as PLCs (programmable logic controllers), sensors and other devices. The data can be transferred to a central database or an MES (Manufacturing Execution System).

Example of a configuration

- **Server**: A standard server with Windows Server 2019, 16 GB RAM and 500 GB SSD.
- **OPC Router instance**: Installation of the OPC Router on the server, configuration of the required plug-ins (e.g. OPC UA, MQTT).
- **Network**: Local area network (LAN) with gigabit Ethernet connections to the machines and devices.
- **Data integration**: Configuration of data connections to the machines, real-time data processing and forwarding of data to a central MES.

Redundancy: Nice to Have

Redundancy can be a useful feature for small use cases as well, increasing the reliability and resilience of the systems. Although redundancy is often used in larger installations, it can also be implemented sensibly in smaller scenarios.

Options for redundancy

- **Local redundancy**: Implementation of a second OPC Router server or a second instance that takes over in the event of a failure of the main instance.
- **Database redundancy**: Use redundant database servers or configure a locally redundant database to avoid data loss.
- **Network redundancy**: Use redundant network connections to ensure connectivity in the event of a network failure.

Example of a simple redundancy configuration

- **Second server**: An additional server that is configured identically to the main server.
- **Failover mechanism**: Setting up a failover mechanism that automatically switches to the second server if the main server fails.
- **Synchronization**: Regular synchronization of data and configurations between the two servers.

Advantages of redundancy

- **Increased reliability**: Minimizes downtime and ensures that data processing continues even in the event of a system error.
- **Fast recovery**: Automatic switchover to the redundant system ensures minimal interruptions to operations.

Security aspects

Security is an important aspect even in small use cases. The following measures should be considered:

- **Network security**: Implement firewalls and network segmentation to prevent unauthorized access.
- **Data encryption**: If possible, data transmission between the machines and the OPC Router instance should be encrypted. However, older systems used in industry often do not support this. In such cases, it is recommended to implement physical separation or segmentation via VLANs. The OPC Router system can be present in multiple networks so that the web management can be accessed from the “outside”. HTTPS should always be configured for OPC Router management and secure passwords should be used.

use-https.md

use-https.md }

- **User access control**: Setting up user accounts and roles to control access to the OPC Router instance and the data. Each user should have their own user, or there should be a connection to e.g. Azure EntraID / Active Directory.

connect-microsoft-entra-id-azure-ad.md

user-management

Backup and updates

Regular backups

- **Automated backups**: Implement automated backups of the OPC Router configuration and databases. This can be done using regular scripts or backup software.
- **External storage locations**: Storing backups in external storage locations, e.g. in the cloud or on external hard disks, to avoid data loss in the event of hardware failure.
- **Recovery testing**: Regularly test backup recovery to ensure that data can be quickly and completely restored in the event of an emergency.
- **Simple backups in virtualized environments**: Virtualized systems offer the option of creating snapshots and backups of the entire virtual machine, which simplifies backup and recovery.

Regular updates

- **Software updates**: Keep the OPC Router and all plug-ins up to date. Install software updates and patches regularly to close security gaps and improve performance.
- **Operating system updates**: Ensure that the server's operating system is regularly updated to close security gaps and ensure stability.
- **Test environment**: First, run updates in a test environment to identify possible compatibility issues before implementing the updates in the production environment.
- **Maintenance windows**: Schedule regular maintenance windows for updates and backups to minimize disruption to operations.
- **Easy updates in container environments**: In Docker environments, updates can be performed by simply updating the image tag and restarting the container, which greatly simplifies maintenance.

Summary

For small use cases, the OPC Router offers a flexible and cost-effective solution for data integration and processing. By implementing a single instance on a standard server and using a local network, companies can ensure efficient and reliable communication between their machines and devices. Taking security aspects into account also ensures that data and systems are protected. Redundancy is a useful feature that, although not mandatory, can further increase reliability and resilience. In addition, the OPC Router offers a scalable platform that grows with the company and supports future expansions.

Outlook: Companies that decide to use the OPC Router in small use cases have the option of scaling up to medium and enterprise solutions in stages. For more information and inspiration, see the chapters on **architecture for medium-sized solutions** and **architecture for enterprise solutions** to further develop and expand your digitization strategy.

Edge device architecture

This chapter describes the architecture recommendations for using the OPC Router on edge devices. These recommendations are intended specifically for companies that want to use a distributed architecture in which data processing and integration take place directly at the edge of the network.

Definition and characteristics of edge devices

Edge devices are specialized hardware devices that are used at the periphery of a network to process and store data locally. This reduces latency and bandwidth usage by only forwarding relevant data to central systems.

Typical characteristics

- Local data processing and integration
- Low latency
- Reduced bandwidth utilization
- High availability and reliability
- Easy scalability

Typical application examples

Example 1: Production monitoring An edge device used in a production plant to collect and process data from machines and sensors in real time.

Example 2: Condition monitoring An edge device that collects data from various sensors and devices in a warehouse or plant to monitor the condition of the equipment and predict maintenance needs.

Minimum hardware and software requirements

Hardware requirements

- Edge device or powerful embedded PC
- Sufficient CPU capacity (at least 4 cores)
- At least 4 GB RAM
- SSD with sufficient capacity for data processing and storage

Software requirements

- Operating system: Linux or Windows IoT
- OPC Router software with the required plug-ins for communication with the existing machines and devices
- Network infrastructure: LAN or WLAN with sufficient bandwidth and stability

Implementation on edge devices

Docker container: The OPC Router can be run in a Docker container on the edge device. This provides portability and isolation of the application, as well as easy management and updates.

Installation and configuration

- **Docker installation**: Make sure that Docker is installed on the edge device.
- **Docker Compose**: Use Docker Compose to configure and orchestrate the OPC Router and the required plug-ins as containers.
- **Configuration**: Customize the configuration of the OPC Router and the plug-ins to the specific requirements of the environment.

Example configuration

- **Docker Compose File**:

```
version: '3'
services:
  opc-router:
    image: opc-router:<Major>.<Minor>
    container_name: opc-router
    ports:
      - “8080:8080”
    volumes:
      - ./config:/config
      - ./data:/data
```

Management of Edge Devices

Portainer

Use Portainer to simplify the management of Docker containers on the edge devices. Portainer provides a user-friendly web interface for managing and monitoring containers.

Deployment templates

Create and use deployment templates to quickly and consistently deploy the OPC Router to multiple edge devices.

Documentation and Resources:

OPC Router Portainer App Template deployment

OPC Router Portainer Edge Template deployment

Configurable environment variables

Integration with industrial platforms

Siemens Industrial Edge:

Use Siemens Industrial Edge to integrate OPC Router into an industrial edge platform designed specifically for use in production environments.

Azure IoT Edge:

Use Azure IoT Edge to integrate OPC Router into the Azure cloud and benefit from the extensive IoT services and capabilities of the Azure platform.

Documentation and Resources:

Siemens Industrial Edge

Security aspects

Security is an important aspect even in small use cases. The following measures should be taken into account:

- **Network security**: Implement firewalls and network segmentation to prevent unauthorized access.
- **Data encryption**: If possible, data transmission between the machines and the OPC Router instance should be encrypted. However, older systems used in industry often do not support this. In such cases, it is recommended to implement physical separation or segmentation via VLANs. The OPC Router system can be present in multiple networks so that the web management can be accessed from the “outside”. HTTPS should always be configured for OPC Router management and secure passwords should be used.

Use HTTPS (Docker

Use HTTPS (Windows

- **User access control**: Setting up user accounts and roles to control access to the OPC Router instance and the data. Each user should have their own user, or there should be a connection to e.g. Azure EntraID / Active Directory.

Connect Microsoft Entra ID/AzureAD

User management

Backup and updates

Regular backups:

- **Automated backups**: Implementing automated backups of the OPC Router configuration and databases on the edge devices.
- **External storage locations**: Storing backups in external storage locations, e.g. in the cloud or on external hard disks.

Regular updates:

- **Software updates**: Keep the OPC Router and all plug-ins used on the edge devices up to date.
- **Easy updates in container environments**: In Docker environments, updates can be performed by simply updating the image tag and restarting the container.

Summary

Using edge devices with the OPC Router provides a flexible and efficient solution for local data processing and integration. By using Docker containers and management tools such as Portainer, companies can easily scale and manage their edge architecture. Taking into account security aspects and regular backups and updates also ensures that data and systems are protected and that the entire system continues to perform and remain secure.

Scaling to medium-sized solutions

This chapter describes the architecture recommendations for scaling the OPC Router from small to medium-sized solutions. These recommendations are specifically intended for companies that operate a growing number of machines or devices, pursue a more comprehensive digitization strategy, and require an advanced solution for data integration and processing.

Transition from small to medium-sized solutions

The OPC Router offers companies a flexible and scalable platform that enables them to start with small use cases and expand them step by step. This approach supports a sustainable digitization strategy in which more and more machines and devices are integrated step by step. The transition from small to medium-sized solutions requires careful planning and adaptation of the existing infrastructure.

Important steps:

- Evaluating current and future requirements
- Planning the scaling of hardware and software
- Implementing a scalable network architecture
- Ensuring data security and system reliability
- Recognizing the right time to transition from small to medium-sized solutions

Advanced hardware and software requirements

Hardware requirements:

- More powerful servers or multiple servers
- Increased CPU capacity (at least 8 cores)
- At least 16 GB RAM
- Extended storage solution, e.g. RAID arrays or SAN

Software requirements:

- Operating system: Windows Server or Linux
- OPC Router software with advanced plug-ins and features
- Network infrastructure: Gigabit Ethernet or higher

Virtualized and containerized environments:

- **Virtualized environments**: Use virtualization solutions such as VMWare or HyperV to run multiple virtual machines on powerful servers. **Containerized environments**: Use Docker and Docker Compose to run the OPC Router in a scalable and portable environment.

Example architecture: Medium-sized production site

Network structure:

- **Central OPC Router instance**: Multiple instances of the OPC Router are run on powerful servers or in virtual machines. These instances manage the communication between the machines and devices in the production facility. **Separation of Concerns (SoC)**: Workflows are distributed across individual OPC Router instances to efficiently manage the load and improve scalability.
- **Redundancy for failover**: Implement redundancy concepts to ensure failover.
- **Machine and device data**: The machines and devices are connected to the OPC Router instances via a robust network (LAN). The data is collected and processed in real time.
- **Data integration**: The OPC Router integrates data from various sources, such as PLCs, sensors and other devices, and transfers it to central databases or MES systems.

Example of a configuration:

- **Server**: Two powerful servers with Windows Server 2019, each with 32 GB RAM and 1 TB SSD in a RAID array.

- **OPC Router instances**: Installation of two OPC Router instances, configuration of the required plug-ins (e.g. OPC UA, MQTT, REST).
- **Network**: Robust network with gigabit Ethernet connections and redundant network structure.
- **Data integration**: Configuration of data connections to the machines, real-time data processing and forwarding of data to central MES systems and databases.

Security aspects

Security is an important aspect even in small use cases. The following measures should be taken into account:

- **Network security**: Implement firewalls and network segmentation to prevent unauthorized access.
- **Data encryption**: If possible, data transmission between the machines and the OPC Router instance should be encrypted. However, older systems used in industry often do not support this. In such cases, it is recommended to implement physical separation or segmentation via VLANs. The OPC Router system can be present in multiple networks so that web management can be accessed from the outside. HTTPS should always be configured for OPC Router management and secure passwords should be used.

Use HTTPS (Docker)

Use HTTPS (Windows)

- **User access control**: Setting up user accounts and roles to control access to the OPC Router instance and the data. Each user should have their own user, or there should be a connection to e.g. Azure EntraID / Active Directory.

Connect Microsoft Entra ID/Azure AD

User management

Backup and updates

Regular backups:

- **Automated backups**: Implementing automated backups of the OPC Router configuration and databases.
- **External storage locations**: Storing backups in external storage locations, e.g. in the cloud or on external hard disks.
- **Simple backups in virtualized environments**: Virtualized systems offer the option of creating snapshots and backups of the entire virtual machine, which simplifies backup and recovery.

Regular updates:

- **Software updates**: Keep the OPC Router and all plug-ins up to date.
- **Operating system updates**: Make sure that the operating system of the server is regularly updated.
- **Easy updates in container environments**: In Docker environments, updates can be performed by simply updating the image tag and restarting the container.

Introduction of “Separation of Concerns” (SoC)

As complexity increases, the introduction of SoC becomes increasingly important. SoC means that different tasks and responsibilities are split into separate components to improve the maintainability and extensibility of the system. In the context of the OPC Router, this can mean that different instances or containers are used for different tasks such as data acquisition, processing and forwarding.

See also: Separation of Concerns in the OPC Router

Advantages of SoC:

- **Easier extensions**: New functions can be added without affecting the entire system.
- **Easy scaling**: Each component can be scaled independently.
- **Improved maintainability**: Problems can be isolated and resolved more easily.

Horizontal scaling and availability

Horizontal scaling:

- Add additional OPC Router instances to distribute the load across multiple servers or containers. Workflows are manually distributed to each instance to manage the load.
- Use separation of concerns to ensure that different tasks are distributed to different instances.

Availability:

- Implement high availability solutions to ensure that the OPC Router remains available even if individual components fail.
- Use redundancy concepts to ensure fail-safe operation.

Test and development environment

As complexity and scale increase, a separate test and development environment becomes more and more important. This environment allows changes and updates to be tested before implementation in the production environment, minimizing risks and ensuring stability.

Recommendations:

- Set up a separate environment that replicates the production environment as closely as possible.
- Run regular tests to ensure the stability and compatibility of new updates and changes.
- Use automated testing tools to increase the efficiency and accuracy of testing.

Summary

For medium-sized solutions, the OPC Router offers a scalable and robust solution for data integration and processing. By implementing multiple instances on powerful servers or in virtualized/container environments and using a robust network, companies can ensure efficient and reliable communication between their machines and devices. The introduction of separation of concerns, the consideration of security aspects, and regular backups and updates also ensure the protection of data and systems, as well as the continuous performance and security of the entire system.

Architecture for Enterprise Solutions

This chapter describes the architecture recommendations for using the OPC Router in large enterprise environments. These recommendations are aimed at companies that operate complex and extensive systems and require a highly available, secure and scalable solution for data integration and processing. The OPC Router is the ideal platform for digitization and offers companies the opportunity to design their solutions flexibly and scalably. Whether the requirements are increasing due to company growth, new ideas or changing regulations and customer requirements, the OPC Router can quickly and flexibly adapt to the new circumstances through configuration.

Definition and characteristics of large enterprise solutions

Enterprise solutions are characterized by their extensive infrastructure, high data volumes and complex availability and security requirements. The OPC Router offers the flexibility and scalability to meet these requirements.

Typical characteristics of large enterprise solutions

- High availability and redundancy
- Comprehensive security measures Integration into existing IT infrastructures
- Scalability and flexibility
- Mandatory test and development environments

High-availability and redundant architecture

High availability and redundancy are critical factors for operating OPC Router instances in enterprise environments. These concepts help to minimize downtime and ensure the reliability of data communication. There are various architectural options for achieving these goals:

1. **Centralized architecture with cross-redundancy**
   - **Definition**: Operation of two or more central OPC Router instances in parallel, with the connected systems also designed to be redundant.
   - **Advantages**: High availability, fast switchover times through “hot standby”, dynamic compensation for failures, centralized administration. **Example**: Redundant connections to OPC UA servers, databases and SAP systems.

2. **Decentralized architecture**
   - **Definition**: Operation of OPC Router instances close to the data source in so-called “fault zones”.
   - **Advantages**: reduced latency, higher fault tolerance, flexibility and scalability.
   - **Implementation**: Local redundancy with edge devices, use of store and forward, local data brokers or databases.

3. **Hybrid architecture**
   - **Definition**: Combination of centralized and decentralized architecture to utilize the advantages of both models.
   - **Advantages**: Maximized availability and fault tolerance, optimized latency.
   - **Implementation**: Centralized servers with cross-redundancy and decentralized edge devices with local redundancy and synchronization.

4. **Kubernetes and container orchestration**
   - **Definition**: Kubernetes is an open-source platform for automating the deployment, scaling and management of containerized applications.
   - **Advantages**: scalability, flexibility, automation.
   - **Challenges**: complexity, resource requirements.
   - **Implementation**: The OPC Router is installed on a Kubernetes cluster using Helm Charts, supported by various configurations and redundancy options for OPC Router and MongoDB instances.

For detailed implementation strategies, practical implementations and specific configuration examples, please refer to the **High-availability and redundant systems** subpage.

Virtualized environments

The use of virtualized environments offers numerous advantages for the operation of the OPC Router in enterprise environments. Virtualization enables resources to be used efficiently, the infrastructure to be scaled flexibly and additional security measures to be implemented.

Operation in virtualized environments

- **Flexibility and scalability**: Virtualized environments offer the flexibility to dynamically allocate resources and easily scale infrastructure.
- **Security measures**: Provide additional security through isolation and easy recoverability.
- **Cost efficiency**: Reduction of hardware costs by consolidating multiple virtual machines on a single physical server.
- **Recoverability**: Easy creation of snapshots and backups of the entire virtual machine.

Recommended platforms

- **VMWare**: Widely used and offers extensive management and security features.
- **HyperV**: Well integrated into Microsoft environments and offers high availability and scalability.

Recommendation for using hypervisors: Regardless of the architecture you use – centralized, decentralized, hybrid or Kubernetes – we recommend using a hypervisor. Hypervisors such as VMWare ESXi or Microsoft HyperV provide a robust and flexible foundation for operating virtualized environments. They enable efficient resource utilization and offer numerous features for managing and securing your virtual machines.

Separation of Concerns (SoC)

The introduction of Separation of Concerns (SoC) is a fundamental and obvious part of the architecture for enterprise solutions. SoC means that different tasks and responsibilities are distributed to separate components to improve the maintainability and expandability of the system.

Implementation of SoC

- **Separation of tasks**: Different tasks such as data acquisition, processing and forwarding are distributed to separate OPC Router instances.
- **Reduced dependencies**: Each instance is configured independently, which makes it easier to expand and scale.
- **Easy maintenance**: Problems can be isolated and resolved more easily, which simplifies maintenance.
- **Flexibility and adaptability**: By separating the tasks, the OPC Router can respond quickly and flexibly to new requirements and changes.

Test and development systems

Test and development systems are essential to ensure the stability and security of the OPC Router in an enterprise environment. These systems allow new features and configurations to be tested before they are deployed in the production environment, and help to identify and resolve potential problems at an early stage.

Importance of test and development systems

- **Risk minimization**: Testing changes in a safe environment allows potential risks and errors to be identified and corrected before they affect the production environment. **Quality assurance**: Ensuring that new features and configurations work as expected and do not have any unexpected side effects.
- **Training and education**: Providing an environment for training and educating IT staff without impacting production systems.

Recommendations for test and development systems

1. **Set up a separate environment**:
    - Set up a separate test and development environment that closely replicates the production environment. Use virtualization or containerization to create and manage these environments efficiently and cost-effectively.
2. **Run regular tests**:

    - Run regular tests to ensure the stability and compatibility of new updates and changes. Use automated testing tools to increase the efficiency and accuracy of testing.

3. **Project export and import**:
    - Use the export and import features of the OPC Router to transfer projects between development, test and production environments. This enables easy and consistent management of configurations and changes.

4. **Set up a development lab**:
    - Set up a development lab where PLCs and other hardware can simulate the machines.
    - Stage external systems such as ERP, MES and databases to create realistic test scenarios.

5. **Version control and deployment pipelines (optional)**:
    - Use version control systems such as Git to manage changes to configurations and projects.
    - Implement deployment pipelines to automate and standardize the transition from development to production environments.

6. **Security checks**:
    - Perform regular security audits and penetration tests in the test environment to ensure that new configurations and updates do not introduce security vulnerabilities.
    - Ensure that the test environment is as well protected as the production environment to ensure realistic results.

7. **Documentation and training**:
    - Document all tests, results and changes thoroughly to have a traceable history.
    - Use the test and development environment for training and exercises to prepare IT staff for new features and changes

Test and development systems are an essential part of a robust and secure IT infrastructure. They enable changes and updates to be tested under realistic conditions before they are transferred to the production environment. By setting up separate test environments, conducting regular tests and security checks, and using project export and import, development labs, and, ideally, version control systems and deployment pipelines, organizations can ensure the quality and security of their OPC Router implementations.

IT Monitoring and Integration

IT monitoring is an essential component for ensuring the performance, availability and security of OPC Router instances in enterprise environments. The OPC Router offers various functions for monitoring and integration with external monitoring tools.

Importance of monitoring

- **Monitoring**: Constant monitoring of system performance and availability.
- **Early warning systems**: Identification and resolution of problems before they lead to failures.

Internal diagnostic data

The OPC Router provides internal diagnostic data that can be transmitted to external systems and monitored via the internal OPC UA server. This enables seamless integration into existing monitoring solutions and provides detailed insights into the operating status of the system.

Self-monitoring with email notification

The OPC Router provides an internal self-monitoring function that automatically sends e-mail notifications when anomalies or errors are detected. This feature ensures that administrators and IT personnel are immediately informed of potential problems and can respond accordingly.

External monitoring tools

It is recommended to use established monitoring tools that are already known and used in the IT infrastructure. These tools offer extensive functions for monitoring and analyzing system performance and availability.

Integrate monitoring into the existing IT infrastructure and alerting systems to ensure comprehensive monitoring and rapid

response to potential problems.

Auditing and security

Auditing and security are essential aspects of operating the OPC Router in enterprise environments. These measures ensure that the system meets the highest standards and is protected against unauthorized access and potential threats.

Audit of changes

- **Tracking**: Logging all changes to systems and configurations.
- **Security review**: Regular review of audit logs to identify security-related events.

Security measures

- **Azure Active Directory / Entra ID**: Use centralized directory services for authentication and authorization.
- **SSL encryption**: Ensure that all connections are SSL-encrypted to ensure data integrity and confidentiality during transmission.
- **Separation of Concerns (SoC)**: Implement SoC to distribute different tasks to separate components and increase security and maintainability. SoC enables a clear separation of responsibilities and reduces the risk of security vulnerabilities affecting multiple system parts.

Regular updates and maintenance

- **Maintenance releases**: Inray regularly provides maintenance releases for the OPC Router versions and checks the dependencies for known vulnerabilities to make updates. **Security updates for included components**: Please also note that security updates for any included components such as the .NET Framework are installed.
- **Host and operating system updates**: Regular updates of the host and operating systems should be carried out to close security gaps and ensure stability.

Summary

For enterprise solutions, the OPC Router offers a highly available, secure and scalable solution for data integration and processing. By implementing cross-redundancy, versioning, Kubernetes and virtualized environments, as well as comprehensive security measures, companies can build an efficient and reliable infrastructure. Integration with existing IT monitoring systems and auditing of changes also ensure continuous monitoring and security of the entire system.

Highly available and redundant architecture

This section discusses the concepts and implementation strategies for highly available and redundant systems in the context of the OPC Router. High availability and redundancy are crucial to minimize downtime and ensure the reliability of data communication. There are various approaches to achieve these goals, including centralized and decentralized architectures, as well as combinations of both approaches.

Centralized architecture with high availability and redundancy

Cross-redundancy

- **Definition**: Cross-redundancy not only involves operating two or more central OPC Router instances in parallel, but also designing the connected systems redundantly. Each instance is able to take over the tasks of the others, using redundant connections to the data sources.
- **Implementation**:
  - **Centralized servers**: Two or more powerful servers in a centralized data center.
  - **Redundant connection**: Configuration of the OPC Router instances with redundant connections to OPC UA servers, databases, SAP systems, etc. For example, a connection is established to two redundant OPC UA servers to enable “hot standby”.
  - **Synchronization**: Regular data and configuration synchronization between the servers.
  - **Failover mechanism**: Automatic switchover to the redundant instance in the event of a server failure.

Advantages of cross-redundancy

- **High availability**: The architecture enables constant availability through parallel operation and fast recovery in the event of a fault.
- **Redundant connection**: The use of redundant connections to critical systems such as OPC UA servers, databases and SAP systems increases reliability.
- **Fast switchover times**: Hot standby allows the redundant instance to be switched over extremely quickly in the event of a failure, minimizing downtime.
- **Dynamic compensation**: Redundancies that are designed to be multiple can dynamically compensate for failing components in the chain.

Challenges of cross-redundancy

- Increased costs due to multiple central servers.
- Complexity in implementing and managing synchronization mechanisms.

Decentralized architecture

Close to the source in a fault zone

- **Definition**: In a decentralized architecture, OPC Router instances are operated as close as possible to the data source (e.g. production line, machine hall). Each instance operates in a defined “fault zone” so that local failures do not affect other zones.
- **Implementation**:
  - **Local servers or edge devices**: Use of local servers or edge devices in the vicinity of the data sources.
  - **Local redundancy**: Setting up redundancy at the local level, e.g. by using duplicate edge devices.
  - **Store and Forward**: Use of the Store and Forward add-on of the OPC Router to store data locally in case of connection interruptions and forward it later.
  - **Local broker/database**: Use of a local broker or database to buffer and synchronize data with central systems.

edge-device-architecture.md

Advantages of decentralized architecture

- Reduced latency times through local data processing.
- Higher fault tolerance through isolated fault zones.
- Flexibility and scalability.

Challenges of decentralized architecture

- Increased complexity due to the management of multiple decentralized instances.
- The need for robust synchronization mechanisms between local and central systems.

Combination of both architectures

Hybrid architecture

- **Definition**: A hybrid architecture combines centralized and decentralized approaches to take advantage of both models.
- **Implementation**:
  - **Centralized servers**: Setting up centralized servers with high availability and redundancy (e.g. cross-redundancy).
  - **Decentralized edge devices**: Use of edge devices close to the data source that are designed with local redundancy.
  - **Data synchronization**: Implementation of synchronization mechanisms between decentralized and centralized instances, e.g. through store and forward or local brokers/databases.

Advantages of the hybrid architecture

- Maximized availability and fault tolerance.
- Optimized latency times through local processing and central administration.

Challenges of hybrid architecture

- Complexity in the implementation and management of synchronized systems.
- Increased costs due to the combination of multiple approaches.

Kubernetes and container orchestration

Definition

- **Definition**: Kubernetes is an open-source platform for automating the deployment, scaling and management of containerized applications. By using Helm Charts, the OPC Router can be efficiently deployed and managed in a Kubernetes cluster.

Implementation

- **Deployment**: The OPC Router Helm Chart is used to install the OPC Router on a Kubernetes cluster. The Helm Chart supports various configurations, including authentication and redundancy.
- **Configuration**: Support for global parameters (e.g. Docker image registry), common parameters (e.g. service account settings) and specific OPC Router parameters (e.g. project repository, environment variables).
- **Redundancy**: Configuration of redundancy for both the OPC Router and MongoDB. For redundancy of the OPC Router, a second pod is provided that is activated in the event of a failure of the main instance. For MongoDB, a redundant database provision is enabled by increasing the number of pods.

Advantages of Kubernetes

- **Scalability**: Easy scaling of OPC Router instances according to requirements.
- **Flexibility**: Ability to manage different configurations and implementations.
- **Automation**: Automated management of deployments, updates and rollbacks.

Challenges of using Kubernetes

- **Complexity**: Implementing and managing Kubernetes clusters can be complex and requires specialized expertise.
- **Resource requirements**: Running a Kubernetes cluster requires significant hardware resources and can be costly.

For detailed implementation strategies, practical implementations and specific configuration examples, please visit the **Kubernetes and Container Orchestration** subpage.

Practical implementation

Centralized architecture

1. **Server provisioning**: Setting up multiple centralized servers.
2. **Configuration synchronization**: Implement mechanisms to synchronize configuration and data between servers.
3. **Failover test**: Regular testing of the failover mechanism to ensure availability.

Decentralized architecture

1. **Edge device deployment**: Install and configure edge devices near the data sources.
2. **Local redundancy**: Implementation of redundant edge devices for local fault tolerance.
3. **Store and Forward**: Activating and configuring the Store and Forward add-on.
4. **Local broker/database**: Set up a local broker or database for data caching and synchronization.

Hybrid architecture

1. **Centralized and decentralized components**: Combination of the steps from the centralized and decentralized architectures.
2. **Synchronization strategies**: Implementation of robust synchronization strategies between the central and decentralized components.
3. **Monitoring**: Regular monitoring and maintenance of both the central and decentralized instances.

Summary

For highly available and redundant systems, the OPC Router offers both centralized and decentralized architecture options that can be combined as needed. A centralized architecture with cross-redundancy provides high availability and centralized management, while a decentralized architecture minimizes latency and increases fault tolerance. A hybrid architecture combines the advantages of both approaches to ensure maximum flexibility and reliability.

Kubernetes and container orchestration

Advantages of Kubernetes

The use of Kubernetes offer a great amount of advantages for the deployment and management of the OPC Router :

- **Evolutivity:** Kubernetes enables to make the number of OPC Router instances evolving, based on the actual needs.
- **Flexibility:** The use of containers enables to manage and deploy easily different versions and configurations of the OPC Router.
- **Automation:** Kubernetes automatise multiple administratives tasks, including the deploiements, the rollbacks and the scalings, which simplify the management and reduce the errors.

Helm Chart for OPC Router

The Helm Chart for the OPC Router make the deployment and the management of the OPC Router in a Kubernetes Cluster easier. Here is the main functions and possibilities of configuration :

Deployment:

- **Helm Chart:** Enable to deploy the OPC Router on each Kubernetes Cluster.
- **Multiple possible configurations:** Taking charge of multiple options of configuration for answering to specific needs.

Configuration:

- **Global Parameters:** Settings such as the global Docker image registry and StorageClass.
- **Common Parameters:** Settings for the customisation of the name and settings of the service account.
- **OPC Router Parameters:** Configurations for the OPC Router image, the environment variables and the project repository.
- **MongoDB Parameters:** Settings for the MongoDB integration, including authentication and redundancy.

Redundancy:

- **OPC Router Redundancy:** Taking charge of the Redundancy mode to run a secondary pod with an OPC Router runtime, that will activate if there is a failure of the main pod.
- **MongoDB Redundancy:** Possibility to run multiples MongoDB pods to make the database redundant and increase reliability.

Best Practice:

To use fully the advantages of Kubernetes and Helm Charts, the following Best Practices should be observed with care.

Private Repositories:

- Ensure, that the Git repositories are private to prevent unauthorised access.
- Connection of the internal Git system of the OPC Router with hosted Git servers such as GitHub, BitBucket or Microsoft Azure DevOps.

Secrets Management:

- Secrets and passwords should never be stored in the Git repository.
- The OPC Router always save the secrets in a separate local vault to ensure that they are not versioned.

Regular Updates and Monitoring:

- Regular updates of the Kubernetes cluster and the Helm Charts
- Survey the Clusters to ensure that all the components function and evolve correctly.

Practical realisation:

Deployment with Helm Chart:

1. **Preparation:** Make sure that Kubernetes and Helm are installed and configured.
2. **Add the Helm repository:** Add the repository OPC Router Helm and update it :

```
helm repo add opc-router https://opc-router.github.io/helm-charts
helm repo update
```

3. **Installation of the OPC Router:** Use the following command to install the OPC Router :

```
helm install my-opcrouter opc-router/opc-router --set I_do_accept_the_EULA=true
```

4. **Security configuration:** Add the authentication and the security parameters to secure the installation :

```
helm install my-opcrouter opc-router/opc-router \
  --set webManagement.auth.disable=false \
  --set webManagement.auth.initialUser.name=<'Username for the initial user of the Web Management'> \
  --set webManagement.auth.initialUser.password=<'Password for the initial user of the Web Management'> \
  --set mongodb.auth.enabled=true \
  --set mongodb.auth.rootPassword=<mongodb root Password> \
  --set mongodb.auth.replicaSetKey=<mongodb replicaset key> \
  --set I_do_accept_the_EULA=true
```

Redundancy and Scaling:

1. **Activate the OPC Router redundancy:** Configure the redundant mode of the OPC Router into the Helm Chart, as soon as he is available.
2. **MongoDB Redundancy:** Ensure that MongoDB works with the redundant mode by increasing the number of pods :

```
--set mongodb.replicaCount=2
```

Launch a project from the Git repository:

- Reference the project repository and the path in the Helm Chart to load an OPC Router project from a Git repository:

```
--set project.projectRepo=<'URL to the Git Repository'> \
--set project.projectPath=<'Path to .rpe Project file'> \
--set project.configPath=<'Path to the configuration file of the Project'> \
--set project.auth.ssh_key=<'Private SSH Key'>
```

Summary

The use of Kubernetes and Helm Charts offer a powerful and flexible way to make the OPC Router working in an evolutive and manageable environment. Through the best practices and by using the posibilities of configuration, companies can maximize the reliability and efficiency of their OPC Router deployments.

Settings

The following options can be set by opening “Settings” in the project configuration. Select a runtime like “Local service” and click on “options”.

Runtime

Connection time-out	Time (in s) that the connection may take before a connection error is returned.
System Metrics update Interval (s)	How often system data is transferred to the OPC Router License Portal

Message group

Storage time for events	Time range in which the data are kept.
Storage time for transferred values	Time range in which the data are kept in case of faults.
Log transferred values	Activate if transferred values are to be recorded
Size of ring buffer	Select the needed size of ring buffer.
Maximum log size for transferred values (GB)	Specification of the maximum log size for transferred values in GB. Setting how far the OPC Router is allowed to fill the status databases. We recommend 8 or 9 GB for the database.
Maximum log size for events (GB)	Specify the maximum log size for events in GB.
Maximum amount of signs status	Limitation of the number of displayed characters of transferred values in the status view. This only affects arrays and strings.

Project backup

Project backup	Select when to perform an automatic project backup.
Maximum number of stored backup files	Number of project backups that are kept. If this number is exceeded in the target directory, the oldest project backup is automatically deleted.
Backup directory	Destination directory where the project backup will be saved.

Log output

Log all transfers	Writes all transfers to log file. This option overrides deactivation of logging in single connections.
Zip log files	This setting affects only the log files of the service. At the change of the day, the service automatically compresses the log file of the last day into a zip file. The associated text file will be deleted afterwards. Depending on the project, the required hard disk space can be significantly reduced.

Error	Logs errors. Recommended during normal operation.
Warning	Logs warnings. Recommend during normal operation.
Trace	Detailed logging to trace errors. Should be deactivated during normal operation.
Debug	Logs debug information for developers. Should be deactivated during normal operation.
Log folder	Folder for log files (default: *<OPC Router-Program folder>*\LogDir).

You can log the Router's activities in several levels. During normal operations, we recommend to log warnings and errors. The log file remains small, and if an error occurs, you have got detailed information. If an error occurs, activated “Trace” to log more information to find the error's cause. Open the log file with right click on the router's tray icon, then “Open log file”.

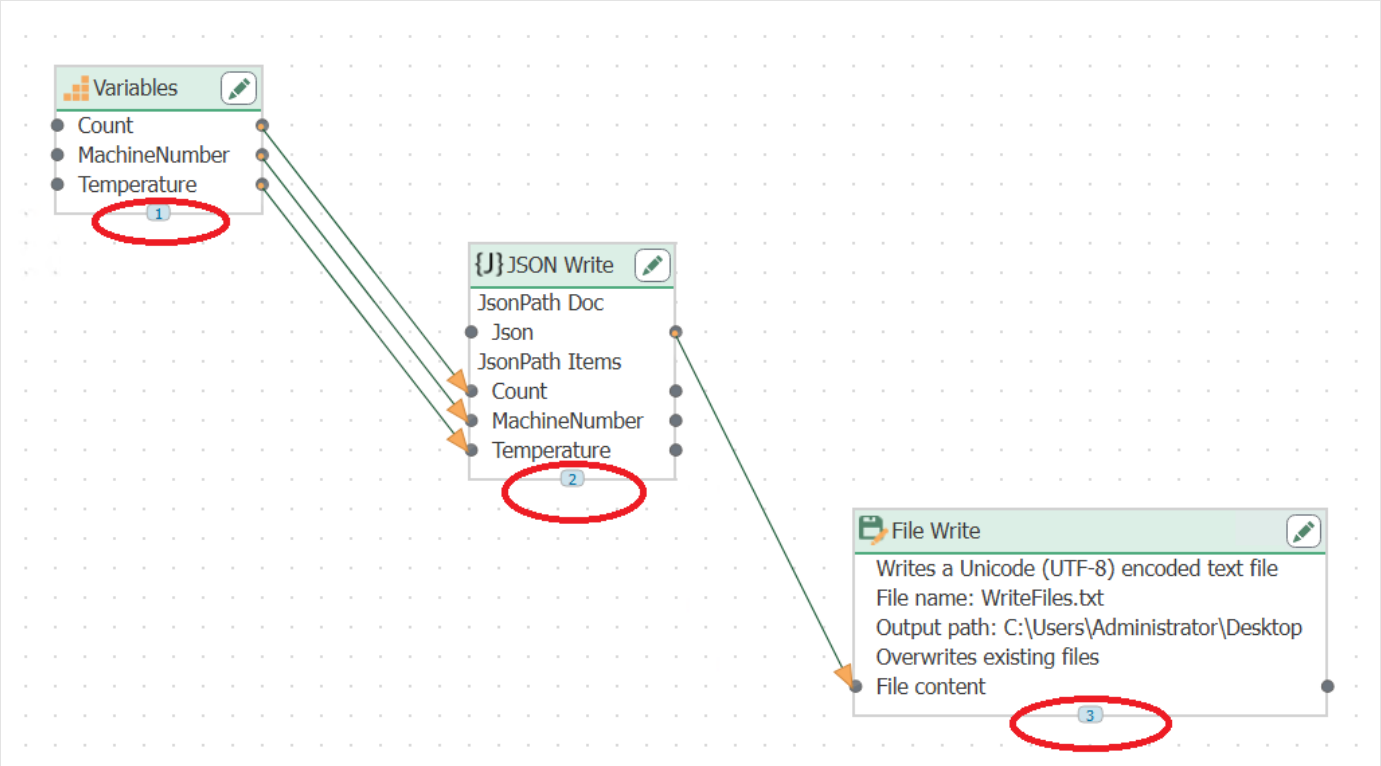
Every time an OPC Router process is started, it issues a message with its current version.

Language of Configuration

Use menu “Extras – Select language” to switch the language. You have to restart OPC Router Configuration to complete switching.

Show transfer steps

If you want to see in which order your transfer objects are executed in the connection, you can switch it on or off under “Information - Show transfer steps”.



Transfer steps

Resetting the database to factory defaults

In the expert mode: At “File - Database settings” you have the possibility to reset the database to the delivery state at the tab

“Delete database” content. In this area you can still select whether you want to delete all connections, all plug-ins or all router settings. If you select Delete all connections, all templates will be deleted!

Notification group

Via the menu item “Extras - Settings” you will find the notification area at the button for the basic settings.

The mail can be used for the following scenarios:

- transfer error
- plug-in error, plug-in again responsive
- connection error (plug-in / trigger error)
- runtime was stopped / started

At this point, enter all the shipping information and select the areas to which the notification applies in the list of events. You can also give a name for the notification via the settings. Click on the pen button. To rename an existing notification, click the plus sign. With the On / Off-ToggleSwitch you activate the notification or deactivate it.

![Notification](</assets/Notification (1).png>)

Settings

consolidation interval	The amount of time before a message is sent.
Timespan which is used until a message gets send	The length of time in which you can send. Events outside of this range are collected and sent later. Enter a start time and an end time and, if necessary, activate a weekday by activating an appropriate checkbox.
E-Mail sender (mail server)	Select a mail server as sender from the drop-down menu. To add a mail server, click on the button with the plus sign and make the settings. To edit or delete a mail server, select the appropriate buttons.
Receiver	At this point, enter the recipients who should receive this notification.

E-mail sender name

Name	Give the e-mail notification a name.

SMTP-Server

Server	Name of the server.
Port	Port on the server, which is used for the mail traffic.
Use Secure Socket Layer benutzen (SSL)	Uses SSL encryption. The server must support this, the appropriate port must be set.
Windows authentification	Alternatively, uses the user under which the router runtime is running (default: “system”).
Username	Username for logging on to the server (SMTP or NT authentication).
Password	Password to log in to the server (SMTP or NT authentication).
Domain	Domain to which the user is registered. (For login to the server via NT authentication)

E-Mail settings

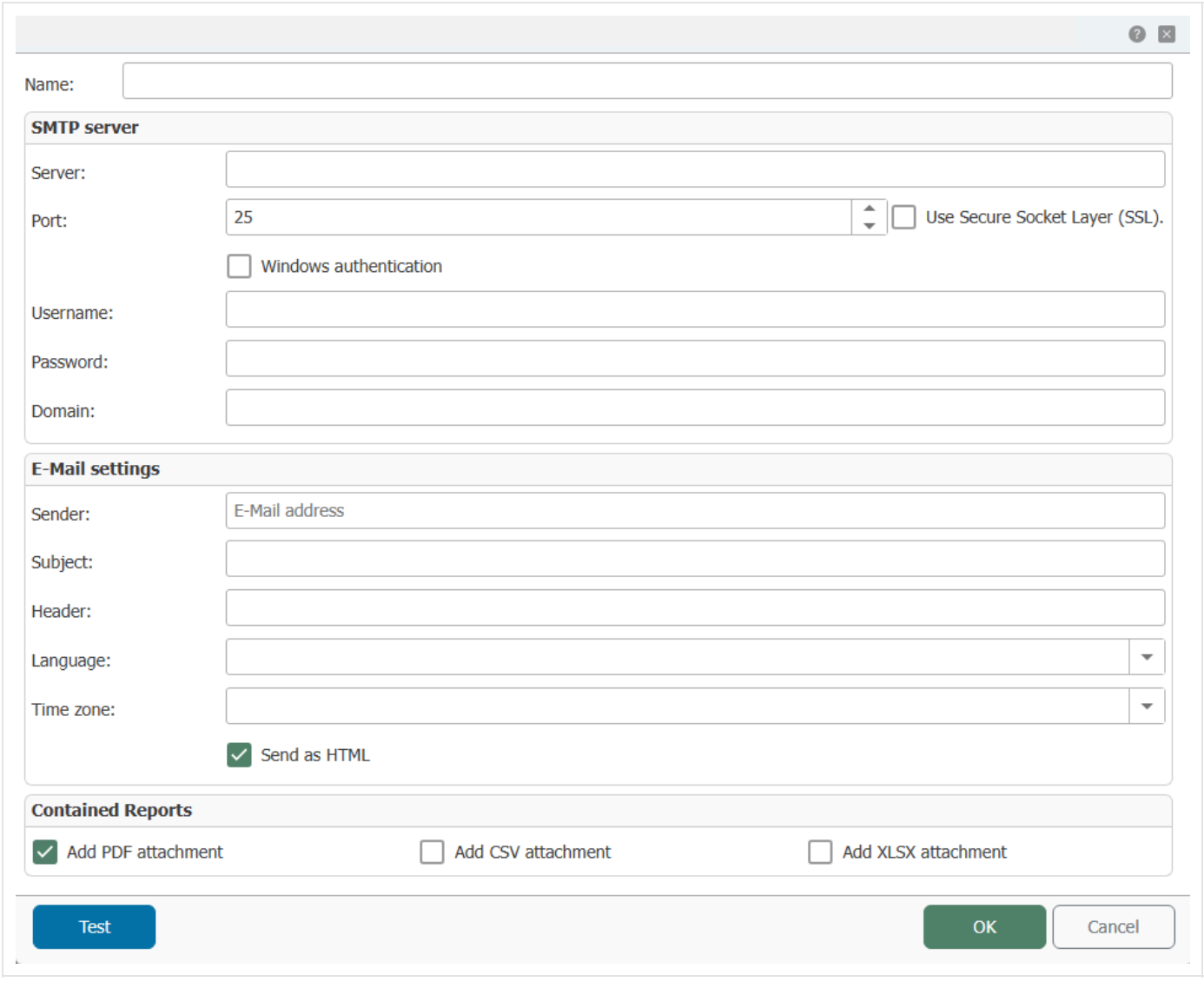
Sender	Enter the sender e-mail address. The sender must have the format of a mail address (name@domain.xx).
Subject	Enter a subject for your notification. Max 255 characters are possible.
Header	Extended entry of a header possible.
Language	Select a language from the drop-down menu.
Time zone	Select a time zone from the drop-down menu.
Send as HTML	Activate the checkbox if you want to send the notification in HTML format.

Enthaltene Reports

Add PDF attachment	Activate the checkbox if you want to send a PDF file.
Add CSV attachment	Activate the checkbox if you want to send a CSV file.
Add XLSX attachment	Activate the checkbox if you want to send a XLSX file

The “Test” button allows you to verify that the server connection is working.

Confirm your entries with “OK”.

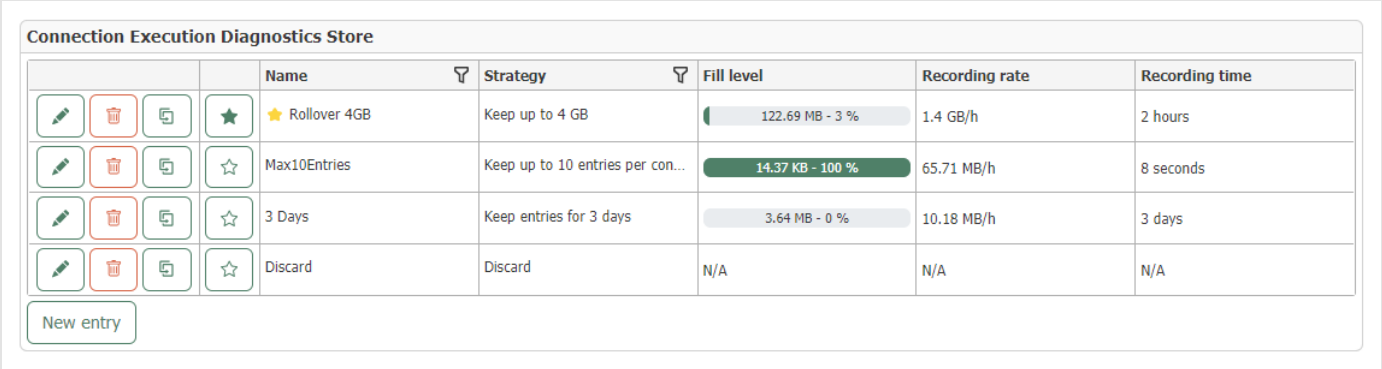

Name:
SMTP server
Server:
Port:
25
Use Secure Socket Layer (SSL).
Windows authentication
Username:
Password:
Domain:
E-Mail settings
Sender:
E-Mail address
Subject:
Header:
Language:
Time zone:
Send as HTML
Contained Reports
Add PDF attachment
Add CSV attachment
Add XLSX attachment
Test
OK
Cancel

Status storage options

Connection Execution Diagnostics Storages

New diagnostic storages for connection execution can be created here. The values that are transmitted in a connection are saved in these diagnostic storages. The storad values are being displayed when you click on a point in the status chart of a connection.

Preconfigured Connection Execution Diagnostics Storages



Diagnostic Storages overview

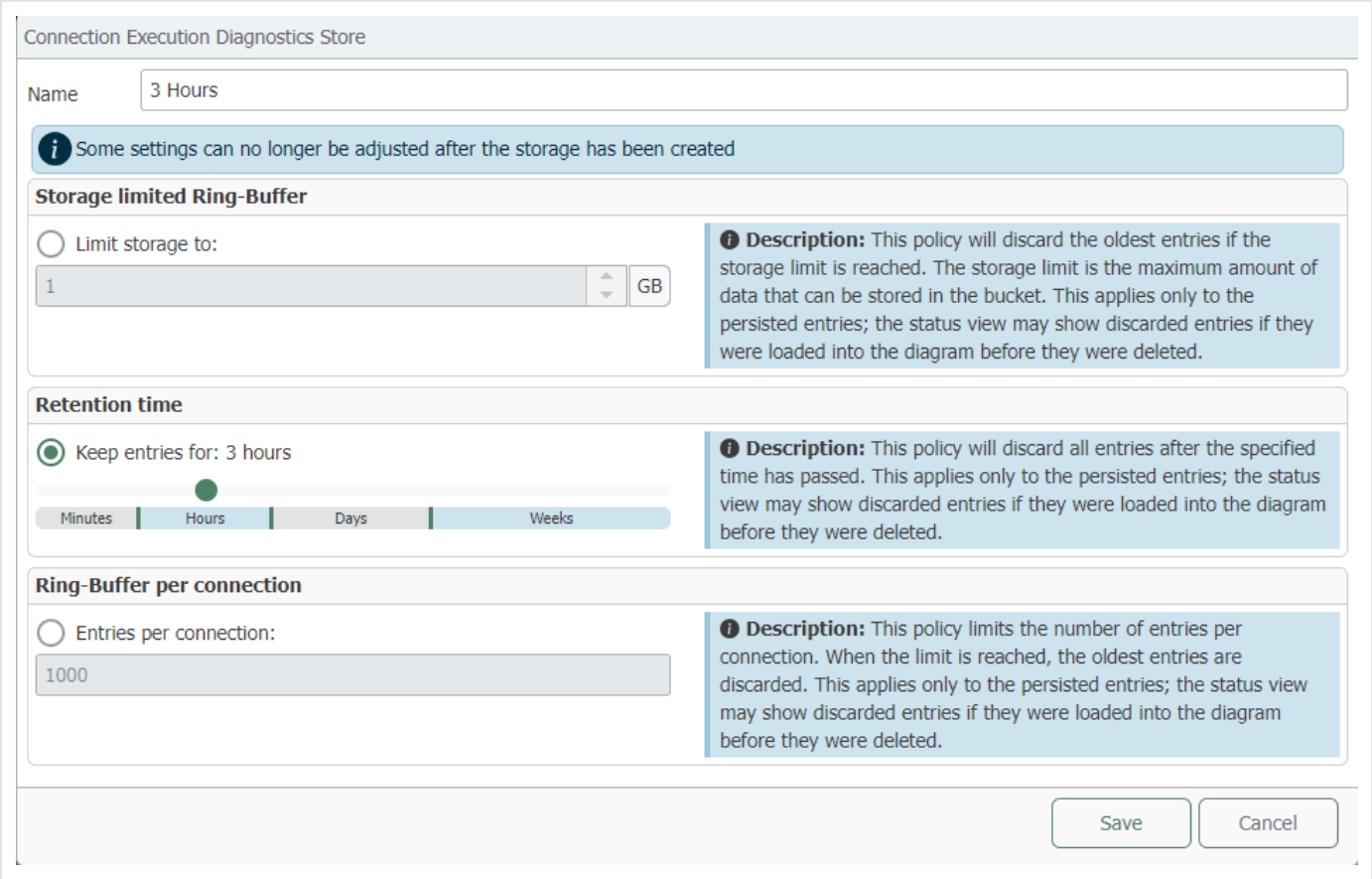
Rollover 4GB (default)	Saves a maximum of 4 GB of data. When the limit is reached old data is being overwritten
Max10Entries	Saves at most values from the last 10 executions per connection.
3 Days	Saves data for 3 days. Older data is being deleted.
Discard	No data is being saved. Discard is well suited if no storage of the transferred data is required. This can save memory space.

Overview of the diagnostic memory for connection execution

The diagnostic memories are listed in a clear table.

On the left-hand side of each row, there are buttons for editing, deleting and duplicating the diagnostic memories. The star button can be used to set a diagnostic memory as the default. You can find out how these memories can be used for connections here: settings-for-connections-and-templates.md.

Create a Connection Execution Diagnostics Storage



Creating a new Diagnostics Storage

The following properties can be configured for a Diagnostics Storage:

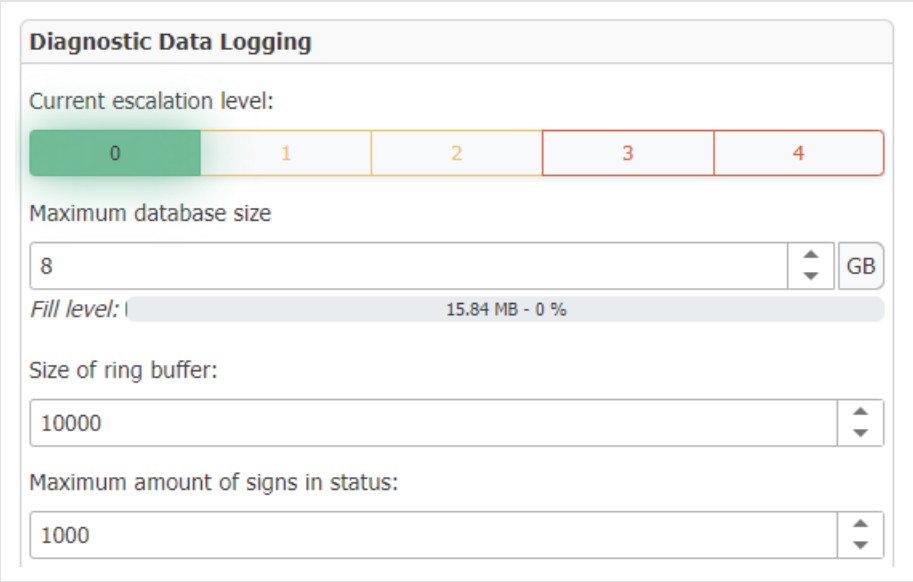
Name	The name for the diagnostics storage. It is used to identify the Diagnostics Storage when projecting. **Note:** The name cannot be changed later.
Type of limitation	**Storage limited Ring-Buffer:** The amount of data that is stored before data is overwritten can be specified here. **Retention time:** Here you can specify how long the data is stored before it is being deleted. **Ring-Buffer per connection:** Here you can specify the maximum number of transfers per connection for which data is being saved before data is overwritten.

After creation, the newly created storage is displayed in the overview.

Diagnostic Data Logging

Here you can conveniently see what the current escalation level is.

For more information see: Escalation levels



Diagnostic Data Loggin with current escalation level

The following options can be configured for Diagnostic Data Logging

Maximum database size	The maximum amount of data in GB that can be stored. Default value: 8 GB.
Size of ring buffer	The maximum number of entries in the ring buffer. Default value: 2000 entries.
Maximum amound of characters in the status	The maximum number of characters that are saved for a value in the status display. Default value: 1000 characters

Transfer execution diagnostics

Here you can configure which transfer execution diagnostic data should be recorded and how much of it should be stored.

The following data can be recorded:

Notifications	History of sent notifications. Default value: 512 MB.
Connection status	The status history of connections. e.g. "Ready" or "Plug-in Error". Default value: 256 MB.
Plug-in status	The status history of plug-ins. e.g. "Ready" or "Error". Default value: 256 MB.

Diagnosis of the runtime service

Here you can configure which runtime service diagnostic data should be recorded and how much of it should be stored.

The following data can be recorded:

Service Events	History of Service Events. Default value: 256 MB.
Service Status	Status history of the runtime service. Default value: 256 MB.
Service Health	Health of the runtime service. Default value: 256 MB.
Performance metrics	Data on the performance of the OPC Router. Default value: 256 MB.

If enough data is available, an estimated recording rate and recording time are being shown for the diagnostic data of the transmission execution and the runtime service. This can look like this, for example:

Data recording estimates

Approximately 3.46 MB of data is being recorded per hour. This means that the data can be stored for approximately 3 days before it is being overwritten.

Certificate management

Via the menu item Extras–Settings you will find the certificate management area next to the button for the basic settings. In this area you can import certificates, create your own certificates or view existing certificates.

The following buttons are available:

Import: With this button you have the possibility to import certificates from your PC into the OPC Router certificate administration. They will be available later in the drop-down menus of the individual applications.

View: Select a certificate in the list and click the “View” button to display the details and the path for the certificate.

Add private key: The private key is the secret key that will later be used to sign the encrypted connection. The private key is created by you and should not be made public. A private key can only be added for OPC Router certificates and not for trusted certificates.

Remove: Select a certificate and click the “Remove” button to delete this certificate from the list.

Create: With this button you can create your own certificates. Enter the appropriate properties and confirm your entries with “Create”. You can only create OPC Router certificates and not trusted certificates.

Revoke: Trusted certificates can be invalidated using the “Revoke” button. You can then only re-import it to use it as a valid trusted certificate again.

![Redundanz](</assets/Certificate_Management (1).png>)

Redundancy

You can find the Redundancy settings via the parameters. This allows the OPC Router to be protected against failures.

If the secondary OPC Router loses the connection to the primary OPC Router, the connections are activated and start transferring. This is shown in green in the status display. If the client can re-establish the connection to the primary OPC Router, the connections switch back to standby. This is shown in blue in the status display.

Two types of redundancy can be selected in the OPC Router.

Secure Redundancy

- Secure redundancy with keys that are exchanged.
- Should always be used when possible.

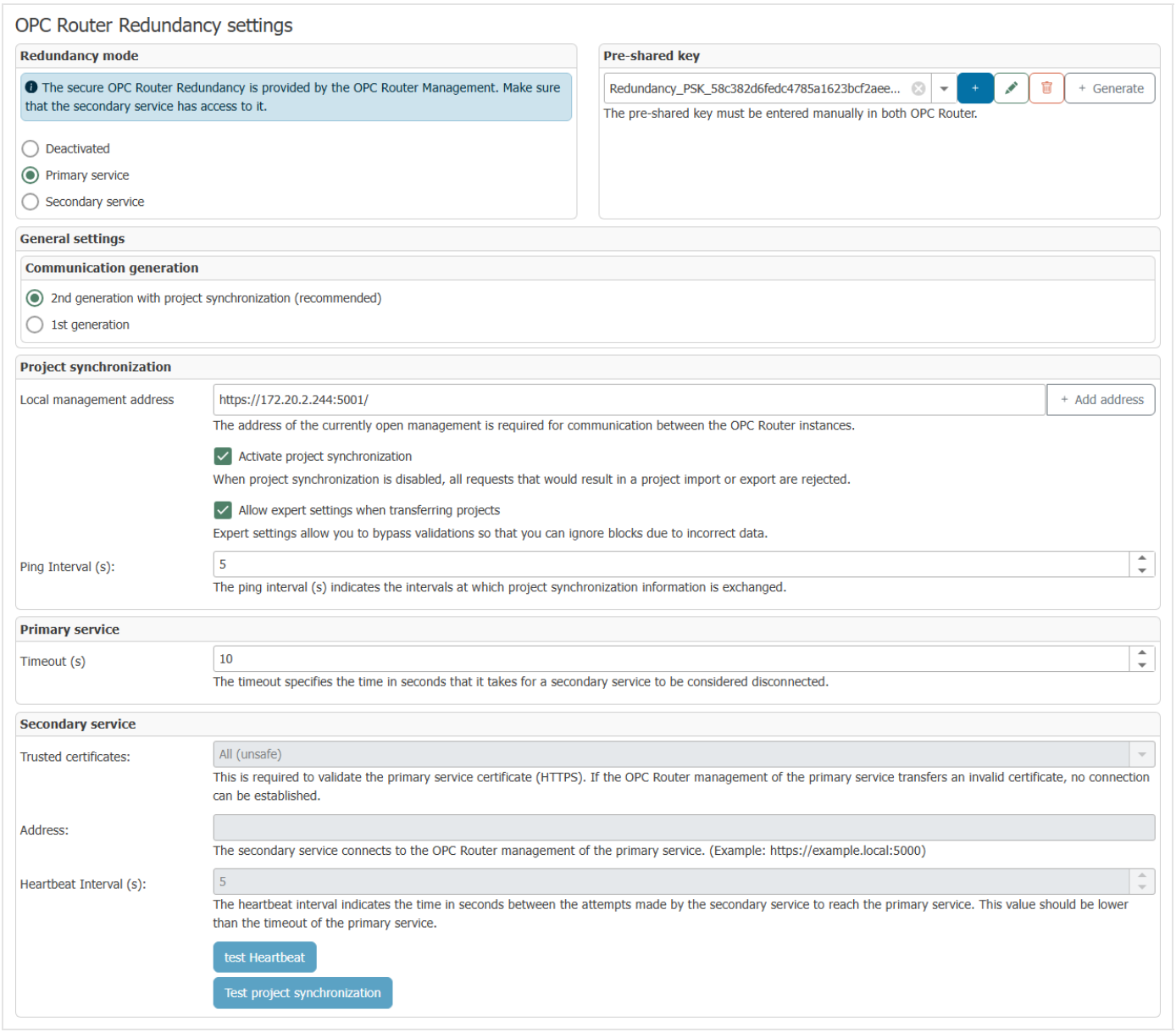
Secure Redundancy

Legacy Redundancy

- Is included for backwards compatibility
- No longer supported as of version 5.4

Legacy Redundancy

Secure Redundancy

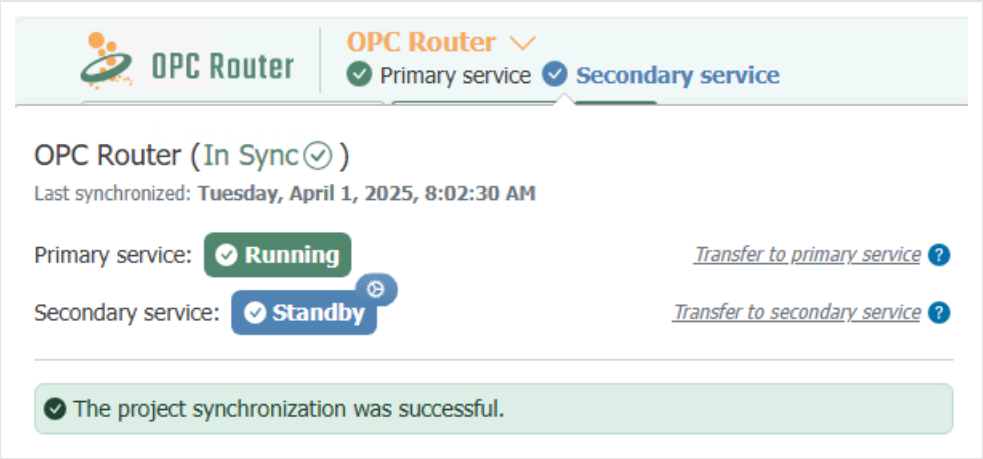


OPC Router redundancy settings

Configuration of the Secure Redundancy

Project synchronisation

In order for the primary and secondary services to be able to synchronise a project, it must be set in both how their respective management in the network can be achieved for the other. The actual synchronisation can be carried out later via the redundancy status in the header menu:



In addition to the basic status, it is also possible to view whether or when a project synchronisation has taken place and to trigger it if required.

Redundancy mode

Primary service

The primary service receives heartbeat signals from the secondary service. If the heartbeat signals are interrupted, a warning is displayed. If the primary service is no longer available, the secondary service takes over.

Pre-shared key	Character string that is used to authenticate the OPC Router. A secure character string can be generated using the ‘Generate’ button. This key must be the same for the primary and secondary service. An empty key is invalid and prevents the connection from being established.	
Timeout (s)	Time in seconds after which the secondary service is considered disconnected if no heartbeat has been sent.	

Secondary service

The secondary service is on standby as long as the primary service is available and starts as soon as the primary service is no longer available. The connection to the opposite service can be tested via "Test heartbeat". Test project synchronisation" can be used to check whether all the necessary communication paths of the OPC routers involved can be established and the necessary data can be exchanged. It is not possible to check whether the necessary settings have been made or whether synchronisation has actually been successful.

For the connection test to work, the redundancy configuration must be switched to productive.

Pre-shared key	Character string that is used to authenticate the OPC Router. A secure character string can be generated using the ‘Generate’ button. This key must be the same for the primary and secondary service. An empty key is invalid and prevents the connection from being established.
Trusted certificates	If an HTTPS address is specified, the certificates to be trusted can be selected here: • All (unsafe): Every certificate is accepted. • Router: A connection is authorised if the certificate is in the router's certificate management and is trusted. • Windows: A connection is authorised if the certificate is available in the Windows certificate store.

Address	Address of the primary service. This is specified in URL format (http(s):host name/IP address:port). The port used to access the Web Management (the web interface of the OPC Router) must be specified as the port. **Example:** https://example.local:5001
Heartbeat Interval (s)	The interval in seconds in which the secondary service attempts to reach the primary service. **Note:** This value must be below the timeout of the primary service so that the connection in the primary service is not constantly regarded as disconnected.

Deactivated

In this mode the redundancy is inactive.

Note: The Web Management port of the primary service must be enabled in the firewall so that the secondary service can connect.

The status changes caused by the redundancy are logged. A log entry is created when the redundancy becomes active (Primary router has failed) or becomes inactive again (Primary router is accessible again).

Environment variables

Several settings for redundancy behaviour can be set via Configurable environment variables for redundancy.

Settings set via environment variables always overwrite settings made via the UI.

Warning: SAP®-Trigger can be problematic in redundant operation with identical user data. Problems arise because the plug-ins are also initialised in the secondary system and want to access the same user data.

Legacy Redundancy

Via the menu item Settings, you will find the redundancy settings. This way, the OPC Router can be protected against failures.

There is a “Primary OPC Router” and a “Secondary OPC Router”. ~~*Here only the port in the firewall must be released.*~~ If the connection leaves from the secondary OPC Router to the primary OPC Router, the connections are switched to active and begin to transfer. This is shown in green in the status display. If the secondary OPC Router can reconnect to the Primary OPC Router, the connections switch back to standby. This is shown in blue in the status display.

Redundancy settigns	Select the mode with that the OPC Router should run. • **Primary OPC Router:** Default setting of the OPC Router • **Secondary OPC Router:** starts in standby mode and monitors the primary OPC Router
Primary hostname	Enter the IP/DNS of the primary OPC Router.
Check period (s)	Enter the interval in which the secondary OPC Router checks the connection to the primary OPC Router.
Reconnect interval	Enter an interval in which the secondary OPC Router checks whether the primary OPC Router is reachable again.

Note: The port in the firewall must be released.

The state changes due to the redundancy are logged. A log entry is created when the redundancy becomes active (master has failed) or becomes inactive again (master is reachable again).

Attention: SAP® triggers and ThingWorx® triggers can be problematic in redundancy operation with identical user data. There are problems because the plug-ins are also initialized in the secondary system and want to access the same user data, which can lead to errors in the assignment of sessions with SAP® and ThingWorx® connections.

HTTP(s) failover concepts for REST, OPC UA Server & WebManagement in the OPC Router

What is failover?

Failover ensures that requests are automatically redirected to another available server in the event of a server failure. A reverse proxy can perform health checks to check which endpoints are active.

Use case example for failover with OPC router redundancy

Situation

The two OPC routers of a redundancy pair provide the same REST endpoint.

Here are projects from the primary and secondary service.

Primary Service Project

Secondary Service Project

Requirement

The REST endpoint should be reliably accessible at an address. Requests to this address should be forwarded to the active OPC router of the redundancy pair.

Solution

A reverse proxy, e.g. Nginx, is configured so that requests to a specific address can be forwarded to both OPC routers. Health checks are performed to determine which endpoint is currently active.

The configuration of a failover for an OPC router redundancy pair with Nginx and Traefik is described below.

Redundancy failover with Nginx

What is Nginx?

Nginx is a powerful, open-source web and reverse proxy server known for its performance and scalability. In addition to the open source version, there is **Nginx Plus**, a commercial solution with extended features such as security functions and support. Both are used for caching, load balancing and the provision of modern web applications.

There are several ways to set up a failover for OPC router redundancy with Nginx. The setup of a failover with Nginx Plus and Nginx is described below.

Failover with Nginx Plus

This is an example of an Nginx configuration that enables failover for an OPC router redundancy pair.

```
events { }

http {
    upstream opcrouter-service {
        zone opcrouter-service 64k;
        # Defines the servers to which a request can be forwarded.
        server opcrouter-primary:8080;
        server opcrouter-secondary:8080;
    }

    server {
        listen 80;

        location /api {
            # Replaces /api/ with /services/MyRestServer/api/
            rewrite ^/api/(.*) /services/MyRestServer/api/$1 break;
            # Forward the request to one of the endpoints defined in opcrouter-service.
            proxy_pass http://opcrouter-service;

            # Check the health of the endpoints
            health_check
            # Interval in seconds at which the health of the endpoints is checked.
            interval=5
            # Defines how many consecutive times the request may fail before the endpoint is considered "Unhealthy".
            fails=1
            # Defines how many times the request must be successful in succession before the endpoint is considered "Healthy".
            passes=1
            # The url used to determine the health status.
            uri=/health/runtime/ready;
        }
    }
}
```

Failover with Nginx

Active health checks with `health_check` are not available in the free version. Therefore, a passive health check is used in the following example.

```
events { }

http {
    upstream opcrouter-service {
        # Defines the servers to which a request can be forwarded.
        server opcrouter-primary:8080 max_fails=2 fail_timeout=5s;
        server opcrouter-secondary:8080 max_fails=2 fail_timeout=5s;
        # max_fails is the number of failures that may occur within the fail_timeout before the server is temporarily (for fail_timeout) considered
"Unhealthy".
    }

    server {
        listen 80;
        location /api {
            rewrite ^/api/(.*) /services/MyRestServer/api/$1 break;
            proxy_pass http://opcrouter-service;
            # Specifies in which cases the request should be repeated with another server. In this case for errors, timeouts, and http codes 404
and 503
            proxy_next_upstream error timeout http_404 http_503;
        }
    }
}
```

Docker Compose example for Redundancy failover with Nginx

The Nginx configurations described on page Redundancy failover with Nginx can be used in Docker Composer to create an OPC router redundancy pair with failover.

The files that are used as a project can be found on the HTTP(s) failover concepts for REST, OPC UA Server & WebManagement in the OPC Router page.

```
services:
  nginx_plus:
    # Nginx Plus
    image: private-registry.nginx.com/nginx-plus/base:<Tag>
    # Alternatively Nginx
    # image: nginx:latest
    restart: always
    networks:
      - opcrouter-net
    ports:
      - "5080:80"

    environment:
      # Only required for Nginx Plus
      - NGINX_LICENSE_JWT=<Nginx license JWT>
    volumes:
      # Make nginx.conf available in the container
      - ./nginx.conf:/etc/nginx/nginx.conf:ro

  opcrouter-primary:
    image: opcrouter/service:latest
    restart: always
    networks:
      - opcrouter-net
    ports:
      - "5085:8080"
    environment:
      - OR_I_ACCEPT_EULA=true
      - INITIAL_USERNAME=admin
      - INITIAL_PASSWORD=SuperSecret123
      - OR_REDUNDANCY_MODE=primary
      - OR_REDUNDANCY_LOCAL_MANAGEMENT_ADDRESS=http://opcrouter-primary:8080
      - OR_REDUNDANCY_SHARED_KEY=LUyLxk4kZdvYW9kM6ZSBWbBatdj9RGow9Bp
      - OR_IMPORT_SOURCE=/inray/REST_Primary.rpe
    volumes:
      # Make project available in the container
      - ./REST_Primary.rpe:/inray/REST_Primary.rpe

  opcrouter-secondary:
    image: opcrouter/service:latest
    restart: always
    networks:
      - opcrouter-net
    ports:
      - "5086:8080"
    environment:
      - OR_I_ACCEPT_EULA=true
      - INITIAL_USERNAME=admin
      - INITIAL_PASSWORD=SuperSecret123
      - OR_REDUNDANCY_MODE=secondary
      - OR_REDUNDANCY_SHARED_KEY=LUyLxk4kZdvYW9kM6ZSBWbBatdj9RGow9Bp
      - OR_REDUNDANCY_ADDRESS=http://opcrouter-primary:8080
      - OR_REDUNDANCY_LOCAL_MANAGEMENT_ADDRESS=http://opcrouter-secondary:8080
      - OR_REDUNDANCY_PRIMARY_TIMEOUT=30
      - OR_REDUNDANCY_SECONDARY_HEARTBEAT_INTERVAL=10
      - OR_IMPORT_SOURCE=/inray/REST_Secondary.rpe
    volumes:
      # Make project available in the container
      - ./REST_Secondary.rpe:/inray/REST_Secondary.rpe

networks:
  opcrouter-net:
```

Redundancy failover with Traefik

What is Traefik?

Traefik is a modern, open-source reverse proxy and load balancer for microservice-oriented architectures. It supports multiple backends such as Docker and Kubernetes, offers automatic service discovery, SSL/TLS encryption and easy management via a dashboard or API.

There are several ways to configure Traefik, as different "providers" can be used for a configuration, e.g. files and Docker.

In the following, the configuration of Traefik with files is discussed first, as this type of configuration can be used universally. Later, an example will show how Traefik can be configured in Docker Compose.

Configuration of failover with files

The general configuration of Traefik for the use of configuration files could look like this. In this example, all configuration files in the `/etc/traefik/traefik_configs` folder are used to configure Traefik.

```
api:
  dashboard: true
  insecure: true

ping: {}

providers:
  file:
    # Specifies the directory in which the configuration files are located.
    directory: /etc/traefik/traefik_configs

entryPoints:
  web:
    address: ":80"
```

A failover configuration file for an OPC router redundancy pair could look like this. This file must be in the configured directory for configuration files.

```
http:
  routers:
    api-router:
      # Responds to the prefix /api
      rule: "PathPrefix(`/api`)"
      # Sends requests to opcrouter-service
      service: opcrouter-service
      entryPoints:
        - web
      middlewares:
        - api-rewrite

  middlewares:
    api-rewrite:
      # Replaces /api/ with /services/MyRestServer/api/
      replacePathRegex:
        regex: "^/api/(.*)"
        replacement: "/services/MyRestServer/api/$1"

  services:
    opcrouter-service:
      loadBalancer:
        # Server of the redundancy pair
        servers:
          - url: "http://opcrouter-primary:8080"
          - url: "http://opcrouter-secondary:8080"
        # Sends requests to the specified path to test availability
        healthCheck:
          path: "/health/runtime/ready"
          interval: "5s"
          timeout: "2s"
```

Docker Compose example for Redundancy failover with Traefik

Docker Compose example for Redundancy failover with Nginx

The following example shows how Traefik can be configured in Docker Compose. No other configuration files are required. The necessary configurations are passed as command parameters or defined via labels.

The files used as a project can be found on the page HTTP(s) failover concepts for REST, OPC UA Server & WebManagement in the OPC Router.

```
services:
  traefik:
    image: traefik:v3.0
    restart: always
    networks:
      - opcrouter-net
    # Configurations as command parameters
    command:
      - “--api.dashboard=true”
      - “--api.insecure=true”
      - “--entrypoints.web.address=:80”
      - “--providers.docker=true”
      - “--providers.docker.exposedbydefault=false”
      - “--ping”
    ports:
      - “5080:80”
      - “5081:8080” # Traefik Dashboard (optional)
    volumes:
      - “/var/run/docker.sock:/var/run/docker.sock:ro”

  opcrouter-primary:
    image: inrayhub.azurecr.io/opcrouter-runtime:5.3.6001.68
    restart: always
    networks:
      - opcrouter-net
    ports:
      - “5085:8080”
    environment:
      - OR_I_ACCEPT_EULA=true
      - INITIAL_USERNAME=admin
      - INITIAL_PASSWORD=SuperSecret123
      - OR_REDUNDANCY_MODE=1
      - OR_REDUNDANCY_LOCAL_MANAGEMENT_ADDRESS=http://opcrouter-primary:8080
      - OR_REDUNDANCY_SHARED_KEY=LUyLxk4kZdvYW9kM6ZSBWbBatdj9RGow9Bp
      - OR_IMPORT_SOURCE=/inray/REST_Primary.rpe
volumes:
      # Make project available in container
      - ./REST_Primary.rpe:/inray/REST_Primary.rpe
    # Configurations via labels
    labels:
      - “traefik.enable=true”
      # Responds to the prefix /api
      - “traefik.http.routers.api-router.rule=PathPrefix(`/api`)”
      - “traefik.http.routers.api-router.entrypoints=web”
      - “traefik.http.routers.api-router.service=opcrouter-service”
      - “traefik.http.routers.api-router.middlewares=api-rewrite”
      # Replaces /api/ with /services/MyRestServer/api/
      - “traefik.http.middlewares.api-rewrite.replacepathregex.regex=^/api/(.*)”
      - “traefik.http.middlewares.api-rewrite.replacepathregex.replacement=/services/MyRestServer/api/$$1”
      # Load balancing configuration
      - “traefik.http.services.opcrouter-service.loadbalancer.server.port=8080”
```

```
      # Check availability by sending requests to the specified path
      - “traefik.http.services.opcrouter-service.loadbalancer.healthcheck.path=/health/runtime/ready”
      - “traefik.http.services.opcrouter-service.loadbalancer.healthcheck.interval=5s”
      - “traefik.http.services.opcrouter-service.loadbalancer.healthcheck.timeout=2s”

opcrouter-secondary:
    image: inrayhub.azurecr.io/opcrouter-runtime:5.3.6001.68
    restart: always
    networks:
      - opcrouter-net
    ports:
      - “5086:8080”
    environment:
      - OR_I_ACCEPT_EULA=true
      - INITIAL_USERNAME=admin
      - INITIAL_PASSWORD=SuperSecret123
      - OR_REDUNDANCY_MODE=2
      - OR_REDUNDANCY_SHARED_KEY=LUyLxk4kZdvYW9kM6ZSBWbBatdj9RGow9Bp
      - OR_REDUNDANCY_ADDRESS=http://opcrouter-primary:8080
- OR_REDUNDANCY_LOCAL_MANAGEMENT_ADDRESS=http://opcrouter-secondary:8080
- OR_REDUNDANCY_PRIMARY_TIMEOUT=30
      - OR_REDUNDANCY_SECONDARY_HEARTBEAT_INTERVAL=10
      - OR_IMPORT_SOURCE=/inray/REST_Secondary.rpe
    volumes:
      # Make project available in container
      - ./REST_Secondary.rpe:/inray/REST_Secondary.rpe
    # Configurations via labels
    labels:
      - “traefik.enable=true”
      # Responds to the prefix /api
      - “traefik.http.routers.api-router.rule=PathPrefix(`/api`)”
      - “traefik.http.routers.api-router.entrypoints=web”
      - “traefik.http.routers.api-router.service=opcrouter-service”
      - “traefik.http.routers.api-router.middlewares=api-rewrite”
      # Replaces /api/ with /services/MyRestServer/api/
      - “traefik.http.middlewares.api-rewrite.replacepathregex.regex=^/api/(.*)”
      - “traefik.http.middlewares.api-rewrite.replacepathregex.replacement=/services/MyRestServer/api/$$1”
      # Load balancing configuration
      - “traefik.http.services.opcrouter-service.loadbalancer.server.port=8080”
      # Check availability by sending requests to the specified path
      - “traefik.http.services.opcrouter-service.loadbalancer.healthcheck.path=/health/runtime/ready”
      - “traefik.http.services.opcrouter-service.loadbalancer.healthcheck.interval=5s”
      - “traefik.http.services.opcrouter-service.loadbalancer.healthcheck.timeout=2s”


networks:
  opcrouter-net:
```

Health Endpoints

The OPC router provides health endpoints that can be used to query data about the status of the OPC router, e.g. the redundancy state. Both the Runtime and the Web Management provide health endpoints.

The possible values for a status are:

- Healthy
- Unhealthy
- Degraded

Health endpoints of the Web Management

The Web Management provides the following health endpoints.

ENDPOINT	DESCRIPTION
/health	General state of the Web Management
/health/live	Liveness probe
/health/ready	Readiness probe
/health/details	Details on the status in JSON format
/health/runtime<Runtime health endpoint>	Reverse proxy to the health endpoints of the Runtime

Health Endpoints of the Runtime

Configuration

The Heath endpoints of the Runtime can be configured using the following environment variables.

ENVIRONMENT VARIABLE	DESCRIPTION	DEFAULT VALUE
`OR_ENABLE_HEALTH_ENDPOINT`	Activates the provision of all health endpoints	`true`
`OR_RUNTIME_HEALTH_ENDPOINT_PORT`	Port on which the Runtime provides the health endpoints	53200

The port configured with `OR_RUNTIME_HEALTH_ENDPOINT_PORT` is only accessible from localhost, e.g. from the Docker container or the Windows server in/on which the OPC router is running.

Endpoints

The Runtime provides the following health endpoints.

ENDPOINT	DESCRIPTION
/	General state of the Runtime
/live	Liveliness probe

ENDPOINT	DESCRIPTION
/ready	Readiness probe and general state of the Runtime Returns the status code 200 if all health checks are successful. Otherwise, the status code 503 is returned.
/details	Details of all health checks in JSON format
/RedundancyState	Current Redundancy state in JSON format e.g. `{"status": "Ready"}` The possible values for the redundancy status are: • Ready • Standby • LicenseError
/TransferManager	Transfer Manager health check
/RuntimeLicenseManager	License Manager health check
/HistoricalDiagnosisStorage	Circular Storage health check
/TelemetryManager	Telemetry Manager health check

Store and Forward

Via the menu item Extras - Settings you will find the Store and Forward button.

Store and Forward can be used to cache data when servers are unreachable.

To enable Store anf Forward, click the slider at the top so that it is turned ON.

Attention: Even with Store and Forward enabled, it must be additionally enabled in connections you want to.

Settings

Enable storage time	Specifies the maximum time in hours that the router keeps the recorded values. After this time has elapsed, without the plug-in coming out of the error state again, the values that are too old are discarded.
Storage in GB	Maximum storage capacity that the OPC Router provides for the buffered values. Indication in gigabyte.
Behavior when memory capacity is reached	• Connection error: The configured connection switches to error state. • Ignore: No further data is recorded.
Activate store and forward as default for new connections	Enable this option if you want the Store and Forward option to be enabled for newly created connections.

Dynamic instances API

Via the menu item Extras - Settings you can find the button Dynamic Instances API Settings.

Dynamic instances allow you to generate as well as remove connections at runtime.

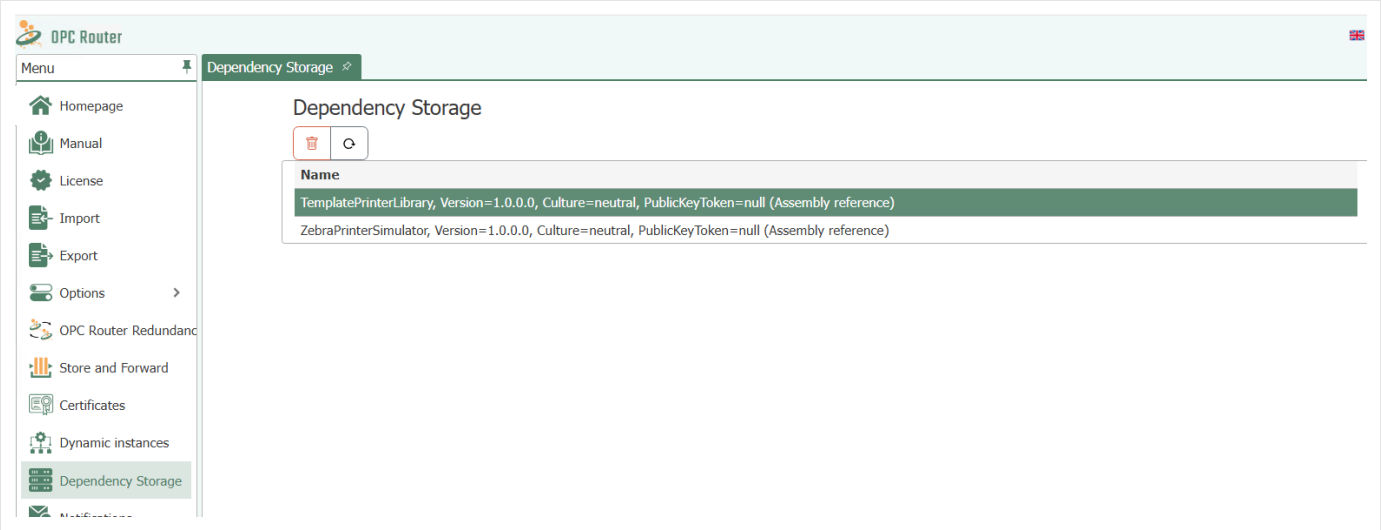
More information about dynamic instances and how to use them correctly can be found here: Dynamic instances

Settings

Host	Specify the DNS or the IP of the database host
Database	Specify the name of the database where the tables for the dynamic instances should be stored
User data	Enter the user data for the database. When using the Windows user data the OPC Router runtime must be logged on in the correct context.

Dependency Storage

In Dependency Storage, you can manage all added assemblies. You can update the list or delete those that have already been added.



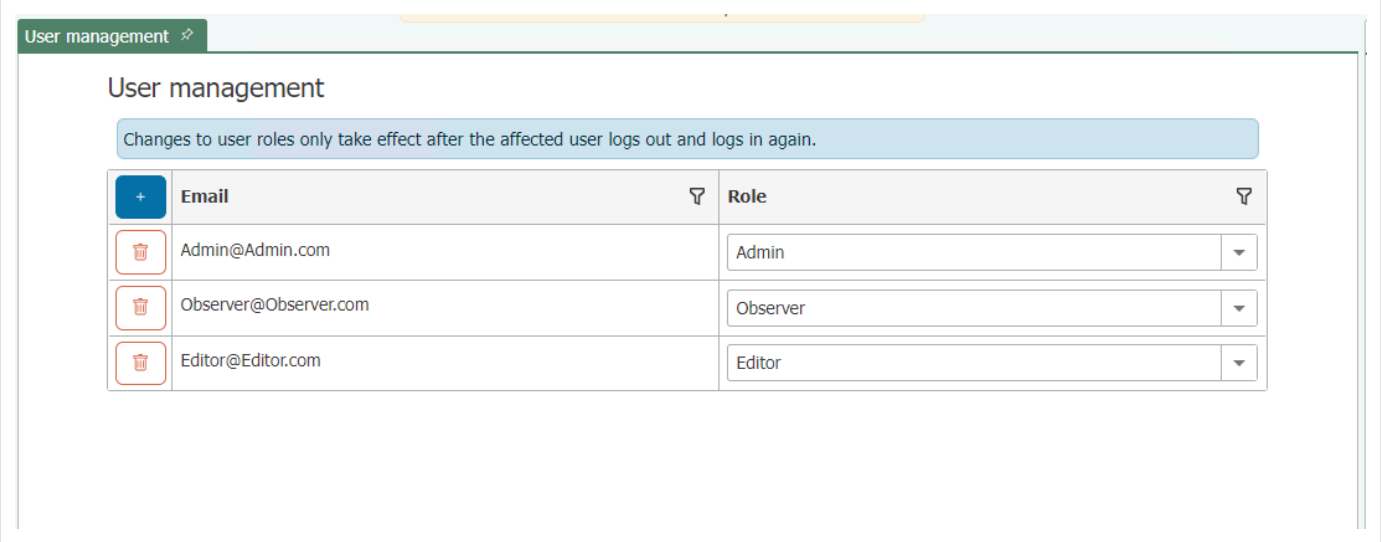
Dependency storage

Note: The assemblies are added in the Script Editor.

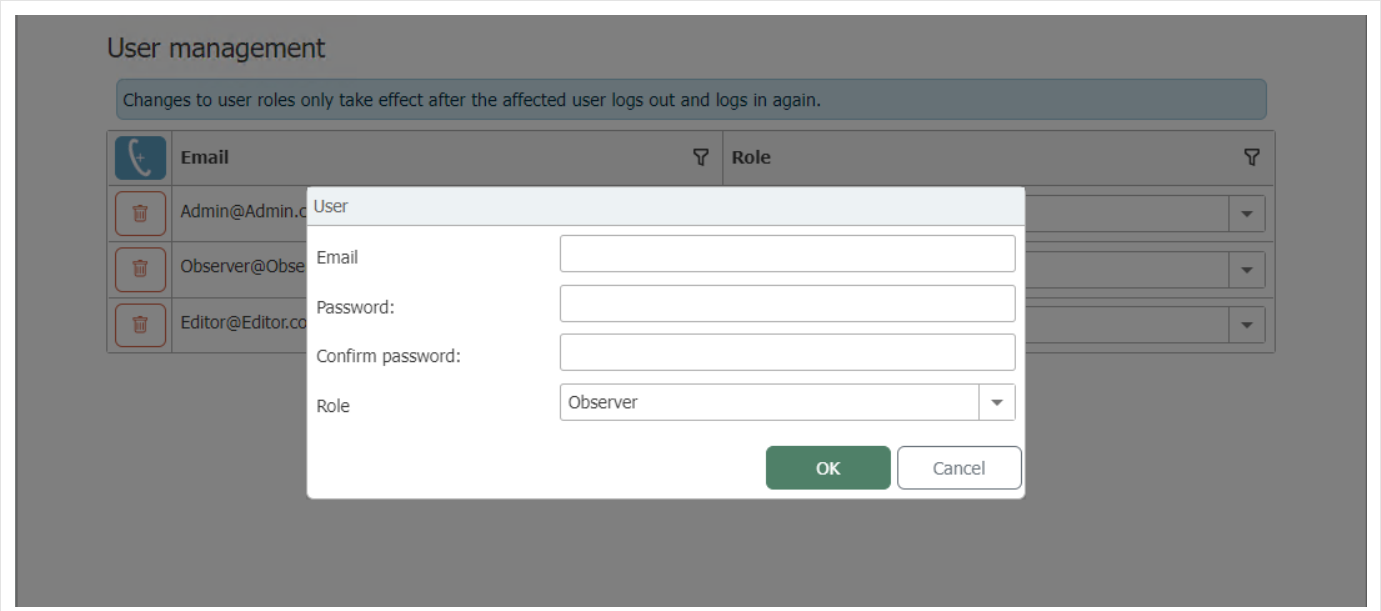
User management

In the user management, internal users can be created, deleted, and edited. External users, such as Windows Authentication users and Microsoft Entra ID users (or Azure Ad), are exempted from this process if these integrations have been set up during installation.

Once the initial user creation is done, it will be listed here. Although this user can be deleted or modified, after restarting the management services, it will be restored with the *Admin* role to prevent complete lockout from the management.



New internal users can be created by clicking on the plus sign at the top left of the table, which opens a new dialog.



Here, email address and password must be provided, with only one user per email address permitted. Additionally, a user role must be selected. If the dialog is confirmed by OK and all entered data is valid, the new user should be visible in the table.

User roles can be changed via the dropdown in the table, and users can be removed using the delete button.

Changes to user roles are not immediate. Existing stored login tokens (saved in browser cookies) remain valid with the old role until they are either deleted or revalidated. This happens through a logout and subsequent login.

Login Options

1. **Email Address and Password:** The classic login option using an email address and a personal password is a widely used and familiar method. Users must enter their email address and associated password to access the OPC Router. This method provides a simple and direct form of authentication.

2. **Windows User Integration:** For integration into existing Windows environments, the OPC Router supports login with Windows user accounts. This allows users to authenticate using their existing Windows credentials without needing separate access credentials for the OPC Router. This option promotes easy and efficient usage in enterprises with a Windows-based IT infrastructure.
3. **Azure AD / Entra ID:** For organizations utilizing cloud-based identity services, the OPC Router offers the option to log in via Azure Active Directory (Azure AD) or Entra ID. This option requires prior configuration during the installation phase of the OPC Router and enables secure and centralized management of user identities and access rights within a cloud infrastructure.

Security Notes:

When accessing the OPC Router from another PC, we strongly recommend enabling the use of HTTPS to encrypt data transmission and ensure communication security. Details on activating HTTPS can be found in the OPC Router installation guide. This measure is particularly important when transferring sensitive data or when accessing via insecure networks, such as the internet.

User Roles

Starting from version 5.2.0, OPC Router 5 provides user authorization through user roles.

Users from older versions are automatically assigned the *Admin* role.

Currently, there are three possible user roles: Observer, Editor, and Admin.

PERMISSION	OBSERVER	EDITOR	ADMIN
View Runtime Status	true	true	true
Manage Connections	false	true	true
Manage Connection Templates	false	true	true
Manage Connection Template Instances	false	true	true
Manage Plugins	false	false	true
Download Log Files	true	true	true
Generate Diagnostic Reports	false	true	true
Publish to Runtime	false	true	true
Control Runtime Service	false	true	true
Import Projects	false	false	true
Export Projects	false	true	true
Manage General Options	false	false	true
Manage License	false	false	true
Manage Database Options	false	false	true
Manage Redundancy Settings	false	false	true
Manage StoreAndForward	false	false	true
Manage Certificates	false	false	true
Manage Dynamic Instance API	false	false	true
Manage Dependency Store	false	false	true
Manage Notifications	false	false	true
Manage Users	false	false	true
Access Expert Mode	false	false	true

Versioning

Versioning of Configurations

Versioning of configurations is essential to make the operation of the OPC Router secure and traceable in enterprise environments.\ It offers several advantages:\ **Traceability**: All changes to the system configuration are documented, so you can always see who made what changes and when.\ **Security:** In the event of problems or misconfigurations, it is possible to quickly restore an earlier, stable version.\ **Compliance:** Many industries have regulatory requirements that require documentation and traceability of system changes.

Integrated Versioning in the OPC Router

The OPC Router supports Git internally and uses YAML configuration files, providing effective configuration management and versioning. This built-in versioning provides an out-of-the-box solution for tracking changes and rolling back to previous versions.

Benefits of integrated versioning

Ease of use: No additional tools or setup required.\ **Automatic versioning**: Every change is automatically versioned and documented.\ **Fast Recovery**: If necessary, it's easy to revert to a previous version.

Extensions through integration with hosted Git servers

Integrated versioning can be made even more powerful by connecting to hosted Git servers such as GitHub, Bitbucket, or Microsoft Azure DevOps. These external repositories provide additional features such as centralized management, advanced access control, and comprehensive change tracking.

Implementation Strategies Integration with Hosted Git Servers

Benefits: Centralized management, access control, and comprehensive change tracking. **Implementation:** Connect the OPC Router's internal Git system to a hosted Git server (e.g., GitHub, Bitbucket, Microsoft Azure DevOps).

It is important to ensure that the repositories are private to prevent unauthorized access.

Automated versioning

Supported by the OPC Router: Automatic versioning and change tracking directly in the OPC router.\ **Automated Backups**: Set up automated backup processes to regularly back up configuration files.

Best Practices

To take full advantage of versioning, the following best practices should be followed:

Commits: Commit the current configuration to the local Git repository on a regular basis.\ **Pushes**: Push the commits to the hosted Git server on a regular basis, e.g. daily or after every significant change. This can be done manually using a Git client such as Git Console, GitHub Desktop, or SourceTree.\ **Keep change logs**: Keep a detailed log of all changes made, including the date, time, and name of the person responsible.\ Use tools that can automate this logging.\ **Test environment**: Implement a test environment where changes can be validated before being deployed to production.\ Use staging environments to gradually introduce changes and monitor their effects.\ **Security of secrets**: Secrets or passwords should never be stored in the Git repository. The OPC Router always stores secrets in a separate local vault to ensure that they are not versioned.

Practical Implementation Integration with Hosted Git Servers

Setting up the Git server: Set up a hosted Git server (e.g., GitHub, Bitbucket, Microsoft Azure DevOps) to manage the configuration files. Make sure the repositories are private.\ **Connect the OPC Router**: Configure the OPC router to use the hosted Git server for versioning.\ **Management and Monitoring:** Use the Git server to manage, track, and monitor all configuration changes.

Automated versioning

Enable in OPC Router: Enable automatic versioning and change tracking in the OPC Router.\ **Automated Commits:** Set up

commit routines to back up configuration files on a regular basis.\ **Push Processes**: Manually or automatically push commits to the hosted Git server.\ Monitoring: Regularly checking and maintaining version control systems to ensure integrity and availability.

Summary

Configuration versioning is an essential part of operating the OPC Router in enterprise environments.

It ensures that changes are traceable, increases security through easy rollback, and meets regulatory requirements.

By leveraging the OPC Router's built-in versioning and extending it with hosted Git servers, organizations can significantly improve the reliability and maintainability of their OPC Router instances.

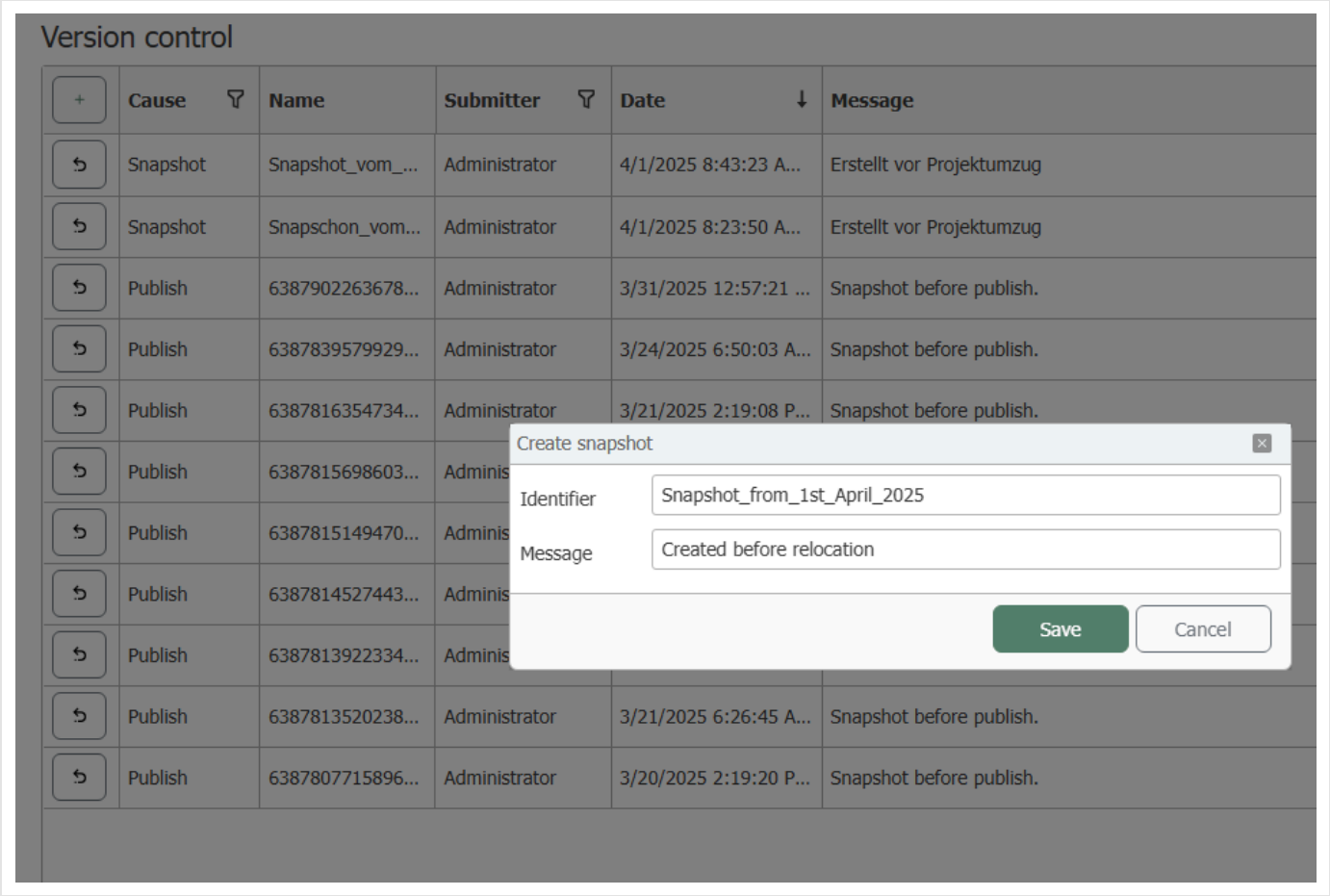
Using versioning

To use versioning in the OPC Router, navigate to the settings and click on the “Versioning” option.

The various snapshots are now displayed in the OPC Router.\ As soon as the project is published, a new snapshot is automatically created.\ At the top left, you can create a new snapshot using the button![](</assets/image (40).png>).

The button is only active if there are changes in the project.

Enter a name for the snapshot and a message as a comment to help you identify the snapshot in the future.



The arrow symbol ![](</assets/image (42).png>) shows you which status is currently loaded.\ If no entry with an arrow symbol is displayed, it means that no snapshot has been loaded yet.

You can use the return symbol ![](</assets/image (43).png>) to select and restore any of the available snapshots.

Plug-ins

The OPC Router implements communication with a wide variety of systems. This implementation is carried out by the extensions "Plugins", "Add-ons" and "Data transformations (ETL tools)". Each extension is configured according to the specific requirements of the target system. This variety of connections allows you to collect and merge data and make the information available in the right place at the right time.

The modular OPC Router server license is valid for any number of clients, data points and connections and pays off even when using just a few functions.

Connect your systems with each other without programming, automate processes and implement your Industry 4.0 system integration.

Cloud / IoT

To communicate with devices, you can use these plug-ins. Among other things, they are used with IoT, "Internet of Things".

Apache Kafka

MQTT

REST

Apache Kafka

The Apache Kafka messaging protocol allows data to be transferred between devices.

You can use the Apache Kafka plug-in to generate queries from message log connections.

This operation is standard:

1. Open the plug-in by clicking on "Apache Kafka" in the "Plug-ins" menu.
2. Click on the plus button or double-click on an existing entry.
3. Make the necessary settings (see section “Plug-in”).
4. Confirm with “OK”.

The Apache Kafka connection is now available for configuring a connection.

Use the “Check connection” button to check your connection.

Apache Kafka Plug-in

You set the following properties in the configuration of the Apache Kafka plug-in:

NAME	ASSIGN A UNIQUE NAME TO THE APACHE KAFKA CONNECTION.
Broker connection	Enter the address of the broker. The port in the address is the port on the server that is used for the connection with Apache Kafka.
SSL	Verwendet SSL-Verschlüsselung. Der Server muss dies unterstützen, der entsprechende Port eingestellt sein.
Client ID	Display of the Client ID. The button „Generate“ creates a new ID.
Group ID	Display of the Group ID. The button „Generate“ creates a new ID.
User	Username for logging in to the server.
Passwort	Password for logging in to the server.
SASL Mechanism	Used for authentication. SASL authentication is supported over both unencrypted connections and TLS connections. You have the following options: • **PLAIN:** Authentication with user and password. They are stored in the local Kafka configuration. • **SCRAM-SHA-256/512:** credentials are stored centrally in ZooKeeper. This option can be used when ZooKeeper cluster nodes are running in an isolated private network. • **GSSAPI:** Authentication at a Kerberos-Server.
Client certificate	Selection of a client certificate from the drop-down list, which the OPC Router also sends.
Trusted certificates	Select which certificates you want to accept. For example, only Windows certificates or only router certificates.
Connection timeout (ms)	Timeout in millisecond before the transfer gets cancelled.
Enable GZip	Enables zipping the messages. You can also define the compression level.
Wait for transmission result	Only acknowledge the message to the broker if the entire transfer was OK.
Consume interval (s)	Interval for the trigger to check for changed values of the topic.

Apache Kafka Transfer Object

To enable a connection proceed as follows.

![Kafka-Transferobjekt](</assets/en (1) (2).png>)

After double-clicking the transfer object, set the following properties:

BROKER	**SELECT A CONFIGURED BROKER CONNECTION FROM THE DROP-DOWN MENU.**
Topic	Enter the topic.
Dynamic topic	Activate if different topics are required. Topic can be defined variably via Input.

Confirm your entries with „OK“.

Apache Kafka trigger

This trigger fires when the Apache Kafka topic changes. Returns the payload and key.

![Apache Kafka-Trigger](</assets/en (2) (2).png>)

The individual properties are:

BROKER	**SELECT A CONFIGURED BROKER CONNECTION FROM THE DROP-DOWN MENU.**
Topic	Enter the topic.
Wait for transfer	Select whether and in what form to wait for the transfer.
Trigger at service start	Activate if the Router should trigger this topic if the OPC Router-service gets started.

"Snowflake AI Data Cloud" plugin

The **Snowflake plugin** enables bidirectional data exchange between OPC Router and the cloud data platform Snowflake. Data can be written to Snowflake as well as read from Snowflake. Various methods are available to specifically control costs, performance and latency.

Architecture and mode of operation

All operations are always executed via a **Snowflake Warehouse**. The warehouse handles the processing and calculation of all SQL queries and write operations.

Cost note: Every read or write operation via the warehouse incurs **Snowflake credits**. This also applies when *Auto Resume / Suspend* is enabled – costs are incurred as soon as the warehouse is active.

For **pure ingest scenarios** without queries, a **cost-optimised alternative** via Snowpipe is recommended.

Cost-optimised ingest variants (write-only)

For write-only scenarios, in which the OPC Router only inserts data into Snowflake, there are two options available:

1. REST plug-in → Snowpipe

The OPC Router can send JSON or CSV data to a Snowpipe endpoint via the REST plug-in. Snowpipe loads this data asynchronously into the target tables without keeping a warehouse active.

Advantages

- No active warehouse required
- High cost efficiency for continuous ingest
- Supports event-driven transfers (e.g. from production systems)

2. Cloud File Access plug-in → S3 / Stage

Alternatively, OPC Router can write data directly to a cloud stage (e.g. Amazon S3). Snowpipe or a copy command in Snowflake then automatically reads the data.

Advantages

- Batch processing of larger data volumes
- Ideal for periodic or buffered writing
- No permanent warehouse utilisation

Bidirectional use

The Snowflake plug-in not only supports writing, but also reading and returning data. This allows closed-loop scenarios (e.g. feedback from cloud analyses back to machines) to be implemented.

Two common approaches are available:

1. Trigger via transfer state column

The Snowflake transfer status trigger initiates a transfer when a value in the monitored table is marked as not yet transferred. Once the transfer is complete, the record is marked as transferred or failed. Failed transfers can be repeated as often as desired using an additional counter.

Process

1. The trigger monitors a defined table in the database.

2. A data record is marked as not transferred (e.g. with 0) or already added as such.
3. As soon as such a data record is detected, a transfer is triggered.
4. After successful transfer, the data record is marked as transferred (e.g. 1) or failed (e.g. 2).
5. Failed transfers can optionally be repeated multiple times, controlled by a repeat column and number.

Advantages

- Automated transfer process based on status values.
- Repeat mechanism for failed transfers increases reliability.
- Flexible configuration of status values and repeat logic.
- Option to limit the transfer quantity per run.

Notes

- The monitored table must have a single primary key.
- The status column and repeat column must be initialised with 0 or the configured value – NULL values are not permitted.
- Sorting and filtering of the data records to be transferred can be configured.

2. Trigger via data change in table

Alternatively, the OPC Router can detect data changes by evaluating a column such as updated_at or sequence_id.

Process

1. Periodic polling for changed data records.
2. Processing of new or changed data records.
3. Storage of the last checkpoint in the router.

Advantages

- No additional status field required
- Easy to integrate into existing structures

Notes

- The monitored table must have a single primary key.
- Unique sorting (e.g. ORDER BY updated_at, id) recommended.
- Duplicate detection via hash or unique ID.
- Select small batch sizes for high change rates.
- Optionally, a transfer can be triggered directly when connecting to the database.

Best practices

- Choose warehouse size carefully: smaller clusters for continuous streams, larger ones only for batch loads.
- Make active use of suspend: automatic deactivation significantly reduces running costs.
- Outsource asynchronous ingest if real-time is not required (Snowpipe, REST).
- Version schema changes to ensure stable transfers.

SCENARIO	METHOD	WAREHOUSE REQUIRED	COST-OPTIMISED	RECOMMENDED FOR
Synchronous inserts/selects	Warehouse connection	□	□	Validations, small amounts of data
Asynchronous ingest (REST)	REST plug-in → Snowpipe	□	□	Event/log data, edge → cloud

SCENARIO	METHOD	WAREHOUSE REQUIRED	COST-OPTIMISED	RECOMMENDED FOR
Batch ingest (file upload)	Cloud File Access → S3/Stage	□	□	Large data volumes, periodic uploads
Read data from Snowflake	SQL trigger, state or timestamp column	□	□	Retransfer or data feedback

See also

- Snowflake plugin
- Snowflake transfer objects
- Snowflake triggers
- REST Plug-in
- Cloud File Access Plug-in

"Snowflake AI Data Cloud" plugin

↑Snowflake

- Connection settings
- Advanced connection settings
- Advanced

Connection settings

The following entries must be made in the "Snowflake AI Data Cloud" plugin to successfully connect to Snowflake

INPUT FIELD	DESCRIPTION
Account identifier	The account identifier that can be read in Snowflake
User	The user name used to log in to Snowflake
Auth input	*Password** or *Private key pair***
*Password	If *Password* is selected: The password used to log in to Snowflake
**Private key file	If *Private Key Pair* is selected: Select private key file
**Private Key Password	If *Private Key Pair* is selected: Password for the selected file
Database	Select an existing database or manually enter a future database
Schema	Optional selection of a schema

Advanced connection settings

The following connection parameters can also be entered or changed in the ‘Snowflake AI Data Cloud’ plugin. All settings on this page are optional.

INPUT FIELD	DESCRIPTION
Host	Specify the server if it differs from the Snowflake standard.
Allow underscores in host	Select if underscores are to be allowed in the host.
Role	Select a role or manually enter one that will exist in the future.
Connection timeout	If the connection to Snowflake cannot be established, the predefined waiting time expires before the connection attempt is considered unsuccessful.
Retry timeout	Timeout for retries.
Polling interval	Time interval between cyclical status requests to Snowflake

Advanced

The following advanced settings can also be entered or changed in the ‘Snowflake AI Data Cloud’ plugin. All settings on this page are optional.

INPUT FIELD	DESCRIPTION
File transfer memory treshold	Maximum permissible storage limit for data transfers, specified in bytes
QueryTag	Optional tagging of SQL queries
Validate default parameters	Enable validation
Pooling enabled	Enable pooling.
Keep alive	Enable keep alive.

Snowflake transfer objects

Snowflake databases are available to you as both sources and destinations. The required database must be configured as a plug-in instance. Create one of the Snowflake transfer objects within the connection as described. The following transfer objects are available:

- **Snowflake Select**: Queries one or more records (from a table or view). You must define filter and sort criteria accordingly. The transfer object thus serves as a data source.
- **Snowflake Insert**: During data transfer, a new record is inserted into the table. The transfer object thus acts as the transfer destination. Incomplete arrays are filled in during a database insert. Example: If a single entry from a timestamp and several entries from an array are to be read into a database, the timestamp can be used for the individual variables of the array. The timestamp is therefore automatically used multiple times until all variables have been read. Therefore, the following applies: During an insert, the highest InsertCount is always determined. Entries that have a smaller count are then extended by a factor of X so that the values in the table are filled in. Also selectable for views.
- **Snowflake Update**: During data transfer, one (or more) existing records in the table are updated. You must define a filter accordingly. The transfer object thus acts as the transfer target. Also selectable for views.
- **Snowflake Delete**: The data transfer deletes one or more records from the table. You must define a corresponding filter. Also selectable for views.
- **Snowflake Stored procedure**: Triggers a stored procedure as soon as the trigger condition is met. You can use input parameters and return values as targets or sources here. Select the required TO according to your requirements and set the properties queried in the respective window:

PROPERTY	INSERT	UPDATE	SELECT	DELETE	STORED PROCEDURE	EXPLANATION
Data source	✔	✔	✔	✔	✔	Data source in which the following table, view or stored procedure can be found
Table	✔	✔	✔	✔		Table on which the TO-dependent operation is to be performed. Table on which the selected operation is performed, or name of the stored procedure. **Note**: The specified table must have a primary key. Tables without a primary key are not supported.
Stored Procedure					✔	Stored procedure to be executed
Columns/Parameters	✔	✔	✔		✔	Select the columns/parameters that are to be connected to the TO as sources or targets
Query result set					✔	If return values are expected, they can be read out with ‘Query result set’ and can then be selected to offer return values as a source at the TO.
Filter		✔	✔	✔		Configure a filter that suits your requirements to narrow down which data is to be selected/updated/deleted. There are options for linking the conditions using AND and OR operators. An explanation of the various filter operators can be found here.

PROPERTY	INSERT	UPDATE	SELECT	DELETE	STORED PROCEDURE	EXPLANATION
Filter/Filter group Edit		✔	✔	✔		The possible entries change depending on which element is selected under 'Filter'. **Link**: Defines the AND and OR operators between the filters/filter groups. **Column**: Column to be filtered **Comparison operator** for value comparison
Expected number of incoming data	✔					Optional: A check is performed at runtime to determine whether the appropriately configured expected number of records was provided for the insert.
Query			✔			Configurations for record count, sorting and marking. Details in the chapter Select-TO-Tab 'Query'

Important: If 'Stored Procedure' is selected as the type, the OPC Router calls the procedure immediately once with default parameters (String = NULL; Int = 0, etc.) to read the structure of the result set. A rollback is performed. The procedure must therefore be transaction-safe and must not return any errors during the standard call.

Note: If the type is set to 'Update', the transfer object provides the number of updated rows (as the data source).

Special features in Select-TO

Select-TO tab 'Query' {#select-to-query}

Sort order	Ascending or descending.
Sort by column	Column by which the query result is sorted.
Do not change records	(only possible for queries from tables) No changes are made to the transferred records.
Mark records after transfer	(only possible for queries from tables) In a corresponding column of the source table, '1' is entered for each transferred record and '2' for each faulty transfer. **Note:** A simple primary key on the numeric column 'ID' is required for this in the relevant table.
Column for marking	(only possible for queries from tables) Column in which the marker for the successful or faulty data transfer is entered. The column should be automatically filled with '0' as the default value.
Marking values	Here you can specify the values to be written in the marking column. The marking values must be unique.

Count failed transfers	Counts the failed transfers. **Note:** Under ‘Error column’, select the column in which the number of failed transfers is to be counted up.
Delete records after successful transfer	(only possible for queries from tables) Of course, transferred records can also be deleted from the table. **Caution**: The records are irrevocably deleted from the source table.
Cancel transfer (if query result is empty)	Cancels the data transfer if the query result is empty.
Trigger read error (if query result is empty)	Cancels the data transfer if the query result is empty and triggers the error log (in the router status and the log file).
Return empty data record	Returns an empty data record if the query result is empty.
Limit number of data records to	Only transfers the number of data records specified here. The sorting must be set accordingly. Example: Only the most recent data record is to be transferred from a query result. The sorting would have to be set to a field with a ‘timestamp’ in descending order, and the number of data records would have to be limited to ‘1’.

Mark transferred data records

Mark transferred data records. Provide columns in your tables in which OPC Router can mark transferred data records (‘transfer flag’ not transferred, transferred, transfer error). The transfer flag columns should be assigned the default value ‘0’ and must not accept NULL so that correct assignment to the three statuses is always guaranteed.

Sort select results

Sort the table in ascending order by the transfer flag. This way, the non-transferred data records are transferred first; faulty individual transfers are only retried at the end of the transfer. For performance optimisation, you can filter by the transfer flag column (transfer flag = 0 or = 2).

Snowflake triggers

↑Snowflake

- Snowflake transfer status trigger
- Snowflake data change trigger

Snowflake Transfer Status Trigger

The Snowflake Transfer Status Trigger initiates a transfer when a value in the monitored table is marked as not yet transferred.

Once the transfer is complete, the record is marked as transferred or failed. Failed transfers can optionally be retried as many times as desired using an additional counter.

For the Snowflake Transfer Status Trigger to work, the monitored table must have a primary key.

Properties

General tab

TERM	EXPLANATION	
Data source	Database plug-in instance to be used.	
Table	Table to be monitored.	
Marking Column	Column in the table where the transfer status is stored. The column must be initialised with 0 or another configured value for "Not transferred". It does not work with NULL.	
Marking values	Here you can set your own values for the different transfer statuses.	
Repeat failed transfers	If this option is active, failed transfers are repeated as often as specified.	
Number of retries	Specifies how many times failed transfers should be repeated. **Note:** The column must be initialised with 0. It does not work with NULL.	
Delay on delay	Time interval before a transfer is retried.	
Sort order	Here you can set how the values are to be sorted. If "None" is selected, no sorting is performed.	
Sort column	The column to be sorted.	
Limit number of records to	If this option is enabled, you can set the maximum number of records that can be transferred at one time.	
Do not use change notifications	If this option is enabled, the polling interval specified in the plug-in will be used for queries even for databases with change notifications.	

Filter tab

Here, filters and filter groups can be created and configured graphically. The SQL statement is constructed from the tree. Unlike with the DB transfer object, the filter values must be entered here. The "Add" button is used to add filter criteria and the "Add Group" button to add filter groups. The button with the minus sign deletes the respective filter or filter group. **Filter** A filter consists of three elements:

- **Column**: The column to which the filter refers.
- **Filter operator**: The operator determines how the filter value is used to filter the records.
- **Filter value**: The value used to filter the records.

An explanation of the different filter operators can be found here.

"AND"/"OR" Clicking on the `AND` or `OR` text changes the type of group to the other.

- `AND` means that **all** conditions in the group must be met for the group as a whole to apply.
- `OR` means that **one** condition in the group must be met for the group as a whole to apply.

"Columns" tab

Here you can select the columns whose values are to be used in the connection. The columns can be selected and deselected either by double-clicking or by using the buttons between the available and used columns.

Snowflake Data Change Trigger

The Snowflake Data Change Trigger monitors query results and triggers a transfer when changes occur. To do this, a select query is defined in the Snowflake Data Change Trigger. The columns to be monitored are specified in the "General" tab. The "Filter" tab allows you to define more precisely when a transfer should be triggered. For example, a transfer should only be triggered when a column has a specific value.

Properties:

"General" tab

TERM	EXPLANATION
DB access data	The configured instances of the Snowflake plug-in are available for selection here.
Table	Table from which columns are to be monitored. **Note:** The specified table must have a primary key. Tables without a primary key are not supported.
Available	All available columns in the selected table that have not yet been selected for the "Used" area. Select the columns to be monitored.
Used	All columns of the selected table that have already been selected. You can add or remove fields using the arrows between the "Available:" and "Used:" areas or by double-clicking.
Trigger on connect	If this option is set, a transfer is triggered when the router connects to the database.
Do not use change notifications	If this option is set, the polling interval configured in the plug-in is used for querying instead of the change notifications.

"Filter" tab

Here, filters and filter groups can be created and configured graphically. The SQL statement is constructed from the tree. Unlike with the DB transfer object, the filter values must be entered here. Use the "Add" button to add individual filter criteria and the "Add Group" button to add a filter group. The button with the minus sign deletes the respective filter or filter group.

Filter A filter consists of three elements:

- **Column**: The column to which the filter refers.
- **Filter operator**: The operator determines how the filter value is used to filter the records.
- **Filter value**: The value used to filter the records. An explanation of the various filter operators can be found here.

"AND"/"OR" Clicking on the `AND` or `OR` text changes the type of group to the other.

- `AND` means that **all** conditions in the group must be met for the group as a whole to apply.
- `OR` means that **one** condition in the group must be met for the group as a whole to apply.

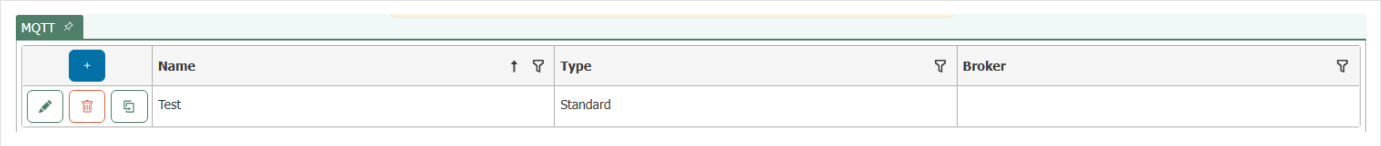
MQTT

The MQTT machine-to-machine messaging protocol allows telemetry data to be transferred between devices.

You can use the MQTT plug-in to generate queries from message log connections.

This operation is standard:

1. Open the plug-in by clicking on “MQTT” in the Plug-ins menu.



1. Click the plus button or double-click the existing entry.
2. Make the necessary settings (see “Plug-in” section).
3. Confirm with “OK”.

The MQTT connection is now available for configuring a connection.

Use the “Check connection” button to check your connection.

MQTT Plug-in

The following properties can be set in the configuration of the MQTT plug-in. Legacy / Migration Up to version **5.4**, the MQTT implementation could be selected in the **Expert setting** → **Implementation** area. The following options were available:

- `Legacy Automatic`
- `Legacy Strider Mqtt`
- `Legacy MQTTnet`
- `MQTTnet` (recommended) Starting with version **5.5**, a **migration note** appears for old configurations, recommending switching to **MQTT .NET V2**. Starting with version **5.6**, only **MQTT .NET V2** is available.

Tab: Router Settings

FIELD	DESCRIPTION
Name	Assign a unique name to the MQTT connection.
Cloud Profile	Select the desired profile: **MQTT Standard**, **AWS**, **Azure IoT**, **Wago Cloud**.

Test Connection can be used to check the current connection.

Cloud profiles

The cloud profiles extend the standard configuration with additional tabs:

- AWS IoT Cloud
- Azure IoT Hub

For more information on using MQTT and the cloud profiles, please refer to our Knowledge Base: ** [https://www.opc-router.de/knowledge-base/**](https://www.opc-router.de/knowledge-base/)

Tab: Connection settings

Connection details

FIELD	DESCRIPTION
Broker connection	Enter the address of the MQTT broker.
Port	TCP port of the broker (default: 8883).
MQTT version	Select the protocol version (e.g., MQTT v5 or MQTT v3.1.1).
Quality of Service type	Select the default QoS: **0** – maximum once, **1** – minimum once, or **2** – exactly once.

info Performance For details on latency times, network round trips (RTT), and internal synchronization during publish operations, see MQTT Performance.

Authentication

FIELD	DESCRIPTION
Username	Username for logging in to the broker.

FIELD	DESCRIPTION
Password	Selection of a secret. The buttons can be used to create, edit, or delete a secret.
Client ID	Unique client ID. Supports placeholders: `${hostname}`, `#[INSTANCE]`, `# [INSTANCE:<numeric value>]`, `#[RAND:<numeric value>]`. This is helpful for **connection pooling** or **project synchronization** to automatically generate different IDs per instance. Example: `MyInstance-${hostname}-#[RAND:4]`
Buttons	**Random** – generates a random ID. **Client ID Template** – inserts the default template into the input field.

Enable TLS/SSL

FIELD	DESCRIPTION
SSL protocol	Select from supported protocols (e.g., TLSv1.2, TLSv1.3).
Client certificate	Select a client certificate (or *Without*).
Trusted certificates	Specifies which certificates are accepted (e.g., *All (unsafe)*, *Windows*, *Router*).

Tab: Storage

FIELD	DESCRIPTION
Enable MQTT Data Storage	When enabled, the **last received MQTT message** for each topic is stored on the hard disk. This feature is required to use the **MQTT Storage Read Transfer Object**.
Pattern	A subscription is created for each topic pattern added. Supports wildcards `+` and `#`.
Note	Free text to describe the purpose or origin of a topic.

info How it works When the **MQTT Data Storage** function is enabled, the OPC Router creates automatic subscriptions for all entered topics. The received messages are stored persistently – **only the last message per topic** (old values are overwritten). These stored values can be read out again using the **MQTT Storage Read transfer object**.

Tab: FirstWill

FIELD	DESCRIPTION
Enable First Will	Enables the sending of a First Will when establishing a connection.
Retain First Will	Broker stores message for later delivery.
Topic	Target topic for the First Will.
Payload	Content of the First Will message.
Quality of Service	Service quality (0, 1, or 2).

Tab: Last Will

FIELD	DESCRIPTION
Enable Last Will	Activates the sending of a message when the connection is terminated.
Retain Last Will	Broker stores message for later delivery.
Send Last Will on disconnect	Sends Last Will even when disconnecting manually.
Topic	Target topic for Last Will.
Payload	Content of Last Will message.
Quality of Service	Service quality (0, 1, or 2).

Tab: Advanced

Connection Pool

FIELD	DESCRIPTION
Enable Connection Pooling	When enabled, one connection is used for subscriptions and multiple connections are used for publish operations. Useful for high message volumes.
Min. connections	Minimum number of connections.
Max. connections	Maximum number of connections.
Connection idle timeout	Time in minutes after which inactive connections are closed.

Subscription/Trigger

FIELD	DESCRIPTION
Wait for transfer	Router only responds to the broker after the transfer has been completed or failed.
Enable message deduplication	Prevents multiple triggers for identical messages within a time window.
Deduplication time window	Time in seconds during which identical messages are ignored.

Session options

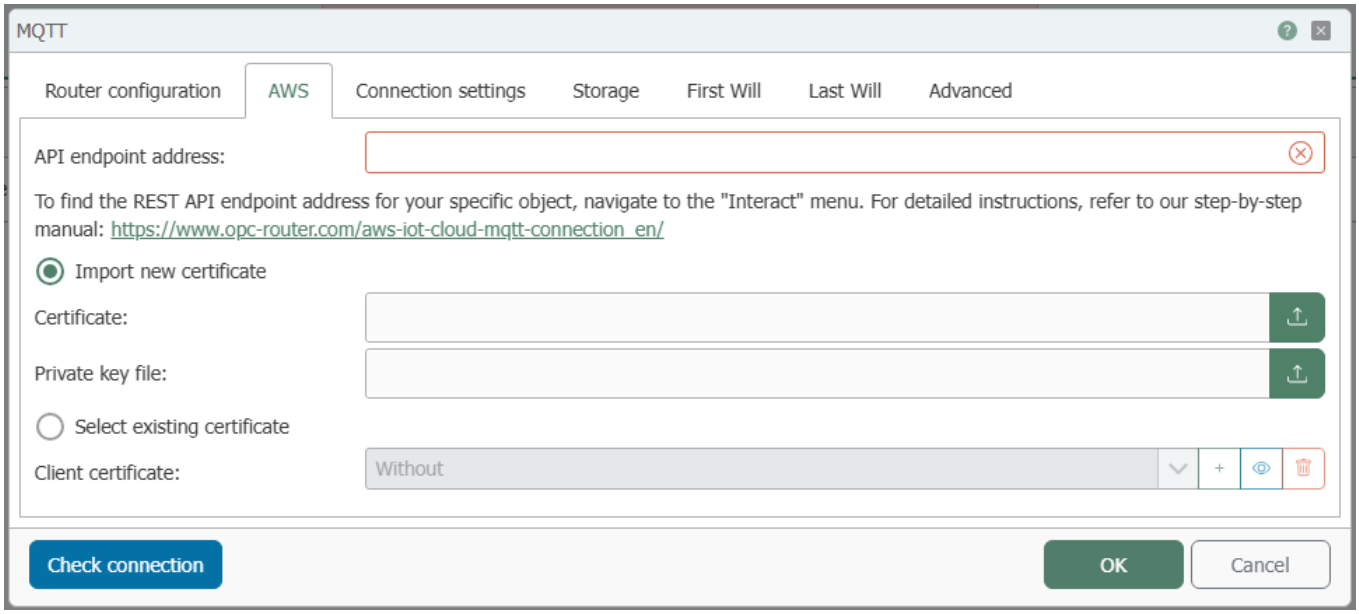
FIELD	DESCRIPTION
CleanSession	Starts a new session without stored messages (disable for persistent sessions).
Keep alive (s)	Ping interval in seconds.
Timeout (ms)	Waiting time in milliseconds for reconnection.

MQTT Cloud Profile: AWS IoT

This chapter describes the **additional "AWS" tab** of the MQTT plug-in. All general MQTT settings (standard profile) can be found under **MQTT Plug-in Configuration**.

> Principle: The AWS tab supplements the standard configuration with fields for endpoint and client certificates. The other tabs (Connection settings, Storage, First Will, Last Will, Advanced) are configured as in the standard profile.

Tab: AWS



FIELD	DESCRIPTION
API endpoint address	REST API endpoint of the AWS IoT object (Thing). You can find the value in AWS IoT under **Manage → Things → (Your Thing) → Interact**.
Import new certificate	Imports a **new** client certificate with associated **private key** into the OPC Router's certificate management.
Certificate	File selection for the **device certificate** (`*.crt` / `*.pem`).
Private key file	File selection for the **private key** (`*.key`).
Select existing certificate	Uses a **previously imported** client certificate from certificate management.
Client certificate	Selection of the client certificate to be used (or *Without*).
Check connection	Checks the connection with the currently set values.

Note The link note in the dialog leads directly to the **[Knowledge Base article]**. In AWS IoT, you can find the **API endpoint** on the **Interact** page of the selected thing.

Prerequisites in AWS IoT Core

1. **Create thing**: *Manage* → *Things* → *Register a thing* → *Create a single thing* and name it.
2. **Create/activate certificates**: *Create certificate* → Download **certificate (CRT/PEM) and private key (KEY)** → **Activate**.
3. **Create and assign policy**: *Secure* → *Policies* → Create policy (e.g., `iot:*` to `*` as starting point) and **Attach** the policy to the certificate.

> The exact policy should correspond to your security concept. Restrict actions/ARNs as far as possible.

Manage certificates in OPC Router

1. **Import certificate**: *Tools* → *Settings* → *Certificate Management* → *Client Certificates* → *Import*.
2. **Assign private key**: Select imported certificate → *Add private key* → Select KEY file.

3. Check certificate: Check display in certificate list.

Configuration in the MQTT plug-in (AWS tab)

1. Open **Plug-ins** → **MQTT** and create/edit an MQTT connection.
2. **Router settings**: Assign a name, select **Cloud profile: AWS**.
3. **AWS**:
4. Enter **API endpoint address** (copied from *Interact*).

- Select **Select existing certificate** or use *Import new certificate* and specify the certificate + key.
    - Run **Check connection**.
- **Connection settings / First Will / Last Will / Advanced**: Configure as required (as in the standard profile).

5. Save with OK.

Typical settings / notes

- **Port & TLS**: AWS IoT uses TLS; **Enable TLS/SSL** should be active in *Connection settings*.
- **Client ID**: Use **placeholders** for parallel instances (e. g. `MyInstance-${hostname}-#[RAND:4]`).
- **QoS**: See **MQTT Performance** for effects on latency and execution time.

* Storage: For “last value” queries, enable MQTT Data Storage and define the desired patterns. Read about MQTT Storage Read Transfer Object.

Troubleshooting

SYMPTOM	CAUSE / SOLUTION
Handshake/Authentication failed	Certificate/key not imported correctly or wrong certificate selected. Check certificate management and re-import if necessary.
Access denied	Policy missing or too restrictive. Check in AWS IoT under **Secure** → **Policies** and **Attach** to the certificate.

SYMPTOM	CAUSE / SOLUTION
No route to host / Timeout	Endpoint incorrect, port/firewall blocked or TLS disabled. Recopy endpoint from **Interact**, check TLS/port.
Duplicate Client ID	Multiple instances with the same client ID. Use a placeholder or **Random**.

See also

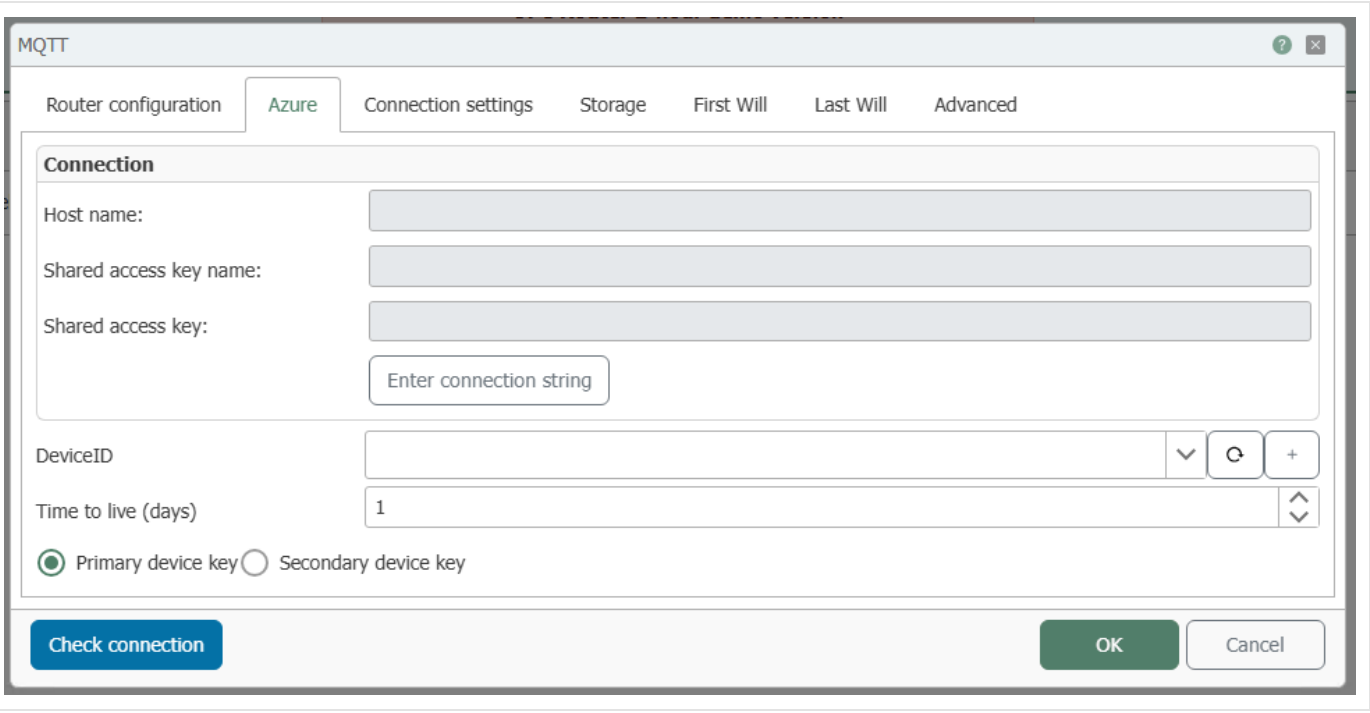
- **Main configuration**: MQTT Plug-in Configuration
- **Performance/RTT/QoS**: MQTT Performance
- **Storage Read**: MQTT Storage Read Transfer Object

MQTT Cloud Profile: Azure IoT Hub

This chapter describes the **additional "Azure" tab** of the MQTT plug-in. All general MQTT settings (default profile) can be found under **MQTT Plug-in Configuration**.

> Principle: The Azure tab supplements the standard configuration with fields for authentication via shared access keys (SAS tokens) and device IDs for secure access to the Azure IoT Hub.

Tab: Azure



Quick start (according to tutorial) Recommended introduction according to step-by-step instructions: **Enter connection string**, **Select/create device**, **Check connection**. For details, see **[Knowledge Base article]**.

FIELD	DESCRIPTION
Host name	Host name of the Azure IoT Hub. Displayed in Azure in the *IoT Hub → Overview → Hostname* area (e.g., `myhub.azure-devices.net`).
Shared access key name	Name of the policy (e.g., `iothubowner` or user-defined policy) from the Azure IoT Hub.
Shared access key	The corresponding key for the selected policy. Displayed in Azure under *Shared access policies*.
Enter connection string	Opens a pop-up window for direct entry of the complete connection string (`HostName=...;SharedAccessKeyName=...;SharedAccessKey=...`).
Device ID	Unique device ID in the IoT Hub. Displayed or newly created in the Azure portal under *IoT devices*.
Time to live (days)	Lifetime of the generated token in days. After expiration, a new token is automatically created from the access key.

FIELD	DESCRIPTION
Primary / Secondary device key	Select whether the primary or secondary token is used for authentication.
Check connection	Tests the connection to the specified IoT Hub and token.

Prerequisites in Azure IoT Hub

1. **Create IoT Hub:** Via *Azure Portal → Create a resource → Internet of Things → IoT Hub*.
2. **Create device:** *IoT Hub → Devices → + New→* Define device ID → *Save*.
3. **Check access policy:** Under *Shared access policies*, ensure that a policy with write permissions (`iothubowner` or your own policy) is available.
4. **Copy access data:**
5. Hostname (e.g., `example.azure-devices.net`)

- Shared Access Key Name (policy)

* Shared Access Key (token)

Certificates / Security

- Azure IoT uses TLS by default for all MQTT connections (port **8883**).
- Therefore, **TLS/SSL must be enabled** in the **Connection settings**.

* Manual certificate selection is usually not necessary, as Azure secures the connection using public root certificates.

Configuration in the MQTT plug-in (Azure tab)

1. Open **Plug-ins** → **MQTT** and create/edit a connection.
2. **Router settings**: Select the **Azure IoT** cloud profile.
3. **Azure** tab (recommended according to the tutorial):
4. Click **Enter connection string** and insert the *Primary connection string* of the desired policy (e.g., `iothubowner`).
5. Select **DeviceID** or create a new one using **+**.
    1. Run **Check connection**.
6. **Alternative**: Enter hostname / shared access key name / key manually (instead of connection string).

5. Check Connection settings / First Will / Last Will / Advanced if necessary; then save with OK.

Typical settings / notes

- **QoS:** See **MQTT Performance** for details on latency and throughput.
- **Storage:** For persistent “Last Value” queries, enable **MQTT Data Storage** (see MQTT Storage Read Transfer Object).
- **Client ID:** Should correspond to the device ID if multiple devices are connected to the same IoT Hub in parallel.

* TLS port: Default is 8883, alternatively WebSocket via port 443 (not

recommended for OPC Router).

Troubleshooting

SYMPTOM	CAUSE / SOLUTION
401 Unauthorized	Incorrect or expired SAS token. Check *time to live* and regenerate token if necessary.
Device not found	Device ID does not match Azure IoT Hub. Check in the device list.
Connection refused (Port 8883)	TLS disabled or port blocked. Check firewall and TLS settings.
Access denied (Policy)	Missing policy permission. Enable read/write access in Azure Policy.

See also

- **Main configuration:** MQTT Plug-in Configuration
- **Performance / QoS / RTT:** MQTT Performance
- **Storage Read:** MQTT Storage Read Transfer Object

MQTT Performance

This page describes the performance characteristics of the MQTT plug-in in OPC Router based on real benchmark measurements (see benchmark details).

Factors influencing performance

Main factors

- **Round Trip Time (RTT)** – Latency between OPC Router and broker. Has a direct impact on throughput and latency, especially with QoS 1 and QoS 2.
- **QoS Level** – Determines the number of acknowledgment steps between client and broker.
- **Connection Pooling** – Increases throughput for parallel transfers.
- **Broker Performance** – Internal queues, topic filters, and CPU/I/O load.

* Router Configuration – Number of transfers per second, activated options (deduplication, wait-for-transfer, etc.).

Quality of Service (QoS)

QOS	DESCRIPTION	COMMUNICATION PHASES	TYPICAL RTT	RELATIVE SPEED	TYPICAL LOCAL LATENCY
0	Fire & forget – no confirmation	1 (Client → Broker)	0.5 × RTT	**Very high**	< 10 ms
1	Receipt confirmation (PUBACK)	2 (Client ↔ Broker)	1 × RTT	**Medium**	20–80 ms
2	4-way handshake (PUBREC/PUBREL/PUBCOMP)	4 (Client ↔ Broker)	2 × RTT	**Low**	40–150 ms

Network Round Trip Time (RTT)

RTT (Round Trip Time) is the time it takes for a data packet to travel from the OPC Router to the MQTT broker and back again. It depends heavily on the network environment and directly influences the performance of MQTT communication.

ENVIRONMENT	TYPICAL RTT	DESCRIPTION
Local (same host)	0.5–2 ms	Local Mosquitto broker or Docker container on the same machine
Local area network (LAN)	1–5 ms	Broker in the same network segment or data center
Corporate network (Corp-VPN)	10–40 ms	VPN or MPLS-based site connection, additional encryption latency
Cloud (hyperscaler)	30–100 ms	MQTT broker in Azure, AWS, or Google Cloud, depending on region and peering
Wide area network (WAN)	80–250 ms	International or mobile connections, possibly via public networks

info Note

With **QoS 1** or **QoS 2**, the effective transmission time increases linearly with the *RTT*, as each acknowledgment round requires a complete round-trip time.

Connection Pooling

Connection pooling distributes publish operations across multiple parallel MQTT connections, significantly increasing the throughput rate.

PARAMETER	DESCRIPTION
Min. connections	Minimum number of simultaneously open connections
Max. connections	Upper limit for parallel MQTT connections (adjustable up to 100)
Idle timeout	Time until inactive connections are closed

Advantages

- Significantly higher throughput for many transfers
- Load distribution across multiple threads
- Avoidance of blockages with QoS 1/2 ### Recommendation | Scenario | Recommended setting | | ---------- | ---------- | | Few transfers / simple telemetry | Disable pool (single connection) | | Medium to high load | **Min. = 5**, **Max. = 50** |

| Very high load or parallel topics | Min. = 10, Max. = 100 |

Real benchmark results (local)

Measured with Mosquitto 2.0.22, OPC Router 5.5, Windows Sandbox (i7-13700, 4 GB RAM), 100 byte payload. See MQTT benchmark results

TRANSFER FLOWS	QOS	AVG. THROUGHPUT [MSG/S]	MAX. [MSG/S]	COMMENT
1	0	64	65	Base load without pool
1	1	63	65	Hardly any difference to QoS 0 (local)
10	0	640	650	Linear scaling with number of connections
10	1	616	648	Slight overhead due to PUBACK
50	0	2670	3117	Strong parallelization, slight fluctuations
100	0	4230	5286	Maximum tested rate without pool
50 (pool)	0	3025	3230	+13% increase due to pooling
100 (pool)	0	5556	6118	**Highest measured throughput**

info Note All values are from local benchmarks (RTT≈1ms). In real networks, throughput is reduced proportionally to RTT, as each QoS confirmation requires additional runtime.

Optimization recommendations

1. **QoS 0** for fast, non-critical data (e.g., telemetry, measured values)
2. **QoS 1** as standard for delivery reliability with acceptable performance
3. **QoS 2** only if absolute accuracy is required
4. Enable **connection pooling** for high parallelism or MQTT 5 batch transfers
5. Disable **deduplication** if data often remains the same
6. Only reduce **Cycle Time** (expert setting) if CPU load remains stable

7. Adjust Keep Alive/timeout values to the RTT of the network

See also

- MQTT benchmark results
- MQTT Plug-in Configuration
- MQTT Storage Read Transfer Object

MQTT Transfer Object

To enable a connection proceed as follows.

After double-clicking the transfer object, set the following properties:

Broker connection	Select a configured broker connection from the drop-down menu.
Quality of Service	Selection how often the query should be executed.
Payload data type	Selection of the data type Byte Array or String.
Payload encoding	Selection of the encoding for the MQTT query.
Topic	Enter the topic.
Retain	When activated, the query result remains visible until a new one is generated.
Dynamic topic	Activate if different topics are required. Topic can be defined variably via Input.

Confirm your entries with “OK”.

MQTT trigger

In the property mask you can set the binding parameters for your MQTT query:

If topics are used multiple times in different connections via triggers, the individual connections may be triggered more than once.

This also applies to the use of wildcards.

The individual properties are:

Broker connection	Select a configured broker connection from the drop-down menu.
Quality of Service	Selection how often the query should be executed.
Wait for transfer	Select whether and in what form to wait for the transfer.
Payload data type	Selection of the data type Byte Array or String.
Payload encoding	Selection of the encoding for the MQTT query.
Topic	Enter the topic.

REST

With the REST protocol different systems can be connected to the REST interface**.** It works on pure web technologies and on a stateless logic basis for querying and editing data in systems. It is used in many modern application platforms that allow access and data exchange for external systems via the REST interface.

The OPC Router can access and query REST WebServices. So almost any system using REST API can connect, allowing it to retrieve or transfer data from these systems. The OPC Router can also act as a **REST server** via the REST trigger and provide REST functions for calling. This allows you to set up your own **REST API** for any systems that are connected with the available plug-ins.

REST Plug-ins

When it comes to coupling systems, REST (Representational State Transfer) almost always plays a key role. Especially in connection with web-enabled systems, REST is one of the leading standards for system integration. Yet REST is by no means brand new; on the contrary, the concept behind REST is tried and tested. Connecting systems via REST is simple and effective due to its simple design. And because it is stateless, it is easy to scale. In the industrial sector, REST is used in many places. We provide you here with the necessary basic knowledge about REST. Discover everything you need to know about using REST in our practical example and industrial explanations. You can find a dedicated video about this topic by visiting our tutorial stream.

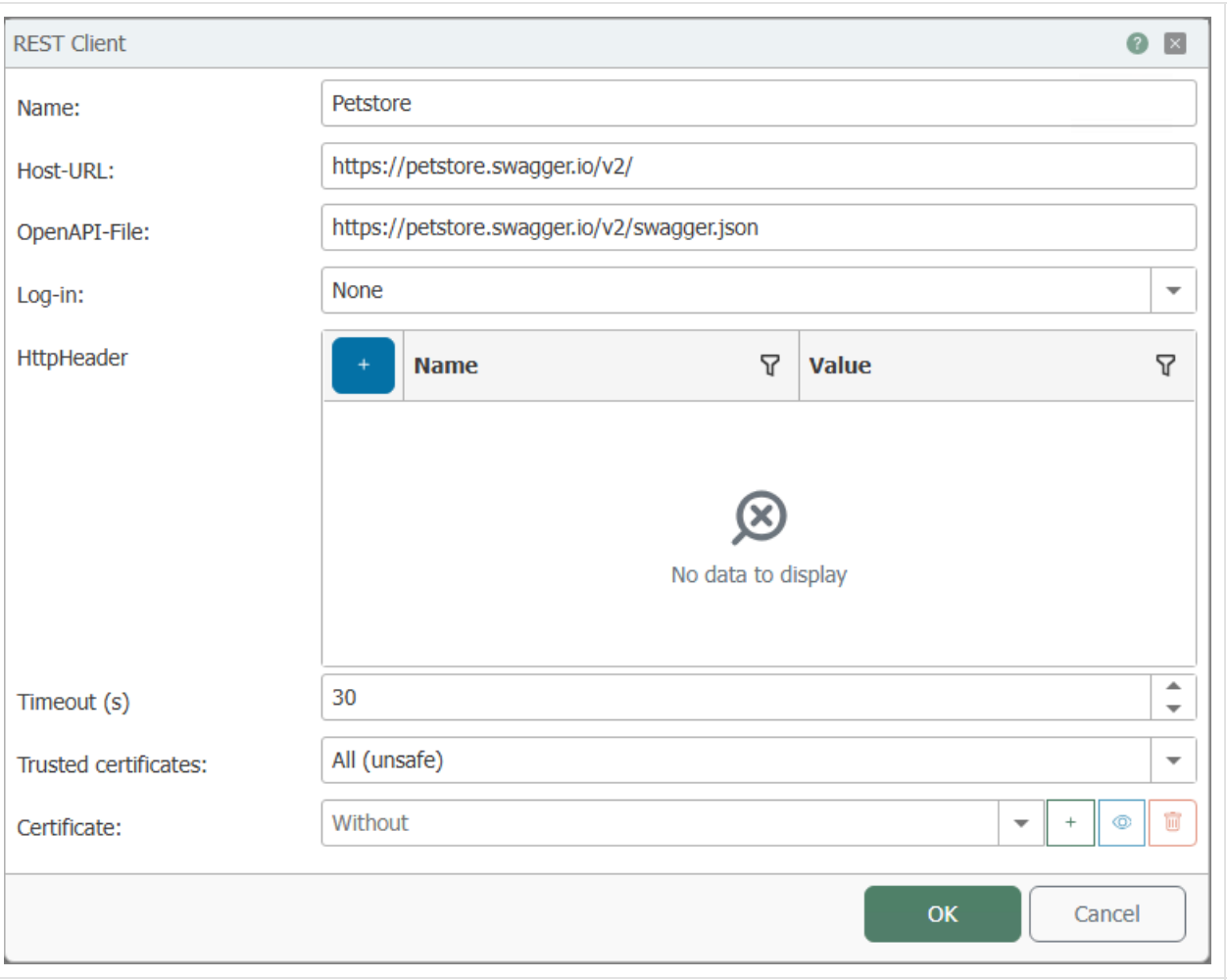
You have the options of calling a REST API with the REST client or providing your own API with the REST server. The configuration of both plug-ins is explained on the following pages.

REST Client

REST Server

Rest Client

With the REST Client, you can call an API to read out data.



The following properties can be set for the REST Client:

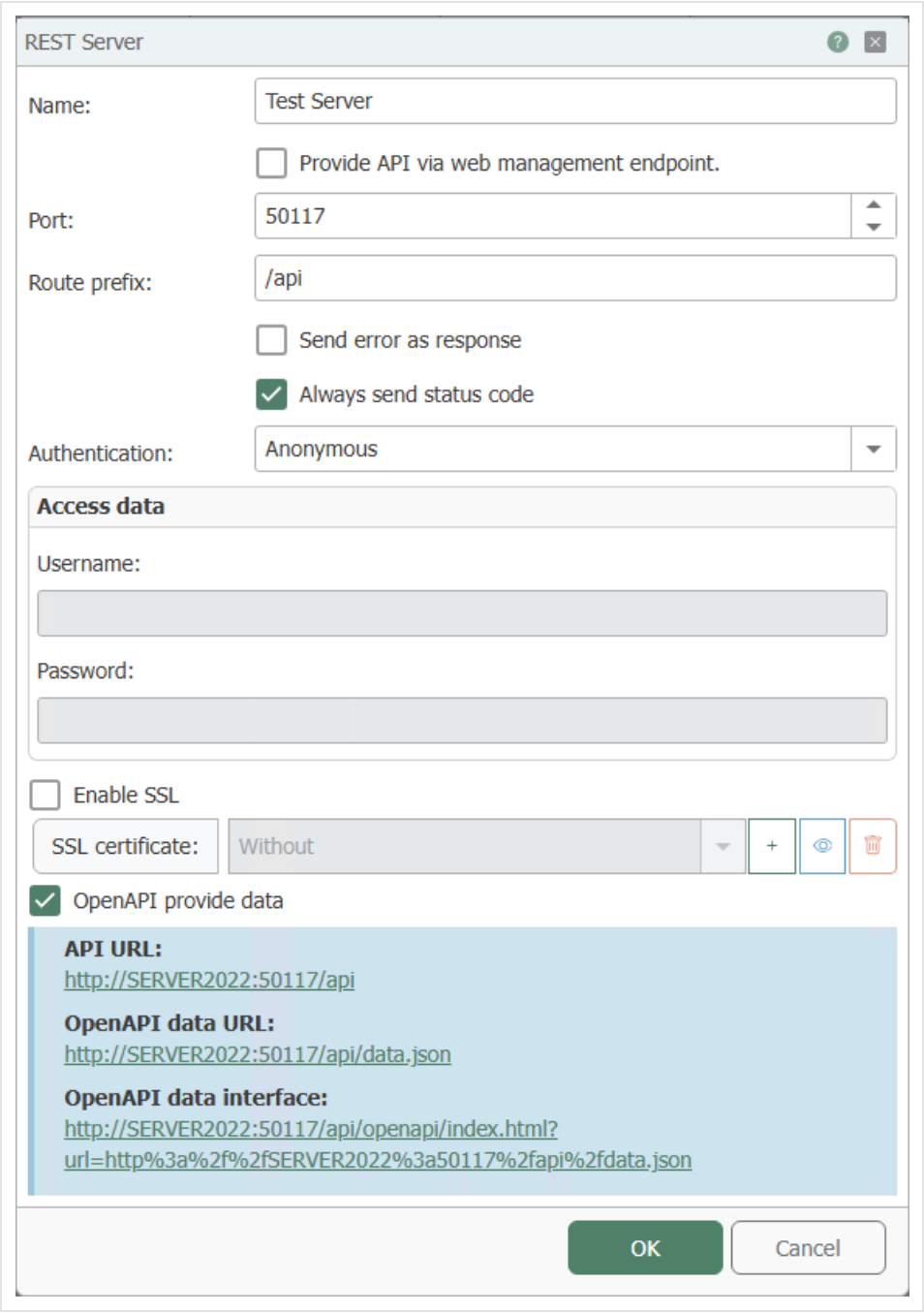
Name	Any name under which this REST client will appear when creating connections or templates. Select a name that is unique for you during connection configuration.
Host URL	URL of the web service you want to connect to.
OpenAPI-File	URL that leads to the description file of the API (programming interface). This is usually a JSON or YAML file. Specifying an OPenAPI file helps later when configuring the transfer object. See
Log-in	If a login to the web service is required, the appropriate type can be selected here. You can obtain the details from the administrator of the web service.
HttpHeader	Fields for transferring the important http parameters.
Timeout	Time in seconds that the connection establishment may take before a connection error is returned.

Trusted certificates	Select which certificates are to be trusted. • **All** (unsafe): Every certificate is being accepted. • **Router**: A connection is allowed if the certificate is in the Router's certificate management and is trusted. • **Windows**: A connection is allowed if the certificate is available in the Windows certificate store.
Certificate	Client certificate to be used for connections. The plus button is used to add a certificate, the delete button is used to remove a certificate. The eye button can be used to display the properties of the selected certificate.

REST Server

REST server

The REST server plug-in can be used to provide APIs that are accessible from outside the OPC Router.



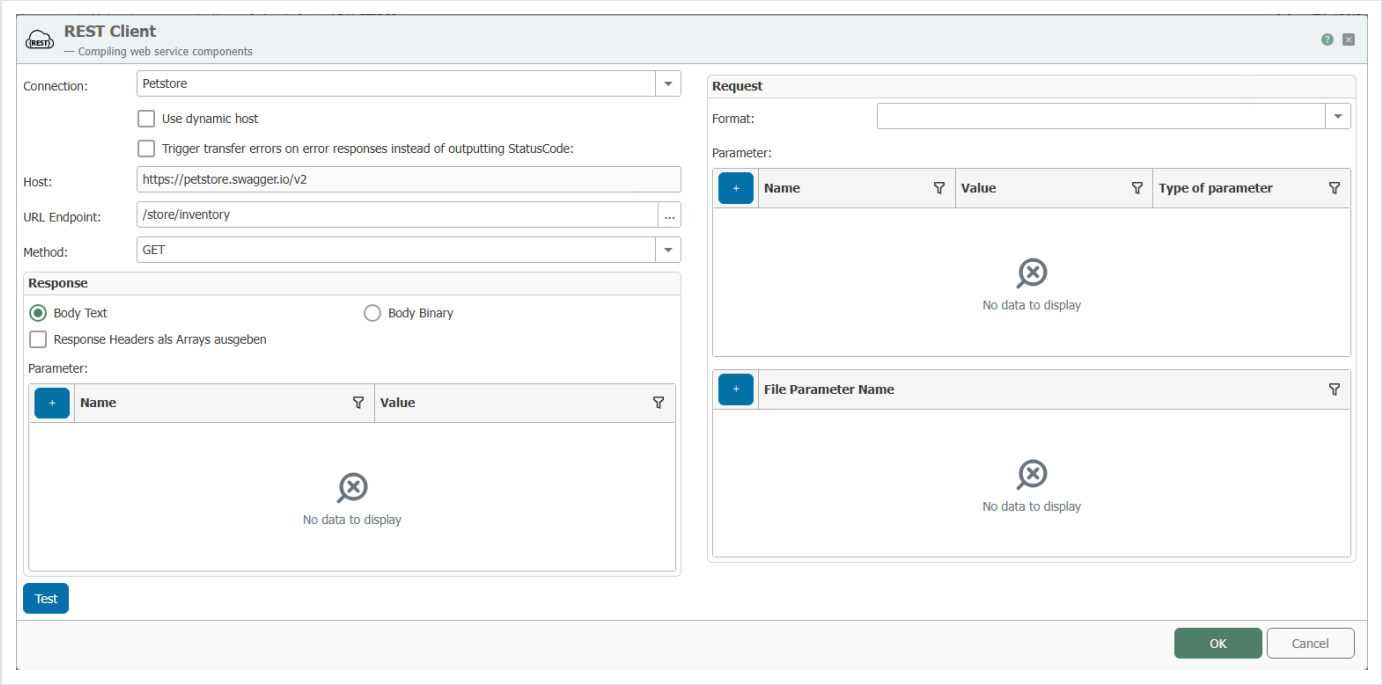
The following properties can be set for the REST Server:

Name	Any name under which the REST server will appear when creating connections or templates. Select a name that is unique for you during connection configuration.
Provide API via web management endpoint	If this option is active, the port of the OPC Router Web Management is used in addition to the selected port to provide the API. Note: The destination address of the API is extended by the prefix `"services/{Plug-in Name}"`.

Port	The port under which the API will be. Note: The port must not already be in use. In addition, it is important to release the port in the firewall if you want to access the API from another device.
Route prefix	The prefix, which must be specified after the IP and with port in order to access the API. Example: If the route prefix is "/api", the full URL would be `{IP/Hostname}:{Port}/api`
Send error as response	If this option is active, the status code is sent as a response if an error occurs.
Always send status code	The status code is always sent, e.g. after a successful transmission.
Authentication	Specify how to authenticate from outside: • Anonymous • Basic • Close API
Username (available for Basic authentication method)	Username for the login.
Password (available for Basic authentication method)	Password for the login.
SSL	If this option is active, SSL is being used for the connection.
SSL certificate (available when SSL is activated)	Certificate to be used for connections. The plus button is used to add a certificate, the delete button is used to remove a certificate. The eye button can be used to display the properties of the selected certificate.
Provide OpenAPI data	Activate this option if the OpenAPI data is to be provided. This is done in the form of a JSON file.

REST Client Transfer Object

With the REST client transfer object, REST APIs can be used in connections. How to create a transfer object is described on the page connection_projecting_project.md.



The following properties can be configured for the REST Client Transfer Obeject:

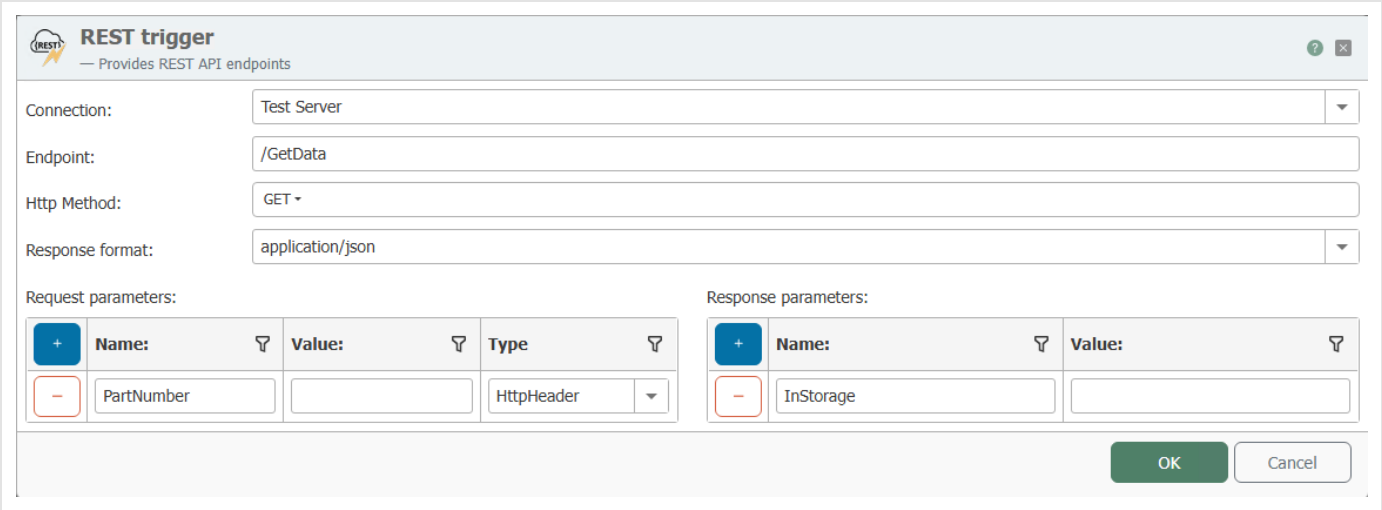
Attention: Please pay attention to upper and lower case when entering the URL. The URLs are case sensitive!

Set of access data	REST client to be used for this transfer object.
Use dynamic host	If this option is active, the host can be passed as a parameter in the connection. This is helpful if requests are to be sent to different hosts. If this option is active, a host address can be entered when using the test function, which can be started with the "Test" Button.
Host	The address specified in the selected set of access data will be displayed here.
URL endpoint	Address for the endpoint. If an OpenAPI file has been specified for the selected connection, the endpoint can be selected in a dropdown (see graphic). Placeholders can be defined within the URL using curly brackets. For example: `/v2/{placeholder1}/user`. These placeholders are automatically added as parameters from the type of URL segment in the request parameter area and can then be written to as usual via the OPC Router interface.
method	Method to be used for the call.
Format	Format to be used for the request. e.g. JSON or XML.
Request parameters	The parameters that are sent with the request.
Response parameters	The expected response parameters of a response can be specified here. These should correspond to the response parameters of the addressed endpoint.

Test	The "Test" button opens a modal in which you can check whether the connection is working correctly. The "Send" button can be used to test the call. If the call works, the expected response (response body) is being displayed and the response parameters are assigned corresponding values.

REST Trigger Transfer Object

The rest trigger provides rest API endpoints. When a request is sent to the provided endpoint, a transfer is triggered.



REST Trigger

The following properties can be configured:

Set of access data	REST server connection via which the endpoint is to be provided.
Endpoint	Determines the part of the URL after the host URL of the selected connection. For example, if the host URL in the connection is "http://Server:50117/api/" and "GetData" is configured here for the endpoint, this call can be accessed under ""http://Server:50117/api/GetData".
Http Method	The Http methods supported by the REST endpoint can be selected here.
Response Format	The format in which the response will be send, e.g. JSON.
Request Parameters	The expected parameters for a call can be specified here. These parameters must be sent with a request for the call to work.
Response Parameter	The parameters to be send sent with the response.

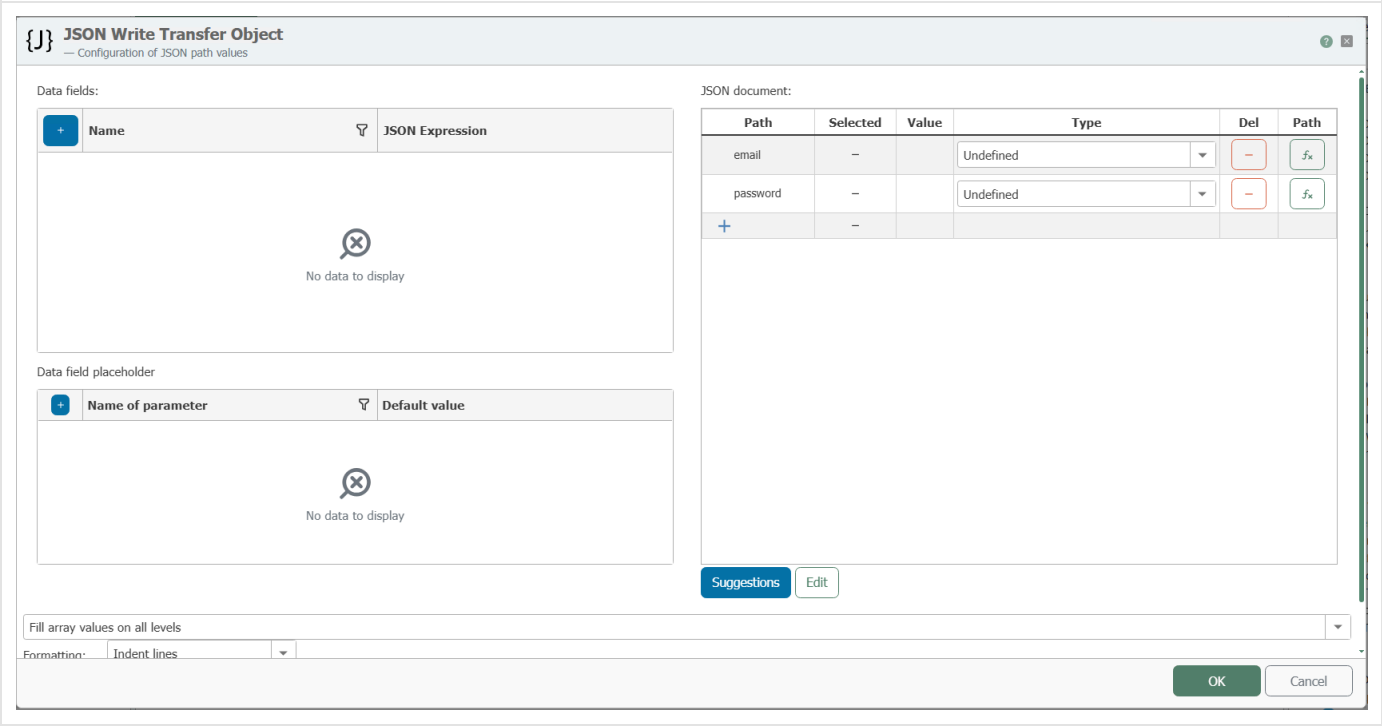
Examples and concepts

Connection for Bearer Token

Connection for Bearer Token

Collect token

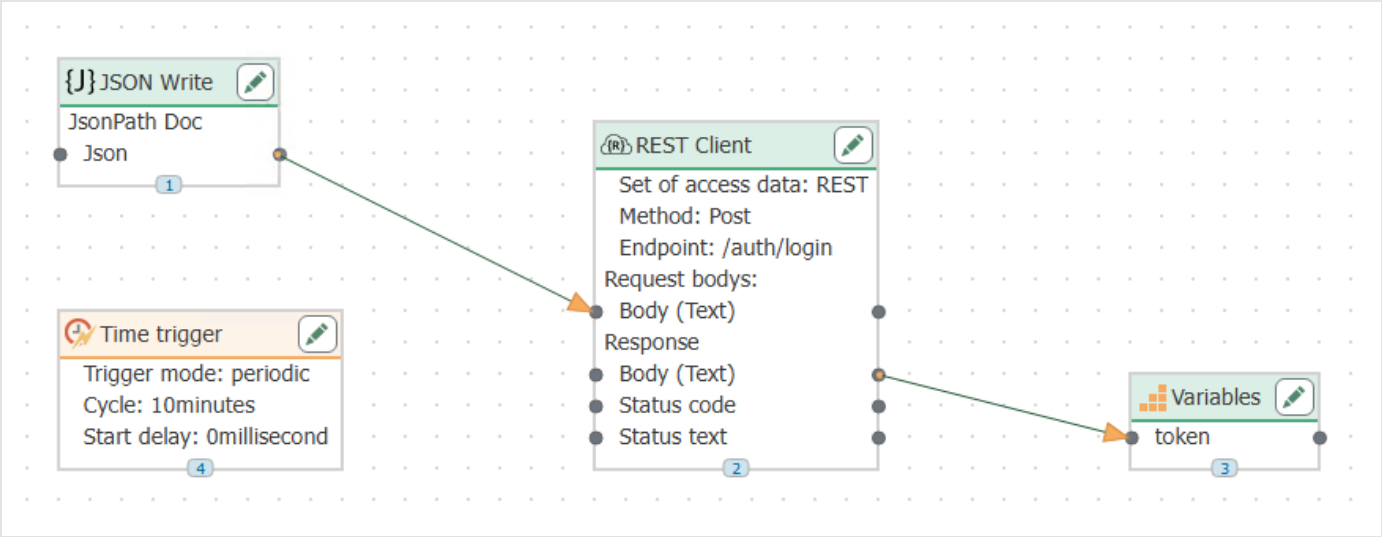
To get the token, write the login data in the JSON Write transfer object in the table below.



JSON Write for Bearer Token

Define a REST transfer object. Include the REST plug-in here. Note that the login is set to None in the REST plug-in. Specify the corresponding URL endpoint. Select POST as the method. In the request, specify the "application/json" in the format. Connect the json parameter from the JSON write transfer object to the body parameter in the request body from the REST transfer object.

Create a variable with the name token, which is type-less. Connect the body parameter in the response to the variable. To execute the connection cyclically, include a time trigger and set the desired cycle.

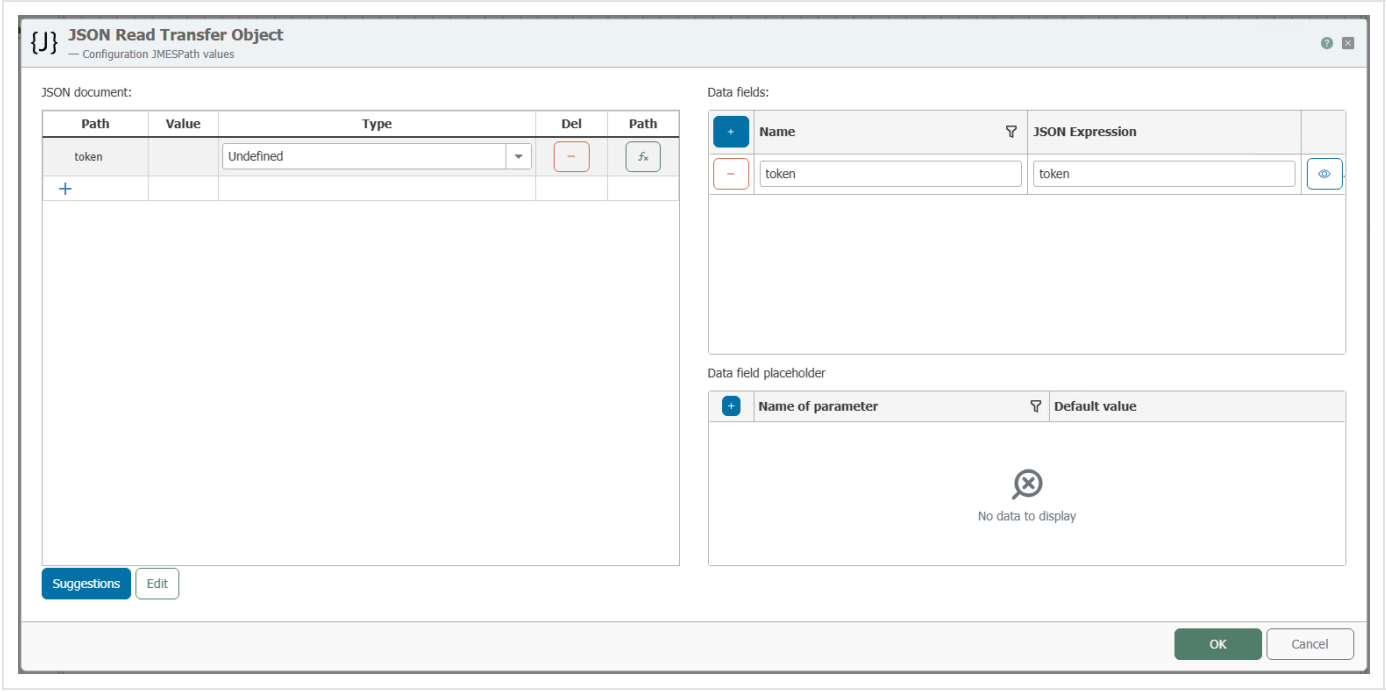


Connection for Bearer Token

Pass token

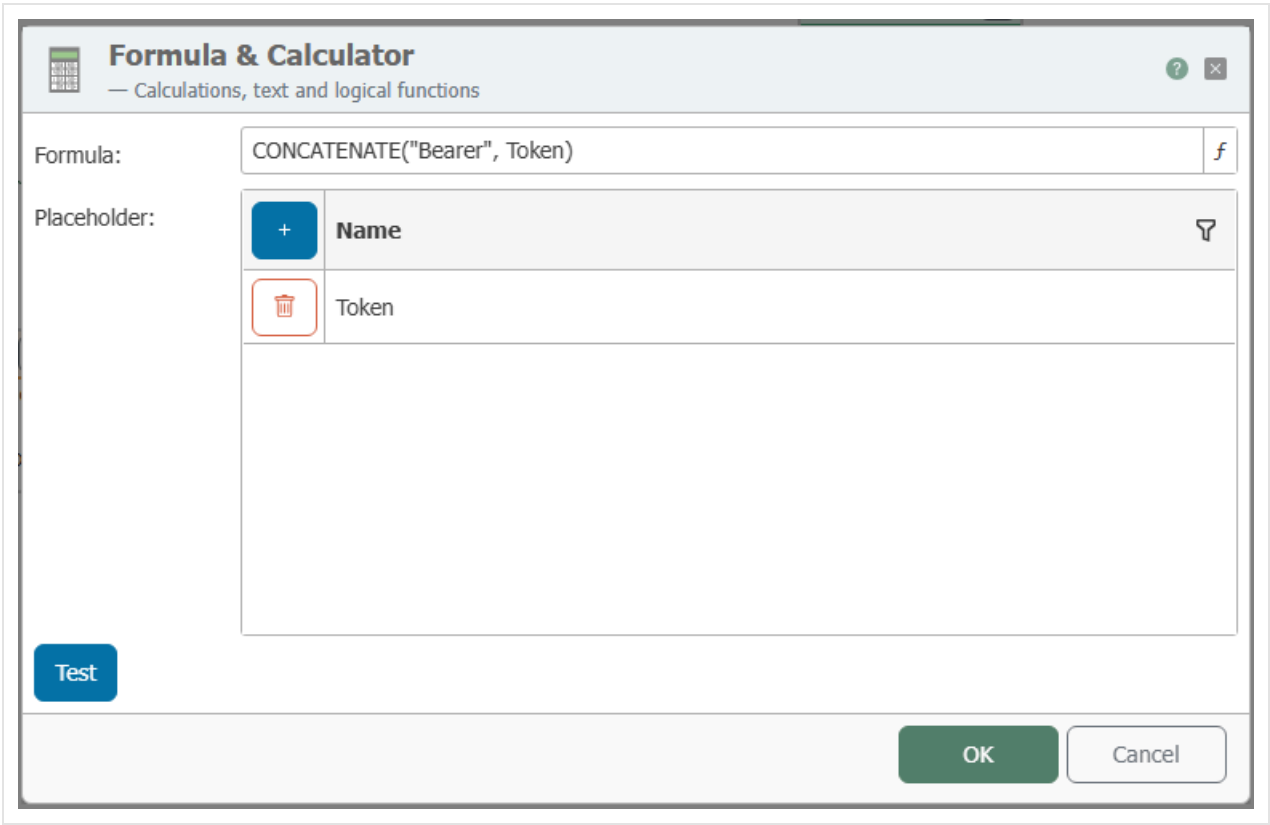
Pass the token in a second connection. Drag a variable into the canvas and select the appropriate variable. Drag a JSON Read

transfer object into the workspace and create a token element.



JSON Read for Bearer Token

Drag the variable to the Json parameter from the JsonPath Doc. Create a calculator transfer object and select the CONCATENATE function via the "More" button. Create a placeholder using the "New" button. Enter the following parameters into the function as shown in the screenshot.

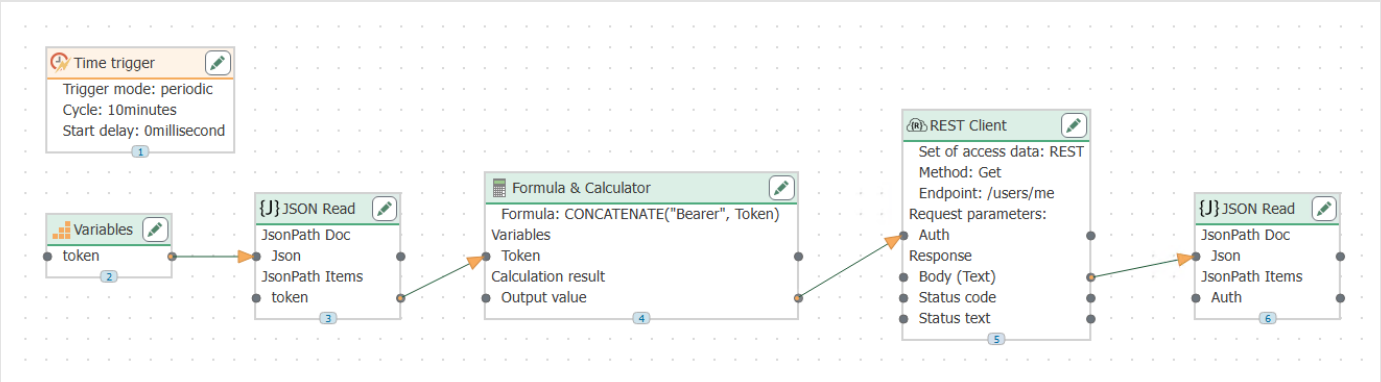


Calculator transfer object for Bearer Token

Connect the parameter from the JsonPath item to the variable of the calculator transfer object.

Create a new transfer object using the GET method. In the request, select the format application/json and create a parameter with the parameter type HttpHeader.

Connect the output value of the computer transfer object to the request parameter of the REST transfer object. Then connect the body parameter of the response of the REST transfer object to a desired other transfer object. To execute the connection cyclically, create a time trigger.



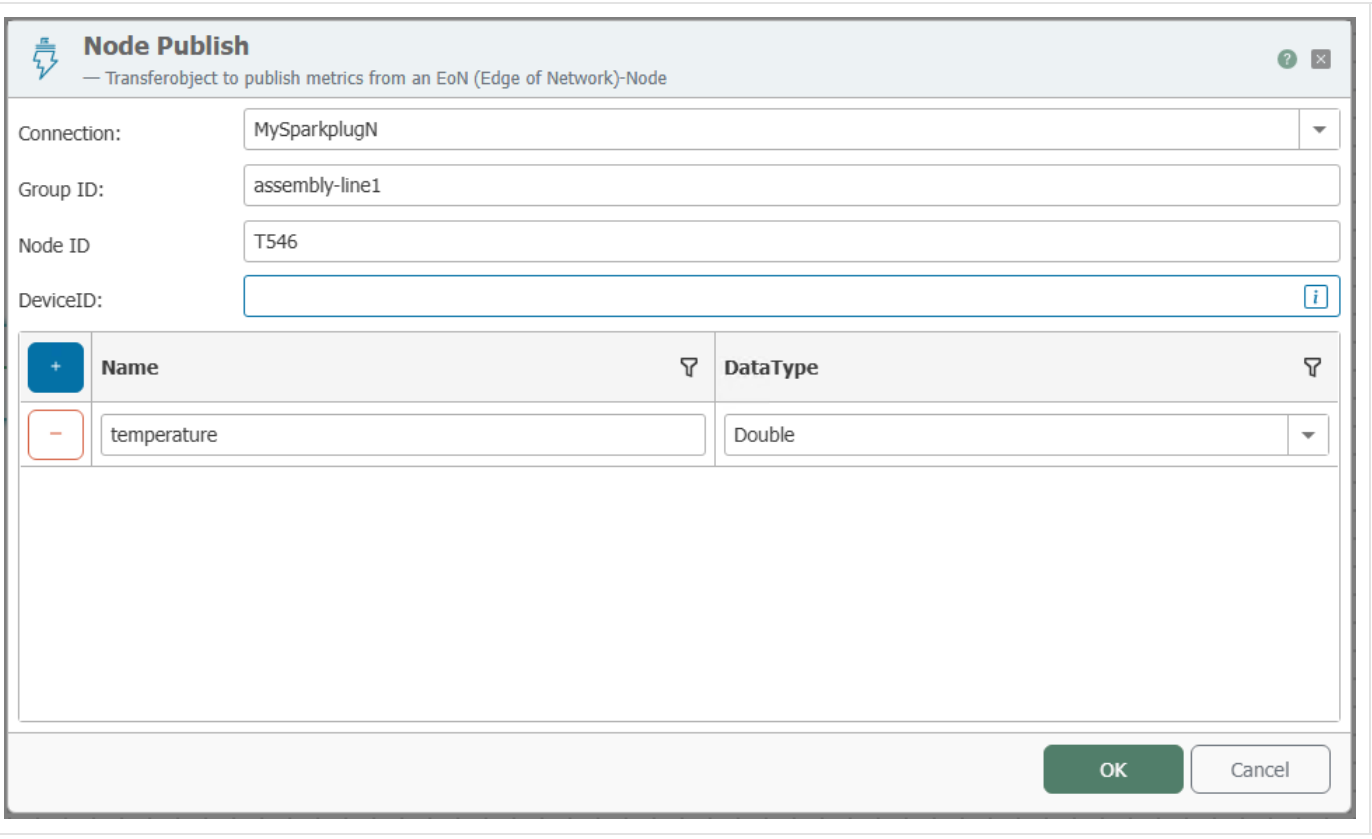
Connection for passing the Bearer Token

Sparkplug

Transfer objects

Node publish

The Sparkplug Node Publish transfer object allows you to send a defined payload to different applications. The different nodes can be narrowed down and reached via the filter options.



Node Publish Transfer object

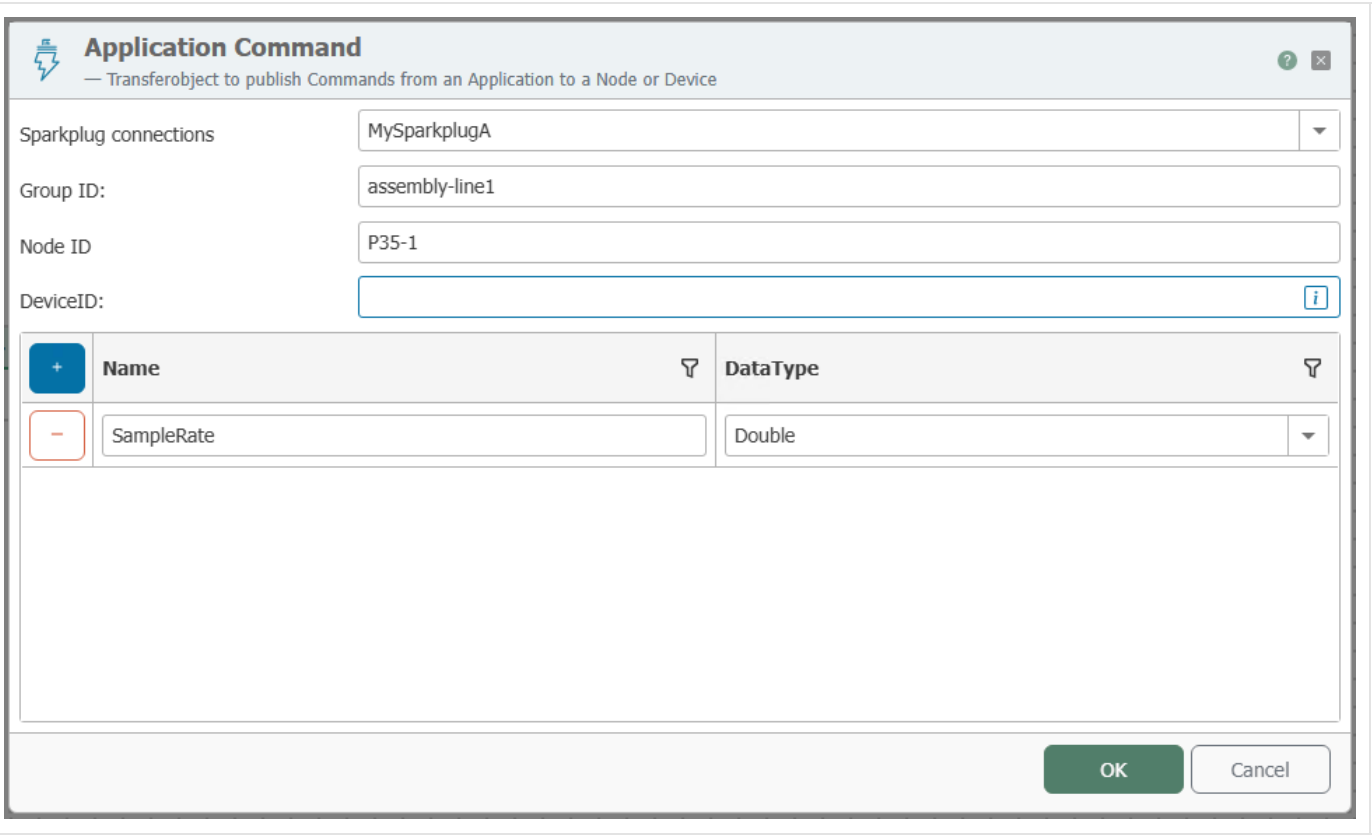
Only connections created for node profiles in the plug-in settings can be selected.

Sparkplug-Anbindung	Select a Sparkplug connection created under Plug-Ins.	
Group ID	Select a Group ID for the transfer object. Leave the field blank to use the settings of the plug-in	
Node ID	Select a Node ID for the transfer object. Leave the field blank to use the settings of the plug-in	
Device ID	Optional	
Items	Enter the payload data for the connection here. These can be supplied with data via arrows, as usual.	

*

Application Command

You can send commands to Sparkplug nodes using the Application Command Transfer object.



Application Command Transfer object

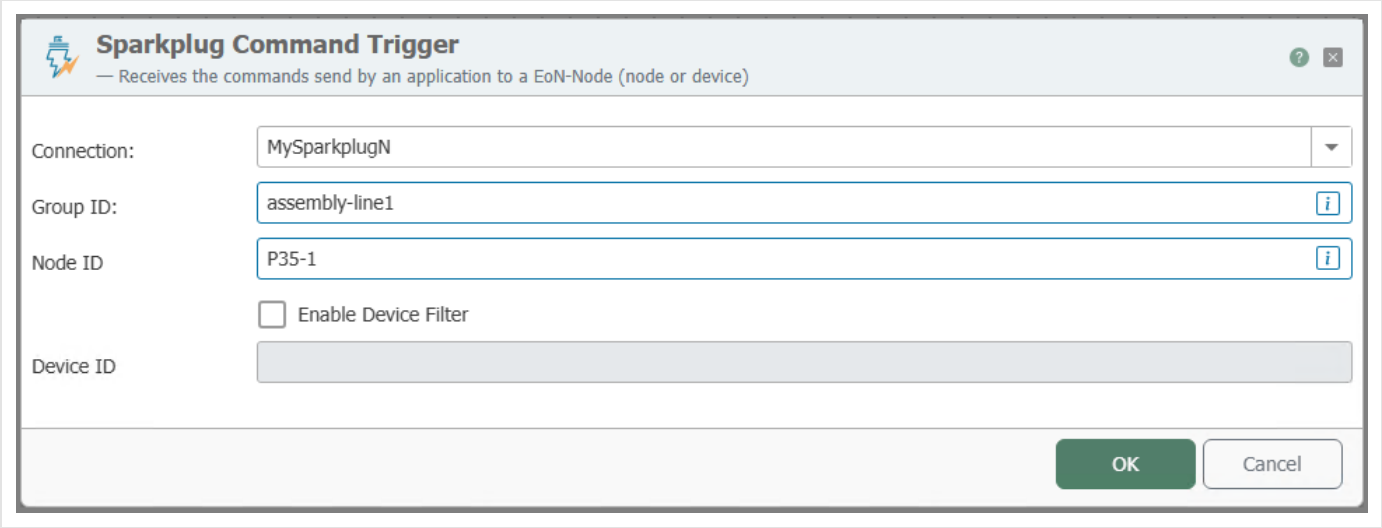
Sparkplug connection	Select a Sparkplug connection created under Plug-Ins.
Group ID	Select a unique Group ID.
Node ID	Select a unique Node ID.
Device ID	Optional
Items	Enter the payload data for the connection here. These can be supplied with data via arrows, as usual.

*

Trigger

Sparkplug Command Trigger

The Sparkplug Command Trigger can be used to receive commands sent by applications to the various nodes. The information contained can be output for further processing.



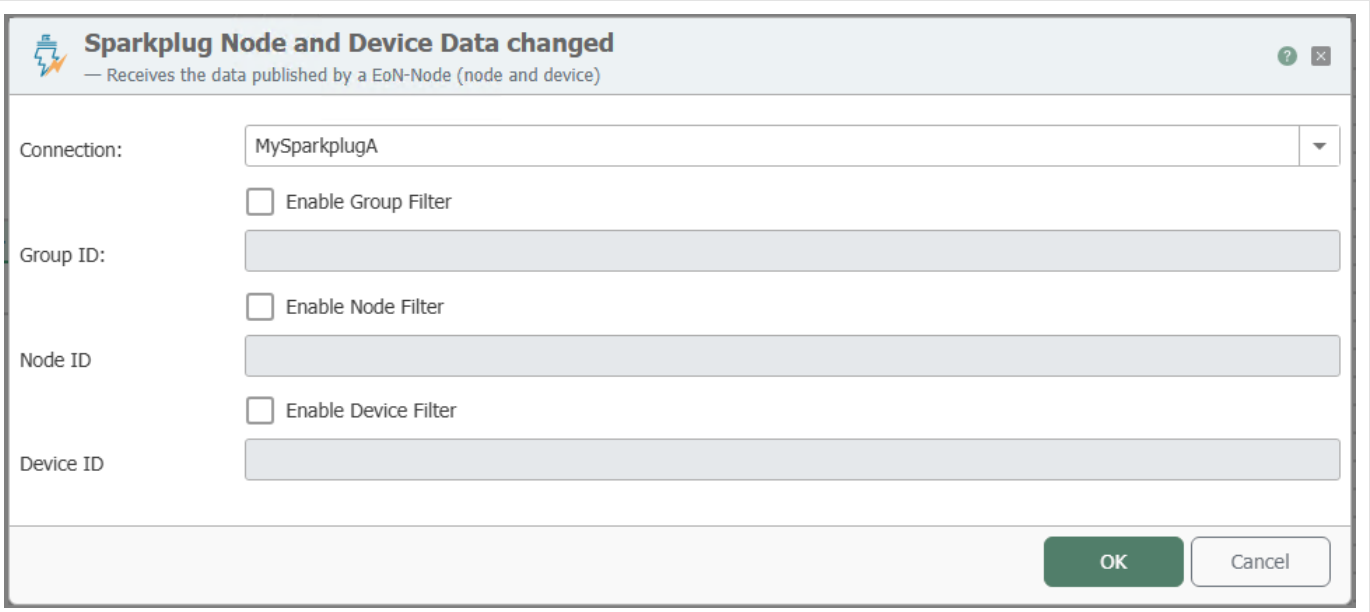
Sparkplug Command Trigger

Only connections created for node profiles in the plug-in settings can be selected.

Sparkplug connection	Select a Sparkplug connection created under Plug-Ins
Group Filter	Specify a group filter for the trigger. **Caution**: The specified filters are of type string and filter mode is 'equal'.
Node Filter	Specify a node filter for the trigger. **Caution**: The specified filters are of type string and filter mode is 'equal'.
Device Filter	Specify a device filter for the trigger. **Caution**: The specified filters are of type string and filter mode is 'equal'.

Sparkplug Node and Device Data changed-Trigger

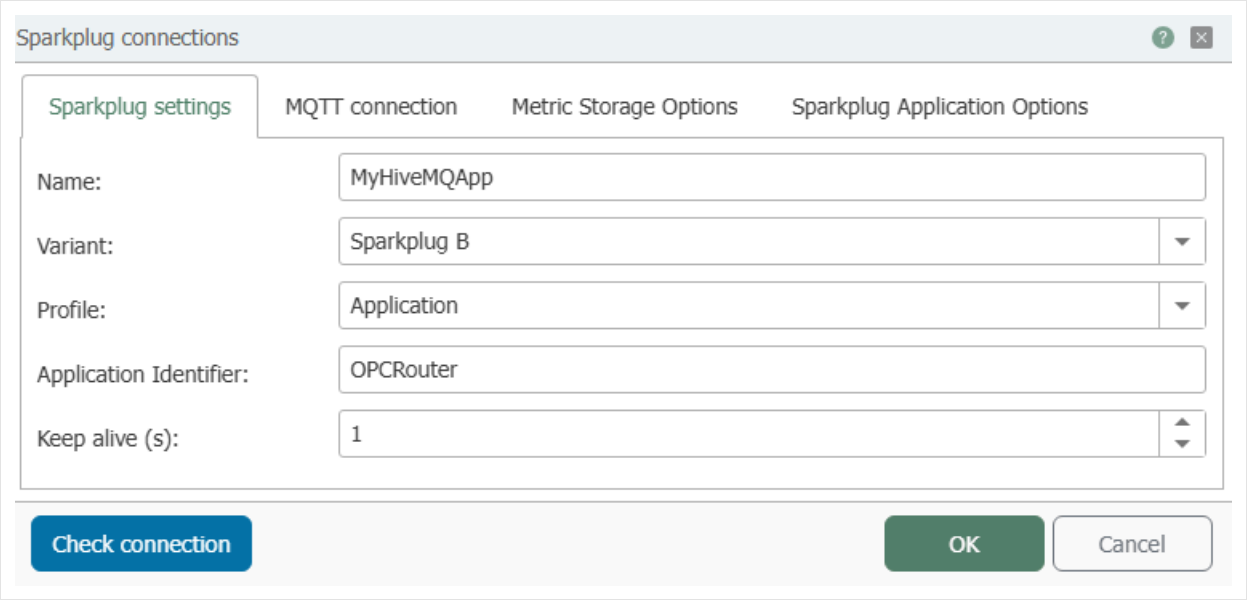
The Sparkplug node and device data changed trigger initiates a connection as soon as a connected device changes its data. Specific devices can be connected via the filters.



Sparkplug Node and Device Data changed trigger

Sparkplug-Anbindung	Select a Sparkplug connection created under Plug-Ins
Group Filter	Specify a group filter for the trigger. **Caution**: The specified filters are of type string and filter mode is 'contain'.
Node Filter	Specify a node filter for the trigger. **Caution**: The specified filters are of type string and filter mode is 'contain'.
Device Filter	Specify a device filter for the trigger. **Caution**: The specified filters are of type string and filter mode is 'contain'.

Sparkplug Plug-in

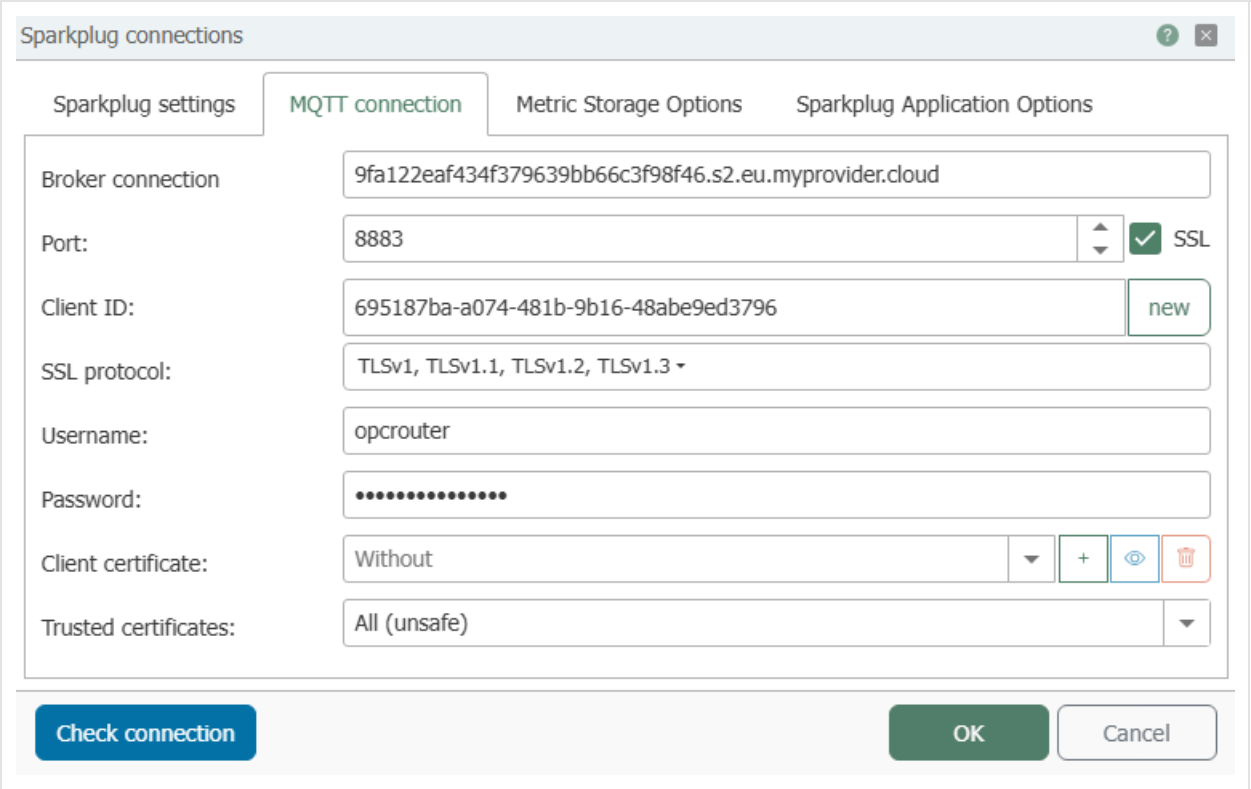


Sparkplug settings in the Sparkplug Plug-in

You set the following properties in the configuration of the Sparkplug Plug-in:

Sparkplug settings:

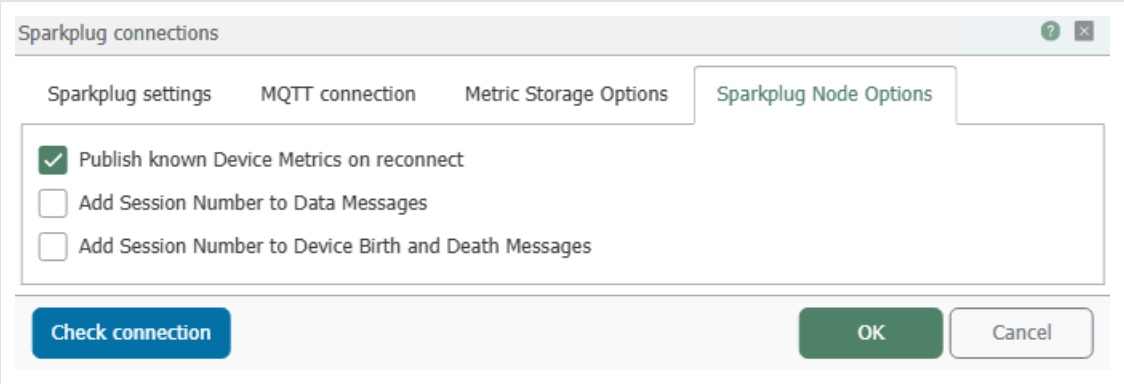
SETTINGS NAME	DESCRIPTION
Name	Assign a unique name to the Sparkplug Connection.
Variant	Select the Sparkplug A or Sparkplug B variant.
Profile	Specify the profile of the Sparkplug connection to be used to log in to the Sparkplug Broker.
Group ID (Only Profile: Node)	Assign a freely selectable unique Group ID to be used as the default.
Node ID (Only by Profile: Node)	Assign a freely selectable unique Node ID to be used as the default.
Application Identifier (Only by Profile: Application)	Assign a freely selectable unique application identifier that is to be used as the default.
Keep alive (s)	Indication of the keep alive in seconds.



MQTT Connection settings in the Sparkplug Plug-in

MQTT connection:

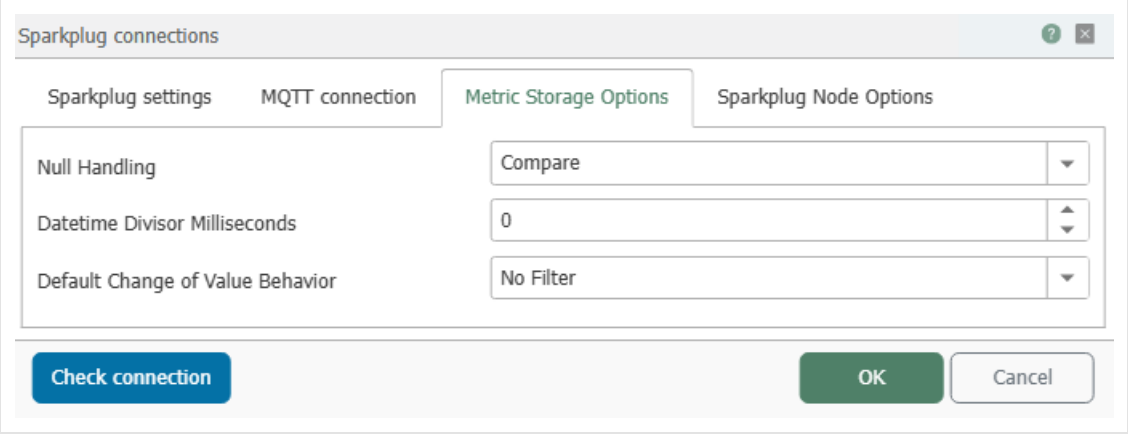
SETTINGS NAME	DESCRIPTION
Broker connection	Enter the adress of the broker.
Port	Port on the server that is used for the MQTT connection.
SSL	Use SSL encryption. The server must support this and the corresponding port must be set.
Client ID	Enter a unique Client ID.
Create a new Client ID	Creates a new unique Client ID, which is displayed in the Client ID field.
SSL protocol	Select the algorithms that may be used **Note:** Some MQTT brokers reject connections if protocols that are too old could be used.
Username	User name for logging in to the server.
Password	Password for logging in to the server.
Client certificate	Selection of a client certificate from the drop-down list, which the OPC Router also sends.
Trusted certificate	Select which certificates are to be accepted. For example, all certificates, only Windows certificates or only router certificates.



Sparkplug Node/Applications Options in the Sparkplug Plug-in

Sparkplug Node/Applications Options:

SETTINGS NAME	DESCRIPTION
Publish known Devicemetrics on reconnect	Determines wheter metrics from known devices are published when reconnecting.
Add Session Number to Data Messages	Determines whether the current session number is added to data messages.
Add Session Number to Device Birth and Death	Adds the current session number to Device Birth and Death messages.



Metric Storage Options in the Sparkplug Plug-in

Metric Storage Options:

SETTINGS NAME	DESCRIPTION
Null Handling	
Datetime Divisor Milliseconds	
Default Change of Value Behavior	

Topfloor

Overview over Plug-ins:

SAP®

ERP Web Services

SAP®

Warning: Docker exclusively supports the SAP NetWeaver protocol

SAP® plug-in connects OPC Router to all reachable SAP® systems (4.0b or higher) via IDoc, BAPI or RFC. Supported are all current SAP® systems (e. g. NetWeaver, ECC 6, R/3 ab 4.0b ...). Prerequisite are either the SAP® log-on pad or library librfc32.dll installed on the computer hosting OPC Router. The library has to be stored at \Windows\System32. If the program is to run in 32-bit mode on 64 bit OS, the 32-version of the DLL has to be in. \Windows\SYSWOW64 .

To use Netweaver, the SAP® NetWeaver RFC SDK must be installed. SAP® files have not be delivered by inray Industriesoftware GmbH. Therefore you need a current SAP® account (status 2018). The following link will help you: https://helpcenter.theobald-software.com/erpconnect/documentation/introduction/saplibraries/

When the librfc32 file is used, Microsoft Visual C++ 2005 Service Pack 1 must be installed on some systems:\
https://www.microsoft.com/en-us/download/details.aspx?id=26347

The following SAP® authorizations are required by the OPC Router:

The following objects are needed at least to establish a connection:

```
S_RFC RFC_TYPE=FUGR ;RFC_NAME=SYST ;ACTVT=16 ;
S_RFC RFC_TYPE=FUGR ;RFC_NAME=RFC1 ;ACTVT=16 ;
```

Look up tables and table meta data:

```
S_RFC          RFC_TYPE=FUGR ;RFC_NAME=SDTX ;ACTVT=16 ;
S_RFC          RFC_TYPE=FUGR ;RFC_NAME=SDIFRUNTIME ;ACTVT=16 ;
S_TABU_DIS      ACTVT=03 ;DICBERCLS=&NC& ;
```

Look up table for IDocs: EDBAS or use the following separate authority check for each table to read:

```
S_RFC          RFC_TYPE=FUGR ;RFC_NAME=SDTX ;ACTVT=16 ;
S_RFC          RFC_TYPE=FUGR ;RFC_NAME=SDIFRUNTIME ;ACTVT=16;
S_TABU_DIS      ACTVT=03 ;DICBERCLS=XXXX ;
S_TABU_NAM      ACTVT = 03; TABLE = DD02V
```

XXXX is the Authority Group for the table. To find out, which authority group belongs to which table look at table `TDDAT` (e.g. with SE16). If the table is not listed there the authority group is `&NC&`. For authorizing specific tables please use authorization object `S_TABU_NAM` instead of `S_TABU_DIS`.

If Custom functions like `Z_XTRACT_IS_TABLE_COMPRESSION`, `Z_XTRACT_IS_TABLE` or `Z_XTRACT_IS_TABLE_JOIN` are used

```
S_RFC          RFC_TYPE=FUGR ;RFC_NAME=XXXX ;ACTVT=16 ;
```

XXXX is here the name of the function group where the custom function module are located.

SAP Query: Look up and executing a query:

```
S_RFC          RFC_TYPE=FUGR ;RFC_NAME=AQRC ;ACTVT=16 ;
```

Defining access data:

1. Open plug-in by clicking “SAP access data” in the project configuration’s plug-in list.
2. Click the plus button or double click on the existing entry.
3. Specify access data. (chapter Plug-in).
4. Click on button “Check connection” to verify the OPC Router is getting access to the server and if access data are correct. This is to exclude one typical source of errors.
5. Confirm with “OK”.

The SAP® server can now be used to define a connection.

Note: In any case, use the switchboard “Test” to check the connection and prevent runtime errors. Note that changes are accepted only after confirmation with “OK”.

Note: When using the NetWeaver protocol, access to the PSE file must be guaranteed for all users. In addition, the environment variable SECUDIR must be defined. The path to the SAP crypto files must be specified as the path. E.g. ‘C:\sapcrypto\nt-x86_64\sec’.

SAP® Plug-in

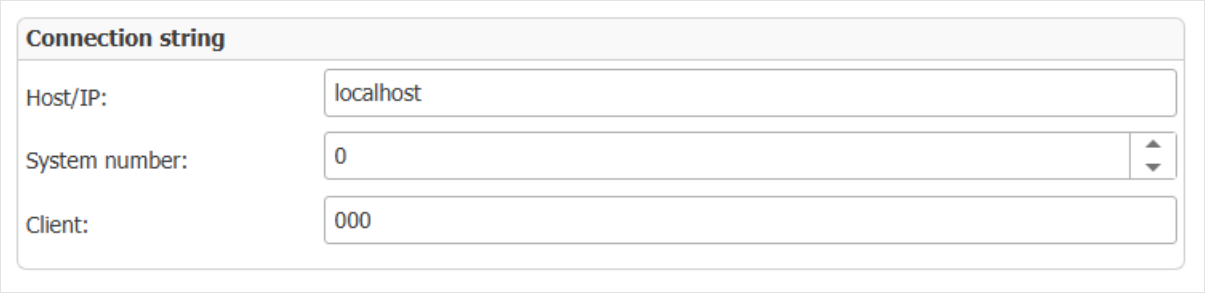
Note: If you want to use load balancing, you need an Enterprise license.

The following properties have to be specified for each set of SAP® access data:

General

NAME	
Name	This set's name. The name will be shown in projection. Select a name you will recognize!
Client protocol	SAP driver used by the OPC Router for communication. Select the client protocol from the drop-down menu: • RFC Protokoll • Netweaver RFC Protokoll
Client libaries	Add here the *.dll files for windows and the *.so files for Docker.
Connection type	Select the desired registration procedure: • Direct log-in • Load Balancing

Direct log-in:



Host/IP	Server (Name or IP) with SAP® system. Explicit log-on is required (no load balancing).
System number	Number of SAP® system.
Client	SAP® client (company code).

Load Balancing:

Message server	Name or IP of SAP® message server
Log-on group	Log-on group (of SAP® system)

System number	Number of SAP® system
Client	SAP® client (company code).

Security

Login method	• Basic Authentication • Secure Network Communications (SNC)

Basic Authentication:

Username	Username to log onto SAP® system.
Password	Password to log onto SAP® system.

Secure Network Communications (SNC):

Contact the support if you want to use the sapscryptto.dll file.

The sapcrypto.dll file is not provided by inray.

Username	Username to log onto SAP® system.
Password	Password to log onto SAP® system.
Log-in	Use your Windows user to log on to the SAP® system.

Own SNC name	Name in format p:CN=... as it can be found in the issued certificate
Name of SNC partner	Entry the SNC partner name in the format p:CN=...
SNC Librarypath	The SNC library path can be configured here. It can be selected in a dialogue window after the '...' button has been clicked.
Use single sign-on	Complete access to all systems with a single login
Security level	Selection of the security level from the drop-down menu e.g. Maximum or Standard.

Authentication method	Selection between NTLM Authentification or Kerberos5 Authentication.
Safety certificate	Select a security certificate from the drop-down list. The plus sign allows you to add another certificate, the minus sign removes one. Another button allows you to display the properties of an existing certificate.

Language and codepage

Language	Enter the language you wish to use here. For example, ‘de’ stands for German, ‘en’ for English, etc. The language you want to use must be available on the SAP system.
IDoc culture	Selection of the IDoc language (Culture) German or English to prevent a crash if an instance in the SAP® plug-in contains an invalid IDoc language. The date notation for example is different: First February in German 01.02.2018, in English 02.01.2018

Register RFC Trigger as

Unicode	Register RFC triggers as unicode
Non-Unicode	Register RFC triggers as non-unicode

If you use SAP Netweaver, you must activate the setting Unicode. If you use the RFC Protocol, the encoding dependent on which librfc32l file you use. The file exists with Unicode (librc32u) and without Unicode (librfc32).

The target in the SAP Transaction SM59 must be changed in Unicode/No Unicode, too. This depends on, what is set in the OPC Router and which library is used.

Expand

IDocHandling	• Use template from SAP system
	  ◦ The standard values from the SAP system as defined in the segments are available here.
	• Create empty IDoc
	  ◦ Only the segments are transferred, but these are empty.

Use additional process for receiving RFC calls and IDocs	If this check box is activated, the OPC Router is not stopped when an error occurs in the SAP trigger (RFC or IDoc) and continues to work.

You can use the button “Test” to check whether your connection is successful.

Note: To use the preferred settings from the used SAP user, the OPC Router need read access on the SAP table USR01.

Configuring the Port

By default, OPC Router uses port “33_<System-Nummer>_” to communicate with the SAP® server, e. g.. “3301”, if System-No = 01. If your SAPO system uses a different port and SAP® GUI is not installed, you have to adjust the port:

1. Backup file C:\Windows\System32\drivers\etc\services.
2. Open file C:\Windows\System32\drivers\etc\services with a text editor (not the backup).
3. Add (or change) “sapgw00”, whereas the port has to have format “XX00” e. g. “3600”.

   ```
   sapgw00      <Portnummer>/tcp
   ```

4. Save the file.

OPC Router will use that specific port.

Transfer objects

Overview over transfer objects:

SAP-IDoc Transfer Object

SAP-BAPI Transfer Object

SAP-RFC Transfer Object

SAP-IDoc Transfer Object

SAP-IDocs (Intermediate Documents) can be used both as source or destination. A set of SAP access data has to be specified within the plug-in configuration. The transfer object is created as described above. The transfer object can send IDocs, the IDoc trigger can receive IDocs.

Open transfer object by double click and configure as follows:

Tab: Settings

SAP access data	Select SAP set of access data.
IDoc type	Select an IDoc type provided by your SAP system.
IDoc Enhancement	Class Extension of IDocs: CIMTYP Customer Extension of the Basic IDoc Type.
Queue Name	
Creation	• **Apply plug-in configuration:** It selects what was selected in the SAP-plug-in by IDoc handling. • **Create empty IDoc:** When an IDoc is loaded from SAP, the IDoc is sent, but the existing elements are emptied and only their own changes are sent. • **Use template from SAP system:** The loaded IDoc from SAP is used but edited with its own values.

Tab: Parameters

Parameter selection	Please mark the segments you want to read or write.

You might write several records at once to the IDoc; the segments needed will be created as required. The other way round, the IDoc might deliver several records at once.

SAP-BAPI Transfer Object

SAP BAPI can be used both as source or destination. A set of SAP access data has to be specified within the plug-in configuration. The transfer object is created as described above.

Open transfer object by double click and configure as follows:

Tab Settings

SAP access data	Select SAP set of access data.
Output	Provides the results of the BAPI call as a JSON document for further processing.
Business object	Select Business Object.
Methods	Select one of Business Object's methods.
Object keys used	Object keys define an object positively and are added automatically. Required if method depends on instance.

Registerkarte: Parameters

Please specify a function	Select the required parameters by setting the appropriate check mark. You can also take whole categories if you place the checkmark accordingly. **Note:** The specified table must have a primary key. Tables without primary keys are not supported.
Copy JSON format	Copies a JSON document to the clipboard whose structure corresponds to the function call.

Parameters are categorized:

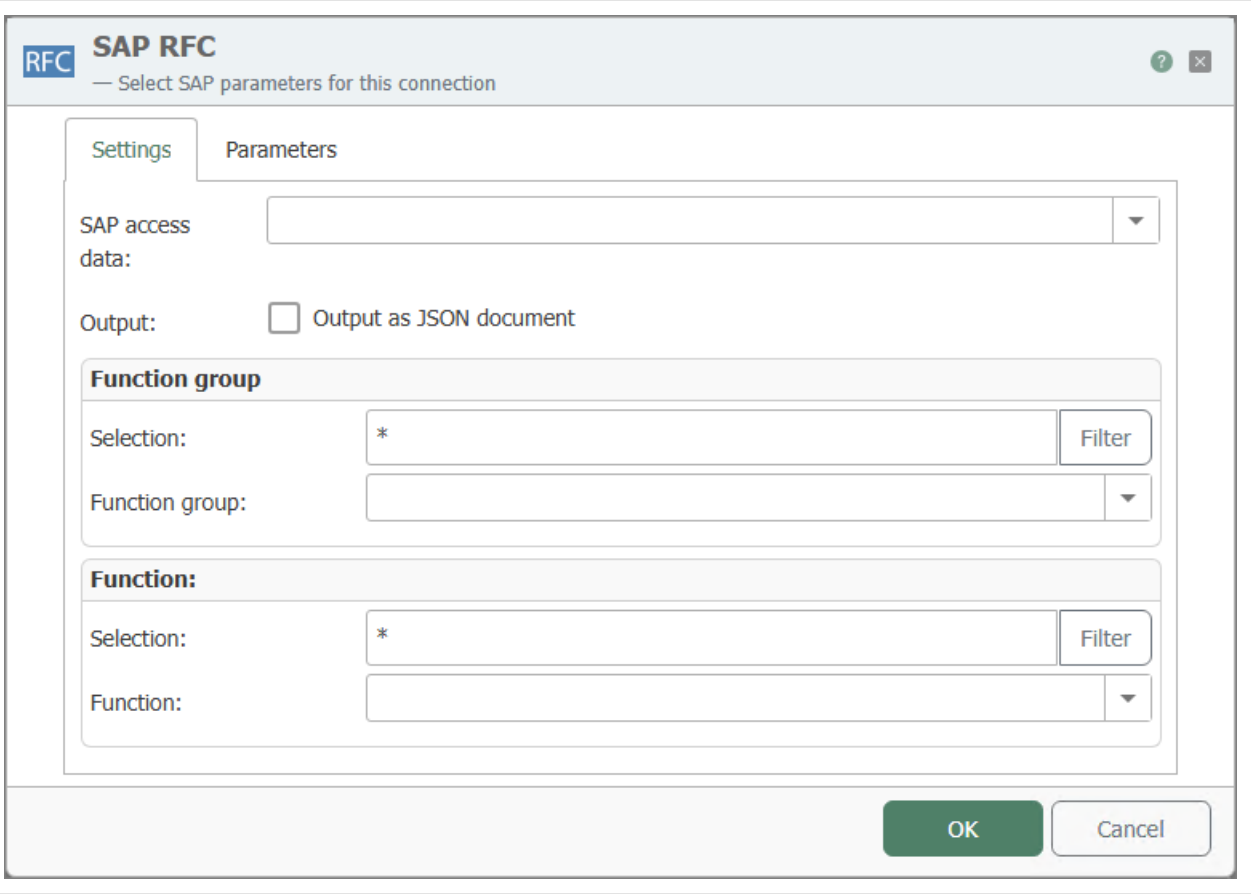
- Import: Data transfer into SAP system
- Export: Data transfer from SAP system
- Change: Data transfer into SAP system and return of changed values.

Categories may contain sub-categories, e. g. "Table" containing data tables.

SAP-RFC Transfer Object

SAP RFC can be used both as source or destination. A set of SAP access data has to be specified within the plug-in configuration. The transfer object is created as describe above.

Open transfer object by double click and configure as follows:



Settings

SAP access data	Select SAP set of access data.
Output	Provides the results of the RFC call as a JSON document for further processing.
Function Group	Select one of the function groups available.
Search	Enter a search string (*=place holder) and click “Filter” to ...
Selection	... show only those groups matching you search term. Select here!
Function	Select one of the functions available in your Function Group.
Search	Enter a search string (*=place holder) and click “Filter” to ...
Selection	... show only those functions matching you search term. Select here!

Parameters

Parameters available	Select the required parameters by setting the checkmarks accordingly. You can also take whole categories (see below).
Copy JSON format	Copies a JSON document to the clipboard whose structure corresponds to the function call.

Note: Sollten RFC Funktionen auf dem SAP System geändert werden, nachdem sie im Router projektiert wurden, muss der Router u.U. neu gestartet werden, um diese Änderungen anzuzeigen.

Parameters are categorized:

- Import: Data transfer into SAP system
- Export: Data transfer from SAP system
- Change: Data transfer into SAP system and return of changed values .

Categories may contain sub-categories, e. g. “Table” containing data tables.

Trigger

Overview over trigger:

IDoc trigger

RFC trigger

IDoc trigger

The IDoc trigger will be used if the OPC Router should receive an IDoc from the SAP system. SAP system will trigger the transfer through this object. SAP configuration is described as overview. Please consult you SAP manual.

Properties are:

SAP ACCESS DATA	SET OF SAP ACCESS DATA.
SAP access data	SAP connection via which the relevant function is accessed.
IDoc type	Type of IDoc the Trigger listens. Must be the same as in corresponding transfer object.
IDoc Enhancement	Class extension of the IDocs.
Program ID	Program ID used by SAP system to call destination, identical with program ID of transaction SM59.
Gateway Service	Gateway Service of SAP system
Output	Set this field if you want to receive an output as JSON.

Important: Program ID and function have to be written in CAPITALS. The combination of host, gateway and program ID has to be unique. From version 3.1.1.5 you can use the program ID several times provided gateway-service and/or hosts differ.

Important: If an IDoc is updated/changed in SAP, the trigger must also be adapted to the new IDoc.

Example

IDoc Trigger Settings:

SAP access data	*<SAP access data>*
IDoc Type	PROCESSORDER
Program ID	INRAYTRIGGER
Gateway Service	Gateway Service of SAP system, e. g. 3302

Logical System within SAP transaction BD54:

LogSystem	OPCRIDOC01
Description	OPC Router with IDoc listener

Corresponding SAP Settings, SM59:

Destination	OPCROUTER
Connection Type	T TCP/IP
Activation Type	Registered Server Program
Program ID	INRAYTRIGGER
Gateway host	(no entry!)
Gateway service	(no entry!)

Communication Port in transaction WE21:

(Type)	Transactional RFC
Port	e.g. IDOCPORT
Version	(select appropriated IDoc version)
RFC destination	OPCROUTER

Partner profile in transaction WE20

(Partner type)	LS (Logical System)
Partner no.	OPCRIDOC01

Outbound parameters in WE20:

Receiver Port	IDOCPORT
Outpout mode	Transfer IDoc immed.
IDoc Type/Basic type	PROCESSORDER
Cancel processing after syntax error	(activated)

For testing, via WE19 an IDoc can be send manually providing partner no. and Port. Errors might be found in WE07. Function to call is IDOC_INBOUND_ASYNCHRONOS (as of 4.0), resp. INBOUND_IDOC_PROCESS (before 3.x).

RFC trigger

With RFC trigger, SAP systems can trigger data transfers. Within SAP system, in transaction SM59 the corresponding entry has to be made.

Properties are:

SAP access data	Set of SAP access data.
Program ID	Program ID used by SAP system to call destination, identical with program ID of transaction SM59.

Gateway Service	• RFC protocol: Enter the gateway, e.g. sapgw00 (SAP Gateway 0)
	• Netweaver RFC protocol: Specification of the port
	  ◦ 3300 + Id for basic authentication
	  ◦ 4800 + Id for SNC authentication

Function	Function defined within SAP system that triggers this connection.
Timeout	Time span after which OPC Router will send time-out to SAP system if task has not been finished
Output	Set this field if you want to receive an output as JSON.
Input (Grid)	The rows of this grid are trigger inputs.
Output (Grid)	The rows of this grid are trigger outputs.
Tables (Grid)	The lines of this grid are used as both output and input.
Copy JSON format	Copies a JSON document to the clipboard whose structure corresponds to the function call.

The expected length for sent and received CHAR and NUM fields must be configured via the Edit button. The default length is 0. If you have not changed the length of a parameter, you get information about it when you close the dialog.

Attention: The call of a non-existing function module forcibly leads to the deregistration of the server in the SAP Gateway. The registration can no longer be used and must be performed again, which manifests itself in ERP-Connect by a server restart.

Attention: There is no way to delete functions once they have been made known to the NW Library. This applies to the entire process, across all SAP systems and RFC servers. So you cannot change the name and type of the Import/Export/Tables parameters of a function block that is called by RFC server and identified by its function block name, nor the existence of the block itself. Do not make any changes while the OPC Router service is running. So if you change something it must always be published completely.

return parameters

TRANSFER_OK	• 1 = OK: Successful transfer of the connection • 2 = Aborted: The transfer was aborted from a type transfer object, for example, if no data was contained in a table for a database transfer object. 3 = Error: An error occurred during the execution of the transfer. The request must be resent later and must not be marked as closed by the connected system. • 3 = Error: An error occurred during the execution of the transfer. The request must be sent again later and must not be marked as completed by the connected system.

Important: Program ID and function have to be written in CAPITALS. SAP call is set:

```
Call function '<FUNCTION\>' destination '<SAP-DESTINATION\>'.
```

The combination of host, gateway and program ID has to be unique. The program ID itself can be used multiple times - but only with different gateways / hosts.

You can also use the RFC trigger to create ITAB or STRUCTURE as parameters. Create the inputs and outputs accordingly by selecting the respective type, assigning a name, and choosing Edit to access the editor for the individual fields. Nesting is not possible.

Example

RFC-Trigger Properties:

SAP access data	*<SAP access data>*
Program ID	INRAYTRIGGER
Gateway Service	Gateway Service of SAP system, e. g. 3302
Function	TRIGGERFUNCTION

Corresponding SAP properties, SM59:

Destination	OPCROUTER
Connection Type	T TCP/IP
Activation Type	Registered Server Program
Program ID	INRAYTRIGGER
Gateway-Host	(no entry!)
Gateway-Service	(no entry!)

By calling this function from within SAP system, SAP system will trigger OPC Router:

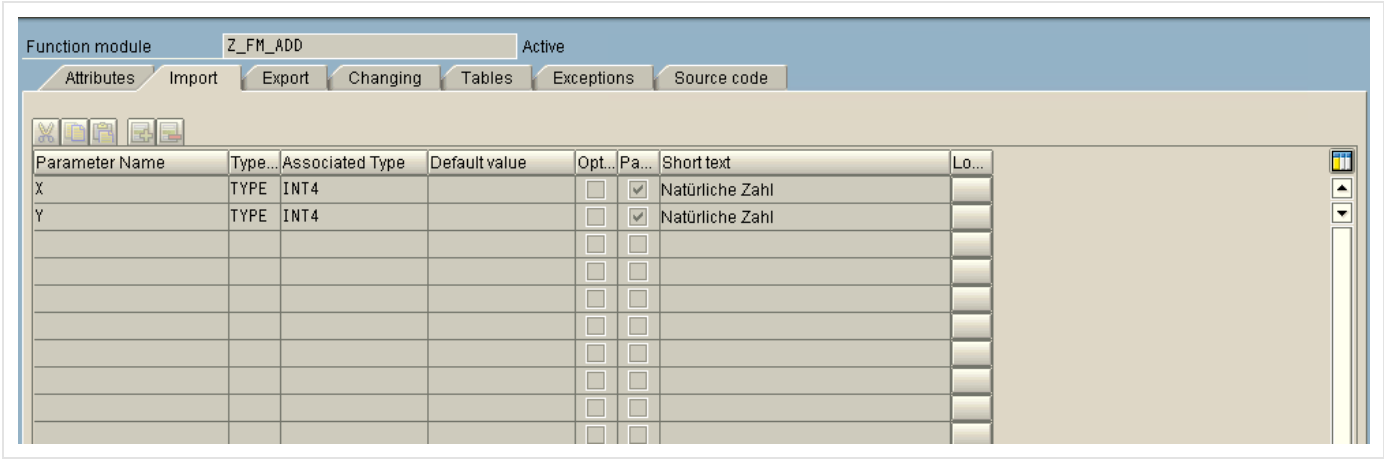
```
Call function 'TRIGGERFUNCTION' destination 'OPCROUTER'.
```

SAP examples

SAP-RFC Transfer Object example

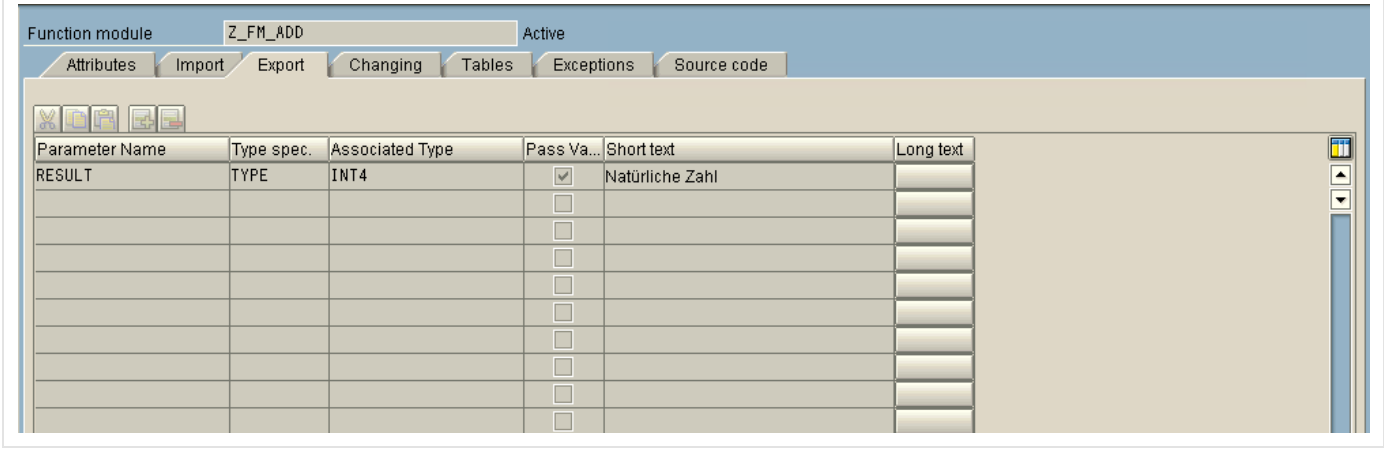
For example, we have a function module with the name Z_FM_ADD, whose import, export and source code can be seen in the following illustration.

Import:



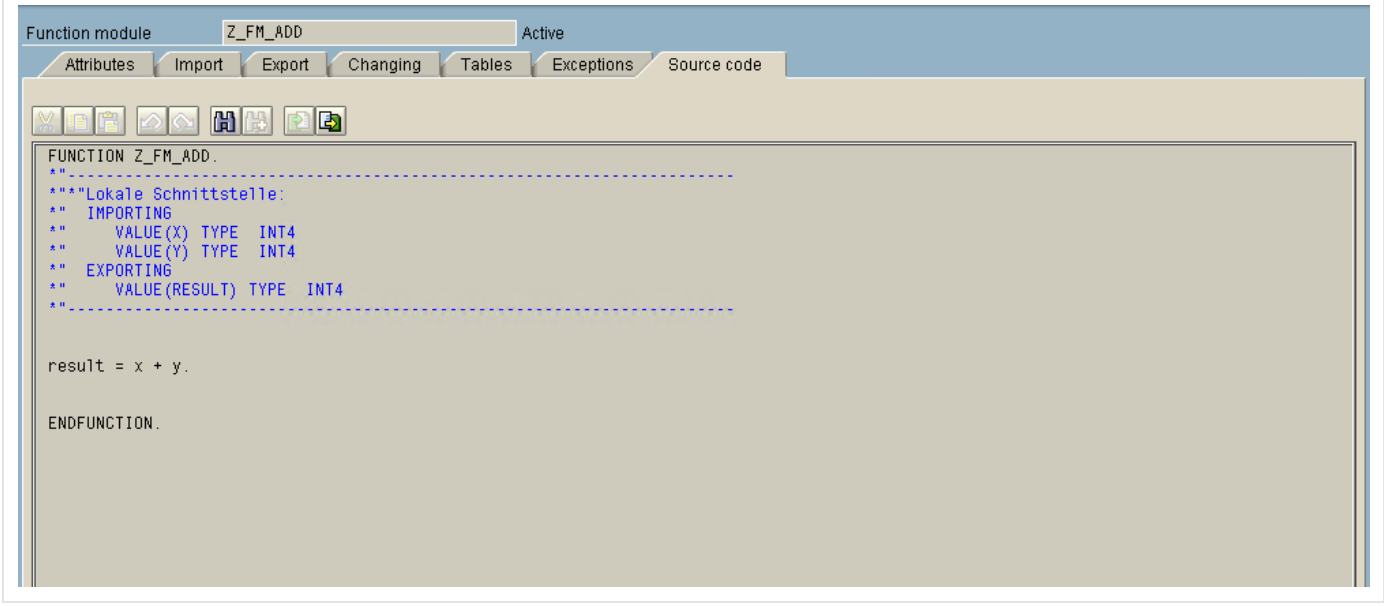
Import

Export:



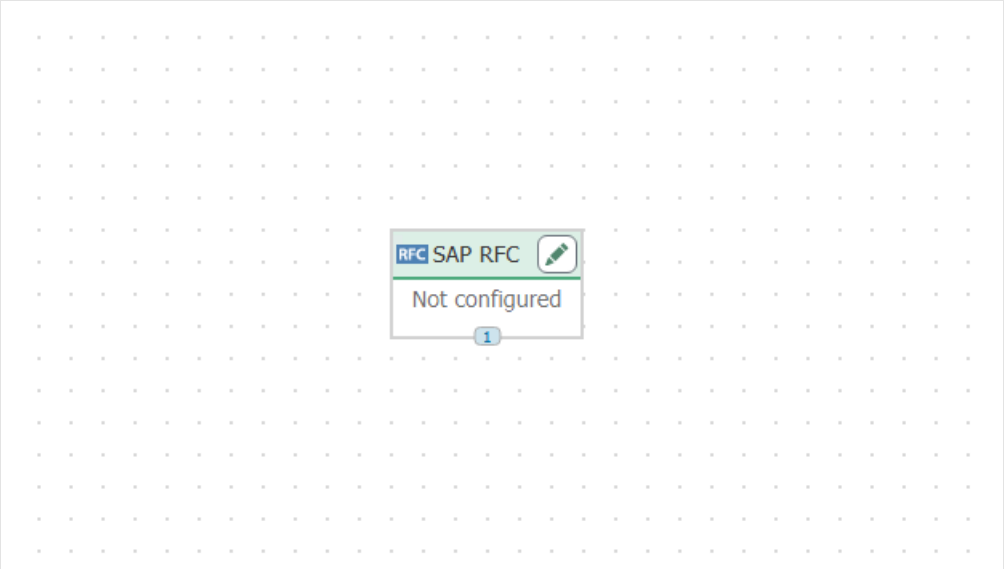
Export

Source code:



Source code

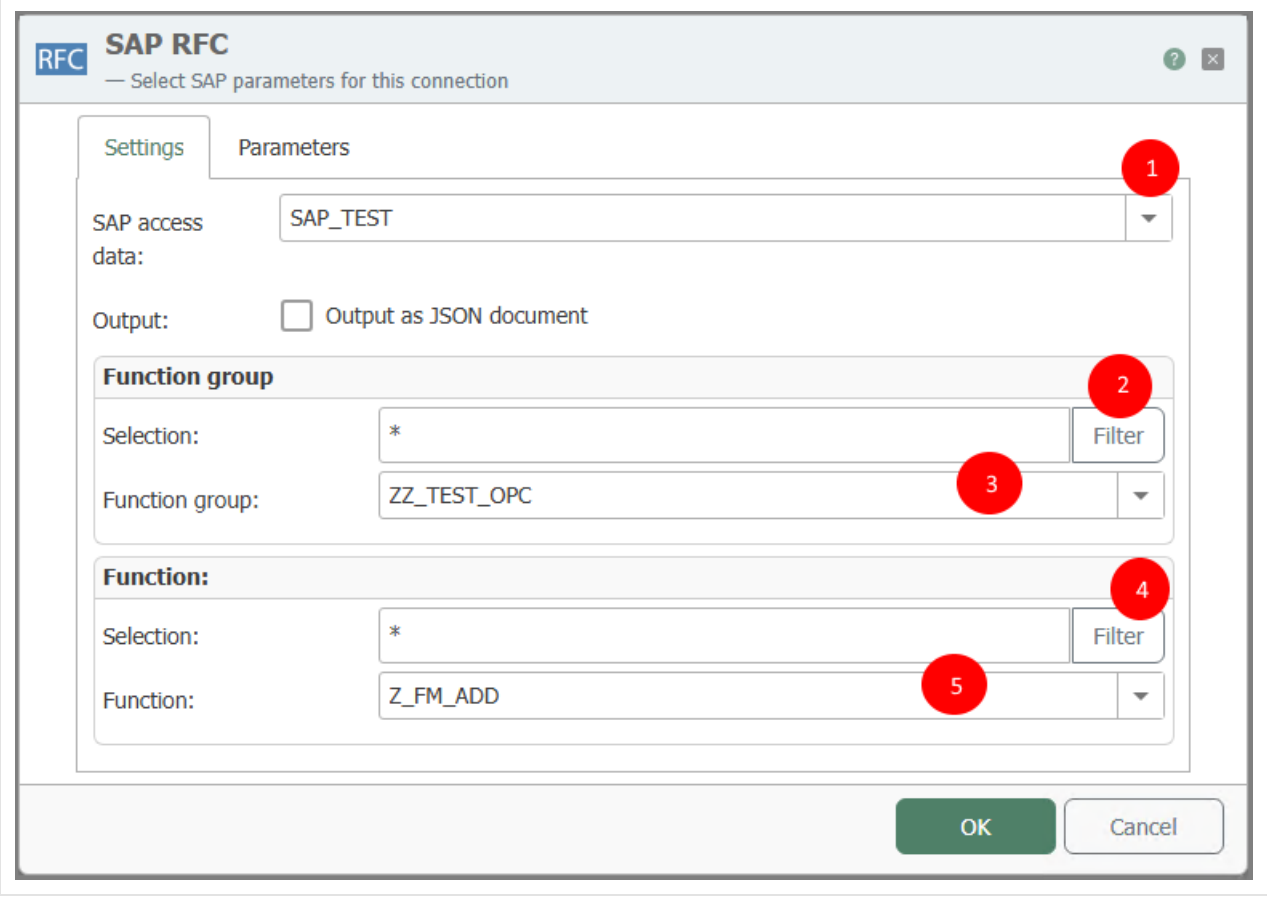
In the OPC Router, select transfer object SAP-RFC for a new connection.



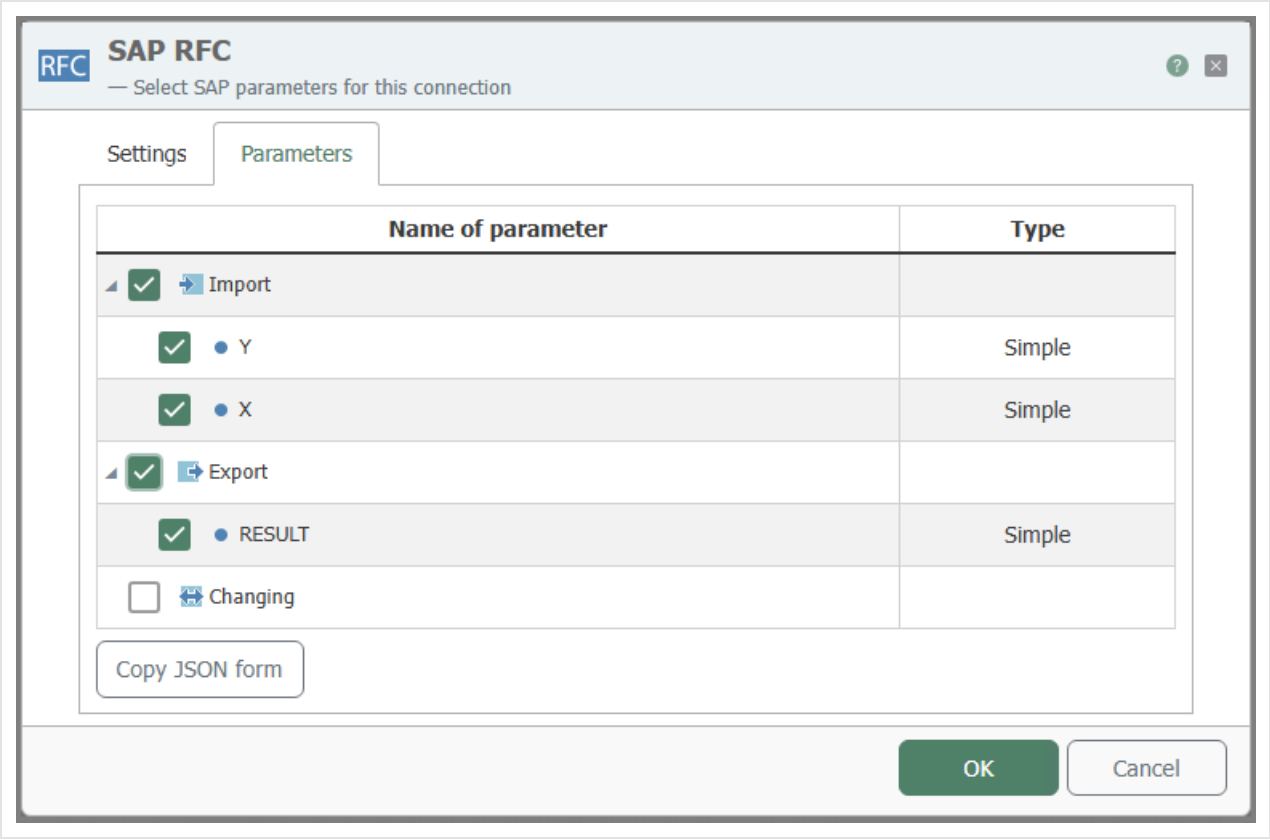
In this example, the Z_FM_ADD module function is located in the ZZ_TEST_OPC function group, and a connection called SAP_TEST was created in the SAP plug-in to access it.

Double-click to open the SAP RFC transfer object and carry out the following steps on the “Settings” tab:

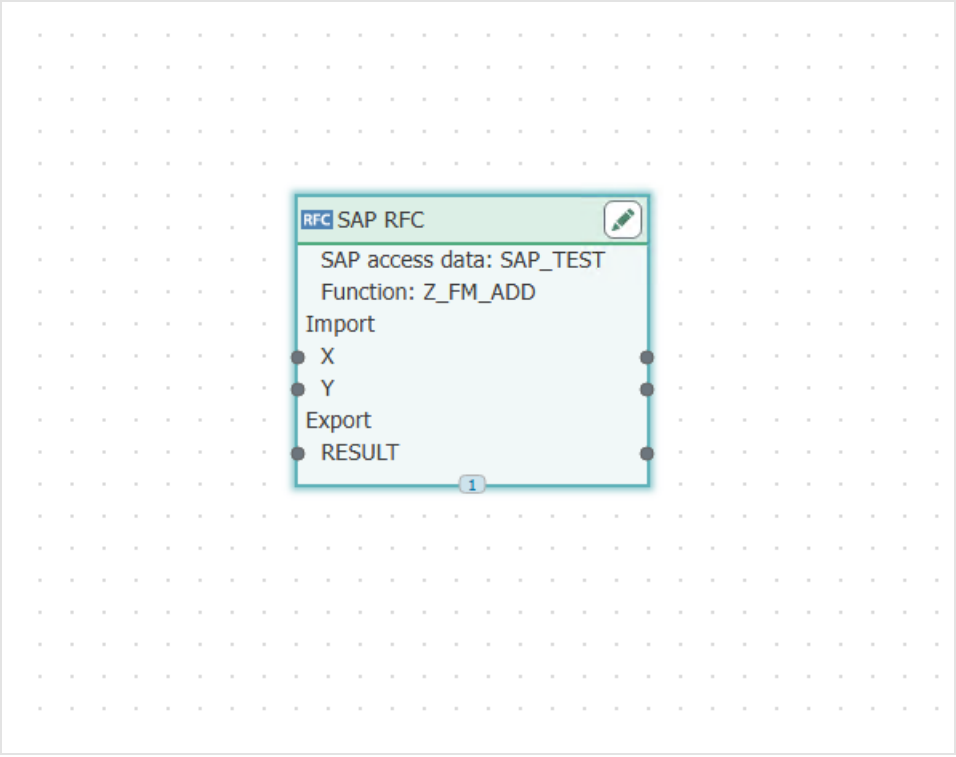
1. Select SAP_TEST in the “SAP access data” field.
2. Click on the “Filter” button in the “Selection” field in the “Function group” area to load the function groups.
3. Select ZZ_TEST_OPC in the “Function group” field.
4. In the “Function” area, click on the “Filter” button in the “Selection” field to load the functions.
5. Select Z_FM_ADD in the “Function” field.



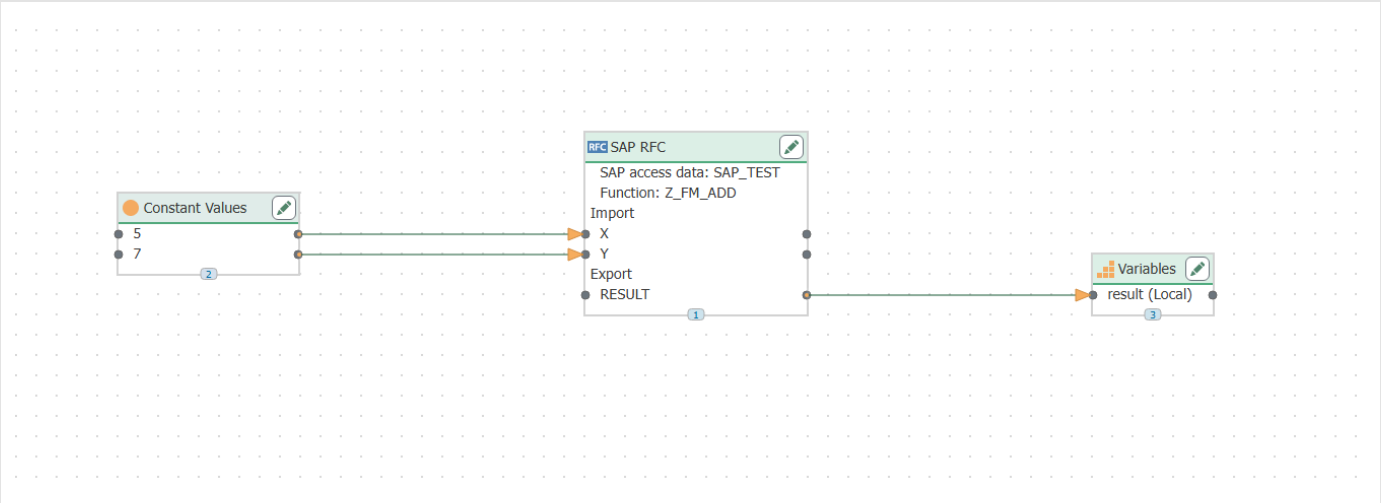
Select imports and exports on the “Parameters” tab.



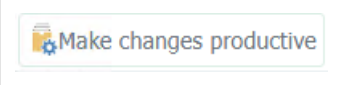
After you have clicked on “OK”, the transfer object SAP-RFC looks as follows:



As shown in the image below, we set “X” and “Y” to constants and enter “RESULT” in a variable.

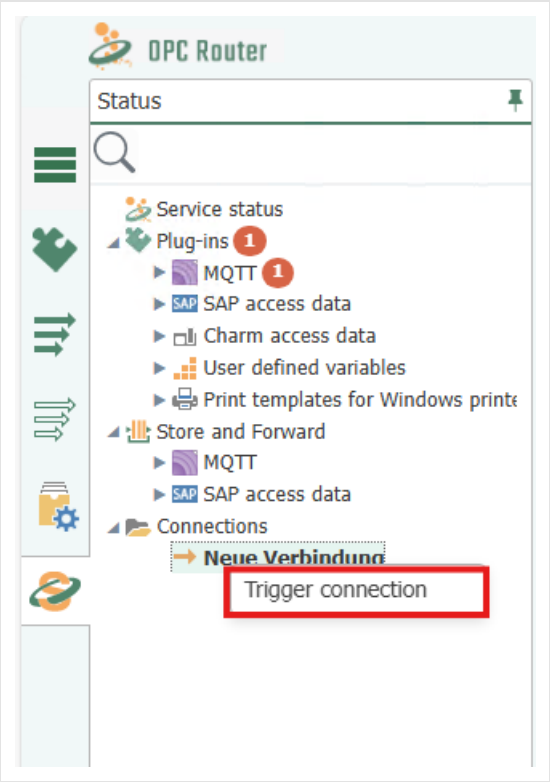


Then click on “Make changes productive” at the top of the page.

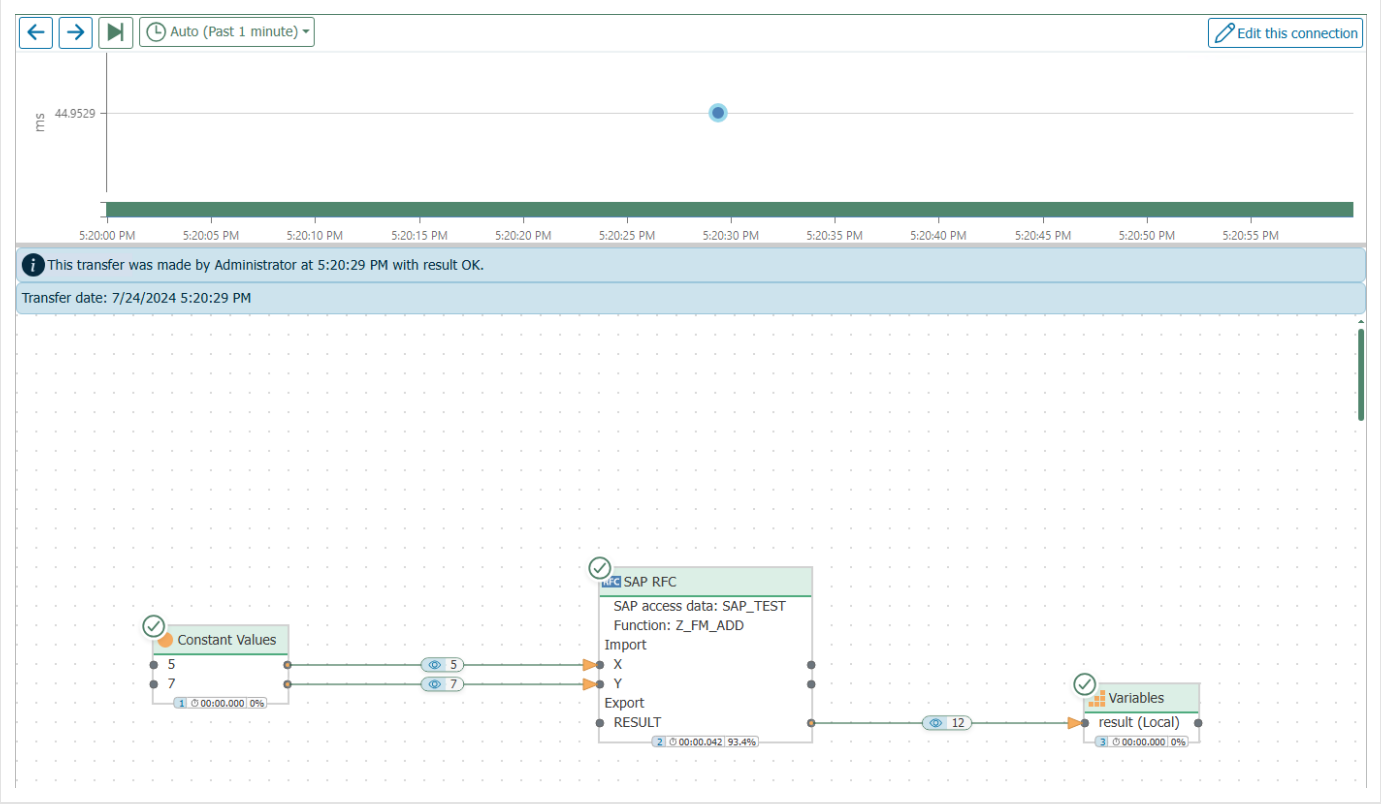


And in the “Publish” area, click “Make changes productive” again.

Then trigger the connection in the Status Indication by right-clicking on your connection and then clicking on “Trigger connection”.



Finally, as expected, the value of “RESULT” is equal to the sum of “X” and “Y”.



ERP Web Services

Via ERP plug-in, you connect to all systems providing a webservice (WSDL) interface.

Defining access data:

1. Open plug-in by clicking “ERP web services” at the plug-in-menu.
2. Chose “Add new set of access data” or double-click an existing entry to edit.
3. Specify access data. (chapter “Plug-in”).
4. Confirm with “OK”.

This set of access data can now be used to define a connection.

With the button “Test” you can check your connection.

ERP Web Service Plug-in

Set the following properties in the ERP plug-in configuration:

Name	Any name under which this web service appears in the connection configuration. Select a name you will recognize!
WSDL-URL	URL of the WSDL file.
Endpoint (optional)	If required, enter an endpoint for your connection.
Protocol	Selection of a transmission protocol from the drop-down menu.
Log-in	If you need to log in to the web service, select the desired type here. The details are available from the administrator of the web service / ERP system. Depending on this property, you can make additional settings: • No authentication: no authentication required, web service freely accessible. • Basic authentication: Login with username / password. • in web service method: login via a web service method. Selection and setting via the button “Log-in parameters” • Windows authentication: Login with Windows (domain) users. When using Integrated Security, make sure that the user context under which the service is running is authorized. The computer with the web service must be a member of the domain, not necessarily at the computer with the OPC Router.
User	Username to log on web service (at basic and NTLM authentication)
Password	Password to log on web service (at basic and NTLM authentication)
Windows authentication	When enabled, the user data where the router service is running.

![Telefonie](</assets/ERP_Plug-in (1).png>)

Authentication in Method

If you use the policy “in web service method”, please set log-in parameters via button “Log-in parameters”.

ERP Web Services Transfer Object

Webservices are both available as destination (input parameters) or source (output parameters). Create an ERP object as described above.

![DB-Transferobjekt](</assets/ERP_TO (1).png>)

Double-click on transfer object to set the following properties:

ERP access data	Select one of the configured web service sets of access data.
Method	Select the method to be called.
Wsdl URL dynamical	Enable to pass the URL dynamically at runtime. However, the method and the parameters must be identical.
Timeout	Time (in ms) that the connection may take before a connection error is returned.

All input and output parameters of the webservice can be used.

Parameter of Type “Array”

Parameters of type “Array” offer two possibilities:

- Activate the property Size of array dynamic to read all elements within the array.
- Limit the number of elements read by defining an number of nodes. The number of nodes corresponds to the number of elements reads. Attention: The webservice, not the OPC Router defines which elements are delivered.

    1. Leave property Size of array dynamic deactivated.
    2. Select the array.
    3. Click on “Add node”. These nodes appear within the array.
    4. Add as many nodes as desired.

        Remove a node by selecting it and click on “Remove node”.

Nested classes with parameters do not support dynamic parameters. Do you still want to use nested classes so must be defined before how many elements the array has since all nodes have to be created by themselves.

Data type “Integer”

The type “integer” represents by XSD-standard an infinite number that can only be stored as a string. If an integer has to be passed, the web service must use the “int” data type, which is read out correctly.

Shopfloor

All plug-ins to devices or machines, which can be used e.g. in workshops.

OPC-UA-Server

OPC Client

OPC-UA-Server

The OPC UA server plug-in publishes its own OPC UA server.

OPC-UA Server Plug-in

Set the following properties of the OPC UA server:

Server properties

NAME	**ANY NAME UNDER WHICH THIS CONNECTION APPEARS IN THE CONNECTION CONFIGURATION. CHOOSE A NAME THAT IS CLEARLY RECOGNIZABLE FOR YOU!**
Port	Select the port via which the OPC UA server is to be reached.
HTTPS port	Selecting the first checkbox adds an HTTPS address to the endpoint address, and its port can be specified in the input field. Selecting the second checkbox changes the HTTP address added in the Endpoint Address section to the Web Management endpoint.
Sample Rate (ms)	Enter the “Sample rate” of the OPC UA server in ms. After this time has elapsed, the values for the output data points are updated.
Server certificate	Select the server certificate. You can create a new certificate using the + sign. The certificates are managed via the OPC Router settings.
Trusted client certificates	Determine which client certificates are permitted. You can create and allow additional certificates via the central certificate management in the OPC Router settings. Alternatively, you can use the Windows certificate management.
Log-in	Specify whether anonymous clients can log on to the OPC UA server. (Limited security, if active)
Endpoint address	The OPC UA server will be accessible via the link displayed.

Security Options:

NONE	**UNENCRYPTED CONNECTIONS WILL BE ALLOWED.**
Basic128 Rsa15	Only connections that are Basic128RSA15 encrypted are permitted. You can use the drop-down field to define whether a signed certificate is sufficient or whether it must also be encrypted.
Basic256 Sha256	Only connections that are Basic256 encrypted are permitted. You can use the drop-down field to define whether a signed certificate is sufficient or whether it must also be encrypted.

Output data:

GLOBAL VARIABLES	**IF ACTIVATED, THE VARIABLES OF THE OPC ROUTER ARE OUTPUT AS DATA POINTS.**
Performance data	If activated, some performance data of the OPC Router Service is output.

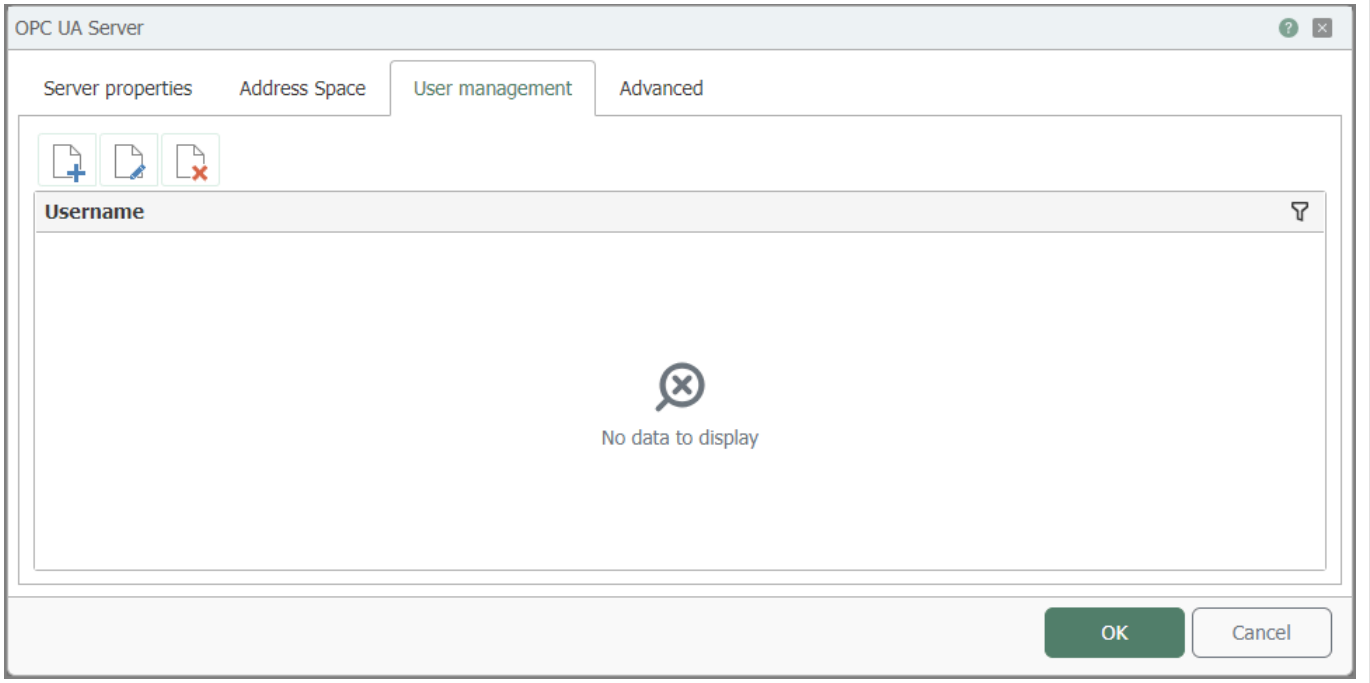
Address space:

Define the data point for your OPC-UA server.

Use the Column “Name” for a random name. Also, you can define folders an objects.

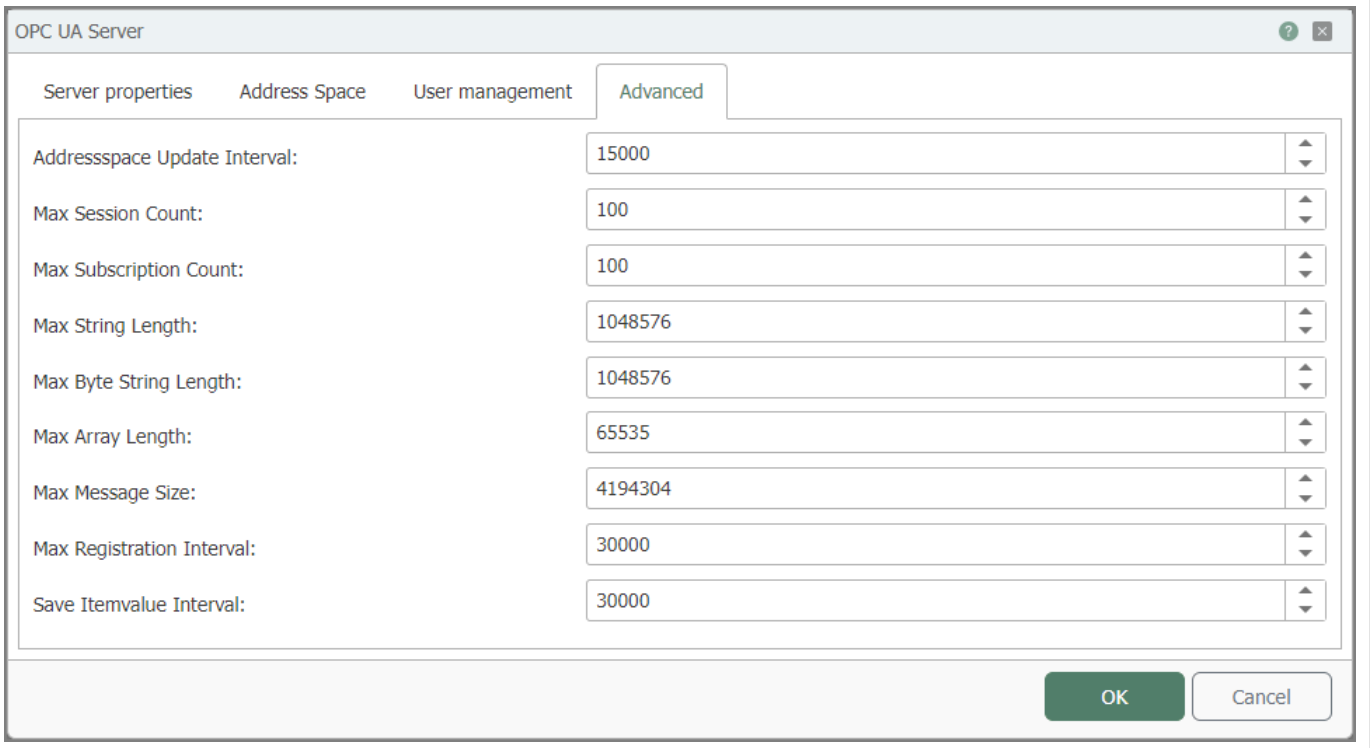
Additionally, you can define default values.

User management:



Use the “User management” to define some users and specify rules for read and write permissions.

Advanced



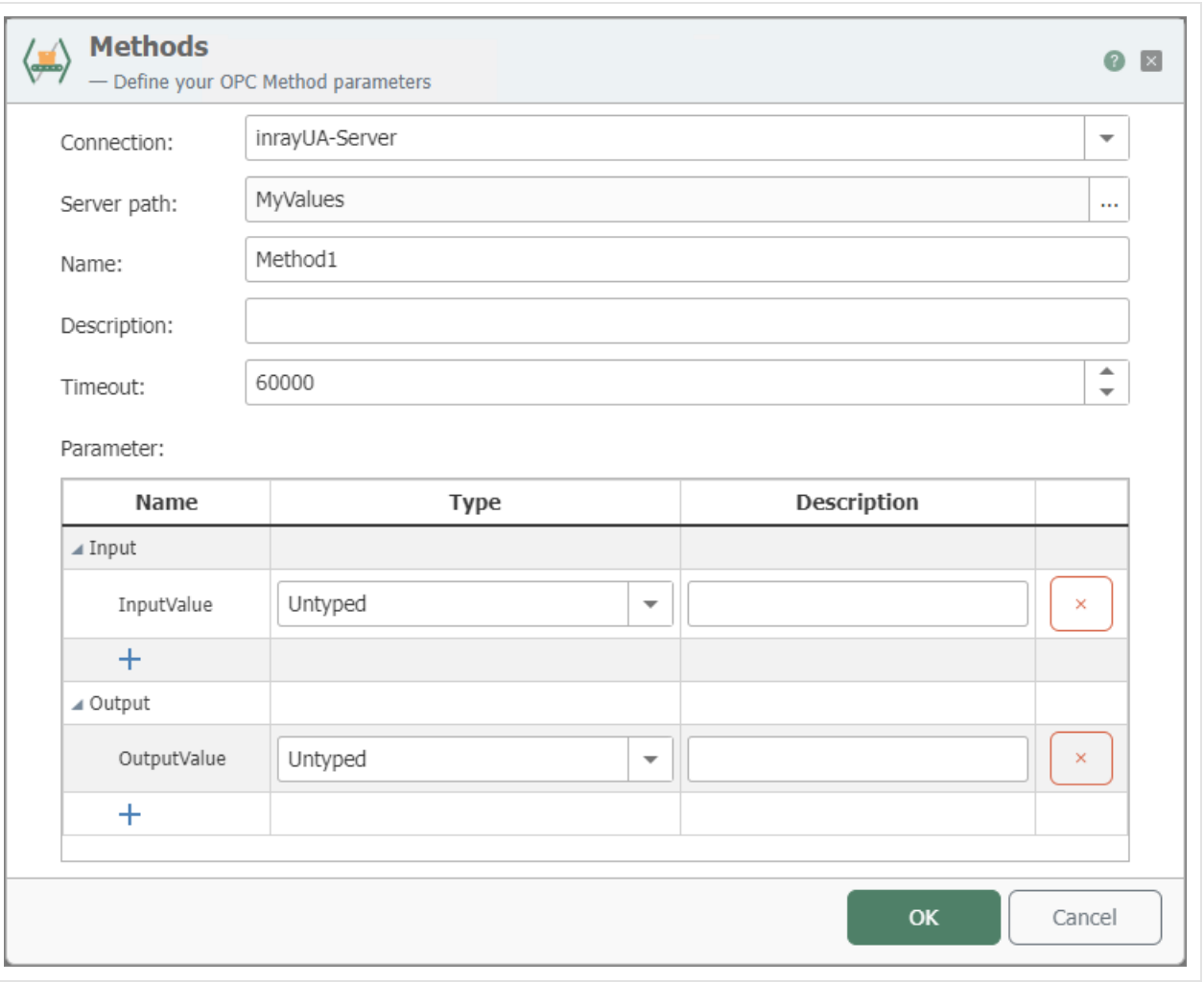
ADDRESS SPACE UPDATE INTERVAL	SPECIFIES THE INTERVAL AT WHICH THE ADDRESS SPACE IS UPDATED.
Max Session Count	Limits the maximum number of sessions to the specified value.
Max Subscription Count	Limits the maximum number of subscriptions to the specified value.

ADDRESS SPACE UPDATE INTERVAL	SPECIFIES THE INTERVAL AT WHICH THE ADDRESS SPACE IS UPDATED.
Maximum string length	Limits the maximum string length (number of characters) to the specified value.
Max byte string length	Limits the maximum number of bytes that a byte sequence can contain to the specified value.
Max array length	Limits the maximum number of elements that an array can contain to the specified value.
Max Message Size	Limits the maximum size of a message (bytes) that can be sent via the OPC UA protocol.
Max Registration Interval	Specifies the maximum interval in which the client must register with the server.
Save Itemvalue Interval	Specifies the interval at which the item values are saved.

Note: A server always starts with an object. Below this, the structure can be freely defined.

OPC methods trigger

By the OPC Methods trigger, users can use the OPC Router to execute functions on the OPC Router.



The individual properties are:

Connection	Select the internal OPC UA server through which the method can be accessed.
Server path	Select the path under which the method is to be found. You can click the button with three dots "..." to use the Tag Browser.
Name	Select a name for your method.
Description	Enter a description for your method.
Timeout	After reaching the timeout (ms) the transfer will get canceled.

Parameter

Input and Output Parameter

Input	Define some input parameters. Set the data types and if you want a description for them.

Output	Define some output parameters. Set the data types and if you want a description for them.

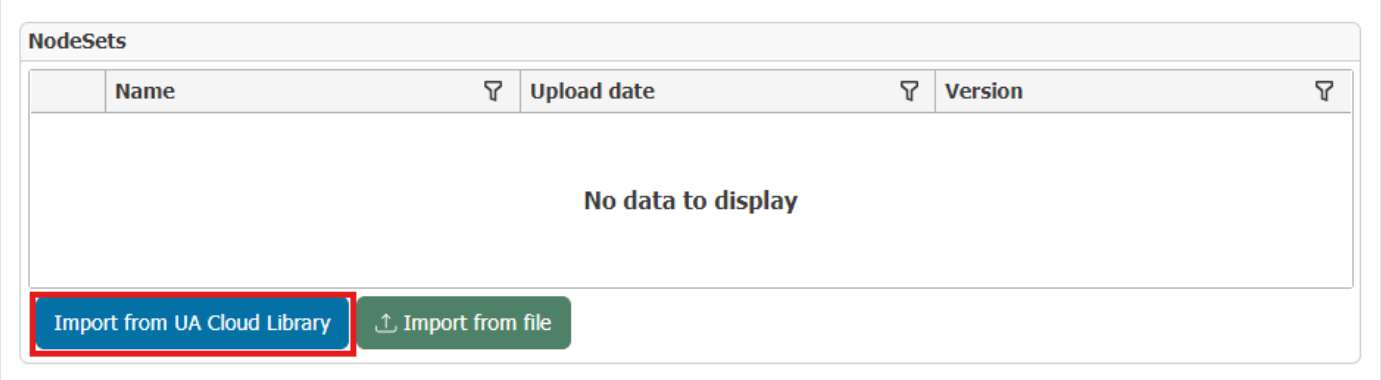
NodeSets

The node sets can be found in the OPC UA Server on the Address Space tab.

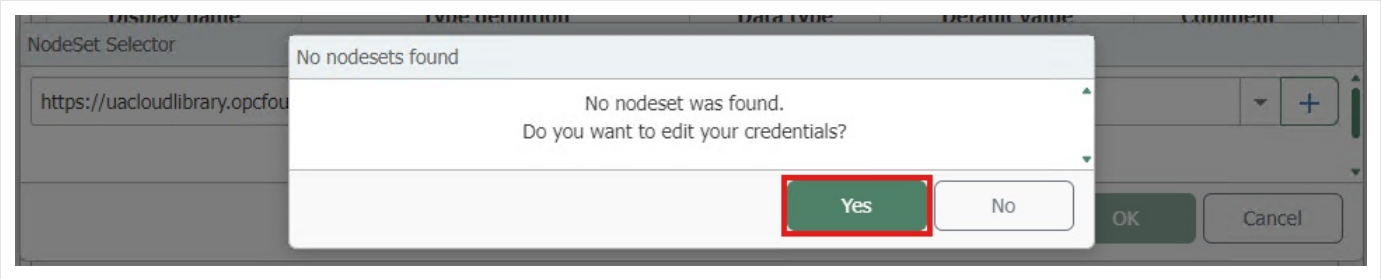
Import from the CloudLib

To use a NodeSet that you do not yet own, you can download it from the CloudLib.

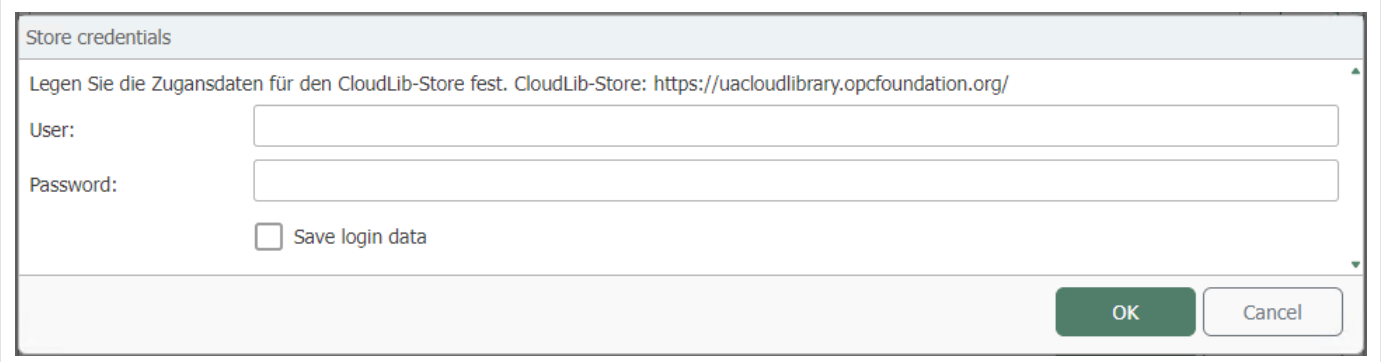
To do this, click on “Import from UA Cloud Library”.



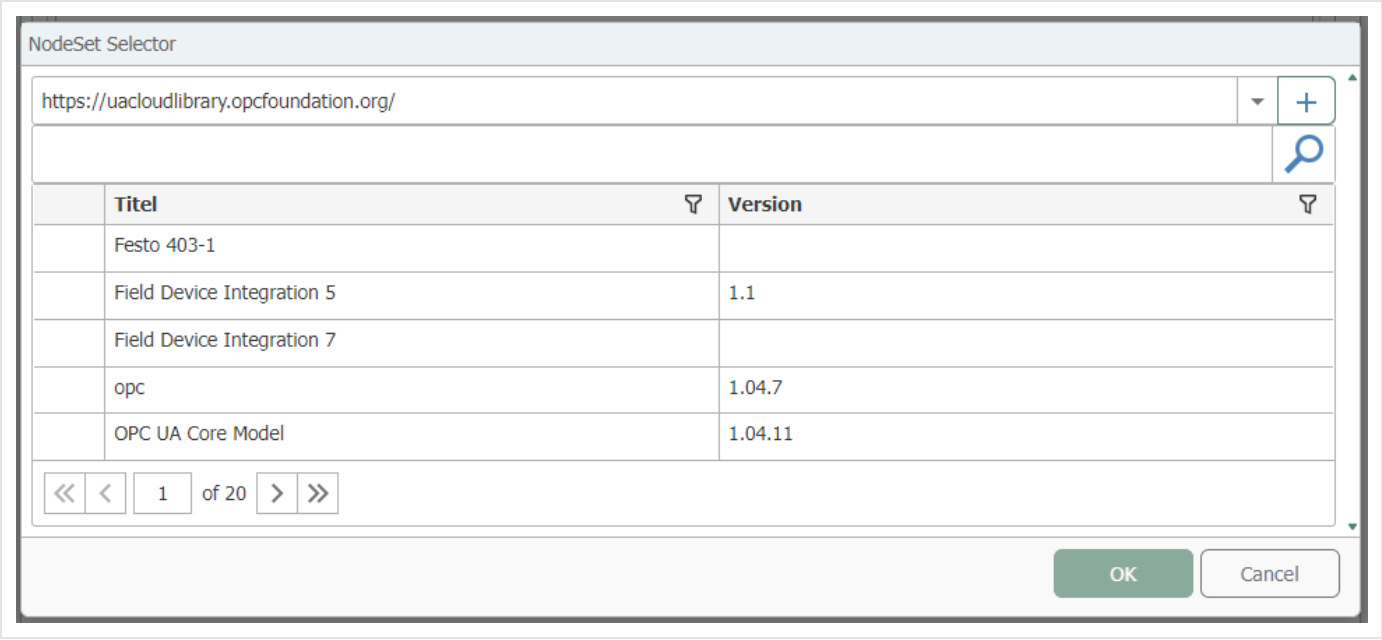
click on “Yes”.



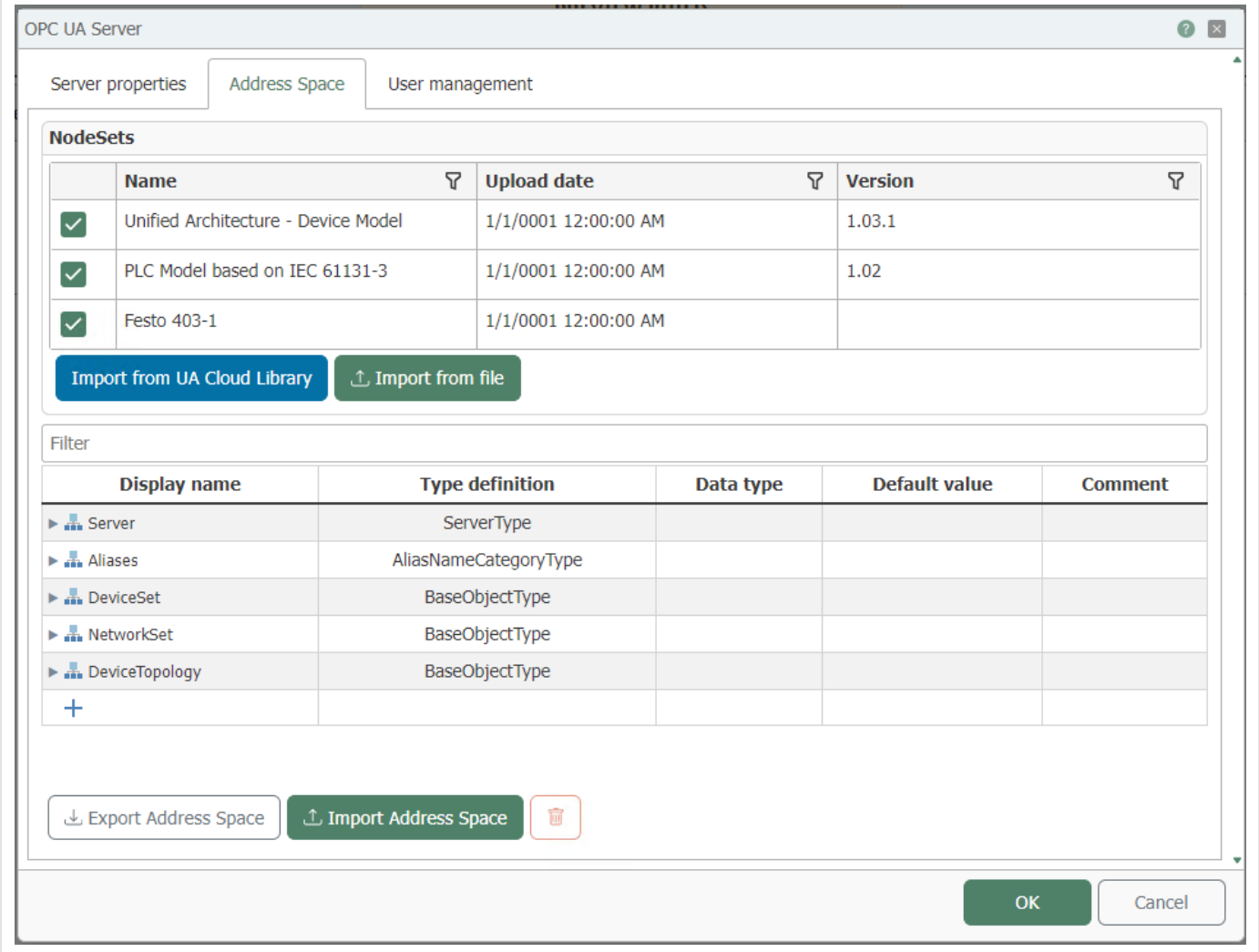
If you are already registered with the OPC Foundation, please enter your username and password. Otherwise, go to https://uacloudlibrary.opcfoundation.org and register.



After entering the username and password and clicking OK, the “NodeSet Selector” window opens. Select the desired NodeSet and click OK.

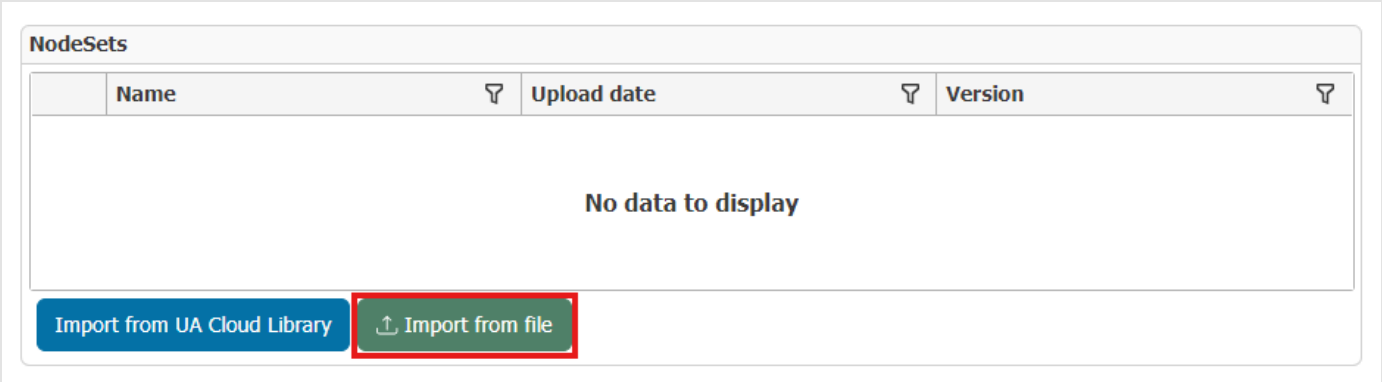


By changing the checkboxes, you can see that NoteSets are added to or removed from the tree.



Import file

Select a file here which you would like to import.



Note: Please note that this must be an XML file.

OPC Client

With the client data can be fetched from the OPC-Server.

OPC UA

- ... for OPC UA the router has to be approve as “Trusted Client”, if an appropriate safety level is set on the OPC server

1. Open plug-in by clicking “OPC-UA Client” in the project configuration’s plug-in list.
2. Click the plus button or double click on the existing entry.
3. Specify access data. (chapter “Properties of OPC UA access data”).
4. Use the “Check connection” button to initialize a connection. If a corresponding safety level is set on the OPC server, the connection will fail. You must then release the OPC router on the OPC server as a trustworthy client. Reinitialize your OPC server, if necessary. Test again to verify successful connection setup.
5. Confirm with “OK”.

The OPC UA can now be used to define a connection.

Introduction to OPC UA

The **OPC Unified Architecture (OPC UA)** is a platform-independent communication standard designed to facilitate reliable data exchange between various devices and systems across different industries. It provides a flexible and scalable framework to enable interoperability through standard data models and communication protocols.

High-Level Capabilities in OPC Router

The **OPC UA Client** plugin within OPC Router acts as a key communication link between OPC UA servers and the various systems connected through OPC Router. This plugin, along with triggers and transfer objects, provides the following capabilities:

- **Seamless Data Transfer**: OPC UA enables both reading and writing of values between systems through its client-server model. The **Transfer Objects** in OPC Router make it possible to define how data is mapped between systems, allowing customized transformations and routing of information.
- **Subscription Management**: The plugin can create **Subscriptions** to monitor data changes or events efficiently, minimizing network load while ensuring systems receive timely updates.
- **Trigger Mechanisms**: **Triggers** in OPC Router can react to data changes, such as value updates or specific conditions. This allows for event-driven workflows where actions are initiated based on data from OPC UA servers.
- **Security and Reliability**: OPC UA is designed with built-in security features, including encryption, user authentication, and redundancy support. These features, managed via OPC Router's security settings, ensure that all communication remains safe and continuous, even in complex environments.

Configuration Overview

The OPC UA Client plugin connects OPC Router to all accessible OPC Servers. If an OPC Server runs on a different computer, the OPC Router must be approved as a "Trusted Client" if an appropriate safety level is set on the OPC server.

The operation is straightforward:

- **Open the Plugin**: Double-click "OPC UA Servers" to open the list of created OPC connections.
- **Add or Edit Connections**: Choose "Add new set as access data" or double-click an existing entry to edit.
- **Specify Access Data**: Enter the necessary access data (refer to the chapter "Properties of OPC UA access data").
- **Check Connection**: Use the "Check Connection" button to initialize a connection. If a corresponding safety level is set on the OPC server, the connection may fail. In this case, approve the OPC Router on the OPC server as a trusted client. Reinitialize your OPC server if necessary, and test again to verify successful connection setup.
- **Confirm and Use**: Confirm with "OK". The OPC server can now be used to define a connection.

opc_ua_plug_in.md

Overview of Triggers and Transfer Objects

Triggers and **Transfer Objects** are core components of OPC Router that facilitate efficient data management and workflow automation between connected systems.

Triggers Overview

Triggers are mechanisms that drive actions in response to specific conditions or events. They enable event-driven workflows, making it possible for OPC Router to respond immediately to data changes. Key types of triggers include:

- **Threshold Trigger**: Activates when a value crosses a specified threshold, which is useful for monitoring key performance indicators or critical values.
- **Bit Trigger**: Reacts to changes in specific bits within a data point, often used in scenarios where bit-level control or monitoring is required.
- **DataChange Trigger (DC Trigger)**: These triggers activate when a data point in an OPC UA server changes, allowing workflows to react in real time to updated information.
- **Multi DataChange Trigger**: Monitors multiple data points and triggers actions when any of them change, providing more comprehensive monitoring capabilities.
- **DataChange Missing Trigger**: Used to detect when expected data changes are missing, which can be critical for ensuring timely updates in certain processes.
- **Message Trigger**: Reacts to messages received from the OPC UA server, often used for logging or notification purposes.
- **OPC Event Trigger**: Triggers actions in response to events from the OPC UA server, enabling event-driven workflows based on server-side occurrences.

Transfer Objects Overview

Transfer Objects (TOs) define how data flows between different sources and targets. They provide a structured way to map, transform, and transfer data efficiently within OPC Router's ecosystem.

- **OPC Data Access Transfer Object**: This Transfer Object allows OPC data points (nodes) to be read and written, providing basic data mapping and transfer capabilities between systems.
- **Structured Transfer Object**: Facilitates the handling of complex data structures, ensuring data is correctly mapped between systems that use structured or nested data formats.
- **Message Transfer Object**: Used for transferring message-based data, often applicable in scenarios where data is communicated in discrete messages rather than continuous values.
- **Methods Transfer Object**: Allows the invocation of methods on the OPC UA server, providing capabilities to execute server-side commands or functions as part of the data workflow.

Triggers and Transfer Objects work in tandem to create robust, automated data workflows that improve efficiency, reduce manual interventions, and enhance data consistency across interconnected systems.

Background Information from OPC UA Specifications

- **Address Space Model**: OPC UA's Address Space is fundamental for representing information. Nodes, such as variables and objects, create a structured and navigable representation of the data, allowing the OPC Router to easily interact with data in a meaningful way.
- **Quality of Service (QoS)**: The quality of data values is crucial in industrial systems. Values are labeled as "Good", "Uncertain", or "Bad" based on their reliability. OPC Router uses these quality attributes to decide on data transfer, avoiding errors when values are deemed invalid.
- **Security and Authentication**: Security is integral to OPC UA, with multi-layer encryption and authentication protocols. OPC Router's **Security Tab** ensures that only authorized users and trusted certificates are used, preventing unauthorized access.
- **Subscription Management**: OPC UA subscriptions allow for efficient monitoring of data changes. This mechanism is particularly useful for applications requiring up-to-date information without the performance hit of continuous polling.

Use Cases

The **OPC UA Client** plugin is highly versatile, enabling a variety of use cases that demonstrate its importance for Industry 4.0

implementations and seamless data integration:

- **OPC UA to Database Integration**: PLCs control machine sequences and processes. For secure communication, we recommend using a trigger with a handshake mechanism, such as the **Message Trigger** or alternatively the **Bit Trigger**. When using the **Message Trigger**, the payload (i.e., the data) is provided along with the trigger, which can then call a stored procedure in a database. The response to the PLC can also be securely transmitted using the response functionality of the Message Trigger, ensuring reliable communication back to the controller. With the OPC UA Client plugin, production data can be read from the PLC via OPC UA, transformed, and stored in an SQL database. Conversely, database data can be sent back to the PLC to adjust production parameters based on customer-specific requirements.
- **Edge Computing and Cloud Integration**: The OPC UA Client plugin can act as an edge computing unit, connecting OPC data sources in local networks, buffering data, and subsequently transferring it to higher-level systems such as ERP or cloud platforms via protocols like MQTT. When the OPC UA Client is used with a publish interval and the DataChange Trigger, high-frequency data changes (e.g., every 5 ms) can be seamlessly transferred to systems like MQTT, InfluxDB, or Kafka. This allows for rapid data processing, with achievable rates of over 2000 value changes per minute. This setup is ideal for environments where on-premises data preprocessing is required before cloud transfer.
- **Shopfloor Integration for Notifications**: The **Message Trigger** within OPC Router ensures communication reliability by acknowledging messages from machines on the shopfloor. For example, when a pallet is completed, the OPC Router can send an acknowledgment back to the PLC, allowing the machine to proceed with the next task. Any error during processing can also trigger a message back to the PLC to stop production and prevent further issues.
- **CSV Data Logging**: OPC UA can be used to read process values and log them into a CSV file for historical analysis. When data changes, it is logged into an existing CSV file or a new one created. This is ideal for scenarios requiring simple data tracking without complex database dependencies.

Summary

The **OPC UA Client** plugin in OPC Router provides extensive tools for connecting to and managing OPC UA server data, ensuring secure and reliable communication. Triggers and Transfer Objects extend the capabilities by enabling dynamic, event-driven workflows and detailed control over data routing and transformation. Through its well-structured tabs, users can set up and customize every aspect of the OPC UA interaction, optimizing data flow for different operational needs.

For more detailed step-by-step instructions on individual tabs and options, refer to the specific setup guide for each part of the configuration.

OPC-UA Client Plug-in

Quickstart Tab

The **Quickstart** tab allows for a quick and straightforward configuration of the basic connection parameters required to connect to an OPC-UA server. The key fields include:

- **Name**: A user-defined identifier for the connection, such as "KepServer" or "Aveva".
- **Endpoint**: The server endpoint URL (e.g., `opc.tcp://aveva:48033` or `opc.tcp://KepServer002.mshome.net:49320`, or an IP address). This field defines where the client should connect.

Note: If encryption is used, the endpoint must be specified exactly as it is stored in the server's certificate. For example, if the certificate contains "Server.domain.local", the endpoint must also be addressed in this way, and not, for example, via the IP. Otherwise, the certificate will be rejected by the OPC server.

- **Accesspath**: An optional field to specify an access path for the connection.

Important: When specifying an Access Path, only data points in the selected folder are displayed in the Tag Browser. You can use the button next to the Access Path to open the Tag Browser and select the folder. The Access Path can only be set if the "Browse Path" function is used for item identification.

- **Timeout (ms)**: Specifies the connection timeout in milliseconds. For example, `15000` ms allows the client 15 seconds to establish a connection.
- **Highest Security Level**: An option to enforce the highest level of security, ensuring that the connection is secured with the best available methods.
- **Authentication**: The type of user authentication used. The options are "Anonymous", "Username", or "Certificate".

Click **Test Connection** to verify the connection parameters before proceeding.

Connection Tab

The **Connection** tab provides additional details about how the client connects to the server. This includes specifying advanced connection paths or additional configuration details that may not be covered in the Quickstart tab.

Security Tab

The **Security** tab helps ensure the integrity and safety of the connection by defining and validating trusted certificates:

- **Trusted Certificates**: Specifies which certificates are trusted. You can import trusted (root) or trusted certificates in the OPC Router settings, or use the Windows/System certificate store to import the Root/Server certificate.
- **Certificate Validation**: Ensures that certificates are verified to add an extra layer of security.
- **Validate Server Application URI**: Validates the server's application URI to ensure correct identity verification.
- **Highest Security Level**: This checkbox ensures that the connection uses the best available security measures.
- **Client Certificate**: Select or create a certificate for the OPC Router to identify itself against the server. If you want to import an existing certificate, you can do this in the OPC Router settings under the Certificate section.
    - Use the **plus** sign to add a new certificate, and the **minus** sign to remove one. Another button allows you to display the properties of an existing certificate.

Redundancy Tab

Note: If you want to use the function to read data from a redundancy server, you need the Enterprise license.

The **Redundancy** tab allows you to configure redundancy settings to ensure a continuous connection:

- **Redundant Server**: The OPC Router can connect to redundant OPC UA servers. This involves specifying an additional server. It is assumed that the node structure and authentication are identical on both servers.
- **Endpoint and Heartbeat Item**: Define the redundant server endpoint and specify a heartbeat item to determine which of the servers is active. If this item has a BadQuality or False value, the other server is assumed to be active.

- **Activate Redundancy**: Activate redundancy settings if the primary server cannot be reached, or if the heartbeat item indicates an issue.
- Use the **Check connection** button to verify data connections in advance.

Subscription Tab

The **Subscription** tab contains settings that define the behavior of subscriptions:

- **Keep Alive Count and Retry**: Specifies how often the server should send keep-alive messages and the number of retries before disconnection.
- **Lifetime Count**: Defines how long a subscription remains active without receiving updates.
- **Priority**: Sets the priority level of the subscription.
- **Subscription Publish Interval**: Sets the interval at which the server publishes data changes.
- **Subscription Minimum Lifetime Interval**: Sets the minimum subscription lifetime interval.
- **Register OPC Tags on Start**: Automatically registers OPC tags at startup.
- **Enable Parallel Subscription Registration**: Enables more efficient subscription handling.

Browse Tab

The **Browse** tab configures how the client interacts with the server's address space:

- **Continue Browse Until Done**: Ensures that browsing continues until all nodes are retrieved.
- **Disable Browse Has Children**: Disables additional checks for nodes with children.
- **Max Browse Continuation Points Override**: Sets the maximum continuation points for browsing.
- **Max Browse References Returned**: Limits the number of references returned per browse operation.
- **Max Nodes Per Browse Override**: Sets the maximum number of nodes per browse operation.
- **Max Nodes Per Translate Browse Path Override**: Limits the number of nodes for translating browse paths.

Read Tab

The **Read** tab is used to configure the behavior for reading data:

- **Max Nodes Per Read Override**: Sets the maximum number of nodes per read request.
- **Read Cycle Warning Threshold**: Defines the threshold for triggering a read cycle warning.
- **Read Max Age**: Sets the maximum age for read values.
- **Parallel Read/Write Activation**: Enables simultaneous read and write operations, improving performance.
- **Transfer OPC null values**: If enabled, invalid OPC values will also be transferred (e.g., as NULL into databases).

Write Tab

The **Write** tab focuses on configuring data writing behavior:

- **Array Handling**: Specifies how arrays should be handled when writing data. Options include:
  - **Pass Original Array**: Send the incomplete, raw array exactly as it is.
  - **Fill Missing Values with Last Used**: Use the value of the most recent value for all missing elements.
  - **Fill Missing Values with Default**: The default value will be used for missing elements (e.g., NULL for INT).
- **Max Nodes Per Write Override**: Sets the maximum nodes allowed per write operation.
- **Max Batched Writes**: Controls the number of writes that can be batched together for efficiency.
- **Asynchronous and Grouped Writes**: Enables asynchronous and grouped writes to optimize data transfer.
- **Publish on Write Activation**: Immediately publishes data when written.
- **Parallel Read/Write Activation**: Allows for simultaneous read and write operations.

Advanced Tab

The **Advanced** tab provides additional settings for experienced users who want to fine-tune the connection:

- **Timeout and Sample Rate**: Set timeouts and data sampling rates to control the connection.
  - **Sample Rate**: Defines the rate (in ms) at which the OPC server refreshes values. Setting "0" will use the smallest

available update rate.

- **Allowed Qualities of Values**: If the quality of the values does not match the selected quality, this will result in a transfer abort in the OPC Router at runtime with an appropriate error message. For example, if "Bad" is allowed, a "" value may be transferred, which can lead to subsequent errors if this value is not valid for the target.
- **Addressing Method**: It is recommended to use "Browse Path" instead of Node ID for addressing. According to the OPC UA specification, it is not mandatory that the Node ID remains persistent between server restarts or reconfigurations. By using Browse Paths, the OPC Router resolves this during startup or when the connection is established (which may be slightly slower initially).

Note: The Browse Path option may be more beneficial when used with templates, although it may increase initialization time, especially in larger projects.

- **Access Optimization**: Enable RegisterNode access for improved performance. This setting registers the nodes and then uses handles from the register for subsequent communication. This reduces the payload size and optimizes the overall communication since the complete NodeId (which can be a long string for some servers) does not need to be transmitted each time.
- **Keep Alive Timeout Override**: Override settings for keep-alive messages.
- **Preferred Culture**: Set the preferred language or culture for communication.
- **Provider Implementation**: Select the OPC UA version for data type definitions. Options include "OPC 1.03 Data Type Dictionary" or "OPC 1.04 Data Type Definition".
- **Enable Local Process Communication**: This setting is used only when accessing the same OPC Router that also acts as the UA Server. In this case, the TCP/IP stack is bypassed, and the client communicates directly in-process with the UA Server, which can improve performance.

OPC Read Mode

The **OPC Read Mode** setting provides three different modes for reading data from the OPC server:

- **OPC Router Cache**: In this mode, the OPC Router itself is responsible for reading the data points. Data points are read at the configured "Sample Rate" and held in an internal cache. The reading process is handled by internally created groups or subscriptions, and an internal thread reads items using a DeviceRead operation from the OPC server. The "Read Time" in the OPC Router status for the transfer object reflects access to the internal OPC Router cache, not the OPC server itself. If the background update takes longer than the configured sample rate, the log file may display "Workcycle took longer than...".
- **Asynchronous (Recommended)**: In this mode, OPC ChangeNotifications are used, meaning the OPC server is responsible for reading the data points and notifying the OPC Router of changes. The OPC Router stores the values in its internal cache and provides them to transfer objects. The read times in the transfer object's status reflect access to the internal cache only. This setting can be combined with the Subscription Publish Interval and Max Notifications per Publish to receive "fast batched" value changes for the DataChange trigger.
- **Synchronous**: In this mode, a "Sync read from Device" is performed directly from the context of the OPC transfer object, bypassing all caches. The times in the transfer object's status reflect the actual access to the device.

Background Information from OPC UA Specifications

The OPC Unified Architecture (OPC UA) is designed as a platform-independent standard to facilitate communication between various systems and devices across all industrial domains. It allows seamless interoperability through standard models for data exchange and interaction, supporting both Client-Server and Publish-Subscribe (PubSub) communication models.

AddressSpace Model: The OPC UA AddressSpace is a core concept that defines how information is structured and accessed within a server. It is an organized collection of Nodes that represent real-world objects, making it easier for clients to understand the structure and interact with the data.

Quality of Service: OPC UA supports defining the quality of the values being transferred, which can affect the behavior of data exchange. For example, qualities like "Good", "Uncertain", or "Bad" influence how clients handle data and what actions are triggered based on data validity.

Subscriptions and MonitoredItems: The Subscription model in OPC UA allows clients to monitor changes to data or events from servers. The Subscription entity, along with MonitoredItems, is used to track data changes efficiently and notify clients about updates, reducing the need for constant polling.

Security and Redundancy: OPC UA has integrated a strong security model, which includes encryption, user authentication, and application-level security measures. The redundancy features allow multiple paths to maintain a connection, ensuring high availability even in the case of network or server failures.

OPC Classic

Attention: OPC Classic is not supported in Docker.

OPC plug-in connects OPC Router to all accessible OPC Servers.

If the OPC server is running on another computer in the network, a correct DCOM configuration is a prerequisite for OPC Classic.

Operation is standard:

1. Open plug-in by clicking “OPC Classic servers” in the project configuration’s plug-in list.
2. Click the plus button or double click on the existing entry.
3. Specify access data. (chapter “Properties of OPC Classic access data").
4. Confirm with “OK”.

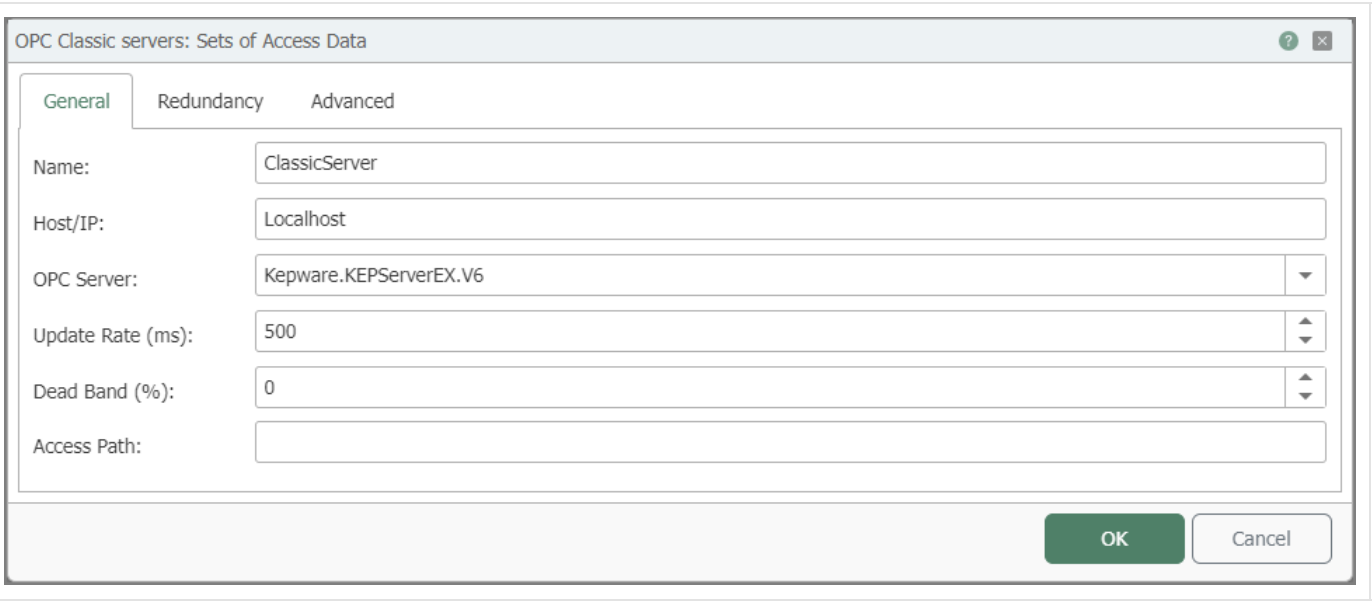
The OPC Classic can now be used to define a connection.

OPC Classic Plug-in

Note: If you want to use the function to read data from a redundancy server, you need the Enterprise license.

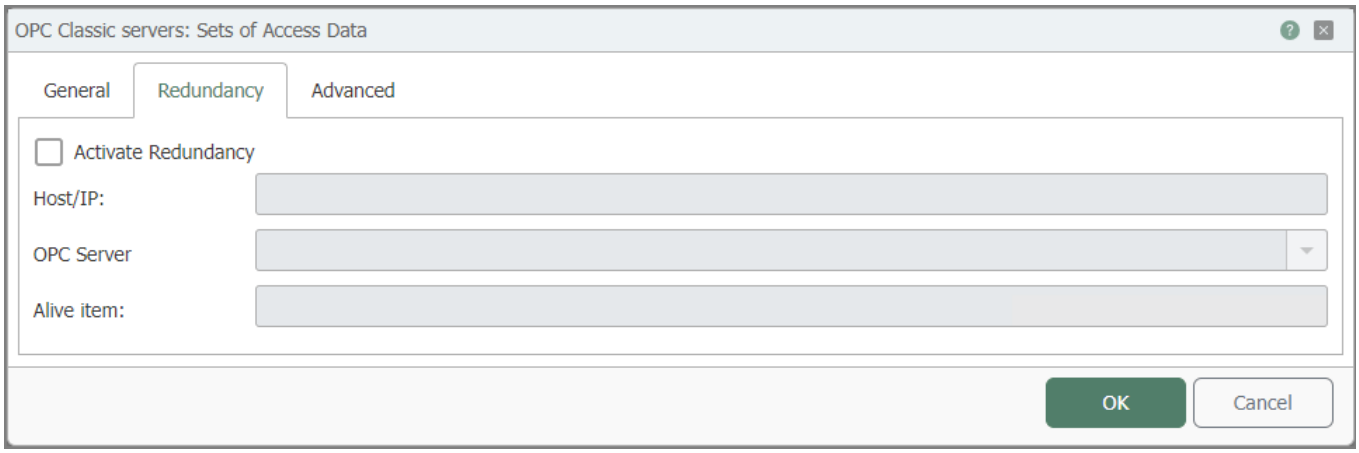
The following properties have to be specified for each set of OPC access data:

General



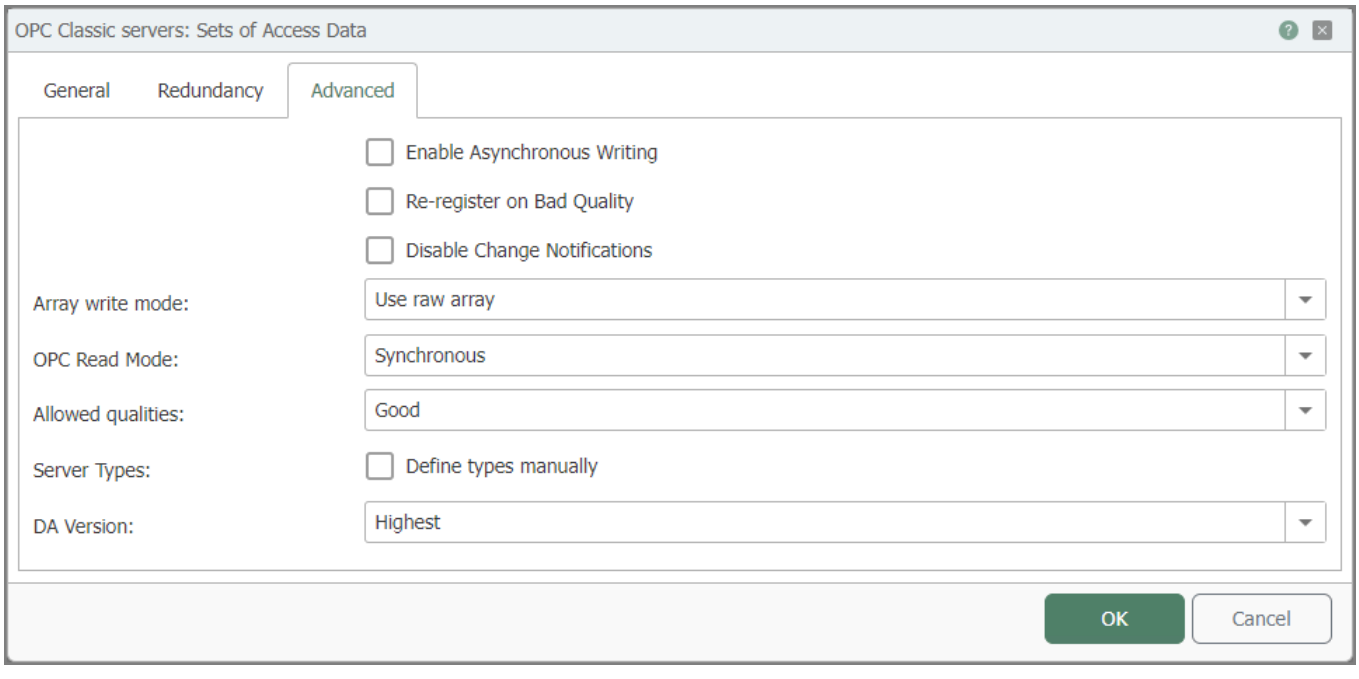
PROPERTY	DESCRIPTION
Name	This set's name. The name will be shown in projection. Select a name you will recognize!
Host/IP	IP or host name of OPC Server's computer.
OPC Server	OPC Servers accessible on computer specified under Host/IP. Select by drop-down
Update Rate (ms)	The update rate is the rate where the OPC server updates its data. Most OPC servers have an internal data memory and use the update rate to update the values in this memory. However, it depends on the OPC server which update rate is used. In the KepServer there is for example the setting Scan Mode. In the properties of the device you can set which scan rate should be used by the server and to what extent the update rate of the client is taken into account.
Dead Band (%)	The Dead Band property ensures that no transfer takes place from the OPC server to the client as long as the value does not exceed a certain threshold. This can be used to relieve the network. This function is optional and does not have to be implemented by all OPC servers.
Access Path	Access path on OPC Server

Redundancy



PROPERTY	DESCRIPTION
Activate Redundancy	Redundancy is enabled when the primary server is unavailable and (optionally) when the primary server's Alive item cannot be read or is set to 0.
Host/IP	IP or host name of OPC Server’s computer.
OPC Server	OPC Servers accessible on computer specified under Host/IP. Select by drop-down
Alive item	A specific node or tag that is monitored to ensure that the connection to the OPC server is still active. This field is used to periodically check the status of the server by reading the value of the specified "alive" element. If the value can be successfully read, it indicates that the server is still responding and the connection is healthy.

Advanced



PROPERTY	DESCRIPTION
Enable Asynchronous Writing	Determines whether asynchronous writing is used. This allows data points to be updated without having to wait for a response from the server before further actions can be performed.

PROPERTY	DESCRIPTION
Re-register on Bad Quality	After an error during reading, the connection to the data point is re-established.
Disable Change Notifications	Deactivates notifications when values are changed. This is necessary for servers with version 1 or version 2 because they do not support change notifications.
Array write mode	The following options are available for the array write mode: • **Use raw array:** This writes the original array to the data point. • **Fill missing values with recent values:** If values are missing, they are filled with those that were last available. • **Fill missing values with standard values:** Here, the missing values in the array are overwritten with a default value. **Note:** The default value is always the default value of the respective data type. The default value for a string is “”.
OPC Read Mode	The OPC read mode defines how data is read out. The following options are available: • **OPC Router Cache:** Data is preferably read from the OPC router cache. This minimizes network traffic, but can result in the data not being the most up-to-date, especially if the values in the data source change frequently. • **Asynchronous:** All data from OPC transfer objects is requested asynchronously. This allows the OPC router to perform other tasks while waiting for a response, which can improve efficiency. • **Synchronous:** With this option, data is read directly from the data source as far as possible. This option generates more network traffic, but provides the most up-to-date data. Please note that this option is currently only implemented in the OPC transfer object.
Allowed qualities	Here you can specify which data qualities are acceptable for the values to be read or written. In OPC UA, the data quality indicates the reliability and validity of the data.
Server Types	Manual type definition in the transfer objects to tell the OPC server which data types are expected.
DA Version	The version of the OPC DA specification of the server. The following options are available: Highest, V3, V2, V1 The version of your OPC server can be found in the corresponding manual.

Note: Pay attention to the firewall of the involved computers. Especially if the OPC-Server and the OPC-Router are running on different computers, it is not possible to use the firewall because of the dynamic port allocation via DCOM. If it is not possible to deactivate the firewalls, please contact inray.

Use the Access Path property to enter only the OPC items later when configuring the connection. This option is particularly useful if you only want to enter the data points during project planning.

To do this, create a separate OPC connection for each controller and define a unique access path in the OPC server, which you specify here.

If you need to connect different controllers with identically named data points, you can also use this variant, e.g. in combination with the connection templates. You would then enter the OPC connection as a variable in the template and assign it accordingly when instantiating the connection.

Transfer objects

Overview over transfer objects:

OPC Data Access

OPC Structures

OPC Messages

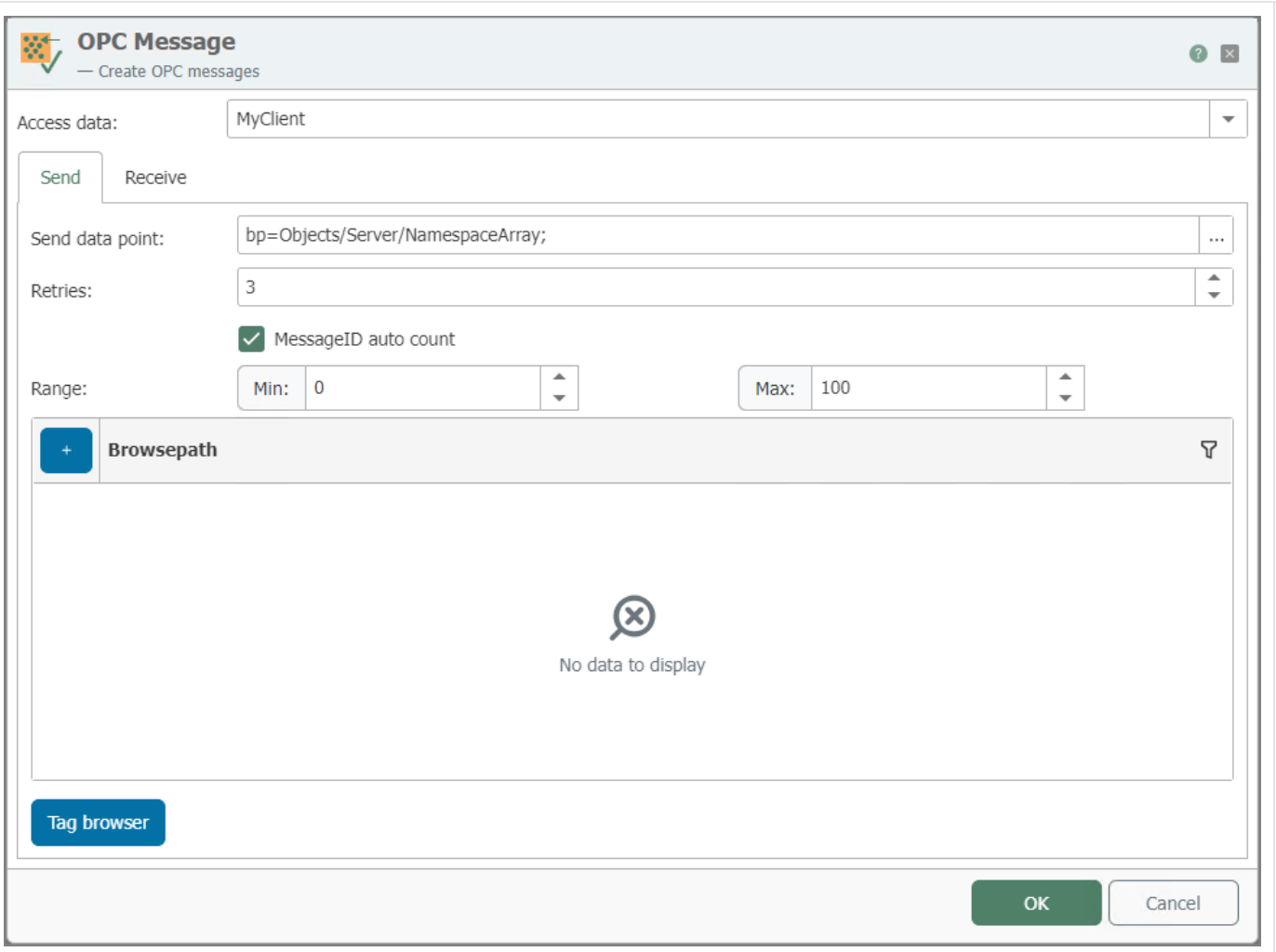
OPC Methods

OPC Message

Collect data in a request and send it to a controller. The controller responds in another data point. Create an OPC message transfer object within the connection as described.

Set the following properties after double-clicking on the transfer object:

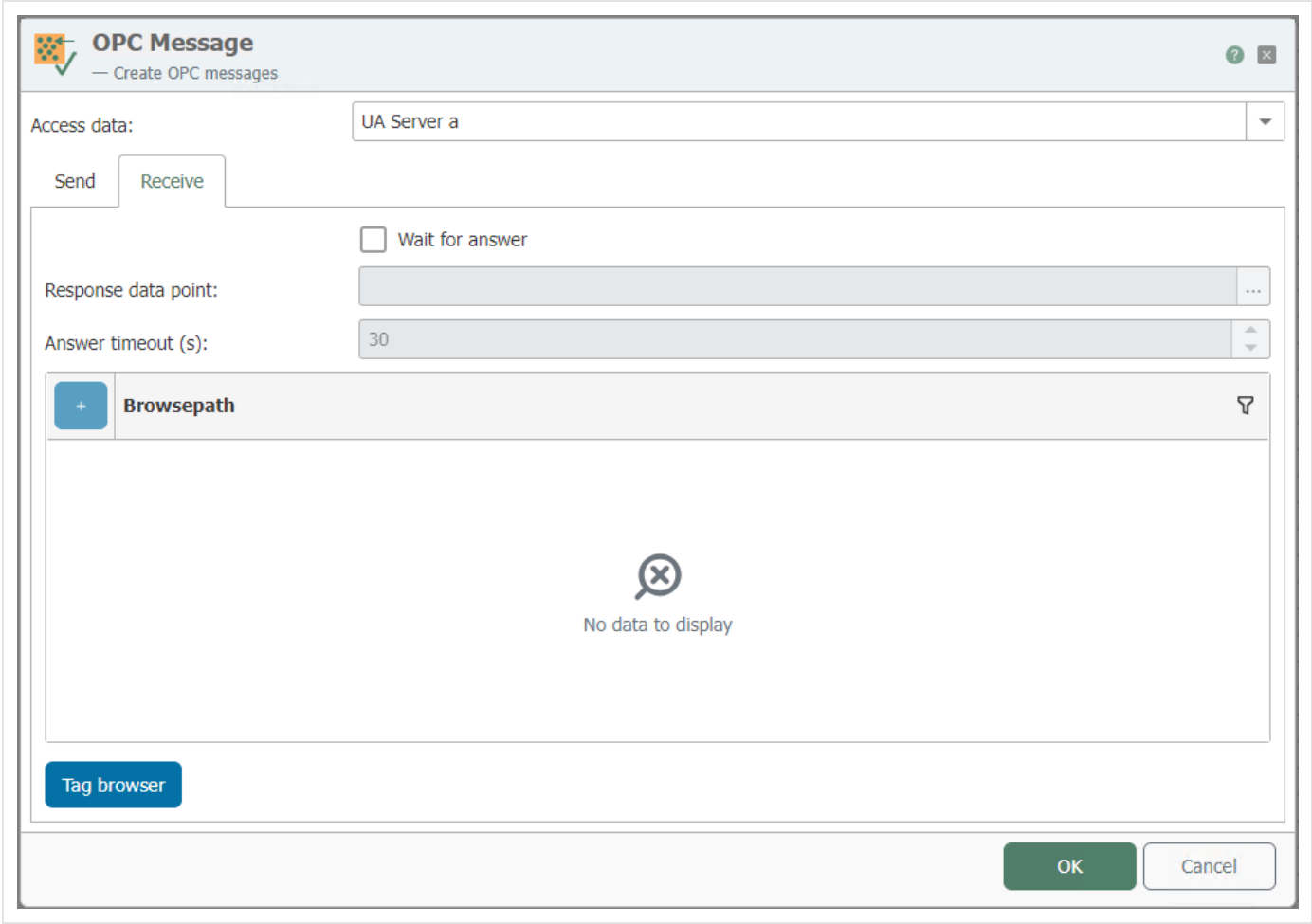
Send



PROPERTY	DESCRIPTION
Access data	The OPC servers are available here as configured instances of the OPC plug-in. After selecting the connection, the OPC router recognizes whether entries are required according to the DA or UA specification. If the OPC connection is to be determined as a template variable at runtime, please refer to the corresponding instructions.
Send data point	Enter the address of the OPC item that is to be incremented after sending. Use the "..." button to use the OPC tag browser to simplify the selection of the data point.
Retries	Number of attempts made to transmit a message if the first attempt fails.
MessageID auto count	Activate this checkbox if the value of the send data point should be increased by 1 for each transfer.
Range	If the option "MessageID auto count" is activated, the range of the MessageID can be defined.

You can compile the data points to be written in the table below.

Receive



PROPERTY	DESCRIPTION
Wait for answer	Activate this checkbox if the transfer should wait for a response.
Response data point	Enter the address of the OPC item that outputs the response counter here. Use the “...” button to use the OPC tag browser to simplify the selection of the data point.
Answer timeout (s)	Number of seconds to wait for a response until a transfer is canceled as faulty.

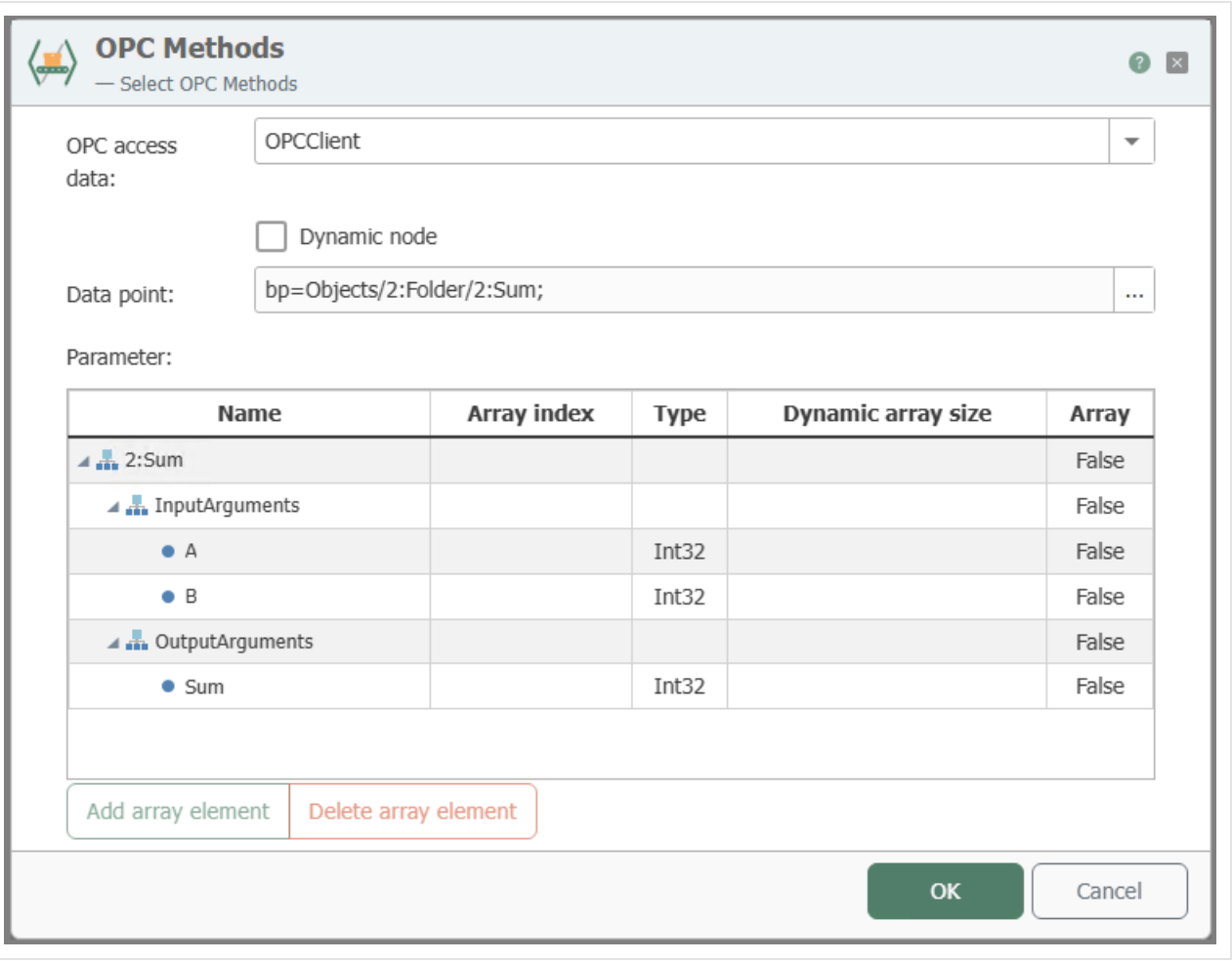
You can compile data points for the response group in the table below.

Using the Tag Browser

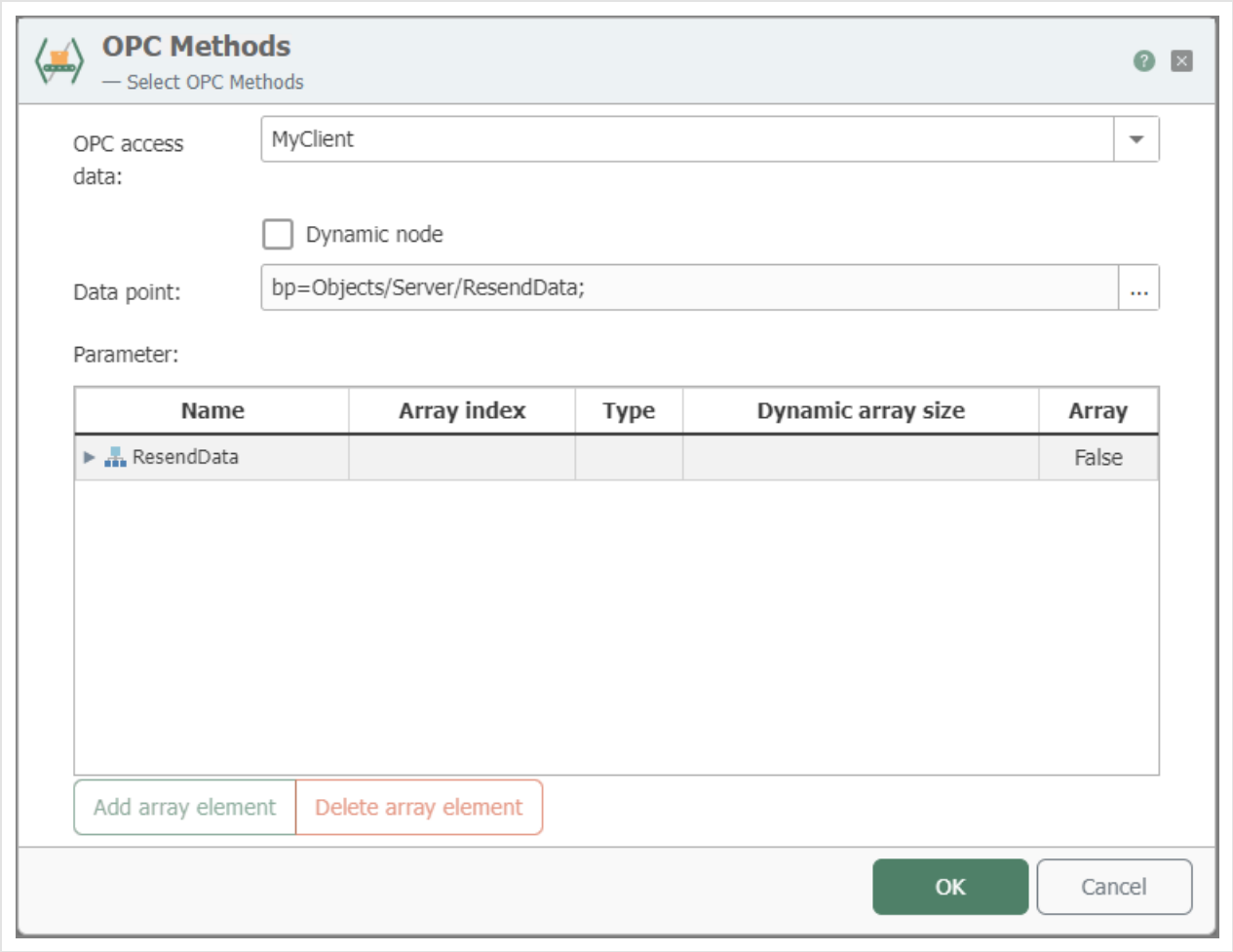
The "Tag browser" button opens another window - the Tag Browser - where the available items are displayed in a graphical structure. You can select multiple OPC items and the selected OPC items will be displayed in the table after they have been applied.

OPC Methods

OPC Methods are provided by some OPC servers. Usually OPC Methods have an input, a output and they are calculating values. The required OPC server has to bis configured as plug-in instance. You can directly configure the transfer object within the connection as described.



OPC Methods Transfer object



Use double click to set up properties:

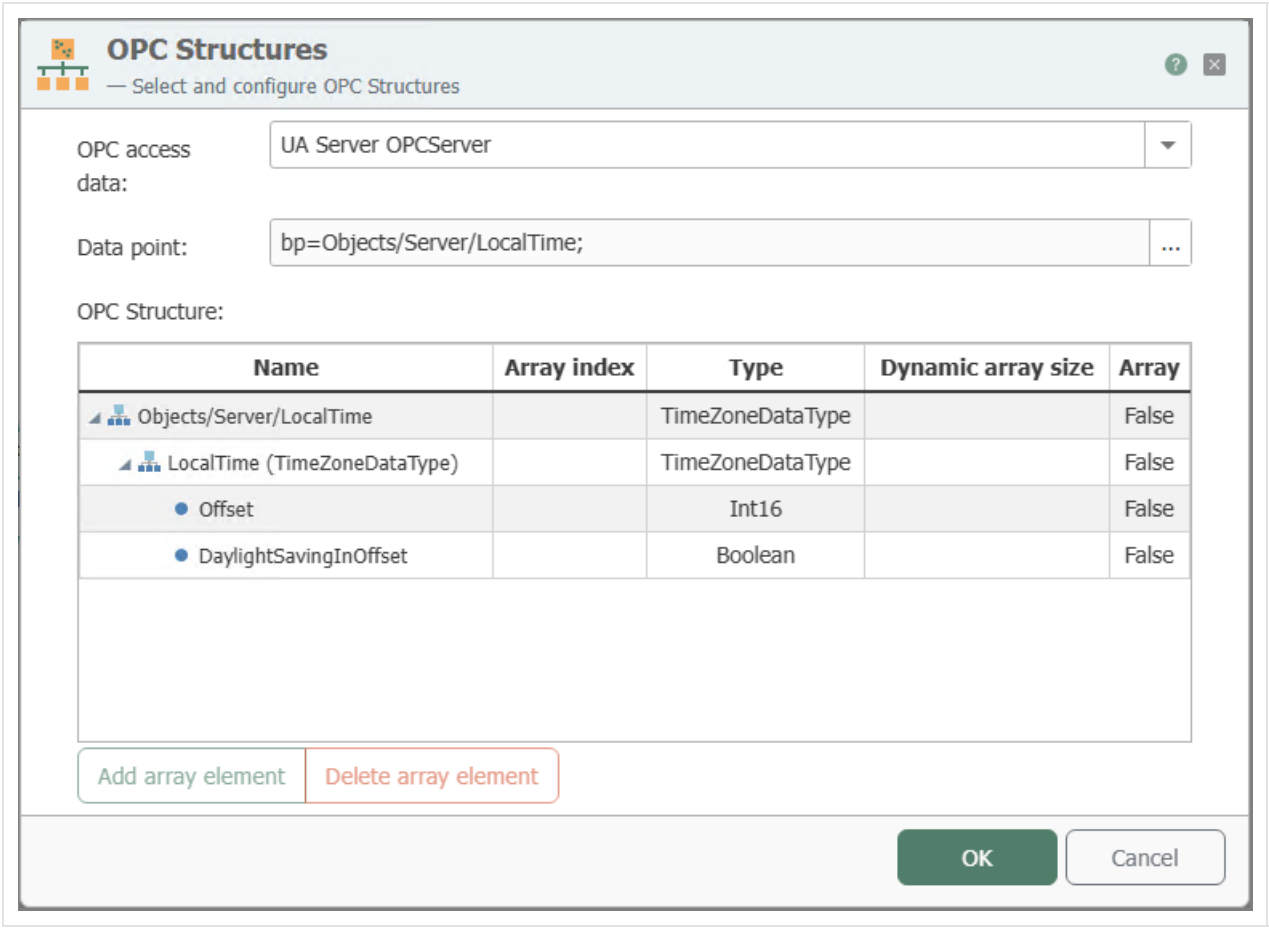
PROPERTY	DESCRIPTION
OPC access data	The OPC servers are available here as configured instances of the OPC plug-in. OPC methods are only supported by OPC UA servers. If the OPC connection is to be determined as a template variable at runtime, please refer to the corresponding instructions.
Dynamic node	Set the field if you want to pass the method dynamically to the transfer object.
Data point	Please enter the address of the method to be used here. Use the "..." button to use the OPC Tag Browser to simplify the selection of the method.
Parameter	Shows you the parameters that the method requires (input) and the parameters that the method outputs (output)

Note: If you check "Dynamic node" you must select a method over the tag browser. This method is a template and must have the same structure and the same data types as the dynamic method.

OPC Structures

OPC Structures are provided by some OPC servers. The required OPC server has to bis configured as plug-in instance. You can directly configure the transfer object within the connection as described.

![OPC structures transfer object](</assets/en (50).png>)



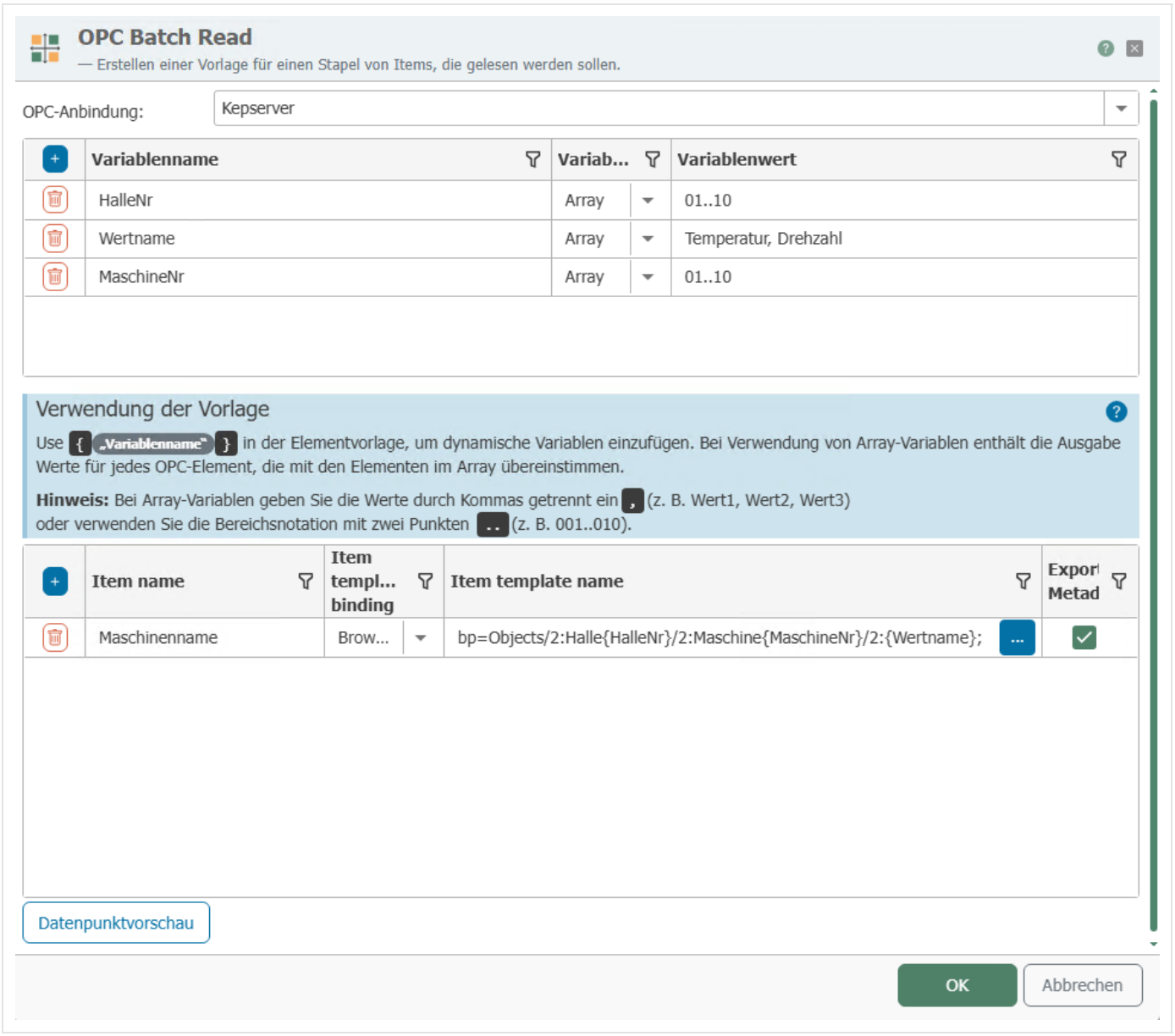
OPC Structures Transfer Object

After double-clicking on the transfer object, set the following properties:

PROPERTY	DESCRIPTION
OPC access data	The OPC servers are available here as configured instances of the OPC plug-in. After selecting the connection, the OPC router recognizes whether entries according to the Classic or UA specification are required. If the OPC connection is to be determined as a template variable at runtime, please refer to the corresponding instructions.
Data point	Enter the required OPC items here. Use the tag browser for convenient selection.
OPC Structure	Preview of the data structure of the selected OPC structure data point.

OPC Batch Read

OPC Batch Read is only available for OPC UA connection.



The OPC Batch Read transfer object makes it easy to read a large number of OPC values that have a standardized structure, without having to define each value individually. The values to be read are specified via variables. The following is a detailed explanation of how to use this transfer object.

Configuration

The following properties can be configured in the OPC Batch Read transfer object:

OPC connection	OPC connection used to access the OPC data point group.

Variables

The following properties can be set for variables:

Type	**Array**: The entered value is passed as an array. **String**: The entered value is adopted as a string.
Value	If the **String** type is selected, the value is passed as entered. If the **Array** type is selected, the value is passed as an array. The value of an **array** variable can be a list of values, such as `temperature, speed`, or a range of numbers. A range of numbers is defined by writing two dots between two numbers, e.g. `1..100` or `001..010`. Number ranges can also be part of a list. E.g. `1..10, 15..20` . Words can also be used together with number ranges in a list.

When specifying a range of numbers, the smaller number must always be on the left-hand side.

Templates

Templates provide an easy way to configure the reading of many values.

The button "..." can be used to open the tag browser to select a path. This function is used to insert a path into the template. This means that the path in which the variables are to be inserted does not have to be typed in manually.

At runtime and in the data point preview, the variables in the template are replaced with the values of the variables. Paths are created with all possible combinations of the values from the variables.

The following properties can be set for templates:

Item name	The name under which all values are output at runtime.
Item-Templatebinding	Here you can select whether the items are to be read by browse path or node ID.
Item-Templatename	The path in which the items to be read can be configured using variables.
Export Metadata	If this option is active, metadata is also output in addition to the value. E.g. the quality ID.

Example

A company has two halls, in each of which three machines are operated. The values “speed” and “temperature” are to be read from each of these machines.

The path for reading each of these values would be structured according to the following pattern:

As a browse path: `bp=Objects/2:Hall{hall number}/2:Machine{machine number}/2:{value name};`

With node ID: `ns=2;s=Hall{hall number}.Machine{machine number}.{value name}`

The path is static except for the three variable values. Variables can be created for these variable values.

For the example, this could look like this:

Variablenname	Variab...	Variablenwert
HalleNr	Array	01..02
Wertname	Array	Temperatur, Drehzahl
MaschineNr	Array	01..03

The tag browser can be used to select a path that matches the pattern for which variables are to be provided.

For the example, this path would be suitable:

Item name	Item templ... binding	Item template name	Expor Metad
Maschinenname	Brow...	bp=Objects/2:Halle01/2:Maschine01/2:Drehzahl;	✓

So that not every hall and every machine has to be added individually, the variables just created can be used here.

The finished path with variables could then look like this:

Item name	Item templ... binding	Item template name	Expor Metad
Maschinenname	Brow...	bp=Objects/2:Halle{HalleNr}/2:Maschine{MaschineNr}/2:{Wertname};	✓

By using variables, it was possible to configure the reading of 12 data points with one template without having to configure each individual path.

It is possible to map many more data points using a template. The example was deliberately kept small for illustrative purposes.

The “Data point preview” button can be used to display a preview of the values that would be read with the current configuration.

Preview - 12 total items

Item Name	Item Template Name	Item Quality	Item Value
Maschinenname	bp=Objects/2:Halle01/2:Maschine01/2:Temperatur;	good	73
Maschinenname	bp=Objects/2:Halle01/2:Maschine01/2:Drehzahl;	good	300
Maschinenname	bp=Objects/2:Halle01/2:Maschine02/2:Temperatur;	good	70
Maschinenname	bp=Objects/2:Halle01/2:Maschine02/2:Drehzahl;	good	300
Maschinenname	bp=Objects/2:Halle01/2:Maschine03/2:Temperatur;	good	108
Maschinenname	bp=Objects/2:Halle01/2:Maschine03/2:Drehzahl;	good	300
Maschinenname	bp=Objects/2:Halle02/2:Maschine01/2:Temperatur;	good	100
Maschinenname	bp=Objects/2:Halle02/2:Maschine01/2:Drehzahl;	good	300
Maschinenname	bp=Objects/2:Halle02/2:Maschine02/2:Temperatur;	good	89
Maschinenname	bp=Objects/2:Halle02/2:Maschine02/2:Drehzahl;	good	300

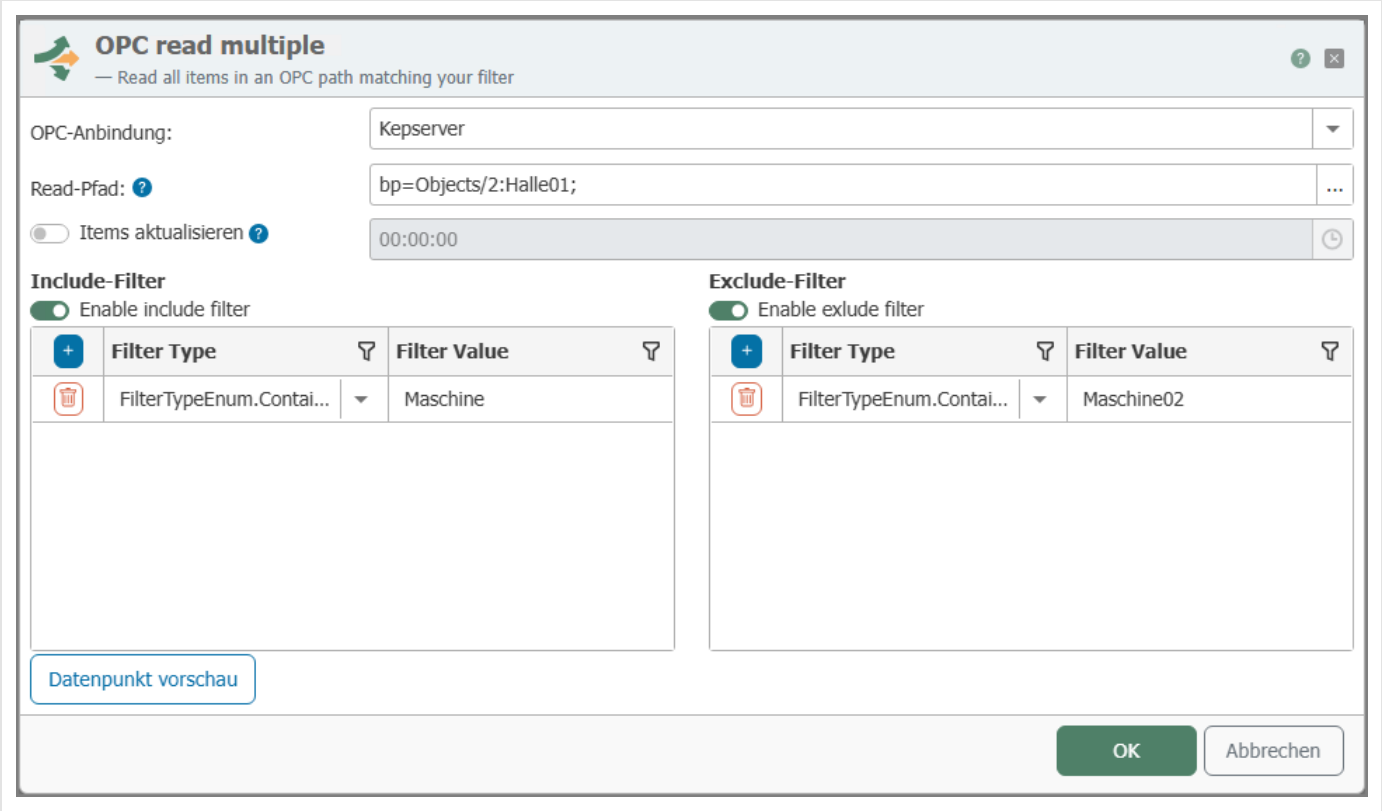
1 2

Schließen

OPC Read Multiple

OPC Read Multiple is only available for OPC UA connection.

The OPC Read Multiple reads data points recursively. Which data points should or should not be read can be configured using filters.



The following properties can be configured:

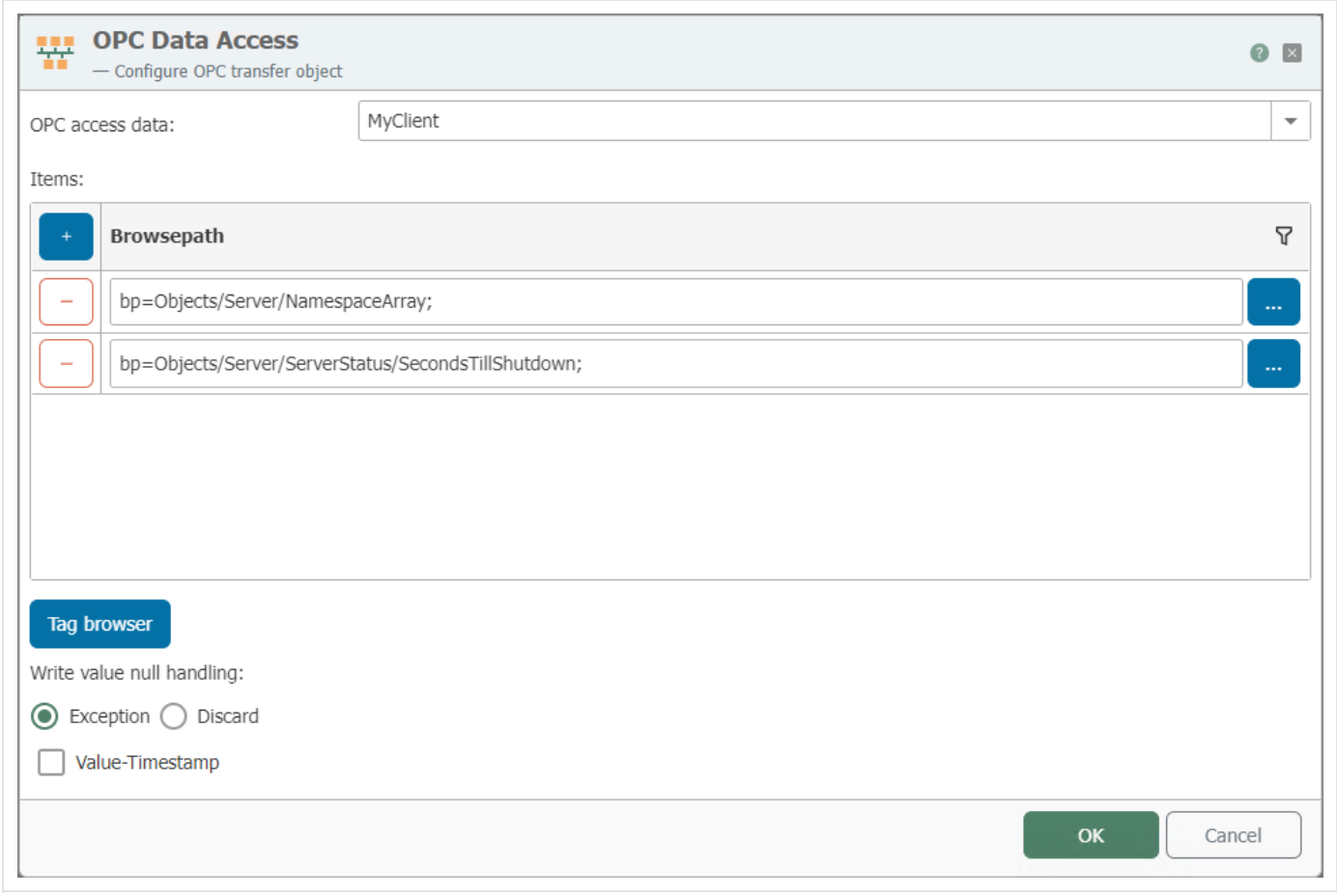
OPC connection	OPC connection used to access the OPC item group.
OPC address	OPC data point group whose values are to be read. Use the "..." button to open the tag browser (see below) to select an item interactively. For OPC UA connections, the node ID is managed in the background.
Refresh items	If active, the data points are re-browsed at this interval.

You can find out how the include and exclude filters work here:

opc-data-point-filters.md

OPC Data Access

OPC servers and thus controllers are available as source and target. The required OPC server must be configured as a plug-in instance. Create an OPC transfer object within the connection as described.



Double-click on transfer object to assign the properties:

PROPERTY	DESCRIPTION
OPC access data	The OPC servers are available here as configured instances of the OPC plug-in. After selecting the connection, the OPC router recognizes whether entries according to the Classic or UA specification are required. If the OPC connection is to be determined as a template variable at runtime, please refer to the corresponding instructions. **Note:** If you use FAS-inMOVE and configure manually at the same time, do not select any OPC connections with the prefix [FiMx] for manual projects: These belong to FAS-inMOVE and can be deleted there. Instead, create a different OPC connection in the plug-in configuration.
Items	Enter OPC items needed. Use the Tag Browser to browse items. **Note:** If Manual type definition is activated in the OPC Classic Server connection, use the drop-down menu to set the expected data type that is to be communicated to the OPC server.
Node ID	Enter Node ID of UA items needed. Use the Tag Browser to browse items.

PROPERTY	DESCRIPTION
Write value null handling	These settings help ensure data integrity by specifying how the system should respond to null values during write operations. • **Exception**: If this option is selected, the system will throw an error when attempting to write a null value. This means the write operation will be halted, and an error message will be generated. • **Discard**: If this option is selected, the system will simply ignore the null value and proceed with the write operation without including the null value.
Value-Timestamp	Indicates the time when the data point was transferred.

If you have checked “Transfer OPC null values” in the OPC UA Plug-in, the corresponding values of the data points will always be transferred if they are null. If “Transfer OPC null values” is not checked, the values are only transferred if the “Quality” of the data point is “Good” and it has a value.

If you have set Browsepath in the OPC UA plug-in, the data points will be displayed as shown in the screenshot above. If you have set Node-ID instead, the name of the data point and the Node-ID will be displayed in the transfer object. You can find more details here.

Within the item’s list, select several items by holding mouse button down (“Ctrl + a” will select all). Selected items can be copied to clipboard and inserted (“Ctrl + c”, “Ctrl + v”).

Note: OPC Classic: You may either enter tags/aliases from OPC server or address data blocks in PLC directly. Just keep in mind to specify the correct Access Path in your Plug-in Configuration.

Using the Tag Browser

The "Tag browser" button opens another window - the Tag Browser - where the available items are displayed in a graphical structure. You can select multiple OPC items and the selected OPC items will be displayed in the table after they have been applied.

OPC HA Read

The OPC HA Read TO can be used to query historical OPC data from servers that support it. The data can be read unprocessed or processed.

General settings

PROPERTY	DESCRIPTION
Access Data	OPC connection used to read the OPC HA data.
Read mode	Either Raw/Modified or Processed
Use start time	Defines the start of the period from which data is read. The time refers to the server's timestamp. An explanation of the syntax can be found here: Time offset input options
Use end time	Sets the end of the period from which data is read. The time refers to the server's timestamp. An explanation of the syntax can be found here: Time offset input options
Include OPC Router timestamp	If this option is enabled, the timestamp of the system on which the OPC Router is running is also transferred when a value is read.
Include server timestamp	If this option is enabled, the timestamp of the system from which the values are read is also transferred when a value is read.
Treat Uncertain Quality as Bad	If this option is enabled, a value with the quality Uncertain is treated as a value with the quality Bad.

If the end time is before the start time, the data is sorted from old to new. If the start time is before the end time, the data is sorted from new to old.

If the start or end time is not set, the earliest possible time is used for the unconfigured value (DateTime.MinValue).

Read mode-specific settings

Raw / Modified

PROPERTY	DESCRIPTION
Read modified data	Specifies whether modified data should be read. **Info**: Modified data is data that has been changed after it was initially written to the server database.
Read maximum N nodes	Specifies the maximum number of nodes to be read.

Processed

PROPERTY	DESCRIPTION
Transfer interval	Specifies the size of the interval to which an aggregation is applied.

PROPERTY	DESCRIPTION
Aggregation	This setting specifies which aggregation is to be used for the values.
Use Server Capabilities Defaults	If this option is enabled, the server's default values are used. The settings for “Use Sloped Extrapolation,” “Accepted Bad-Quality Data,” and “Required Good-Quality Data” are then ignored.
Use Sloped Extrapolation	Specifies whether sloped extrapolation should be used.
Accepted Bad-Quality Data	Specifies the percentage of data that may have bad quality.
Required Good-Quality Data	Specifies the percentage of data that must have good quality.

Time offset input options

General

- Relative values are specified with “+” or “-” and the unit without spaces, e.g. `-1month`
- Absolute values are specified with brackets after the unit, e.g. `year(2020)`
- Only whole numbers are allowed. The time 1.5 hours ago would have to be specified as `-90min` or `-1h -30min`.
- If the year, month, or day are not specified in a relative time specification, the system time is used.
- If the hour, minute, or seconds are not specified, the value `0` is used for them.
- Relative and absolute time specifications can be combined.

Absolute time specifications

PARAMETER	EXPLANATION	VALID VALUES
second(<second>)	Specifies one second.	0-60 (integer)
minute(<minute>)	Specifies a minute.	0-60 (integer)
hour(<hour>)	Specifies an hour.	0-24 (integer)
day<Day>	Specifies a day.	0-12 (integer)
year(<year>)	Specifies a year.	integer

Relative time specifications

PARAMETER	EXPLANATION	VALID VALUES
+/-<second>second	Depending on the sign, the specified number of seconds is added to or subtracted from the current time.	Integer
+/-<minutes>minute	Depending on the sign, the specified number of minutes is added to or subtracted from the current time.	Integer
+/-<hours>hour	Depending on the sign, the specified number of hours is added to or subtracted from the current time.	Integer

PARAMETER	EXPLANATION	VALID VALUES
+/-<days>day	Depending on the sign, the specified number of days is added to or subtracted from the current time.	Integer
+/-<Monate>month	Depending on the sign, the specified number of months is added to or subtracted from the current time.	Integer
+/-<years>year	Depending on the sign, the specified number of years is added to or subtracted from the current time.	

Examples

TARGET	INPUT
Current day 6 a.m.	hour(6)
One month ago	-1month
First day of the month	day(1)
First day of the previous month	-1month day(1)
One year ago	-1year
First day of the year	month(1) day(1)
First day of the previous year	-1year month(1) day(1)

It is not possible to specify the 10th minute of the current hour, because `minute(10)` returns the time `00:10:00`. \ This behavior makes it easier to reset the time to zero. For example, on the first day of the month, the time does not have to be manually set to `0` with all parameters.

Trigger

Overview over trigger:

Threshold

Bit

Data Change Missing

Data Change

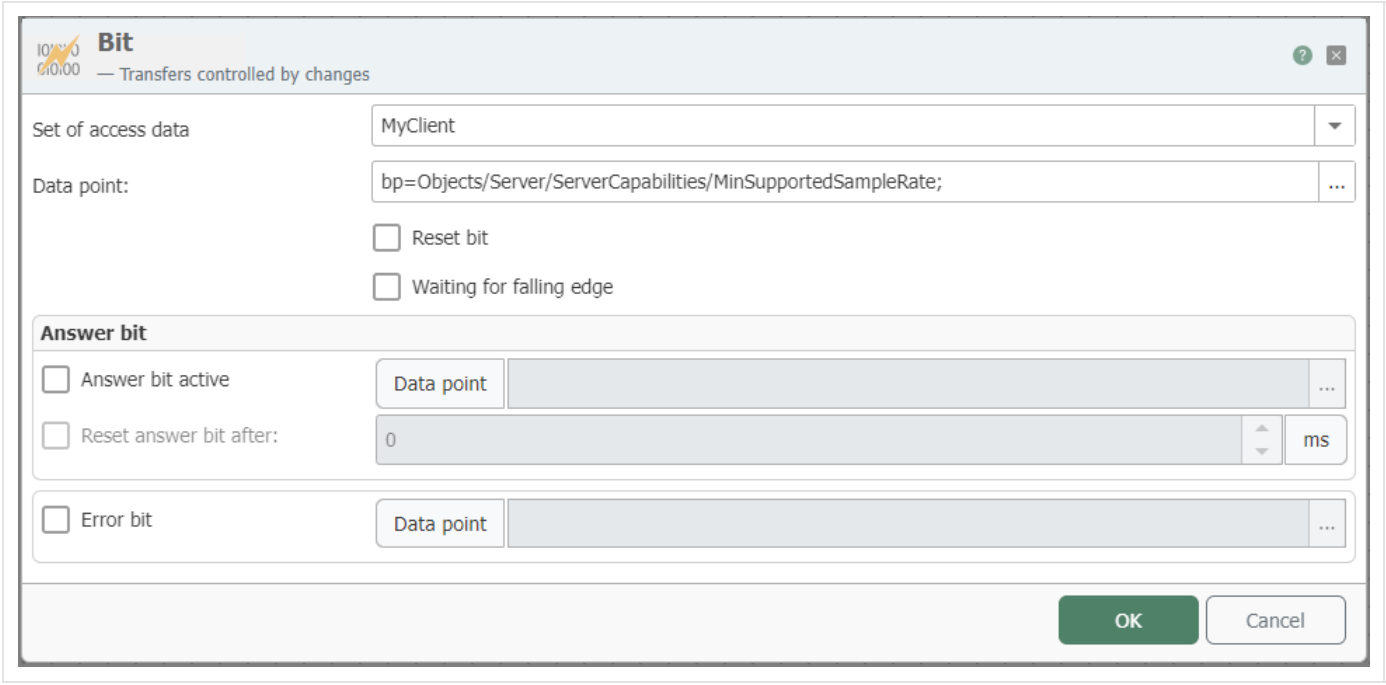
Multi Data Change

Message

Alarm/Event trigger

Bit

The bit trigger triggers the data transfer when the assigned logical data point takes a value 1. If the assigned data point is not a logical data point, the trigger triggers when the data point takes a different value as NULL. You can also send a reply and a fault message to two additional data points if the transfer is incorrect.



The individual properties are:

PROPERTY	DESCRIPTION
OPC access data	OPC connection via which the relevant data point is reached.
Data point	OPC data point that triggers the transfer as soon as it assumes a value other than 0. Use the “...” button to open the tag browser (see below) to interactively select an item. With OPC UA connections, the node ID is managed in the background.
Reset bit	Sets the data point to “0” after the transfer. Should normally be deactivated because a bit should only be written from one side. Instead, use the response bit to inform the controller that the transfer has been completed.
Waiting for falling edge	If you don't want the bit trigger to fire continuously when the bit is active, select this check box. This causes the bit trigger to fire once when the bit is active and then wait for a "falling edge" before firing again.
Answer bit active	Sends a bit to another data point in response that the transfer has been completed.
Data point	OPC data point to which the response is sent. It must be a logical (bit) data point that has the status “0” or “1”. Use the “...” button to open the tag browser (see below) to interactively select an item.
Reset answer bit after	The response bit is reset after the set period of time.
Error bit	Sends a bit to another data point if the data transfer was faulty.

PROPERTY	DESCRIPTION
Data point	OPC data point to which the response is sent. It must be a logical (bit) data point that has the status “0” or “1”. Use the “...” button to open the tag browser (see below) to interactively select an item.

Bit and message triggers for OPC telegrams

Bit and message triggers are used for OPC telegrams if you want to read several related items. The OPC router waits for another read cycle before the items are read out in the OPC transfer object to ensure that the PLC could write all values. On the SPS side the trigger should be written to the end.

Bit and message triggers are used for OPC telegrams if you want to read out several related items. The OPC router waits a further read cycle before reading the items in the OPC transfer object to ensure that the PLC was able to write all values. On the PLC side, the trigger should still be written at the end.

Bit trigger always with response bit

Always use the bit trigger with response bit. If the data point in the PLC changes twice within a read cycle, the OPC router does not notice this. The PLC must react accordingly to the missing response of the router. Standard praxis: A bit may only written from one side, the PLC resets the trigger data point after receiving the response. The OPC router resets the response data point after a set time period.

Always use the bit trigger with response bit. If the data point in the PLC changes twice within a read cycle, the OPC router will not notice this. The PLC must react accordingly to the lack of response from the router.

Standard practice: A bit may only be written from one side, i.e. the PLC resets the trigger data point after receiving the response. The OPC router resets the response data point itself after a period of time to be set.

Timeout Handling

If the response counter has not been written, this must be intercepted by the PLC. One option here would be to generate a timeout and set the send counter again after a certain interval. If the response counter has not been written again after several intervals, the error must be escalated appropriately.

Data Change Missing

The data change missing trigger initiates the data transfer if the value of a data point does not change.

The individual properties are:

PROPERTY	DESCRIPTION
Connection	Connection via which the relevant data point is reached.
Trigger item	Trigger data point whose missing change triggers the transfer. Use the "..." button to open the tag browser (see below) to interactively select an item. With OPC UA connections, the node ID is managed in the background.
Check period	After the test period has elapsed without a value change, the trigger is activated.
Trigger mode	Here you can define whether to trigger once or regularly if there is no value change. (See below)
Output metadata	When activated, you can output the metadata of the last known value. This includes the value of the data point, the quality ID, the quality string, a timestamp and the path and name of the data point.

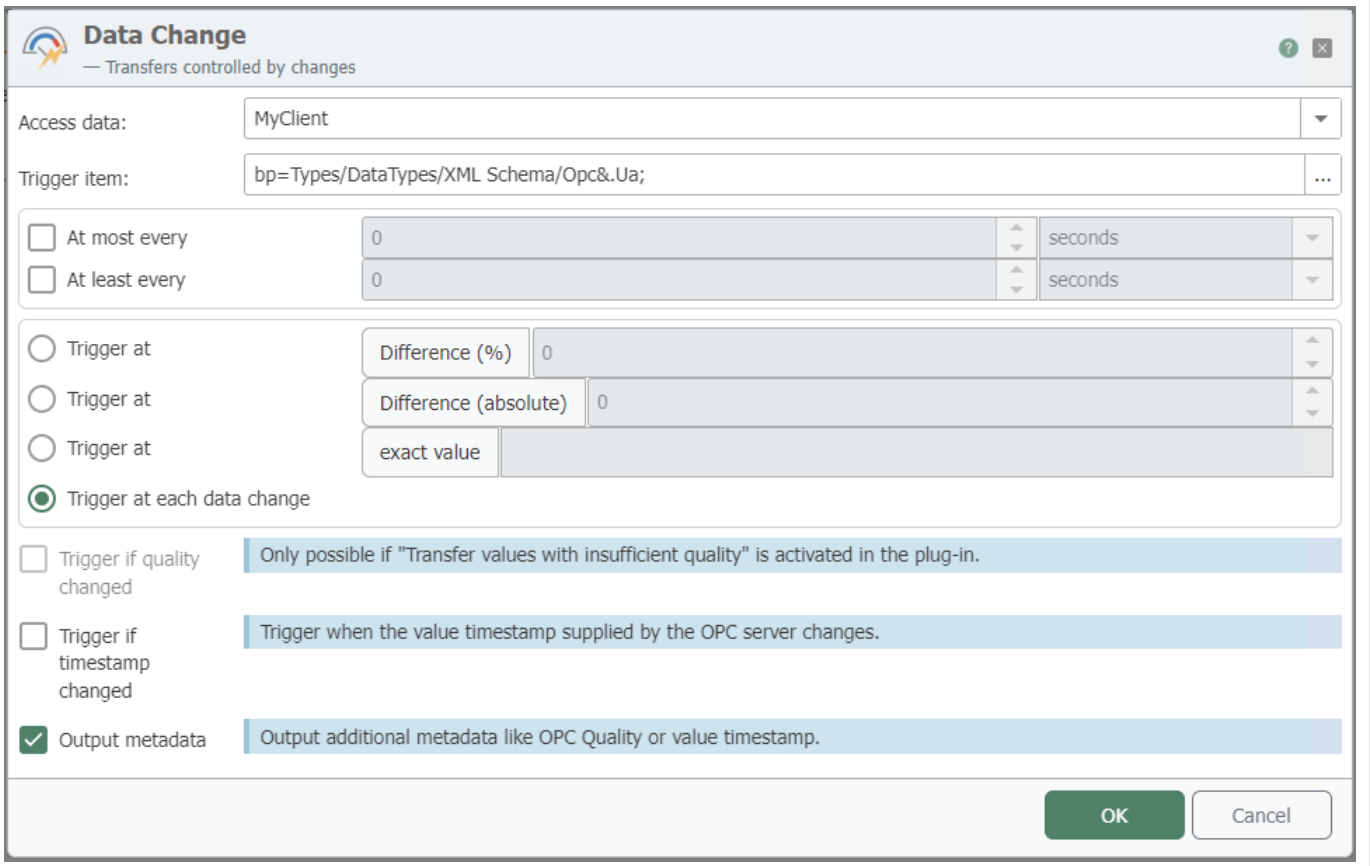
Trigger mode Additional information

PROPERTY	DESCRIPTION
Trigger regularly during a failure	As soon as the set test time has elapsed and no value change has taken place, the trigger is activated. After triggering, the set test time starts to run again and the trigger is triggered again once the test time has elapsed.
Trigger only once during an outage	As soon as the set time for the test period has elapsed and no value change has taken place, the trigger is activated. Only when a value change has taken place again afterwards is it checked again whether no value change has taken place.

Data Change

Note: The Datachange trigger cannot reliably detect every data change. In critical connections for data transfer, please use the message or bit trigger together with a handshake.

The data change trigger initiates the data transfer when the value of a data point changes. You can also define a relative or absolute tolerance as well as a minimum and maximum time interval - forced transfer even if there is no change, or limiting transfers if there are many data changes.



The individual properties are:

PROPERTY	DESCRIPTION
Access data	Select a set of OPC access data. Dependent on plug-in configuration, OPC Router will recognize DA/UA connections.
Trigger item	Trigger data point whose change triggers the transfer. Use the "..." button to open the tag browser (see below) to interactively select an item. With OPC UA connections, the node ID is managed in the background.
At most every	If the value of the data point changes frequently, the transfer number can be limited here via an interval. The entry therefore means: "It is transferred at most every second, for example, even if the value changes more frequently". Specification always as a whole number, no decimal places. Milliseconds, seconds, minutes or hours should therefore be selected as units.
At least every	Time interval in which data is transferred in any case, even if the value of the data point has not changed. Always specified as an integer, no decimal places. The units to be selected are therefore milliseconds, seconds, minutes or hours.
Trigger at ... Difference (%)	The transfer is only triggered if the value of the data point changes by this entry (relative).

PROPERTY	DESCRIPTION
Trigger at ... Difference (absolute)	The transfer is only triggered if the value of the data point changes by this entry (absolute).
Trigger at ... exact value	The transfer is only triggered if the value of the data point changes to this value. If the same value occurs several times, nothing is triggered.
Trigger at each data change	Default. Overrides tolerance settings.
Trigger if quality changed	Activate the checkbox if you want to trigger at quality change.
Trigger if timestamp changed	Activate the checkbox if you want to trigger when the timestamp is changed.
Output metadata	Activate the checkbox if you want the metadata to be output.

Note: Is “Transfer OPC null values” checked than the output of the null value with the data change trigger is “N/A”. Is the box not checked the connection will not been executed when the values are null.

Tip: To set conditions for absolute and relative difference, configure two data range triggers: one for relative and one for absolute difference. The link can be created using AND or OR.

Data length and limit value triggers for analog values

You can set the data length and limit value triggers if you want to transmit a single analog value. Trigger item and transfer item are identical. The transfer starts immediately after the trigger condition has occurred.

Enter a search string (OPC DA only) to hide the tree view and show all items matching the search term. Contains the OPC transfer object other items to be transferred, values can be transferred which do not belong together.

Read Quality Values

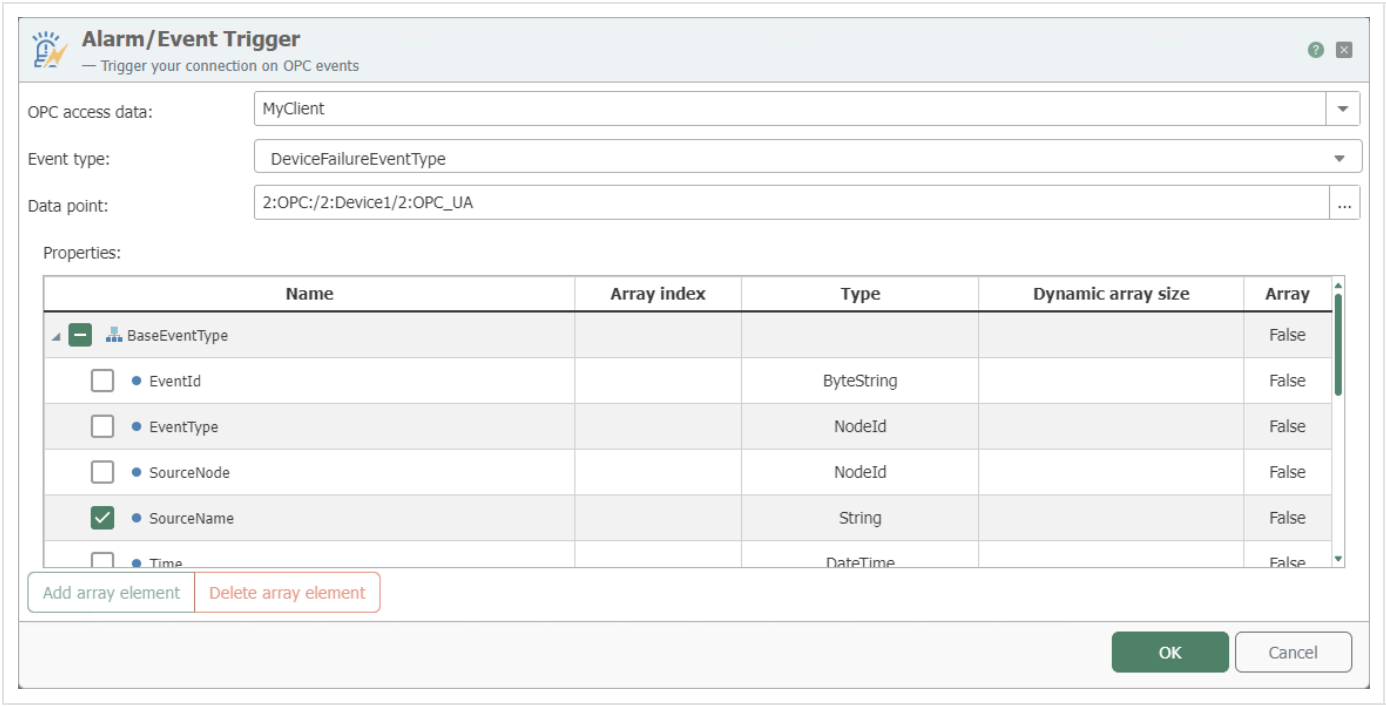
With properties Quality string and Quality ID, the OPC quality values are available and can e. g. be transferred and stored together with the value.

STRING	ID/INTEGER
Timeout	-3
NotInitialized	-2
NotSpecified	-1
bad	0
badConfigurationError	4

STRING	ID/INTEGER
badNotConnected	8
badDeviceFailure	12
badSensorFailure	16
badLastKnownValue	20
badCommFailure	24
badOutOfService	28
badWaitingForInitialData	32
uncertain	64
uncertainLastUsableValue	68
uncertainSensorNotAccurate	80
uncertainEUExceeded	84
uncertainSubNormal	88
good	192
goodLocalOverride	216

Alarm/Event trigger

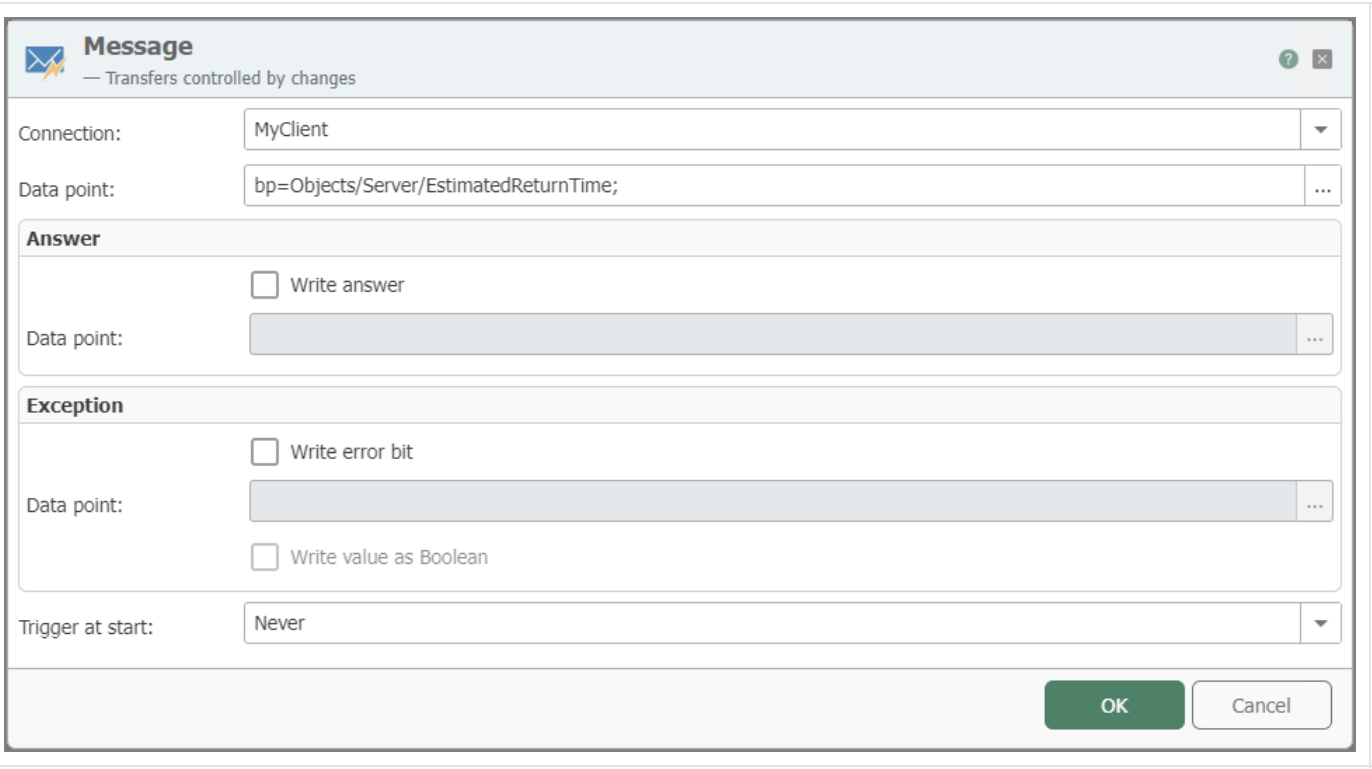
Use the Alarm/Event trigger to execute connections on OPC events.



PROPERTY	DESCRIPTION
OPC access data	Select the OPC UA server that provides the events.
Event type	Select the type of the OPC event.
Data point	Select a datapoint for your OPC event.
Properties	Select the data you want to output if an event executed.

Message

The message trigger initiates the data transfer if the assigned numerical data point assumes a value that is 1 higher. You can also send a response and, if the transfer is incorrect, an error message to two other data points.



The individual properties are:

PROPERTY	DESCRIPTION
Connection	Select a set of OPC access data. Dependent on plug-in configuration, OPC Router will recognize DA/UA connections.
Data point	Item whose increase will execute the transfer. Browse items with tag browser (“...” button). OPC UA node IDs are registered in background.
Write answer	Will transfer OPC item’s value (or set a bit) to another item as confirmation that the transfer executes.
Data point	OPC item the answer is send to. It must be a logical (bit) data point that has state “0” or “1”. Use the button “...” to open the tag browser (see below) for the interactive selection of an item.
Write error bit	Activate to set a bit with transfer errors.
Data point	OPC item (bit) that is set with transfer errors. It must be a logical (bit) data point that has state “0” or “1”. Use the button “...” to open the tag browser (see below) for the interactive selection of an item.
Write value as Boolean	Sends the value as a Boolean to the datapoint.

PROPERTY	DESCRIPTION
Trigger at start	Set when the message trigger should fire at startup: • Always • Never • If message ID and answer ID are not equal

Bit and message triggers for OPC telegrams

Bit and message triggers are used for OPC telegrams if you want to read several related items. The OPC Router waits for another read cycle before the items are read out in the OPC transfer object to ensure that the PLC could write all values. On the SPS side the trigger should be written to the end.

Message trigger values

Record the value of the message trigger. This allows you to track whether records have not been transferred or duplicated. On the PLC side, the data point for the message trigger for new data is to be increased by the value 1.

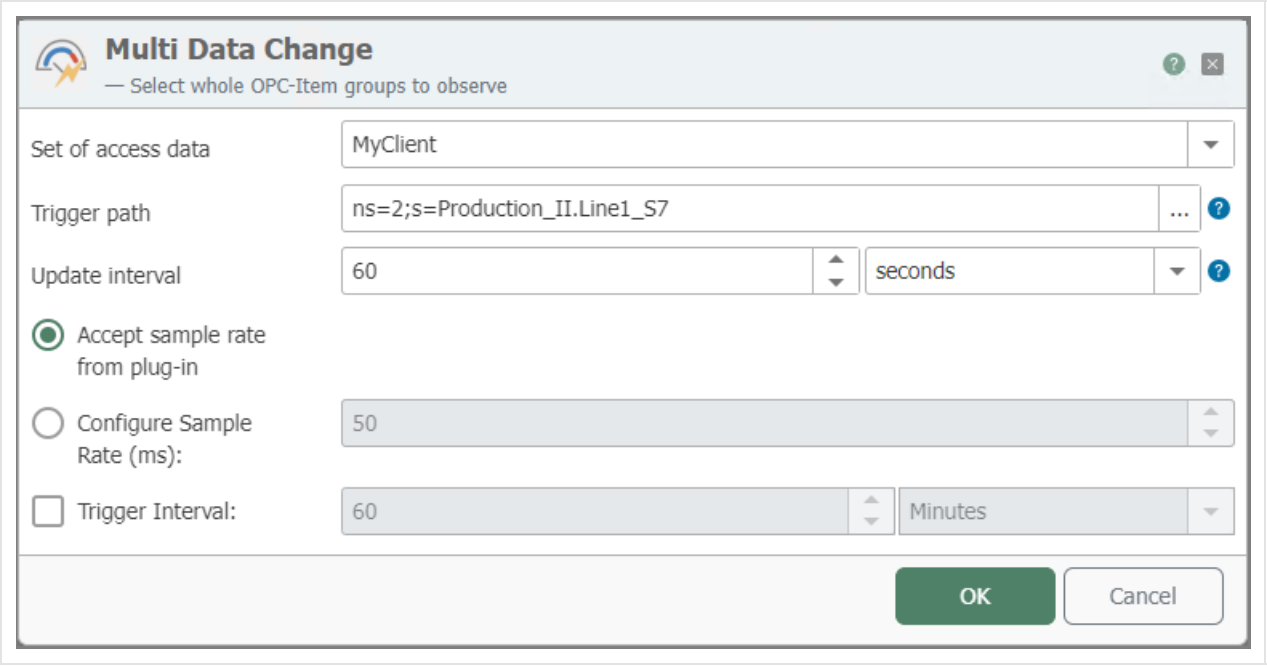
Timeout Handling

If the response counter has not been written, this must be intercepted by the PLC. One possibility here would be to create a timeout and to set the send counter again after a certain interval. If the response counter has not been written again after several intervals, the error must be escalated appropriately.

Multi Data Change

The OPC Multi Data Change Trigger is only available for OPC UA connection.

The multi-datachange trigger initiates the data transfer when a value changes within an OPC data point group.



The individual properties are:

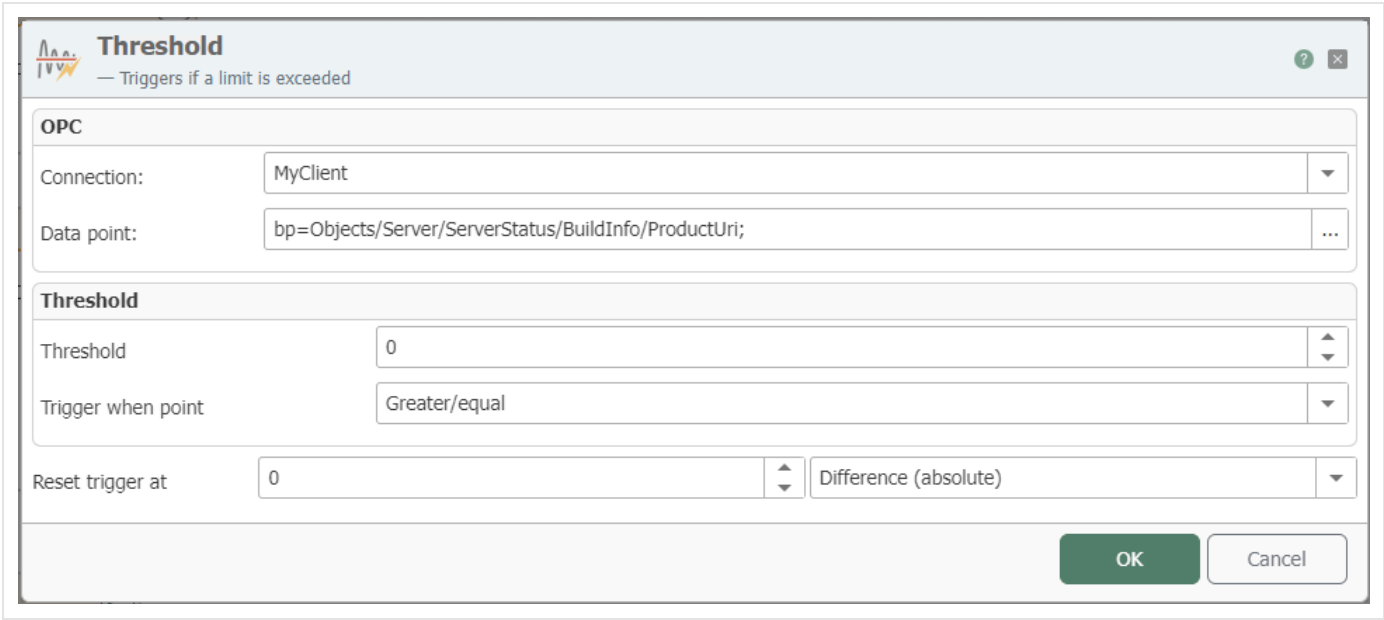
PROPERTY	DESCRIPTION
Set of access data	Select a set of OPC access data. Dependent on plug-in configuration, OPC Router will recognize DA/UA connections.
Trigger path	OPC data point group whose change triggers the transfer. Use the "..." button to open the tag browser (see below) to interactively select an item. For OPC UA connections, the node ID is managed in the background.
Update interval	In this interval, the monitored group is checked for new data points. If new data points have been added or removed, the group is updated at runtime.
Accept sample rate from plug-in	Uses the sample rate configured in the plug-in settings of the OPC connection.
Configure Sample Rate (ms)	You can use this option to overwrite the sample rate selected in the plug-in.
Trigger interval	All data points are transferred at the specified interval, plus the data points that have changed.

Data change and Threshold trigger for analog values

You use data range and Threshold triggers if you want to transfer a single analog value. The trigger item and transfer item are identical. The transfer starts immediately after the trigger condition occurs. If the OPC transfer object contains other items to be transferred, it is therefore possible that values are transferred that do not belong together in terms of content.

Threshold

Threshold trigger executes a data transfer with off-limit condition (of OPC value)



The properties are:

PROPERTY	DESCRIPTION
Connection	Select a set of OPC access data. Dependent on plug-in configuration, OPC Router will recognize DA/UA connections.
Data point	Item whose off-limit condition will execute the transfer. Browse items with tag browser ("..." button). OPC UA node IDs are registered in background.
Threshold	Absolute value.
Trigger when point	Set if threshold has to be under- or overrun.

Hysteresis:

Once the trigger has been activated, it may be desirable, depending on the application context, that the trigger is only activated again when the value of the OPC item has moved away from the limit value again. In this case, set the necessary difference to the limit value here at which the trigger is "armed" again. All specifications refer to the limit value.

Example: The limit value is 20 °C, triggering occurs when it is exceeded. Now the temperature fluctuates around the 20 °C mark, which would trigger the trigger several times. However, only the first exceedance is relevant for the recording. The hysteresis value is therefore set to "Reset trigger from 18 degrees absolute value". The trigger is activated once when the 20 degree mark is exceeded. It will only be triggered again if the temperature has fallen below the 18 degree mark once and then rises above 20 °C again.

Hysteresis options:

RESET TRIGGER AT DIFFERENCE (%)	THE TRIGGER ONLY TRIGGERS AGAIN WHEN THE VALUE DIFFERENCE HAS EXCEEDED THE SPECIFIED PERCENTAGE VALUE.
Reset trigger at difference (absolute)	The trigger only triggers again when the value difference exceeds the specified absolute difference.

RESET TRIGGER AT DIFFERENCE (%)	THE TRIGGER ONLY TRIGGERS AGAIN WHEN THE VALUE DIFFERENCE HAS EXCEEDED THE SPECIFIED PERCENTAGE VALUE.
Reset trigger at absolute value	The trigger only triggers again when the value difference corresponds to the specified value.

Tip: If the connection is to be permanently triggered in the off-limit state, set the hysteresis absolute value to the maximum value the item can take. This way it will trigger in the OPC Router cycle time interval as long as it is in the off-limit condition. Combine with a data change trigger linked with AND to trigger only data changes above the limit. In the example above, the temperature will not exceed 100 degrees, then with threshold = 20 and trigger reset from 100 absolute value, the trigger would always trigger when the temperature is between 20 and 100 degrees, in the frequency of the router's cycle time (settings). You could limit this with an AND data change trigger to only record data changes above the threshold.

Data change and Threshold trigger for analog values

You use data range and Threshold triggers if you want to transfer a single analog value. The trigger item and transfer item are identical. The transfer starts immediately after the trigger condition occurs. If the OPC transfer object contains other items to be transferred, it is therefore possible that values are transferred that do not belong together in terms of content.

import Tabs from '@theme/Tabs'; import TabItem from '@theme/TabItem';

OPC Data Point Filters

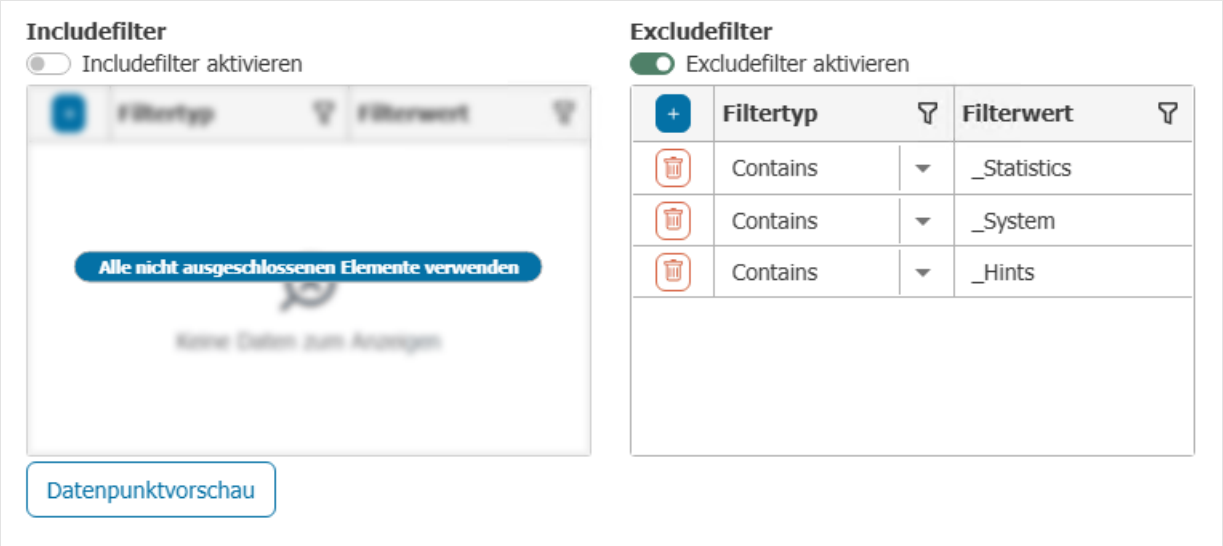
In the multi-data change trigger and the OPC read multiple transfer object, it is possible to create filters for data points to control exactly which data points should be read.

Filter Types

FILTER TYPE	EXPLANATION
Contains	This filter applies if the path or name of the data point contains this value.
Equals	The filter applies if the path or name is exactly the same as the filter.
Regex	The filter applies if the regex expression is found in the path or name.

Configuration options

Recommended if you want to read values from all data points with only a few exceptions, e.g. subdirectories. If only the exclude filter is active, all data points that are not excluded by the exclude filter are read.



Default settings of the filter

This filter is used to read all values from all data points whose path or name does not contain “_Statistics”, “_System” or “_Hints”. **Recommended if** you want to selectively read values from certain data points or subdirectories from a set of data points. If only the include filter is active, all data points to which at least one of the include filters applies will be read.

With this filter, only values of data points would be read whose path or name contains the word “temperature”. **Recommended if** when you want to selectively read values from certain data points or subdirectories from a set of data points, while also excluding some data points or subdirectories. If both filters are active at the same time, all data points to which the include filter applies are read, minus the data points that are excluded by the exclude filter.

With these filters, only values of data points containing the word “machine” would be read, but only if they do not contain the word “machine12”.

Testing and previewing filters

The “Data point preview” button opens an overview of the data points that would be read at runtime.

Vorschau

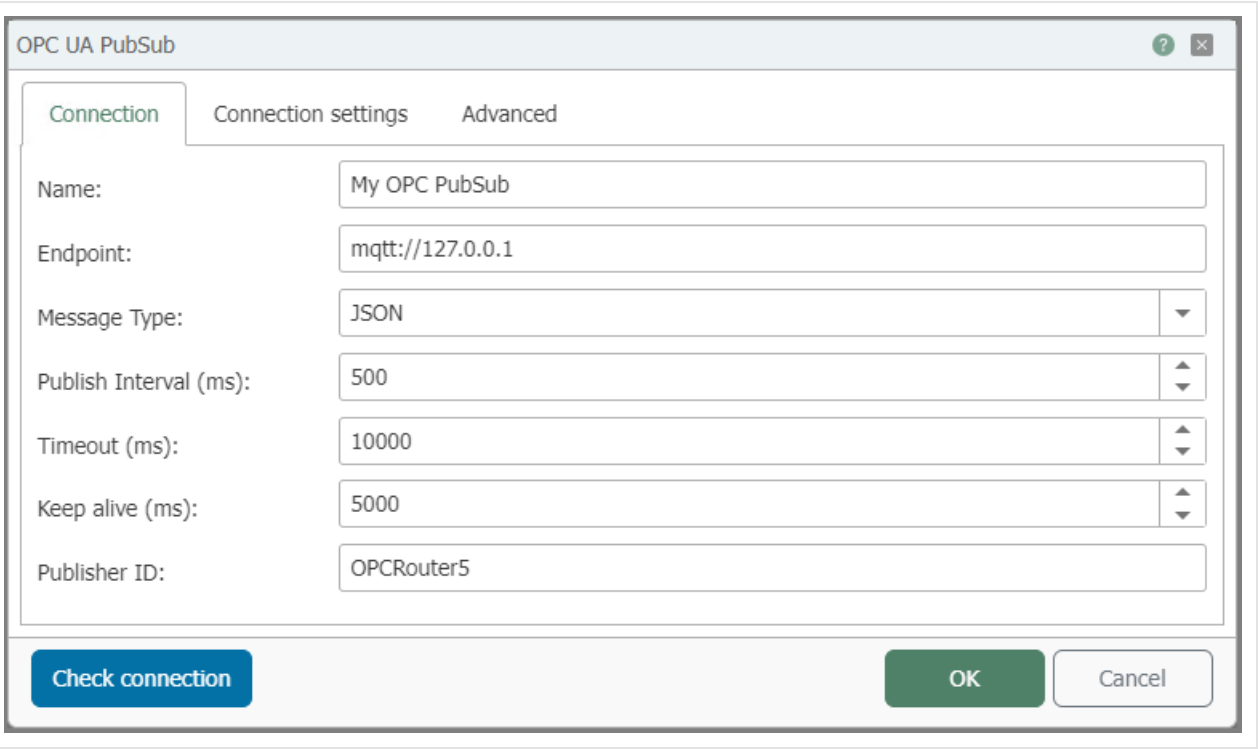
Pfad	Anzeigename	Typ	Identifier
Maschine01/Drehzahl	Drehzahl	Variable	ns=2;s=Halle01.Maschine01.Drehzahl
Maschine01/Temperatur	Temperatur	Variable	ns=2;s=Halle01.Maschine01.Temperatur
Maschine02/Drehzahl	Drehzahl	Variable	ns=2;s=Halle01.Maschine02.Drehzahl
Maschine02/Temperatur	Temperatur	Variable	ns=2;s=Halle01.Maschine02.Temperatur
Maschine03/Drehzahl	Drehzahl	Variable	ns=2;s=Halle01.Maschine03.Drehzahl
Maschine03/Temperatur	Temperatur	Variable	ns=2;s=Halle01.Maschine03.Temperatur
Maschine04/Drehzahl	Drehzahl	Variable	ns=2;s=Halle01.Maschine04.Drehzahl
Maschine04/Temperatur	Temperatur	Variable	ns=2;s=Halle01.Maschine04.Temperatur
Maschine05/Drehzahl	Drehzahl	Variable	ns=2;s=Halle01.Maschine05.Drehzahl
Maschine05/Temperatur	Temperatur	Variable	ns=2;s=Halle01.Maschine05.Temperatur

1 von 32

Schließen

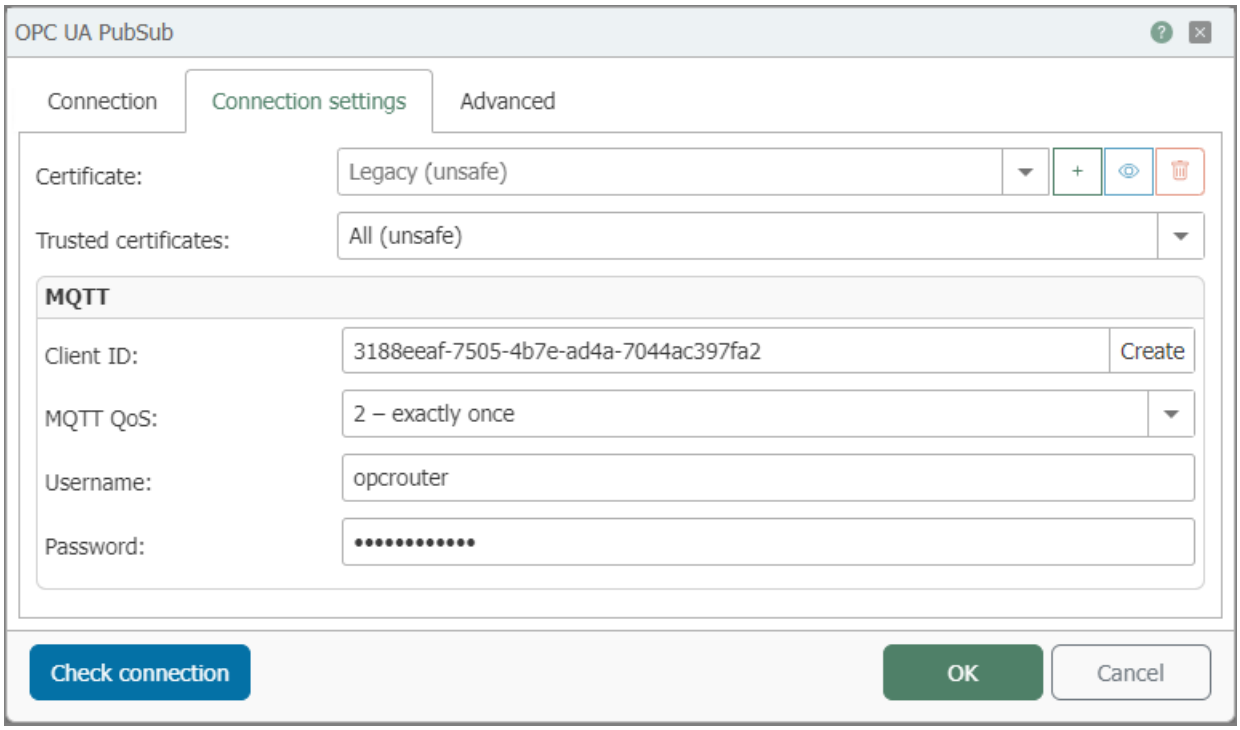
OPC-UA PubSub

Connection



PROPERTY	DESCRIPTION
Name	Enter a name for the plug-in.
Endpoint	Enter the endpoint to the OPC-UA PubSub server.
Message type	Select the message type: • JSON • UADP
Publish interval (ms)	Specify the interval at which you want to publish.
Timeout (ms)	Specify here after how many seconds a timeout occurs.
Keep alive	Period until the next message is sent.
Publisher Id	Enter the ID for the OPC Router.

Connection settings

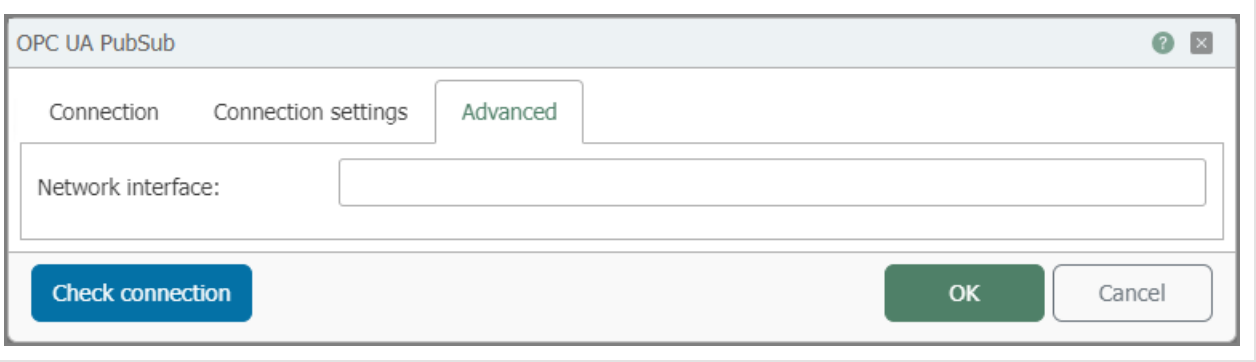


PROPERTY	DESCRIPTION
Certificate	Add a certificate for encrypted commuication.
Trusted certificates	Here you can select which server certificates are to be accepted.

MQTT

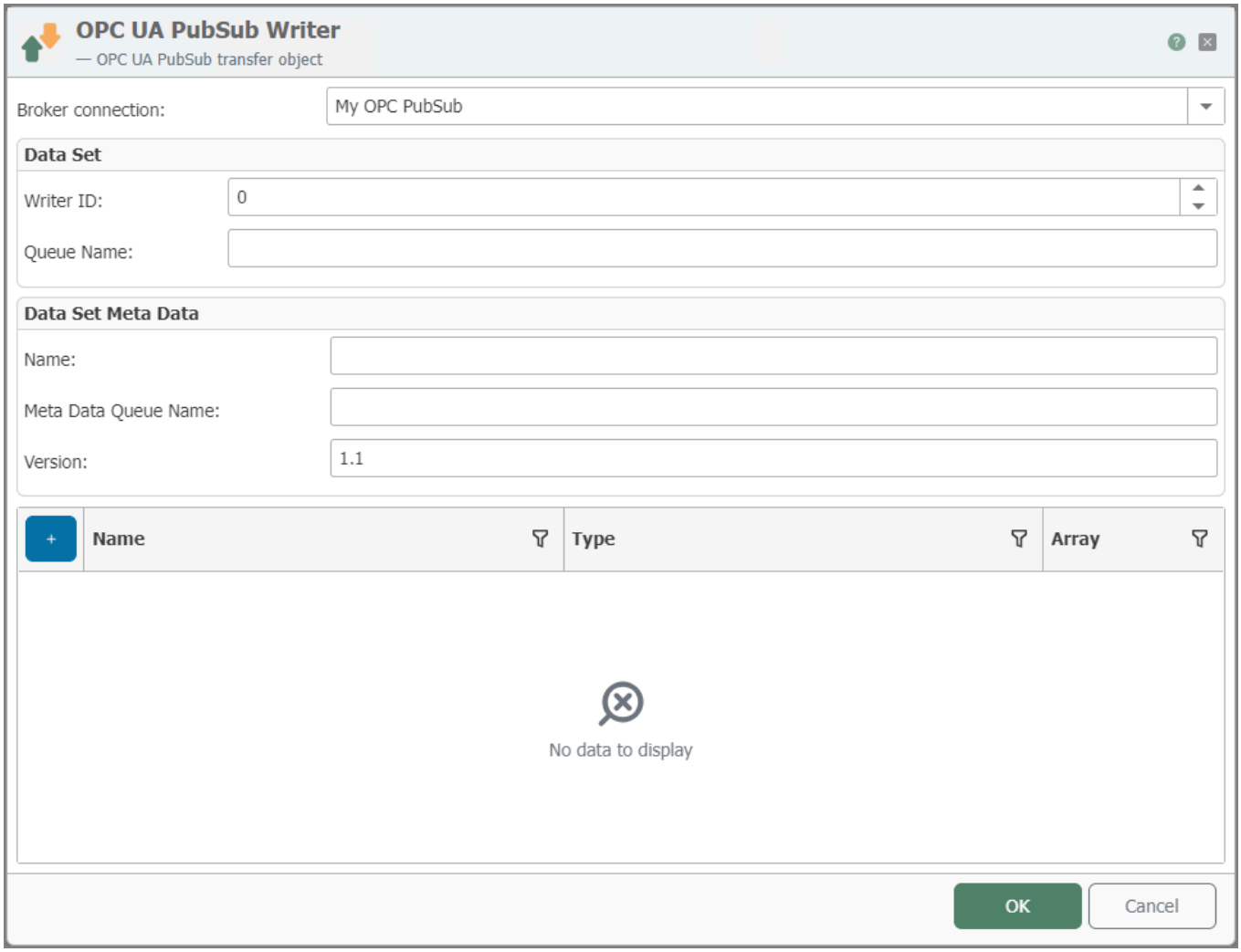
PROPERTY	DESCRIPTION
Client ID	Generate a client id with the button “Generate”.
MQTT QoS	Select the Qos.
Username	Enter a username.
Password	Enter a password.

Advanced



PROPERTY	DESCRIPTION
Network interface	Specifies which network interface is to be used.

OPC-UA Pub/Sub Writer



Connection

PROPERTY	DESCRIPTION
Broker connection	Connect the OPC UA Pub/Sub plug-in here.

Data Set

PROPERTY	DESCRIPTION
Writer ID	The transfer object selects the next available id. Another opinion is that you enter a Writer Id. **Important:** The ID must be between 1 and 65535.
Queue Name	Enter the queue name (topic).

Data Set Meta Data

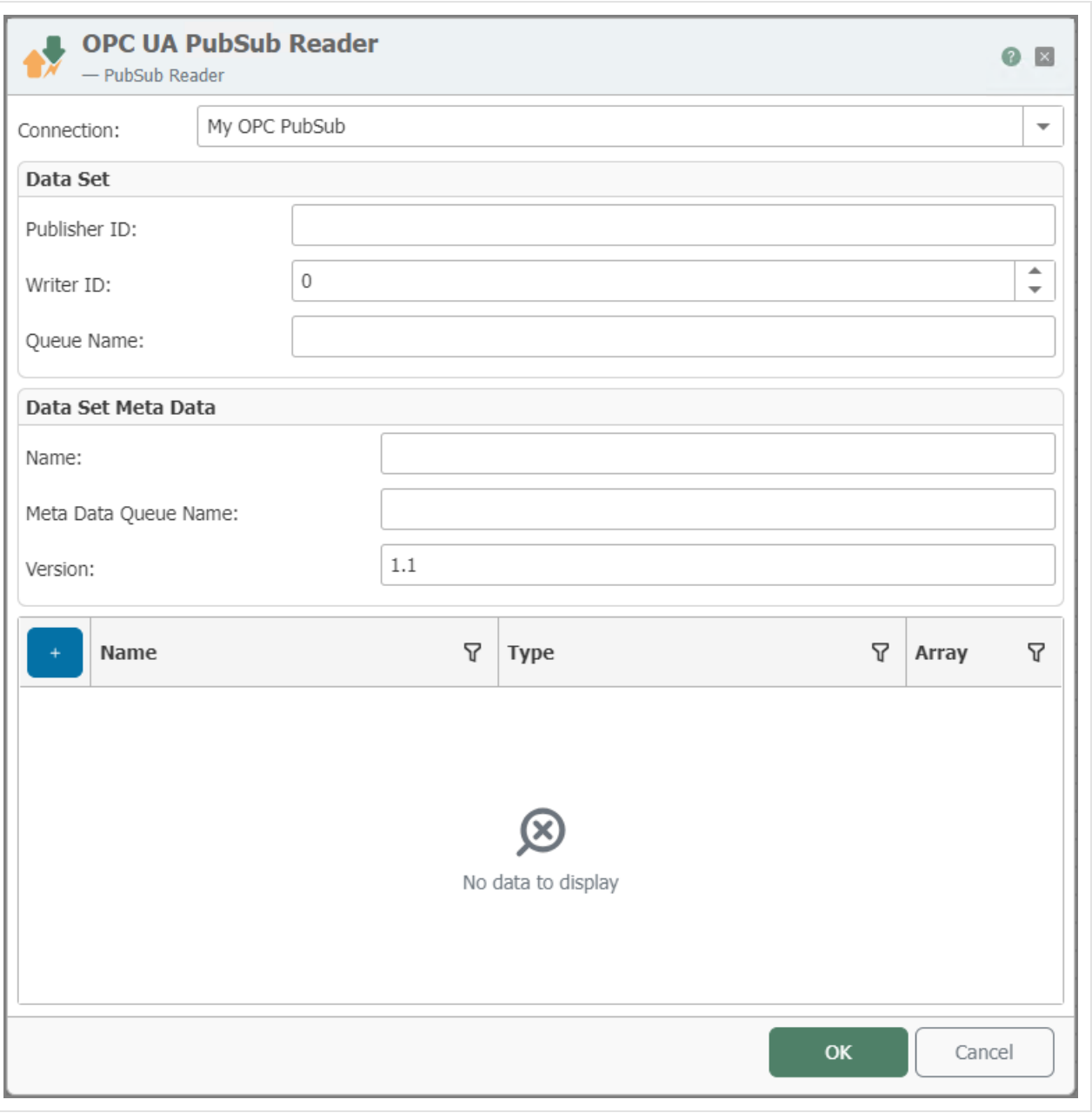
PROPERTY	DESCRIPTION
Name	Name of the data set.

PROPERTY	DESCRIPTION
Meta Data Queue Name	This is therefore a self-defined path where the information on the metadata can be found or where the OPC router should write the metadata if it writes metadata itself.
Version	Version of the data set. The version number must have a major number and a minor number, i.e. 1.0. If you update the data set, you should update the version number, i.e. 1.1.

Fields

PROPERTY	DESCRIPTION
Name	Specify the name of the data point here.
Type	Select the data type here.
Array	Specify whether the data point to be obtained is an array. You can set the following states: • filled field: field is NULL • empty field: field is false • Hook: field is true

OPC-UA Pub/Sub Reader



PROPERTY	DESCRIPTION
Connection	Connect the OPC UA Pub/Sub plug-in here.

Data Set

PROPERTY	DESCRIPTION
Publisher ID	Enter the Publisher Id of the Publisher here.
Writer ID	Enter the Writer Id of the Publisher here. **Note:** If you want to read all data sets of the queue you can type 0 for all writers.
Queue Name	Enter the queue name (topic) to which the subscriber should subscribe to the publisher.

Data Set Meta Data

PROPERTY	DESCRIPTION
Name	Name of the data set. **Note:** If you use “Browse” you don't must enter any value.
Meta Data Queue Name	The “Meta Data Queue Name” that was already specified in the Writer.
Version	Version of the data set. The version number must have major number and a minor number i.e. 1.0. If you update the data set you should update the version number i.e. 1.1. .

Fields

PROPERTY	DESCRIPTION
Name	Specify the name of the data point to be read.
Type	Specify which data type the data point to be read has.
Array	Specify whether the data point to be obtained is an array. You can set the following states: • filled field: field is NULL • empty field: field is false • Hook: field is true

Heidenhain

Heidenhain plug-in connects OPC Router to all accessible Heidenhain controls.

Operation is standard:

1. Open plug-in by double-click “Heidenhain”. The list of created Heidenhain connections is displayed.
2. Chose “Add new set as access data” or double-click an existing entry to edit.
3. Specify access data. (chapter “Plug-in”).
4. Confirm with “OK”.

The Heidenhain server can now be used to define a connection.

Plug-in

The following properties have to be specified for each set of Heidenhain access data:

Heidenhain Access Data

General

Name	This set's name. The name will be shown in projection. Select a name you will recognize!
Host/IP	Heidenhain-Server's address
Type	Heidenhain CNC type.
DNC Protocol	Select DNC Protocol
Port	Select Ethernet port.
Update rate	Select values update rate in milliseconds
Timeout	Select the connections timeout in milliseconds
Transfer OPCNULL values	Check if NULL values should get transferred,

Security

CFGDATAACCESS	If the password is set, the checkbox must be activated and the valid password entered in order to be able to access the configuration.
PCLDATAACCESS	If the password is specified, the checkbox must be activated and the valid password entered in order to be able to access the data of the PLC program and the properties of the PLC runtime.
TABLEDATAACCESS	If the password is set, the checkbox must be activated and the valid password entered in order to able to access ll tables, including that of the tool.

Following control systems are supported:

Important: The machines need an Ethernet interface an the option DNC / option 18 needs to be enabled.

Control system	**from NC Version**	**from DNC Version**
iTNC 530	340420-06, 340422-01	1.6.1
iTNC 530	340480-01	1.6.1
iTNC 530	34049x-03, 60642x-01	1.6.1
NCK based		

TNC 128	771841-01	1.5.1
TNC 320	340551-03, 771851-01	1.5.1
TNC 620	34056x-01, 73498x-01, 81760x-01	1.5.1
TNC 640	34059x-01	1.5.1
MANUALplus 620	548328-05, 54843x-01	1.5.1
CNC PILOT 620	688945-01	1.5.1
CNC PILOT 640	688946-01, 688947-01	1.3.1
MillPlusIT V6x0	53895x-03, 73738x-01	1.5.1
GrindPlusIT	510060-04	1.5.1
GrindPlus 640	73502x-01	1.5.1

Transfer Object

The Heidenhain server's set of access data has to be specified within plug-in configuration. Create a Heidenhain transfer object as described above..

Double-click on transfer object to assign the properties:

Access data	Select a set of Heidenhain access data. Dependent on plug-in configuration.
Items	Enter Heidenhain items needed. Use the Tag Browser to browse items.
Value Timestamp	Returns the timestamp of the transferred value.

Within the item's list, select several items by holding mouse button down ("Ctrl + a" will select all). Selected items can be copied to clipboard and inserted ("Ctrl + c", "Ctrl + v").

Using the Tag Browser

Using the button "Tag Browser", you will open another dialog to browse and select Heidenhain items. The dialog is not modal, you can switch between transfer object and browser.

Enter a search string to hide the tree view and show all items matching the search term.

1. Select the row for the new item.
2. Open the Tag Browser
3. Assign Item by double-click.

Trigger

Heidenhain-Message

The message trigger triggers the data transfer if the assigned numeric data increments by 1. You can also send an answer and, if the transfer is incorrect, an error message to two other data points.

Access data	Select a set of Heidenhain access data.
Item	Item whose increase will execute the transfer. Browse items with tag browser (“...” button).
Write answer	Will transfer Heidenhain item’s value (or set a bit) to another item as confirmation that the transfer executes.
Ttem	Heidenhain item the answer is send to. It must be a logical (bit) data point that has state “0” or “1”. Use the button “...” to open the tag browser (see below) for the interactive selection of an item.
Write error bit	Activate to set a bit with transfer errors.
Item	Heidenhain item (bit) that is set with transfer errors. It must be a logical (bit) data point that has state “0” or “1”. Use the button “...” to open the tag browser (see below) for the interactive selection of an item.
Trigger with start	Release trigger once with OPC Router start.

Heidenhain-Bit

The bit trigger triggers the data transfer when the assigned logical data point takes a value 1. If the assigned data point is not a logical data point, the trigger triggers when the data point takes a different value as NULL. You can also send a reply and a fault message to two additional data points if the transfer is incorrect.

Access data	Heidenhain connection which the relevant data point is reached.
Item	Heidenhain data point that triggers the transfer as soon as it assumes a value other than 0. Use the button “... ” to open the tag browser (see below) for the interactive selection of an item.
Reset bit	Resets the Heidenhain item to zero after the transfer. Should usually be disabled because a bit should only be written from one side. Instead, use the answer bit to tell the controller that the transfer has completed.
Answer bit active	Sends one bit to another data point in response that the transfer has completed.
Item	Heidenhain data point to which the response is sent. It must be a logical (bit) data point that has state “0” or “1”. Use the button “...” to open the tag browser (see below) for the interactive selection of an item.
Reset answer bit after	After the set time, the response bit is reset.
Error bit active	Sends a bit to another data point if the data transfer was faulty.
Item	Heidenhain item (bit) that is set with transfer errors. It must be a logical (bit) data point that has state “0” or “1”. Use the button “...” to open the tag browser (see below) for the interactive selection of an item.
Waiting for failing edge	If you don't want the OPC Router to trigger continuously when the bit is active, activate this checkbox. This causes the OPC Router to trigger only once when the bit is active and then wait for a "falling edge" before triggering again.

Heidenhain-Threshold

Threshold trigger executes a data transfer with off-limit condition (of Heidenhain value)

The properties are:

Access data	Select a set of Heidenhain access data.
Item	Item whose off-limit condition will execute the transfer. Browse items with tag browser ("..." button).
Threshold	Absolute value.
Trigger if item	Set if threshold has to be under- or overrun.
Hysteresis	If the trigger has executed the transfer once, it might be required that the trigger is re-released only if the Heidenhain item's value is clearly different from threshold value. If this is the case, please specify the difference between Heidenhain item and threshold with which the trigger is set active again. Example: The threshold is 20 °C, to be triggered when overrun. Temperature varies between 19 and 21 degrees, but only the first overrun has to be transferred. Hysteresis could be set to 18 degrees absolute value. Thus, the trigger will only release after the temperature has fallen below 18 (after having overrun 20).

Tip: If the connection should be triggered permanently on off-limit condition, set hysteresis absolute value to the maximum value the item can take. Thus, it will trigger in OPC Routers cycle time interval as long as it is in off-limit condition. Combine with a data change trigger linked with AND to only trigger data changes above the limit. In the example above, the temperature will not exceed 100 degrees, then with threshold = 20 and trigger reset from 100 absolute value the trigger would always trigger when the temperature is between 20 and 100 degrees, in the frequency of the cycle time of the router (settings). You could limit this with an AND-linked datachange trigger to record only the data changes above the threshold.

Heidenhain Data-Change-Missing

The Heidenhain Data-Change-Missing-Trigger triggers if a value does not change its value for a specified time.

Access data	Select a set of Heidenhain access data.
Trigger item	Item which should get observed. Browse items with tag browser (“...” button).
Check period	If the data point does not change its value within this time, the trigger triggers.
Trigger mode	Select if the trigger should trigger once or continuous. (Continuous means the selected check period)
Metadata output	Activate this if you want to send the last known data of the data point. It includes the quality ID, the quality string, the timestamp and the name of the data point.

Heidenhain-Data Change

Data change trigger executes transfers with a changing Heidenhain value. You might specify tolerances and time intervals as to optimize the number of transfers – forced transfer, even if no change is made, or limitation of transfers with many data changes.

Access data	Select a set of Heidenhain access data.
Trigger item	Item whose change will execute the transfer. Browse items with tag browser ("..." button).

Use the extended button to define further properties

At most every	Minimum interval in which transfer will be executed even if no data change has occurred. Integers only, unit from milliseconds to hours.
At least every	If data change frequently the number of transfers can be limit. An entry means, e. g. Transfer every second maximum even if data changes more frequently. Integers only. Units from milliseconds to hours.
Trigger at ... difference (%)	Transfer will only execute if data changes by this difference.
Trigger at ... difference (absolute)	Transfer will only execute if data changes by this difference.
Trigger at each data change	Default. Overrides tolerance settings.
Trigger if quality changed	Activate the checkbox if you want to trigger at quality change.
Trigger if timestamp changed	Activate the checkbox if you want to trigger when the timestamp is changed.
Metadata output	Activate the checkbox if you want the metadata to be output.

Tip: To set conditions for relative and absolute tolerance, project two data change triggers linked with AND or OR.

Euromap-77

Plug-in

Please define the following properties to configure set of Euromap-77 connections:

Name	Any name where this connection appears in the connection configuration. Choose a name that is clearly recognizable for you!
Client certificate	Select a client certificate from the drop-down menu. You can use the plus sign to add another certificate and the minus sign to remove one. Another button allows you to display the properties of an existing certificate.
Authentication	Select authentication from the identity verification drop-down menu.
Endpoint	Enter an endpoint for your connection.
Machine name	Select a machine name from the drop-down menu.
Use highest security level	Activate the checkbox if you want to use the highest safety level and make the entries accordingly
Login credentials	If you have selected “Username and password” at authentication, entries are required in this area.
Security settings	If you have not enabled the highest levels of security, then security settings will be required in this area: • MessageSecurityMode • SecurityPolicy If unencrypted communication is required, the client certificate “Without” and MessageSecurityMode “None” must be selected.

Create certificate

Application name	Assign an application name for the certificate.
Organization	Enter an organization as needed.
Organization unit	If necessary, enter an addition to the organization. For example, a place or hall of the organization.
Application URI	Enter the application URI.
DNS name	Enter the DNS name of your server.
Applicant	Enter the applicant.
Expires in (Months)	Specify the validity period of the certificate in months.

Confirm your entries with “Create”. You can now select the client certificate in the profile properties.

Transfer Object

An Euromap object is available to you as transfer source and destination The required profile must be configured as a plug-in instance. Create a transfer object within the connection as explained.

After double-clicking on the transfer object, set the following properties:

Connection	Here the profiles are available as configured instances of the Euromap77 plug-in. Select a profile from the drop-down list.
Endpoint	Select an endpoint from the drop-down list. With the button “Test” you can check if your connection works
Machine	Select a machine from the drop-down list. In the list below you will now get a selection list with all possible information of this machine. Activate the areas that are necessary for your configuration. Now you can use this selected information as transfer information.

Trigger

The Euromap77 trigger triggers the data transfer when the temperature value changes of a data point. The following trigger applications are available:

- Temperature datachange trigger: Monitors temperature changes within the selected injection molding machine.
- Temperature deviation: To monitor temperature-threshold violations of the selected injection molding machine.

The individual properties are:

Profil	Here the profiles are available as configured instances of the Euromap77 plug-in. Select a profile from the drop-down list.
Destination	Select an endpoint from the drop-down list. The “Check” button allows you to verify that your connection is working.
Machine	Select a machine from the drop-down list. In the list below you will now get a selection list with all possible information of this machine. Activate the areas that are necessary for your configuration. You can now use this selected information as transfer information.

By selecting the temperature zone type “InjectionUnit” or “Mould” and selecting the appropriate temperature monitoring, you specify the following properties:

Features temperature datachange trigger

At most every	Time interval in which data is transferred in any case, even if the value of the data point has not changed. Default always as an integer, no decimal places. As units therefore milliseconds, seconds, minutes or hours are to be selected.
At least every	If the value of the data point changes frequently, the transfer number can be limited over an interval. The entry thus means: It is at most e.g. transmitted every second, even if the value changes more often Default always as an integer, no decimal places. As units therefore milliseconds, seconds, minutes or hours are to be selected
Trigger at ... difference (%)	The transfer is only triggered if the value of the data point changes by this entry (relative).
Trigger at ... difference (absolute)	The transfer is only triggered if the value of the data point changes by this entry (absolute).
Trigger at each data change	Default setting, sets trigger at ... difference override.

Features temperature deviation

Trigger if Low-Deviation-Temperature 1 deceeds	When activated, the temperature is only triggered when falling below the low deviation.

Trigger if High-Deviation-Temperature 1 exeeds	When activated, the temperature is triggered only when the highest deviation is exceeded.

Machine mode

This trigger initiates the connection when the machine accepts the status.

Profile	Select a connection that you have created in the plug-in.
Trigger on following status	Select the respective status to be triggered here.

Parts counter

This trigger outputs the number of parts counted.

Profile	Select a connection that you have created in the plug-in.

Note: Although the trigger does not display any other fields, the property is displayed after confirmation in the object.

Production status

This trigger initiates the connection when the following statuses are accepted.

Profile	Select a connection that you have created in the plug-in.
Trigger on following status	Select the respective status to be triggered here.

Charm

Charm devices send results to a defined port on the computer using the OPC Router. The OPC Router monitors this port and accepts the messages (results of measurements (inhibitor, microtoxins, etc.).

No connections are configured in this plug-in. The port to which the Charm device sends is set directly in the trigger.

Basically, a separate port must be created for each Charm device type. Recommended procedure: All charm devices of one type are sent to the same port. In the OPC Router a single connection is created. In addition to the results, the OPC Router also transmits information about the sending Charm device to the database. The logic of how the results are processed is in the database (or another transfer destination).

However, if you want to create a separate connection for each Charm device, a separate port must also be available for each device.

Timeout	When a Charm device establishes a connection, it has the time entered in milliseconds to send all data to the OPC Router. Requires the Charm device longer, the connection is disconnected from the OPC Router.
Maximal message length	When connecting the Charm device to the OPC Router, data is sent. If more than the allowed characters are transferred, the OPC Router disconnects the connection and sees the message as faulty.

Plug-in

The following properties are set in the Charm plug-in configuration for each set of Charm access data

Name	This set’s name. The name will be shown in projection. Select a name you will recognize!
Port	Port on this computer (hosting OPC Router) to which the specific Charm devices will send data.

Trigger

Charm-Error

The charm error trigger triggers a data transfer in the event of a faulty telegram. Since the charm trigger works faster than the error trigger, the property of the trigger trigger for the corresponding port / device must be disabled. Then, in a further connection controlled by the charm error trigger, information about erroneous messages can be treated separately, e.g. a separate database table.

The individual properties are:

Device type	Type of charm reader.
Port	Port on this machine (with the OPC Router) to which the Charm device sends its data.

Confirm your entries with “OK”.

Note: In the Charm device, the IP address of the computer with the OPC Router must be announced together with the port number set here. That's where the charm device has to send the data. The OPC Router then transmits the data record by record.

Charm trigger

Note: In the Charm device, the IP address of the computer with the OPC Router must be announced together with the port number set here. That's where the charm device has to send its data. The OPC Router then transmits the data record by record.

Fields of predefined Charm device types

Name	EZ	EZ-M	EZ-M2	Description
----	--	----	-----	----------
Answer	X	X		Numerical test result including unit where necessary
Assay	X	X		sample type
ChannelNumber	X	X	X	EZ channel number
CommodityName		X	X	Raw material / article
controlLINE	X	X	X	Intensity of the Control Line
CurrentCommoditySelectIndex		X		
CurrentDilutionSelectIndex		X		
CurrentMatrixSelectIndex		X		
CurrentTestSelectIndex		X		
Date	X	X	X	Date of the test
DE_Name		X	X	
Dirt	X	X	X	Degree of contamination of the lens
FailCode	X	X	X	Error code on failed test
Flow	X	X	X	Flow value (Correct application of the milk on the test strip)
Interpretation	X	X	X	Interpretation of test result (positive, negative, etc.)
InterpString	X			(Interpretation in newer firmware)
LotNumber	X	X		LOT number (as entered on the unit)
MatrixNumber			X	Matrix number of the test result
Mode	X	X	X	Mode of the Charm device
OperatorID	X	X		Operator ID (as entered on the device)

Ratio		X	X	Ratio of Extraction Solvent to Extraction
Remmisions	X	X		
Result	X	X	X	Test result (positive, negative, invalid)
RFID			X	
SampleID	X	X	X	Test ID (as entered on the device)
StructVersion	X	X	X	
testLINE	X	X	X	Intensity of the test line
TestOutputName	X	X	X	
TestTemperature	X	X	X	Incubator temperature at test time
TestType			X	
Time	X	X	X	Time of the test
TipBlue	X	X		Displays value “Blue” when the test color is identified
TipGreen	X	X		Displays value “Green” when the test color is identified
TipRed	X	X		Displays value “Red” when the test color is identified
UnitSN	X	X	X	
XLine	X	X	X	Intensity of the X Line
YLine	X	X	X	Intensity of the Y Line
ZLine	X	X	X	Intensity of the Z Line

Scanner

Plug-in

You can set the following properties in the configuration of the scanner plug-in for a barcode reader:

Name	Any name where this device appears in the connection configuration.
COM Port	Enter the COM port for communication with the barcode reader.

Connection settings serial Barcode reader

Baud rate	Select a baud rate. Specifies how often a signal in a communication channel changes state.
Data bits	Specify how many bits of data to transfer.
Parity	Select a parity from the drop-down menu. For example, Even, Mark, Space.
Stop bits	Specify how many stop bits are needed.
Flow control	Select how the data flow should be controlled. Eg. software, hardware

Connection settings Tcp Barcode reader

IP	Enter the IP of your Tcp barcode scanner.
Port	Enter the port of your Tcp barcode scanner.

Trigger

Currently a trigger for the barcode reader is available.

The individual properties are:

Instance	Select an instance from the drop-down menu for the barcode reader whose scan product you want to use.
Values	Select the value to be read, in this case “Barcode”.

RFID

This operation is standard:

1. Open the plug-in by double-clicking on “RFID Reader” in the “Plug-ins” menu.
2. Choose “Add” or double-click an existing entry.
3. Make the necessary settings (see section “Plug-in”).
4. Confirm with “OK”.

You can now make connections with your RFID reader.

Plug-in

The following properties are set in the configuration of the RFID Reader plug-in:

Name	Any name where this connection appears in the connection project planning. Choose a name that you can recognize uniquely!
Client certificate	Select a client certificate from the drop-down menu. Using the plus sign, you can add another certificate, using the minus sign to remove one. Another button allows you to display the properties of an existing certificate.
Authentication process	Select an authentication from the Identity Verification drop-down menu. If you select Username, you must enter Username and Password at Registration information.
Endpoint	Enter an endpoint for your connection `opc.tpc://`
RFID-Reader name	Select your RFID Reader from the drop-down menu.
Scan mode	Selection of scan mode between “Scan manually”, “Scan in regular interval” or “Scan in background”.
Scan interval (ms)	Enter the scan interval in milliseconds.
Trigger if minimum signal strength is reached	Only trigger the scan if the signal strength is sufficient.
Signal strength (RSSI)	Tags with a lower signal strength during scanning are ignored by the OPC Router.
Trigger if an RFID tag was not captured for a defined time.	Specifies how long a tag must be outside the reading field before the OPC Router triggers another transfer for this tag.
Hibernation interval (ms)	Period where a RFID tag is not to be recorded.
Highest security level	Activate if the highest security level has to be used.

Registration information

Username	Enter a username to authenticate yourself.
Password	Enter a password to authenticate yourself.

Security settings

Security mode	No Security = “None”, Signing = “Sign”, Signing and Encrypting with “SignAndEncrypt”.
Encryption	Encryption methods to choose from. Set the highest method available on the OPC server.

Use the “Check connection” button to check whether your connection to the RFID Reader is successful.

Transfer Object

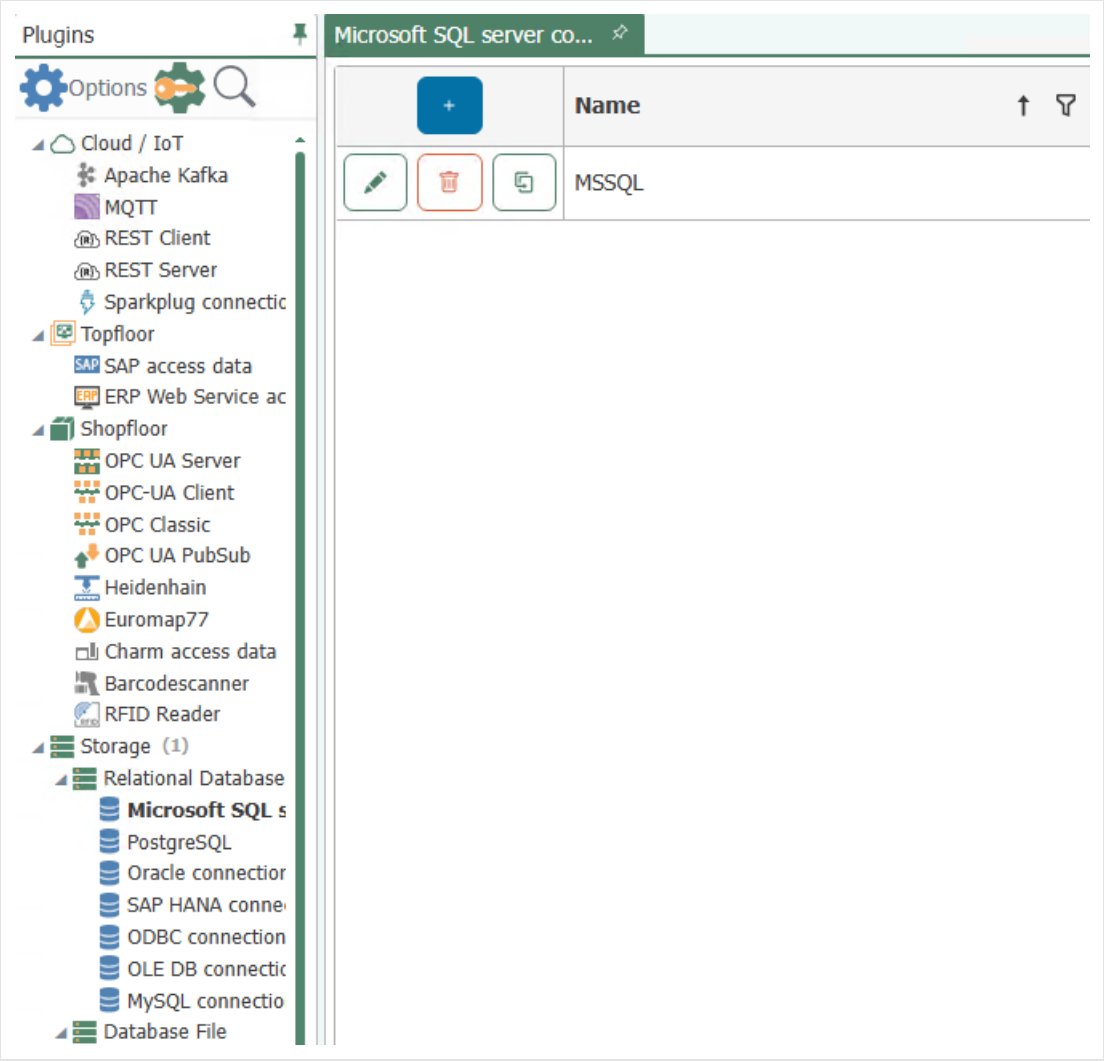
RFID reader transfer objects can synchronize scans, read tags, write, lock, destroy or create passwords. You can directly configure the transfer object within the connection as described.

Trigger

Scan asynchronous trigger of the RFID reader to trigger asynchronous scans. Select an instance from the drop-down menu after double-clicking.

Storage

The database plug-in connects OPC Router to as many database servers as you can reach. The database specific clients are required on the OPC Router's computer and have to be installed manually (e. g. MS SQL Server Native Client, MySQL ODBC driver etc).



Defining access data:

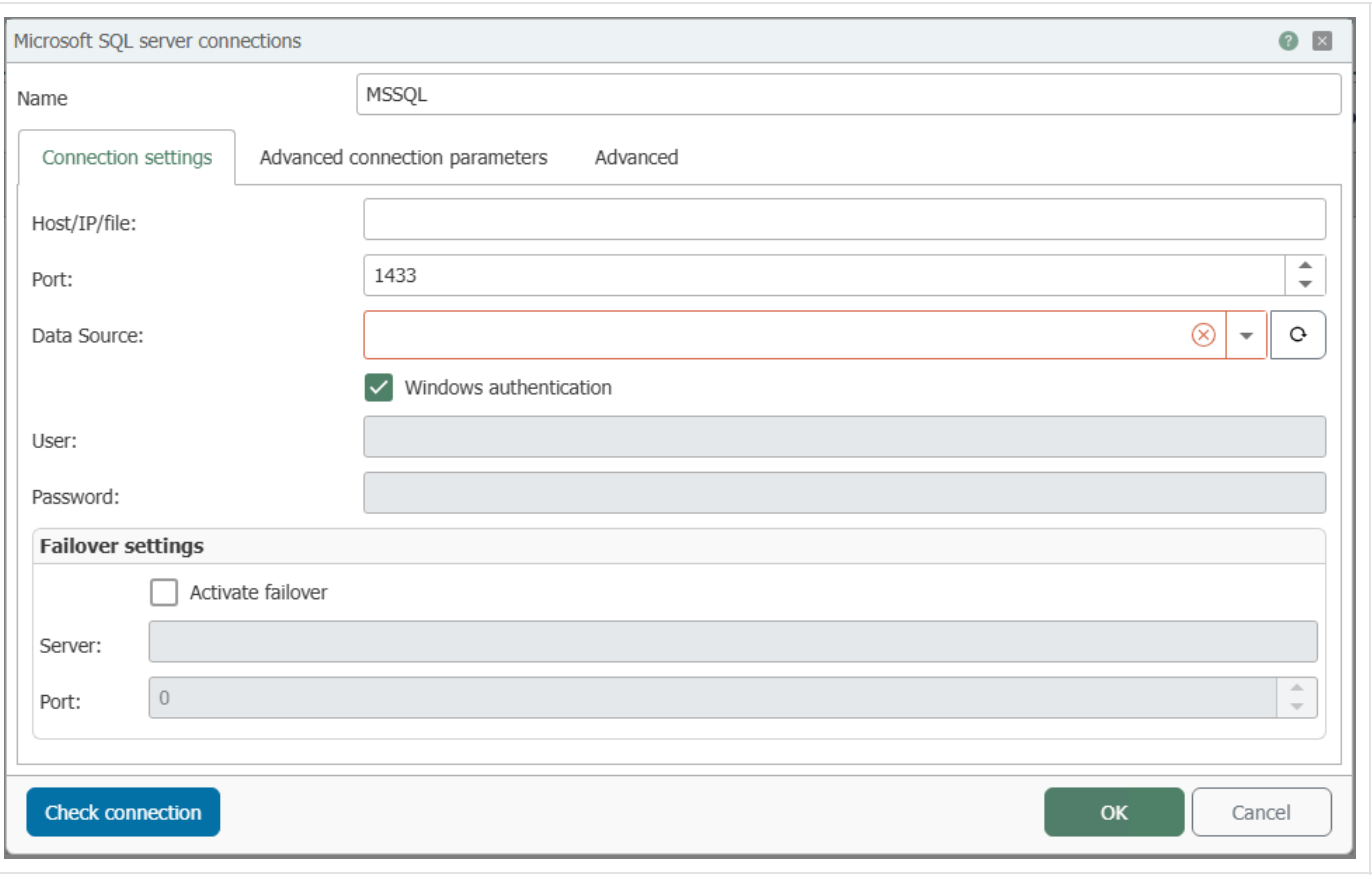
1. Open a plug-in by clicking on a database in the project configuration's plug-in list.
2. Chose "Add" or double-click an existing entry to edit.
3. Specify access data. (chapter "Relational databases").
4. Click on button "Check connection" to verify the OPC Router is getting access to database server and if access data are correct. This is to exclude one typical source of errors.
5. Confirm with "OK".

The database server can now be used to define a connection.

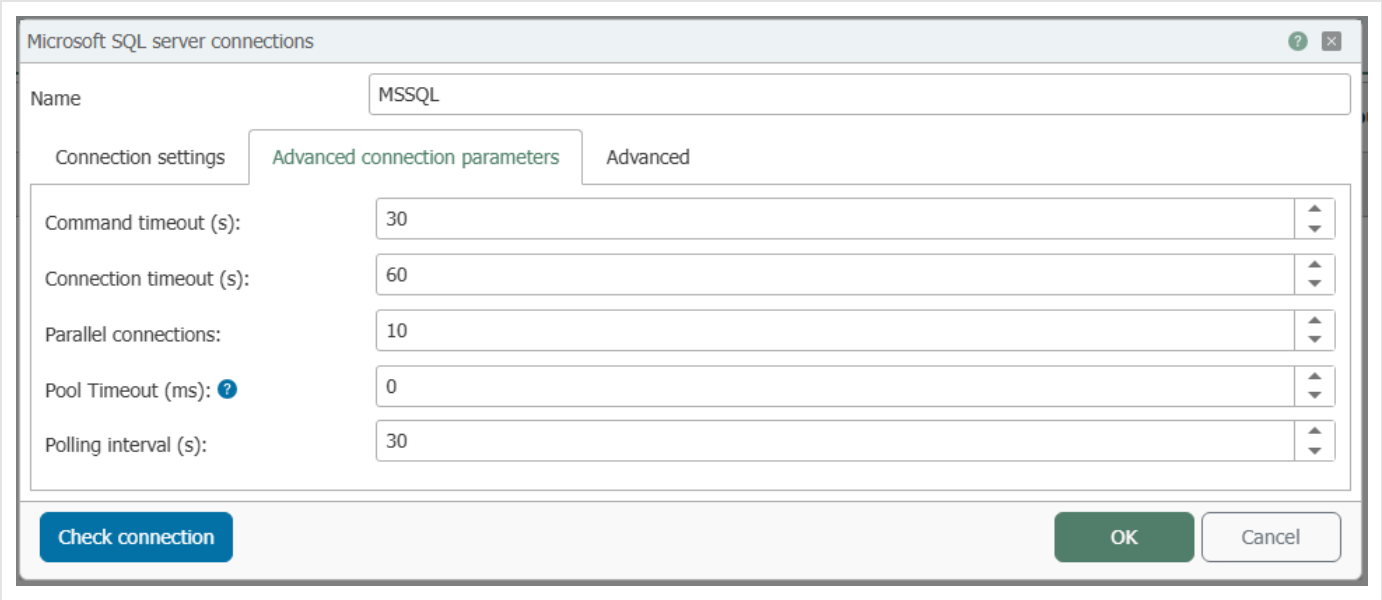
The following properties have to be specified for each set of database access data:

Name	This set's name. The name will be shown in projection. Select a name you will recognize!

Type	Type of database server, select via drop-down: • MS SQL Server • XML_File • My SQL • My SQL 5 • MS Access • Oracle • OLE_DB • ODBC • dBASE • SQLite OLE DB or ODBC drivers for other database servers will be available at the server's manufacturer or vendor. Install according to manufacturer's manual. OLE DB drivers are recommended if available.



MSSQL Connection settings



MSSQL Advanced connection parameters

Properties not applicable for a specific type of server are deactivated. Please refer to database server's manual for server-side configuration. Please contact your database administrator to get the correct setting for the properties. The picture shows connection to MS SQL Server after a successful test.

Relational databases

Overview over relational databases:

Microsoft SQL Server Plug-in

Oracle Plug-in

SAP HANA Plug-in

ODBC Plug-in

OLE DB Plug-in

MySQL Plug-in

Database Transfer Object

Database trigger

Microsoft SQL Server Plug-in

The Microsoft SQL Server plug-in in the OPC Router enables communication with Microsoft SQL servers. With this plug-in, you can read data from existing Microsoft SQL databases, write new data, change data or delete data. You can also transfer data from other OPC Router plug-ins such as OPC UA, MQTT or SAP to Microsoft SQL Servers. The plug-in supports various SQL commands and queries. These can be created graphically.

Authentication

SQL Server authentication

SQL Server authentication is an alternative authentication mode that can be used if Windows authentication is not available. In this mode, a SQL Server login name and password are used to verify the identity of the user. To authenticate at the SQL Server using SQL Server authentication, the user must have a valid SQL Server account with the permissions required to perform the desired tasks.

Windows authentication

With Windows authentication, the user's identity is verified using their Windows user account. This means that the user does not have to log in with a separate SQL Server username and password. Instead, the user is automatically authenticated when they log in with their Windows user account. This authentication is only relevant for Windows systems and does not work with Docker or Linux.

Important! Even if another Windows user was used for the connection test, the OPC Router service user must have the rights required for authentication.

Change Notification Broker

The Change Notification Broker is a function of the SQL Server that makes it possible for the OPC Router to receive notifications when data was changed in a database. The broker uses a queue to send notifications to the OPC Router.

If a trigger is configured and part of a connection in the OPC Router, it connects to the configured database and checks whether the Change Notification Broker is activated. If this is not the case, a warning is being displayed in the corresponding trigger telling that the OPC Router regularly queries the database tables. The query frequency is determined by the property set in the plug-in

In order to successfully configure the database connection, the user establishing the connection must have certain authorizations. Please check whether the user has these authorizations or request them.

The following authorizations are required

ALTER, CREATE MESSAGE TYPE, CREATE CONTRACT, CREATE QUEUE, CREATE SERVICE

To be able to use the Notification Broker without DB_Owner rights, please execute the following SQL script:

```
CREATE SCHEMA [OpcRouterSchema] AUTHORIZATION [OPCRouterUser];
GO
ALTER USER [OPCRouterUser] WITH DEFAULT_SCHEMA = [OpcRouterSchema];
GO
GRANT SELECT to [OPCRouterUser];
GO
GRANT CREATE PROCEDURE to [OPCRouterUser];
GO
GRANT CREATE QUEUE to [OPCRouterUser];
GO
GRANT CREATE SERVICE to [OPCRouterUser];
GO
GRANT VIEW DEFINITION TO [OPCRouterUser];
GO
GRANT SUBSCRIBE QUERY NOTIFICATIONS TO [OPCRouterUser];
GO
GRANT RECEIVE ON QueryNotificationErrorsQueue TO [OPCRouterUser];
GO
GRANT REFERENCES on CONTRACT::[ to [OPCRouterUser];
GO
```

Required roles and rights for database transfer objects

ACTION/ TRANSFEROBJECT	DESCRIPTION	ROLES	RIGHTS
Update		**db_datawriter**	UPDATE
Select	without TransferState	**db_datareader**	SELECT
Select	with TransferState	**db_datareader, db_datawriter**	SELECT, WRITE
Stored Procedure		**db_executer**	EXECUTE
Delete		**db_datawriter**	DELETE
Insert		**db_datawriter**	INSERT

Relevant timeouts

A timeout is a situation in which an action could not be completed within a certain amount of time. In the case of the OPC Router, a timeout can occur during communication with the Microsoft SQL Server.

Types of timeouts

- **Connection timeout**: The connection timeout is the maximum amount of time that the OPC Router waits for a connection to the Microsoft SQL Server.
- **Command Timeout**: The Command Timeout is the maximum amount of time the OPC Router waits for an SQL command to be executed.

If the command timeout is increased, the OPC Router timeout should also be increased. Otherwise the timeouts may overlap and lead to errors.

Causes of timeouts

- **High utilization of the Microsoft SQL Server**: If the Microsoft SQL Server is heavily utilized, there may be delays in the execution of SQL commands.

- **Network problems**: If the network connection between the OPC Router and the Microsoft SQL Server is disrupted, there may be delays or failures in communication.
- **Erroneous SQL commands**: If an SQL command is faulty, it cannot be executed by the SQL Server.

Microsoft SQL Server Performance

The performance of the Microsoft SQL Server is important for the OPC Router as it uses the database to store and retrieve data. If the performance of the SQL Server is impaired, this can lead to delays when reading and writing data.

Factors that influence the performance of the SQL Server:

- **Indexes**: Indexes are data structures that speed up the search for data in a table. If a table has no indexes, the SQL server must search the entire table to find the desired data. This can be very time-consuming if the table has many columns or contains many data records.
- **Fragmentation**: Fragmentation is a condition in which the data in a table is not stored in a continuous order. This can affect the performance of queries, as the SQL server needs more time to read the data.
- **Data organization**: Data organization is another important factor that can affect SQL Server performance. Efficient data organization can help SQL Server find and process data faster.

The maintenance of the Microsoft SQL Server is important for the performance of the OPC Router.

microsoft-sql-server-performance-guide

Using timestamps

When using timestamps, it is important that the same time zone is used everywhere. For example, if you use a timestamp in the UTC time zone, all other data that is compared with this timestamp must also be used in the UTC time zone.

We recommend using UTC

When using a local timestamp, it is recommended to explicitly set the DateTime handling to "Local" in the "Advanced" tab of the Microsoft SQL Server plug-in.

The OPC Router offers three different options for handling timestamps. Please note that the OPC Router must perform data type conversions when transferring data. If no information about the time zone is available during the conversion, the OPC Router normally uses the time zone of the OPC Router service. If the conversion takes place during insertion into the database, the option set here is used.

UNSPECIFIED	**THE OPC ROUTER LEAVES THE TIME ZONE AS IT IS.**
Local	The OPC Router converts the time stamp to the time zone defined by the process.
UTC	The OPC Router converts the timestamp to the UTC time zone.

Properties

NAME	**ANY NAME UNDER WHICH THIS DATABASE SERVER APPEARS IN THE CONNECTION CONFIGURATION. SELECT A NAME THAT IS UNIQUE FOR YOU DURING CONNECTION CONFIGURATION.**

"Connection settings" tab

Host/IP/file	Name or IP address of the Microsoft SQL Server

Port	Select the port for the database connection. The port can be changed if required.
Data source	Name of the database to be used.
Windows authentication	If this option is active, Windows authentication is used instead of the user and password properties. Local user accounts, DCOM users and Windows domain users (NT authentication) are supported. In order to use Windows authentication, the user under which the OPC Router service is started (by default "System") must have access rights to the database. To be able to project, the user under which the configuration environment was started must have access rights to the database. "Test connection" is not conclusive in this case, as testing is not carried out with the "System" user, but with the user who started the configuration environment.
User	Username for logging on to the MS SQL Server
Password	Password for logging on to MS SQL Server
Failover settings	Enter the address and port of a failover server here. (Only the "connection string" is adjusted. The failover cluster is configured directly on the Microsoft SQL Server)

"Advanced connection parameters" tab

COMMAND TIME-OUT	**TIME IN SECONDS THAT THE ROUTER WAITS FOR THE COMPLETION OF AN SQL COMMAND. IT MAY BE NECESSARY TO INCREASE THE VALUE FOR LONG QUERIES. PLEASE NOTE THAT THE CONNECTION TIMEOUT IN THE CONNECTIONS THAT TRIGGER LONG QUERIES MUST ALSO BE INCREASED.**
Connection time-out	Time in seconds that a connection to the database is allowed take.
Parallel connections	Number of parallel connections that the router is allowed to establish to the database. The more parallel connections are allowed, the faster the router is, but the database may be slowed down by it. Ask your database administrator how many parallel connections they allow for the OPC Router. Of course, this also depends on the size of the project.
Polling interval	The interval at which the database is polled. If the broker is activated in the database, polling takes place automatically.

"Advanced" tab

DATETIME HANDLING	**SEE #USING-TIMESTAMPS**
Bracket behavior	Property that determines when square brackets should be used in your Microsoft SQL Server plugin. You can find more information about this here.
Parameter conversion	When this option is active, the types are converted by the OPC Router, if not, they are converted by the target system.
Encrypt	This property determines whether encryption is activated. If the "Trust server certificate" option is disabled and the "Encrypt" option is enabled, the server name (or IP address) in a SQL Server SSL certificate must exactly match the server name (or IP address) specified in the connection string. Otherwise, the connection attempt will fail.
Trust server certificate	This property determines whether SSL is used to encrypt the channel, even if the certificate was not issued by a known certificate authority. If the "Trust server certificate" option is activated and the "Encrypt" option is deactivated, the channel is not encrypted.
Hostname in certificate	This option can be used to specify a different expected CN (name) or SAN (alternative name) for the server certificate if the server name does not match the CN or SAN in the server certificate.

DATETIME HANDLING	SEE #USING-TIMESTAMPS
QuerySpParameterDefaultValue	When this property is activated, the OPC Router queries the default values of the parameters of stored procedures and saves them.

Bracket behavior

- **Enclose if dot present:** This mode is intended for backward compatibility. If a dot (`.`) is contained in an identifier, square brackets are added. This is useful if you have database objects with dots in their names, such as `database.schema.[table.name]`.
- **Never enclose:** In this mode, no square brackets are added, regardless of the identifier. This can be useful if you are sure that your identifiers do not contain special characters or spaces.
- **Always enclose:** In this mode, square brackets are always added. This is useful if your identifiers may contain special characters, spaces or reserved words.

Microsoft SQL Server Performance Guide

For optimum performance, it is recommended that each table has a clustered index, preferably a primary clustered index.

A clustered index is a special type of index that determines the order of records in a table. It specifies how data is physically stored on the hard drive. A clustered index is crucial for database performance as it increases query speed and reduces fragmentation. Every table should have a clustered index, preferably a primary clustered index. A primary clustered index is based on the primary key of the table. The primary key is a column or a combination of columns that uniquely identifies each record in the table. A primary clustered index ensures that data in the table is sorted by the primary key, making it easier to search, insert, update, and delete data. It also prevents duplicate values in the primary key column, enhancing data integrity.

General tips

SQL Server Service Permissions

Administrator rights are required to configure the permissions.

To improve the performance and stability of the SQL Server, you need to make some settings in the local security policy of the computer on which the SQL Server is running. These settings allow the user running the SQL server (usually "network service"/"network service") to have certain rights that normally only administrators have. The following steps show how you can change these settings:

1. Open the "Local Security Policy" program.
2. Click on "Local Policies" and open "User Rights Assignment".
3. Search for the "Lock pages in memory" policy and double-click on it
4. Click on "Add User or Group" and enter the name of the user who runs the SQL server (usually "Network Service").
5. Click on "OK" and close the window.
6. Search for the "Perform volume maintenance tasks" policy, double-click on it and repeat steps 4 and 5.
7. Restart the computer for the changes to take effect.

With these settings, you can prevent the SQL server from having to free up disk space when it needs more memory and from having to wait a long time when it creates or extends files. This can significantly improve the performance and stability of the SQL server.

Table indexes

Clustered Index

A clustered index is a special type of index that determines the order of the data records in a table. It determines how the data is physically stored on the hard disk. A clustered index is important for the performance of a database as it increases query speed and reduces fragmentation. Each table should have a clustered index, preferably a primary clustered index. A primary clustered index is a clustered index based on the primary key of the table. The primary key is a column or combination of columns that uniquely identifies each record in the table. A primary clustered index guarantees that the data in the table is sorted according to the primary key, which makes it easier to search, insert, update and delete data.

Less is more

When optimizing databases, it is advisable to use a balanced number of indexes. Although indexes speed up queries, they also slow down data changes. Each index must be updated with every INSERT, UPDATE or DELETE, which takes up additional time and resources. It is therefore advisable to create only the necessary indices and to check regularly whether they are still efficient.\ \ You can use the script "*Query missing indices*" from the section "*Script collection*" *to identify missing indices in the database*.

Recovery model

The default recovery model for the database should be set to "Full", which means that the transaction log is constantly growing. It is strongly recommended to perform regular transaction log backups, usually every 15 to 30 minutes, to prevent the continuous growth of the transaction log and to create a full backup daily.

The "Full" restore model enables data to be restored since the last backup and at any time using the transaction logs.

However, if you do not need the transaction log or are dependent on a simple restore for certain reasons, you can alternatively change the **Recovery model** to **"Simple "**. Please note that this is associated with the <mark style={{ color: 'red' }}>**risk of data loss**, as only data contained in the last backup can be safely restored.

Fully qualified names in views & procedures

Another performance tip for SQL Server is to fully qualify the database objects by placing the owner (e.g. dbo) in front of the table name. This avoids the SQL Server having to perform name resolution for each object, which can increase execution time. It can also lead to unexpected results if several objects with the same name exist in different schemas. The use of fully qualified names ensures that the correct object is always addressed.

Script collection

Here are some useful SQL scripts that you can use for various purposes.

Note that these scripts may not be suitable for direct use in a production environment and may cause unexpected problems. It is strongly recommended to test them first in a protected environment before applying them to your data.

Query missing indexes

The script only provides suggestions for optimizing the database. It is important to evaluate the results critically and not to accept them blindly. The relevance of the index and the appropriateness of the includes depend on various factors that the script cannot take into account.

```
-- Missing indexes
select
 mig.index_group_handle, mid.index_handle,
 CONVERT (decimal (28,1),
 migs.avg_total_user_cost * migs.avg_user_impact *
 (migs.user_seeks + migs.user_scans)
 ) as improvement_measure,
 'CREATE INDEX missing_index_' + CONVERT (varchar,
 mig.index_group_handle) + '_' + CONVERT (varchar,
 mid.index_handle)
 + ' ON ' + mid.statement
 + ' (' + ISNULL (mid.equality_columns,'')
 + case when mid.equality_columns is not null and
 mid.inequality_columns IS NOT NULL then ',' else '' end
 + ISNULL (mid.inequality_columns, '')
 + ')'
 + ISNULL (' INCLUDE (' + mid.included_columns + ')', '') as
 create_index_statement,
 migs.*, mid.database_id, mid.[object_id]
from sys.dm_db_missing_index_groups mig
inner join sys.dm_db_missing_index_group_stats migs on
 migs.group_handle = mig.index_group_handle
inner join sys.dm_db_missing_index_details mid on
 mig.index_handle = mid.index_handle
where CONVERT (decimal (28,1), migs.avg_total_user_cost *
 migs.avg_user_impact * (migs.user_seeks + migs.user_scans)) > 10
order by migs.avg_total_user_cost * migs.avg_user_impact *
 (migs.user_seeks + migs.user_scans) desc
```

TOP 20 queries by CPU

```
-- Top 20 statements by cumulative CPU within last 1 hour
select last_execution_time, total_worker_time as [Total CPU
 Time], execution_count,
total_worker_time/execution_count as [Avg CPU Time],
text, qp.query_plan
from sys.dm_exec_query_stats as qs
cross apply sys.dm_exec_sql_text(qs.sql_handle) as st
cross apply sys.dm_exec_query_plan(qs.plan_handle) as qp
where DATEDIFF(hour, last_execution_time, getdate()) < 1
-- change hour time frame
order by total_worker_time desc
```

TOP 50 by I/O

```
-- Top 50 statements by I/O
select top 50
(qs.total_logical_reads + qs.total_logical_writes) /
qs.execution_count as average_io, substring
 (qt.text,qs.statement_start_offset/2, (case when
 qs.statement_end_offset = -1 then len(convert(nvarchar(max),
 qt.text)) * 2 else qs.statement_end_offset end -
 qs.statement_start_offset)/2) as query_text, qt.dbid,qt.objectid
 from sys.dm_exec_query_stats qs cross apply sys.dm_exec_sql_text
 (qs.sql_handle) as qt
order by average_io desc
```

Index defragmentation

To optimize the indexes, a maintenance task (index defrag.sql) should be run using the script (set variable `>>@ViewOnly<<` to 0 to execute the actions):

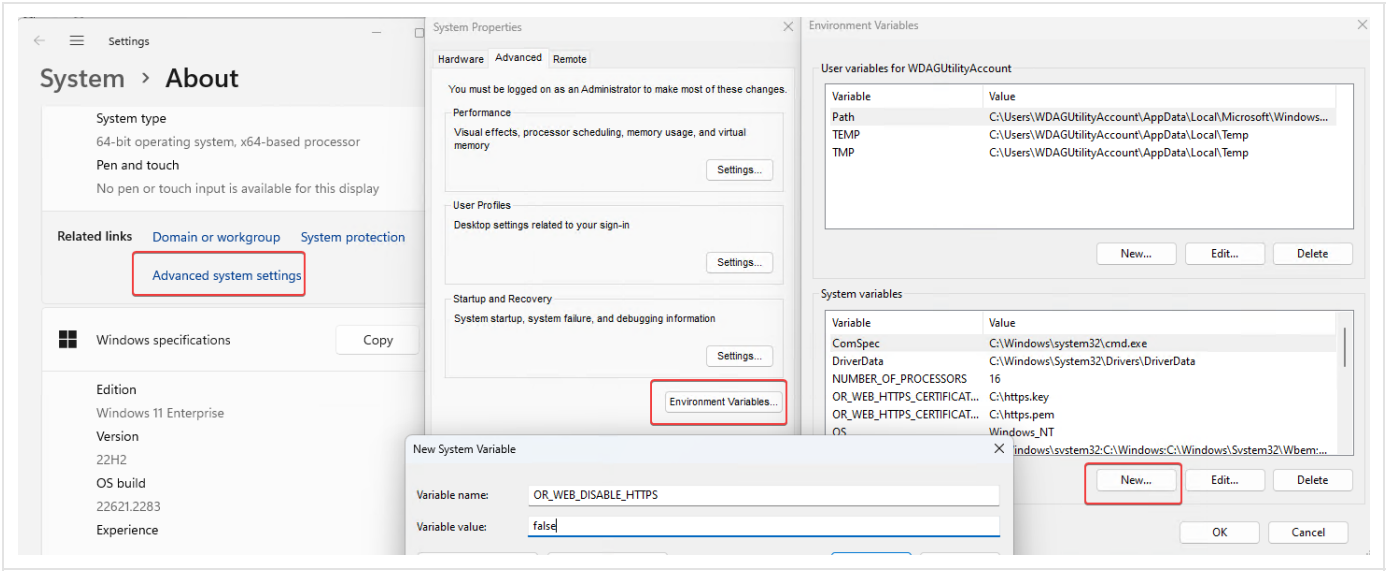
```
-- Specify your Database Name
USE DatabaseName
GO
-- Declare variables
SET NOCOUNT ON
DECLARE @tablename VARCHAR(128)
DECLARE @execstr VARCHAR(255)
DECLARE @objectid INT
DECLARE @indexid INT
DECLARE @frag decimal
DECLARE @maxfrag decimal
DECLARE @IdxName varchar(128)
DECLARE @ViewOnly bit
-- Set to 1 to view proposed actions, set to 0 to Execute
 proposed actions:
SET @ViewOnly=1
-- Decide on the maximum fragmentation to allow for.
SET @maxfrag = 30.0
-- Declare a cursor.
DECLARE tables CURSOR FOR
SELECT CAST(TABLE_SCHEMA AS VARCHAR(100))
+'.'+CAST(TABLE_NAME AS VARCHAR(100))
AS Table_Name
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'BASE TABLE'
-- Create the temporary table.
if exists (select name from tempdb.dbo.sysobjects where name like
 '#fraglist%')
drop table #fraglist
CREATE TABLE #fraglist (
ObjectName CHAR(255),
ObjectId INT,
IndexName CHAR(255),
IndexId INT
```

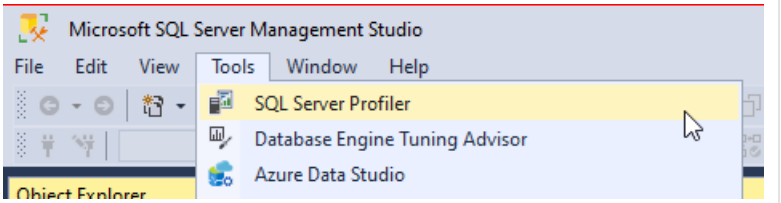
```
Indexid INT,
Lvl INT,
CountPages INT,
CountRows INT,
MinRecSize INT,
MaxRecSize INT,
AvgRecSize INT,
ForRecCount INT,
Extents INT,
ExtentSwitches INT,
AvgFreeBytes INT,
AvgPageDensity INT,
ScanDensity decimal,
BestCount INT,
ActualCount INT,
LogicalFrag decimal,
ExtentFrag decimal)
-- Open the cursor.
OPEN tables
-- Loop through all the tables in the database.
FETCH NEXT
FROM tables
INTO @tablename
WHILE @@FETCH_STATUS = 0
BEGIN
-- Do the showcontig of all indexes of the table
INSERT INTO #fraglist
EXEC ('DBCC SHOWCONTIG ('' + @tablename + '')
WITH FAST, TABLERESULTS, ALL_INDEXES, NO_INFOMSGS')
FETCH NEXT
FROM tables
INTO @tablename
END
-- Close and deallocate the cursor.
CLOSE tables
DEALLOCATE tables
-- Declare the cursor for the list of indexes to be defragged.
DECLARE indexes CURSOR FOR
SELECT ObjectName, ObjectId, IndexId, LogicalFrag, IndexName
FROM #fraglist
WHERE LogicalFrag >= @maxfrag
AND INDEXPROPERTY (ObjectId, IndexName, 'IndexDepth') > 0
-- Open the cursor.
OPEN indexes
-- Loop through the indexes.
FETCH NEXT
FROM indexes
INTO @tablename, @objectid, @indexid, @frag, @IdxName
WHILE @@FETCH_STATUS = 0
BEGIN
IF (@ViewOnly=1)
BEGIN
PRINT 'WOULD be executing ALTER INDEX ' + RTRIM(@IdxName)
 + ' ON ' + RTRIM(@tablename) + ' REORGANIZE WITH
 ( LOB_COMPACTION = ON ) - Fragmentation currently ' +
 RTRIM(CONVERT(VARCHAR(15),@frag)) + '%'
END
ELSE
BEGIN
PRINT 'Now executing ALTER INDEX ' + RTRIM(@IdxName) + ' ON ' +
 RTRIM(@tablename) + ' REORGANIZE WITH ( LOB_COMPACTION = ON ) -
 Fragmentation currently ' + RTRIM(CONVERT(VARCHAR(15),@frag)) +
 '%'
SELECT @execstr = 'ALTER INDEX ' + RTRIM(@IdxName) + ' ON ' +
 RTRIM(@tablename) + ' REORGANIZE WITH ( LOB_COMPACTION = ON )'
EXEC (@execstr)
```

```
END
FETCH NEXT
FROM indexes
INTO @tablename, @objectid, @indexid, @frag, @ldxName
END
-- Close and deallocate the cursor.
CLOSE indexes
DEALLOCATE indexes
-- Delete the temporary table.
DROP TABLE #fraglist
GO
```

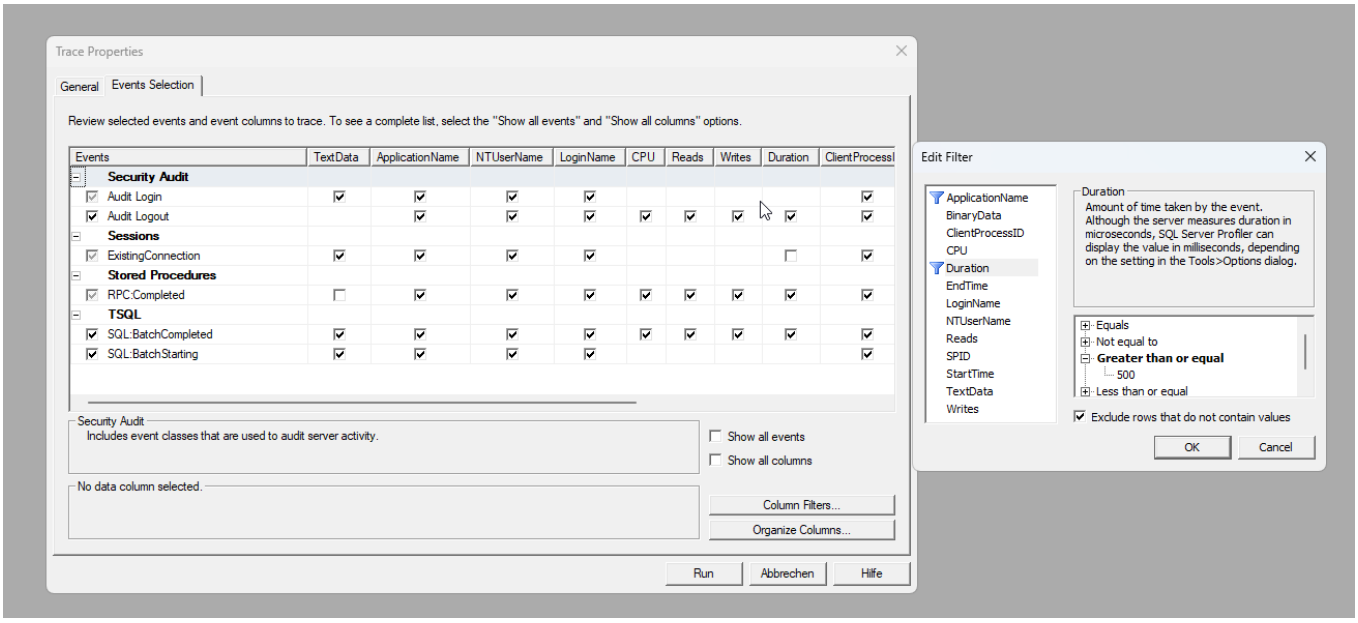
Microsoft SQL Server Performance-Analysis

Start SQL Server Profiler from the Management Studio:





Add a column filter, such as on the "Duration" column, to display all queries that take longer than 500 milliseconds:



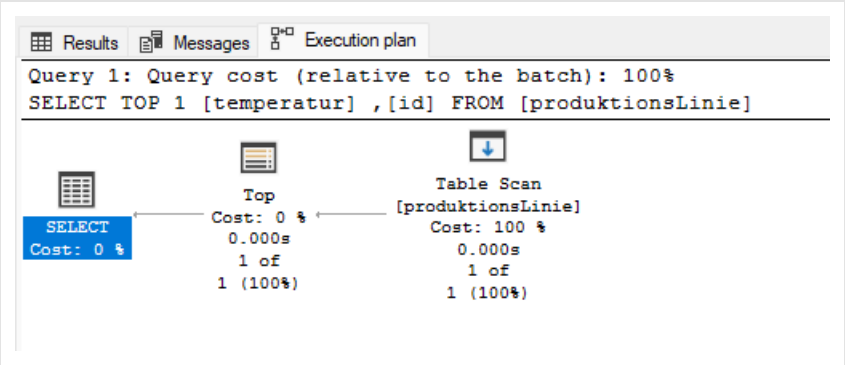
Select the query from the log and copy the query text:

Click the "New query" button in the Management Studio, paste the copied query text into the new query tab and activate the "Include execution plan in output" function:

After executing (1.), another tab "Execution Plan" (2) will appear at the bottom.

In the execution plan, you can see the individual operations of the SQL server and their utilization in detail. You may also receive information about missing indices or similar. This can help you to determine why the execution is taking so long.

If your query is an update or insert, you may also see the stored trigger activities here and, in the case of a select, the stored table joins of a view.



Procedures

Mit dem Microsoft SQL Server können sie sogenannte gespeicherte Prozeduren konfigurieren.

Diese Prozeduren, können Eingangs-Parameter haben, diese Parameter können auch als OUT Parameter festgelegt werden. Der OPC Router unterstützt beides, sodass Sie sowohl Werte übergeben als auch welche zurückbekommen können.

In der Prozedur kann in der Programmiersprache T-SQL komplexe Logik abgebildet werden.

Zusätzlich ermöglicht der OPC Router sogenannte Result-Sets abzufragen. Hierdurch wird das letzte Table-Select, das in der Prozedur ausgeführt wird, als Ergebnis bereitgestellt.

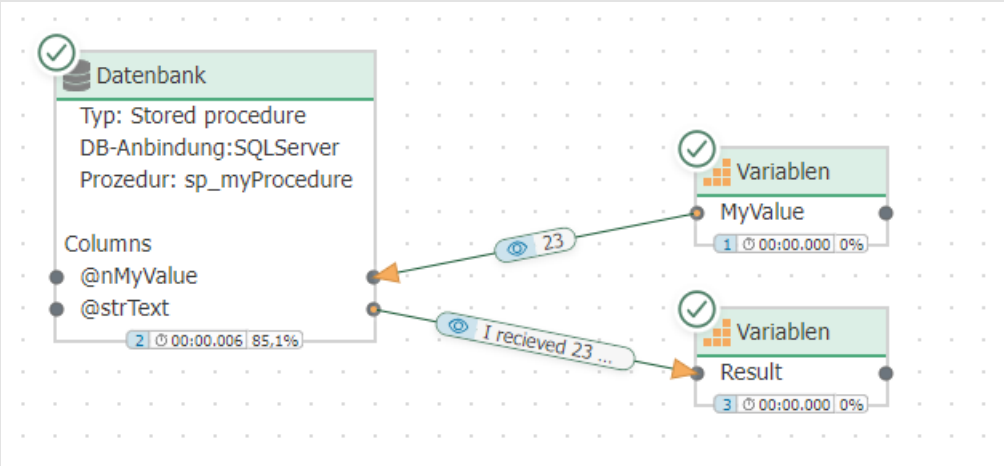
Please note that the OPC Router queries the result set once by invoking the procedure. This is done by the user confirming the "Retrieve Result Set" button in the OPC Router.

The procedure is executed within a transaction, which is subsequently rolled back. To determine the structure of the result set, the procedure is called within a transaction in which all parameters are set to NULL. The transaction is then aborted.

If the parameters have the "Allow Null" option, the procedure is called with NULL values. Otherwise, the procedure is called with default data types, such as "0" for numeric parameters or an empty string for character strings.

Create a procedure

```
CREATE PROCEDURE sp_myProcedure
   @nMyValue INT,
   @strText NVARCHAR(50) OUT
AS
BEGIN
   SET @strText = 'I recieved ' + CAST(@nMyValue as nvarchar)  + ' as value...';
END
GO
```



import Tabs from '@theme/Tabs'; import TabItem from '@theme/TabItem';

Oracle Plug-in

UsernameThe Oracle database is a powerful database for business applications. The OPC Router supports the native connection via the corresponding Oracle client drivers and TNS. You can find a detailed video on this topic under “Connection of database systems” in our tutorial stream.

The following properties can be configured in the Oracle plug-in:

“Connection settings” tab

Name	Name under which the Oracle database server appears in the connection configuration. Select a name that is unique for you during connection configuration.

Connection type

There are three different connection types available. Various properties can be configured depending on the connection type selected.

| | | | | -------- | ---------- | | Host/IP | Host name or IP address of the server. | | Port | Port under which the Oracle database service can be reached. | | Database | The database to which you want to connect.
Note: This property must be set for a successful connection test. | | Schema | The Oracle schema to be used can be selected here. | | | | -------- | ---------- | | TNS name | Name of the TNS configuration to be used. | | Port | Port under which the Oracle database service can be reached. | | Schema | The schema to be used | Only one cloud wallet can be configured. As soon as a Cloud Wallet is used, no other Oracle plug-in can be used | | | | ---------- | ---------- | | Cloud Wallet | Wallet file that contains the access data of the database to which a connection is to be established. | | Database | Database to which a connection is to be established |

Windows authentication	If this option is active, Windows authentication is used instead of the user and password properties. Local user accounts, DCOM users and Windows domain users (NT authentication) are supported. To use Windows authentication, the user under which the OPC Router service is started (by default “System”) must have access rights to the database. To be able to configure, the user under which the configuration environment was started must have access rights to the database.
User	User name for logging in at the Oracle database server
Password	Password for logging in at the Oracle database server

“Advanced connection parameters” tab

Command Time-out	Time in seconds that the router waits for the completion of an SQL command. It may be necessary to increase the value for long-lasting queries. Please note that the connection timeout must also be increased for connections in which long queries are being executed.
Connection Time-out	Time in seconds that a connection to the database may take.

Concurrent connections	Number of concurrent connections that the router may establish to the database. The more simultaneous connections are allowed, the faster the router is, but this can slow down the database. Ask your database administrator how many connections they allow the router. Of course, this also depends on the size of the project.
Polling interval	Interval in seconds at which the database is polled.

“Advanced” tab

Bracket behavior	This property determines when and how square brackets are used in your Oracle plugin. A detailed explanation of the individual options can be found here.
Convert Boolean values	If this option is active, the values “1”, “y”, “t” are converted to true and “0”, “n”, “f” to false values.
Oracle append schema	If this option is active, the schema is added to the identifier. A brief explanation can be found here.
Oracle column data type source	This property determines how the data types of the columns are to be handled. A more detailed explanation can be found here.
Oracle home	Directory in which the Oracle Client software was installed.
Include Oracle packages	If this option is active, stored procedures that are located in packages are also listed.

Bracket behavior

- **Put in quotation marks if dot is present**: This mode is intended for backward compatibility. If a dot (.) is contained in an identifier, quotation marks are added. This is useful if you have database objects with dots in their names, such as `datenbank.schema."tabelle.name"`.
- **Never put in quotation marks**: In this mode, no double quotes are added, regardless of the identifier. This can be useful if you are sure that your identifiers do not contain special characters or spaces.
- **Always use double quotes**: In this mode, double quotes are always added. This is useful if your identifiers may contain special characters, spaces, or reserved words.

Please note that in Oracle, the use of double quotes is used to escape identifiers that would otherwise be interpreted as reserved words. It also allows the use of special characters and spaces in identifiers.

Oracle append schema

- **Legacy:** Same behavior as in previous versions.
- **Yes:** The schema is always being added to the identifier. This is useful if you are working with multiple schemas in the same database and want to ensure that the correct tables and views are used.
- **No**: The schema is not being added to the identifier. This can be useful if you are only working with one schema or if you want to have full control over the SQL statements generated by the plugin.

Oracle column data type source

- **Auto**: The data types are converted to the most appropriate data type when reading. This can simplify the transfer of data to another system.
- **Data Reader**: The data types are retained when reading and are not being converted. This setting is helpful if it is important to retain the exact data types.

SAP HANA

Connection options für SAP Hana

Name	Any name under which this database server appears in the connection configuration. Choose a name that is unique for you during connection configuration.
Host/IP/FIle	Address of the SAP Hana Server
Port	Selection of the Port. He can be changed.
Data Source	Name of the database to use
Schema	Schema
Username	Username to log in to SAP Hana server
Password	Password to log on to SAP Hana server
Command time-out	Time (seconds) that the router waits for a command to complete. For long queries, it may be necessary to increase the value. Note that you also need to increase the value for the individual connections that initiate long queries.
Connection time-out	Time (seconds) that the Router may need to establish connection.
Parallel connections	Number of connections that the Router may establish parallely. The more connections the faster the Router the slower probably the database. Ask your database administrator how many connections the Router may establish. This depends on your project’s scale, of course.
Polling interval	Distance at which the database is queried.

ODBC

Options for ODBC drivers

The ODBC driver needed must be installed and active.

Name	Name of the configuration
Data Source	Name of the database to use
User	Username to log on database
Password	Password to log on database

OLE DB

Options for OLE DB Providers

Attention: OLE DB is not supported in Docker.

OLE DB connections need a connection string. The OLE DB provider needed has to be installed and must be active.

Name	Name of the configuration
Connection string	Select a connection string from the drop-down menu.
Command time-out	Time (seconds) that the Router waits for an SQL command to terminate. Increase if your projects involve commands of long duration (e. g. stored procedures). Note that you also need to increase the value for the individual connections that trigger long queries.
Command time-out	Time (seconds) that the router waits for a command to complete. For long queries, it may be necessary to increase the value. Note that you also need to increase the value for the individual connections that initiate long queries.
Connection time-out	Time (seconds) that the Router may need to establish connection.
Parallel connections	Number of connections that the Router may establish parallely. The more connections the faster the Router the slower probably the database. Ask your database administrator how many connections the Router may establish. This depends on your project’s scale, of course.
Polling interval	Distance at which the database is queried.

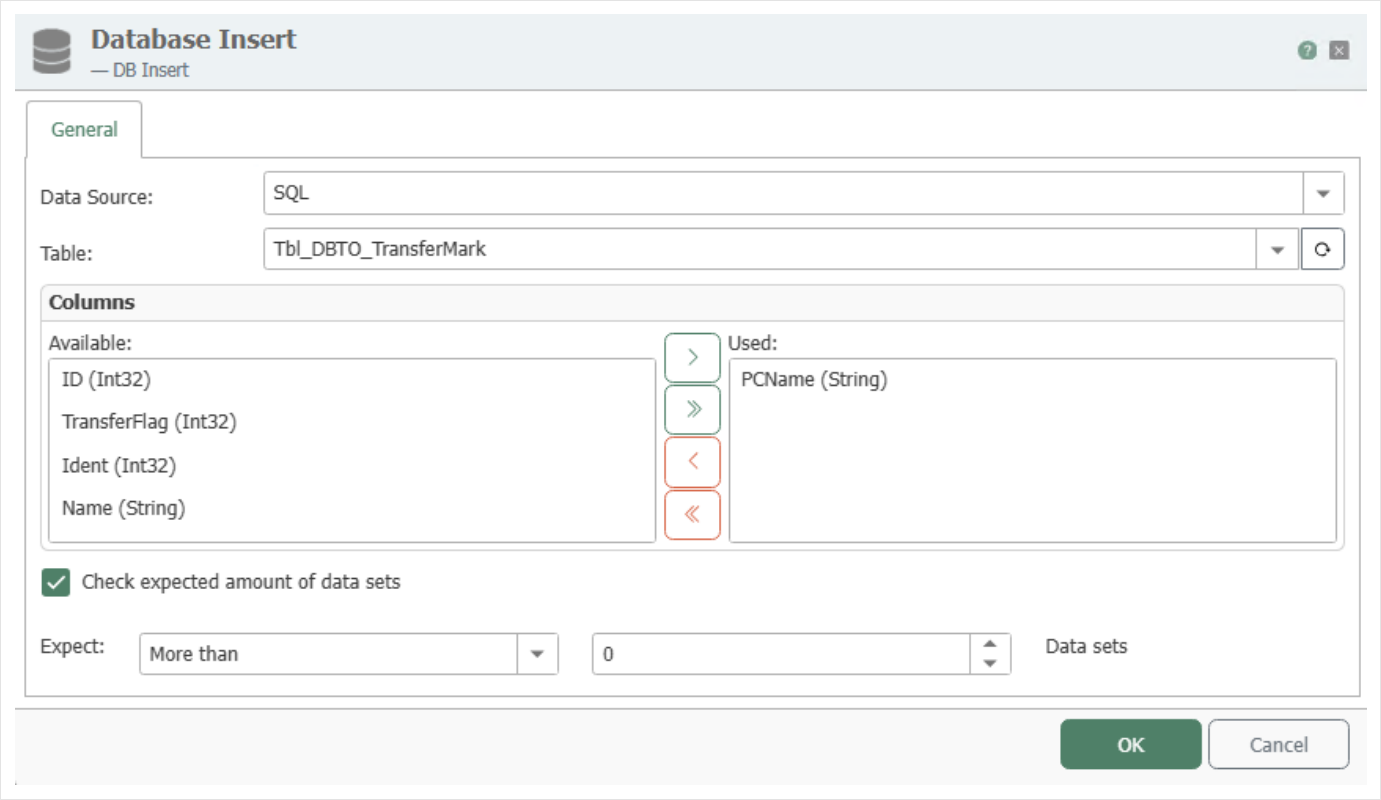
MySQL

Options for MySQL/MySQL-5 Servers

Name	Name of the configuration
Host/IP	IP address of MySQL server.
Port	Selection of the port for the database connection. If necessary, the port can be changed.
Data Source	Database on server to be used.
Windows authentication	Activate to use Windows authentication instead of username and Password. Local accounts, DCOM users a Windows domain users (NT authentication) are supported. To use Windows authentication, the user under which the OPC Router service is running needs access rights to the database server. By default, this is the system user. To be able to define connection, the user under which OPC Router Configuration is running, needs those rights (by default: User logged-on the Windows System). In this case, “Check connection” is valid only for the configuration. Observer carefully if transfers are executed correctly!
User	Username to log on MySQL
Password	Password to log on MySQL
Command time-out	Time (seconds) that the router waits for a command to complete. For long queries, it may be necessary to increase the value. Note that you also need to increase the value for the individual connections that initiate long queries.
Connection time-out	Time (seconds) that the Router may need to establish connection.
Parallel connections	Number of connections that the Router may establish parallely. The more connections the faster the Router the slower probably the database. Ask your database administrator how many connections the Router may establish. This depends on your project’s scale, of course.
Polling interval	Distance at which the database is queried.

Database Transfer Object

Databases are available both as source and destination of data transfers. The database set of access data has to be specified within plug-in configuration. Create a DB transfer object as described above.



Database Transfer Object

Configure the object's properties by double click:

DB access data	Select one of the configured sets of access data
Type	**Select:** Queries one or more rows from a table or a view. You have to define filter and sorting criteria as well. Thus, the transfer object will work as datasource. **Insert:** Inserts a new row into a table. Thus, the transfer object will work as destination. For a database insert, incomplete arrays are padded. For example: If a single entry from a timestamp and several entries from an array should be read into a database, so the time stamp for the individual variables of the array can be done. The timestamp is therefore automatically used multiple times, until all variables have been read. Therefore, applies: An insert always determines the highest InsertCount. Entrys with smaller count are then expanded by a factor of X, so that the values in the table are filled up. Also selectable for views. **Update:** Updates one or several rows within a table. You have to define filter criteria as well. Thus, the transfer object will work as destination. Also selectable for views. **Delete:** Deletes one or several rows within a table. You have to define filter criteria as well. Also selectable for views. **Stored procedure:** Executes a stored procedure as soon as trigger conditions are met. Define parameter values (as destination) and use return values as source.

Important: If the type is set to "Stored Procedure", OPC Router will execute this Procedure immediately, submitting default parameters (string = NULL, int = 0 etc.). Thus, OPC Router detects the structure of the result set. Procedure will be rolled back. This means, that the Stored Procedure called must be transaction safe and must not return errors on default parameters.

Tab General (all types)

Table (View, Stored Procedure)	Table, View or Stored Procedure this object refers to. **Note:** Table has to have a primary key. Tables without primary key are not supported.
Available columns/Columns used	(View/Table operations) **Available columns:** Columns available in table or view, but not assigned to this object yet. **Columns used:** Columns assigned to this object and thus shown as items within this objects (to be used as source or destination). This distinction is made to keep the layout well-arranged, e. g. imagine you only need one out of 30 columns. So only this one column is shown in the connection layout. Move columns by double click onto their names or by the buttons provided (after having selected the columns). Buttons “Up” and “Down” will sort the columns (again, this is to keep the layout well-arranged) **Stored Procedure:** The lists show input and return parameters.

Tab Filter (Select, Update, Delete)

Filter	Shows filters and filter groups. From this tree, the SQL statement is constructed. Filter values can be inserted dynamically during runtime. Use Variables/Constant values to insert constant filter criteria. Use Add-buttons to add filters or groups. Delete will remove the selected entry. Filter types: EQUAL: Compares if values are equal. GREATER: Compares if value is greater. GREATER_OR_EQUAL: Compares if value is greater or equal. SMALER: Compares if value is smaller. SMALER_OR_EQUAL: Compares if value is smaller or equal. NOT_EQUAL: Compares if values are not equal. LIKE: Compares strings. LIKE_WILDCARD: Equal to LIKE %WERT%. LIKE_WILDCARD_END: Equal to LIKE WERT%. IN: Compares multiple values. Value input is an array. NOT_LIKE: Compares if strings are not equal. NOT_LIKE_WILDCARD: Equal to LIKE_WILDCARD but negated. NOT_IN: Compares if multiple values are not equal. Value input is an array.

Edit filter/filter group	Entries are shown dependent on selected entry. **Junction:** Defines whether filters are linked with AND or OR. **Column:** Column to be filtered **Relational operator** to compare values
Preview	Preview of SQL Statement, e. g. WHERE (UtcTimeStamp >_ ?) AND (Value = ?) . Here, all rows would be selected with a TimeStamp newer than the filter and with a value equal to the filter. The values of the filter criteria have to be inserted during runtime. Thus, you can provide e. g. values of OPC items or other tables as criteria. If the filter should be static, simply project a constant value object and transfer its value into the filter.

After confirming with “OK”, you will see filter criteria as elements (items) of this transfer objects. Draw connection arrows from any source to assign values during runtime. With time filters, relative filter criteria like “-1d” (=yesterday) are allowed.

Tab Query (Select)

Sorting order	Ascending/Descending
Sorting by column	Column with sorting criteria.
Do not change records	Transferred records will not be changed.
Mark records after successful transfer	(only with select from table) In the column specified below, for every row transferred successfully, a “1” is inserted, a “2” if an error occurred. **Note:** In the table concerned, a simple primary key to the numerical column "ID" is required.
Column to mark	(only with select from table) Column to mark transferred rows. Default value set by database should be “0”
Count failed transfers	count the failed transfers **Note:** Select at “Error column” the column in which the number of failed transfers is to be counted up.
Delete records after successful transfer	only with select from table) Transferred rows will be deleted. **CAUTION:** Rows in database table will be deleted without recover possibility.

Abort transfer (with empty result)	If the result set is empty, the transfer is canceled.
Return read error (with empty result)	If the result set is empty, the transfer is canceled and an error is logged in log file and status monitor.
Return empty record	Returns an empty record for an empty query result.
Limit number of records	Transfers only the specified number of rows/records. Be sure to have you result set sorted correctly.

Example: If only the newest record is to be transferred, then sorting column should be e. g. a “UtcTimeStamp” with Sorting order “Descending”, the number of records limited to “1”.

Note: If the type is set to “Update”, the transfer object will provide the number of rows updated as source.

Check expected amount of data sets

Activate this option to check the correct size of the data to be used for the insert. If you receive an array of data, for example, you can define here the minimum, maximum or exact number of data sets the array must have for the insert to be carried out.

Select always with filter

Use SQLSelect only with filters (WHERE clause) so as not to load the whole table into memory. In conjunction with transfer marks, you filter, for example, after `<Transferflg> <> 1`

Mark transferred records

Mark transferred records. In your tables, keep columns in which the OPC Router can mark transferred data records (“Transfer flag” not transferred, transferred, transfer error). The transfer flag columns should be set to the default value of “0” and should not be allowed to assume NULL, so that a correct assignment to the three status are always guaranteed.

Index transfer flag

To improve performance, the transfer flag column should be indexed as follows:

- If the OPC Router should filter by `transfer flag = 0`:

```
CREATE NONCLUSTERED INDEX [IX_TransferTable_Transferflag] ON [dbo].[TransferTable]
(
  [Transferflag] ASC
)
WHERE ([Transferflag]=(0))
GO
```

- If the OPC Router should filter by `transfer flag <>1`:

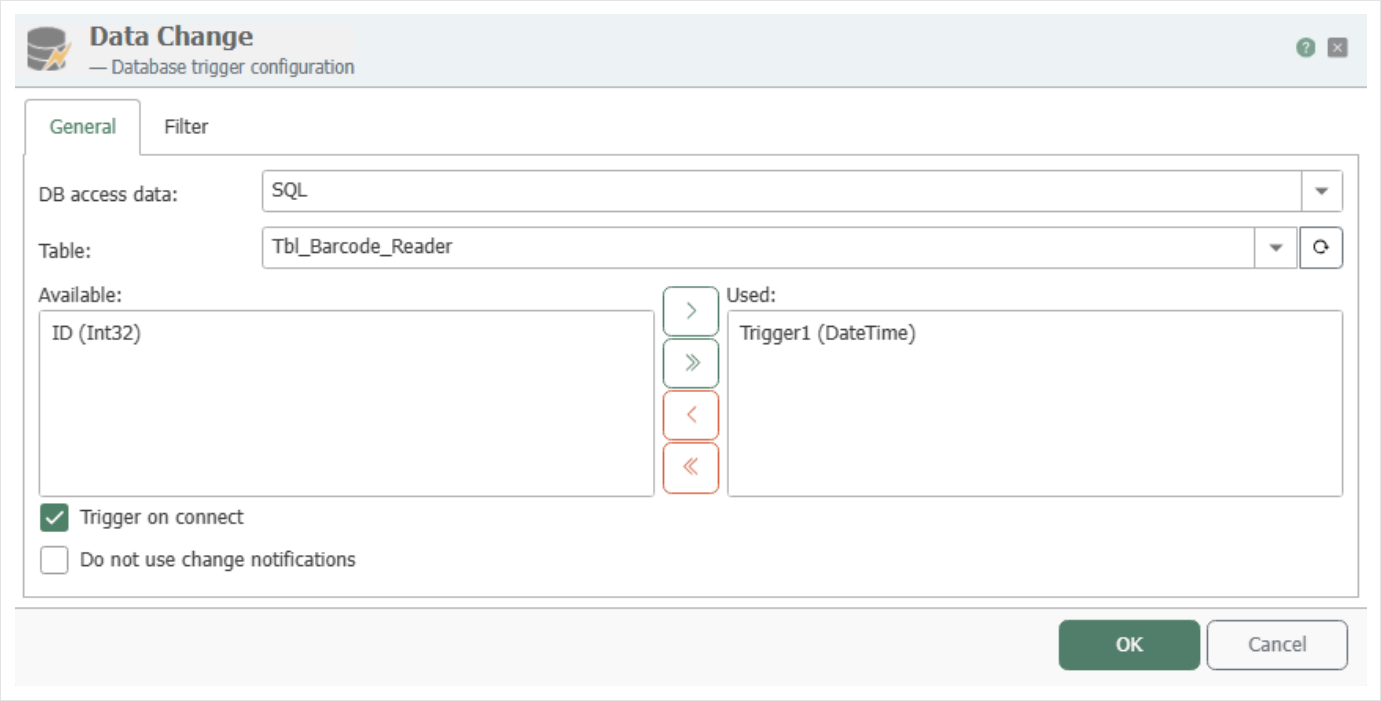
```
CREATE NONCLUSTERED INDEX [IX_TransferTable_Transferflag] ON [dbo].[TransferTable]
(
  [Transferflag] ASC
)
WHERE ([Transferflag]<>(1))
GO
```

Sort select results

Sort the table in ascending order after the transfer flag. This way, the untransferred records are transferred first; erroneous individual transfers are only attempted again at the end of the transfer. For performance optimization, filtering can be done after the transfer flag column (transfer flag = 0 or = 2). A correspondingly filtered index can be created in the database for this purpose.

Database trigger

The database trigger monitors query results and triggers the transfer on change. The trigger defines a select query. If the result of this query changes, the trigger triggers a transfer.



Database Trigger

The individual properties are:

DB-Connection	The configured instances of the DB plug-in are available here.

Tab General (all types)

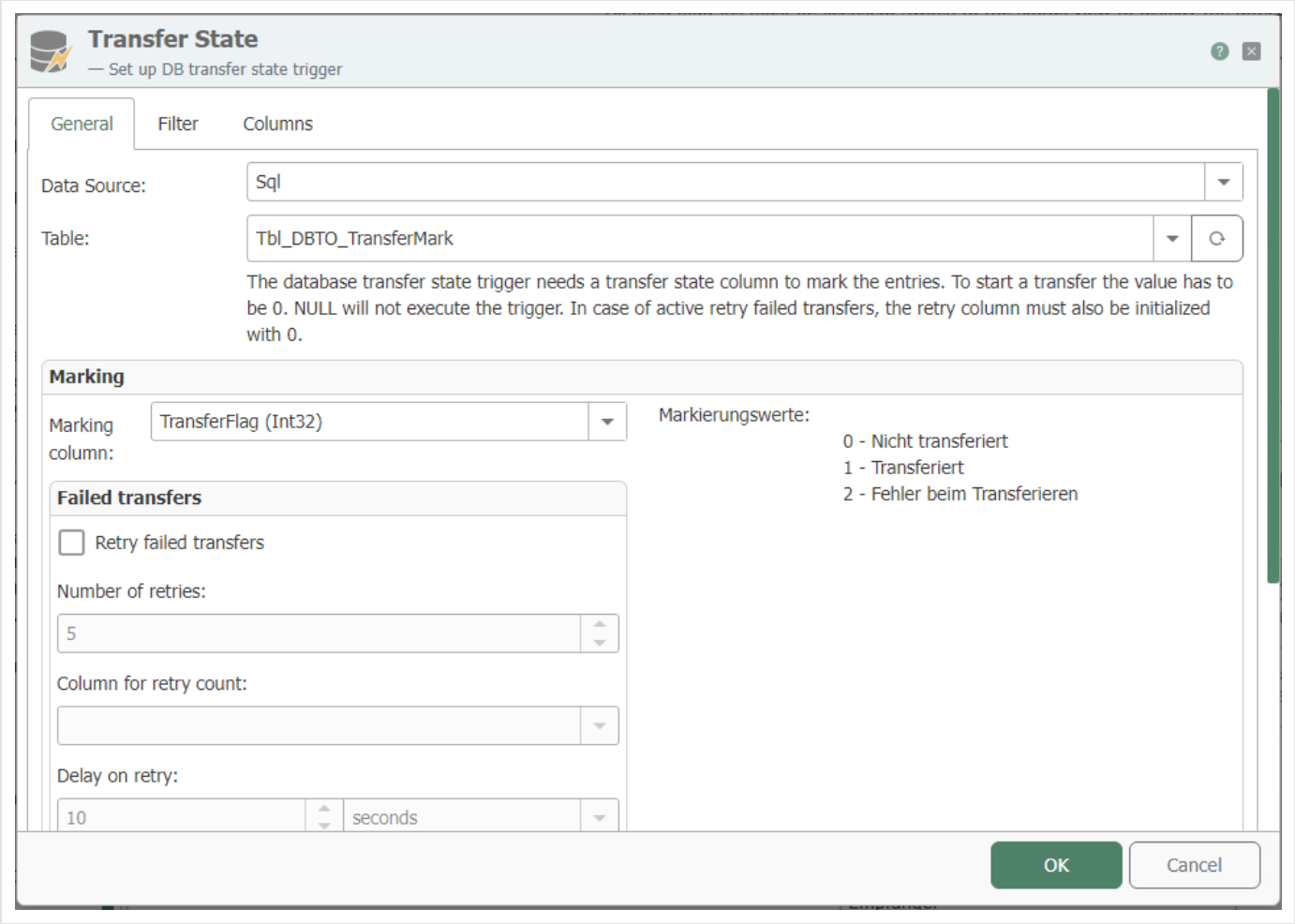
Table	Table being monitored. **Note:** The specified table must have a primary key. Tables without a primary key are not supported.
Available columns	All available columns that have not yet been selected for the panel “Used fields”. Select the columns to monitor.
Used fields	All already selected fields. Use the arrows between “Available columns” and “Used fields” to add or remove fields.
Triggern at OPC Router start	At router start it will be trigger.

Tab: Filter (Select, Update, Delete)

Filter	Displays filters and filter groups graphically. The SQL statement is constructed from this tree at runtime. You must enter the values here, unlike the DB transfer object. Use the buttons to add filter criteria (“Add filter”) and grouped filter criteria (“Add filter group”) to individual filter criteria. “Delete” deletes the selected criterion. **Important:** The filter set here must clearly specify a data record in the “table”, which is then monitored.
Edit filter	The possible entries change depending on which element is selected at “Filter”. **Column:** Column to filter on. **Relational operator** for the comparison of values. **Value**, after being filtered.
Preview	Preview the SQL statement.

Transfer State Trigger

The transfer status trigger initiates connections as soon as a record created in the connected database is marked as ready for transfer via a flag. After the transfer is complete, the data set is marked as successful or failed. Failed transfers can optionally be repeated as often as desired via an additional counter.



transfer state trigger

The transfer status trigger requires a primary key in the monitored table.

Composite primary keys are not supported.

Tab: General

Column to mark	Select the transfer column to be monitored in the table. The transfer status trigger triggers connections only if the value in the transfer column is 0. NULL values are ignored.	
Retry failed transfers	Retry the execution after an error, please activate this checkbox.	
Number of retries	Specifies the number of retries to start a new transfer attempt in case of failed transfers.	
Retry count column	Select a column in the table where the retries can be incremented. Select a column with an integer data type. **Attention**: The Transfer Status Trigger triggers the connection only if there is a value < "Number of retries" as well as not NULL in the field of the retry count column.	

Retry delay	Time period that is waited until a new transfer attempt is started.	

Tab: Filter

Filter	Graphically represents filters and filter groups. The SQL statement is constructed from this tree at runtime. Use the buttons to add single filter criteria ("Filter") and grouped filter criteria ("Filter group"). "Delete" deletes the selected criterion. Filter types: EQUAL: Compares whether values are equal. GREATER: Compares whether the entered value is greater. GREATER_OR_EQUAL: Compares whether the entered value is greater or equal. SMALER: Compares whether the entered value is less. SMALER_OR_EQUAL: Compares whether the entered value is less than or equal to. NOT_EQUAL: Compares whether the entered value is not equal. LIKE: Compares strings. LIKE_WILDCARD: Corresponds to LIKE %VALUE%. LIKE_WILDCARD_END: Corresponds to LIKE VALUE%. IN: Compares with multiple values. The values must be passed as an array. NOT_LIKE: Compares if strings are different. NOT_LIKE_WILDCARD: Equivalent to LIKE_WILDCARD but negated. NOT_IN: Compares if multiple values are not equal. The values must be passed as an array.
Edit filter	The possible entries change depending on which element is selected under "Filter". Link: Sets the AND and OR operators between the filters/filter groups. Column: Column to be filtered by Comparison operator for the value comparison
Preview	Preview the SQL statement, e.g. WHERE (TIMESTAMP >_ ?) AND (VALUE = ?) . In the example, all records would be selected whose Timestamp field is greater than a timestamp to be set (= newer than date/time) and whose Value field is equal to a value to be set. The values for the filters can be defined dynamically at runtime (e.g. values from OPC items or other tables) or you can define a variable already at design time (which again could be changed at runtime).

Tab: Output columns

Available columns	Columns (data fields) that are displayed as elements (items) of the trigger and are thus available for transfers. This distinction primarily serves the overview in the project planning area - e.g. if only one column of 30 existing ones is required for the connection. Move the columns by double-clicking on the column name or via the "<" and ">" button (after selecting the column). "<<" and ">>" move all columns. Use the "Up" and "Down" buttons to sort the columns used. This also serves the clarity in the display.

Examples and concepts

Database to other systems

When you want to securely transfer data, it is a common practice to do so using a handshake procedure. For databases like Microsoft SQL Server, the "Mark Transferred Values" function can be used for this purpose. In this process, the transfer status is recorded in a column of the table. The OPC Router defines the following states:

0 Not Transferred

1 Transferred

2 Transfer Error

It is advisable to configure the transfer column in the underlying table to disallow null values and accept "0" as the default value.

Certainly, the value doesn't have to be set as a default "0" (transfer pending); it can also be set as a default "1" (no transfer pending). In this case, it can be updated to "0" later in the process, for example, by an action initiated by a user when a transfer is intended to take place.

This typically ensures that applications to which this column is added will continue to run smoothly. When the SQL Server performs an insert, it can automatically populate this column with a value. As a result, the OPC Router is aware that this record has not been transferred yet and will transmit it in the next cycle.

You can add multiple transfer status columns to a table if you want to use the same record in different process steps. This can be helpful, for example, when you want to print a record at different stations or for similar scenarios.

Repetition in case of an error

The OPC Router supports the possibility of retrying in case of an error using the Select Transfer object and the Transfer State Trigger, for example, when the target system is not reachable. To implement this, you can use an additional column to keep track of the error count. When the error count exceeds a specified threshold, the transfer is automatically canceled, and the record is marked as final-error. Additionally, a time delay (delay) for retries in case of errors can be configured.

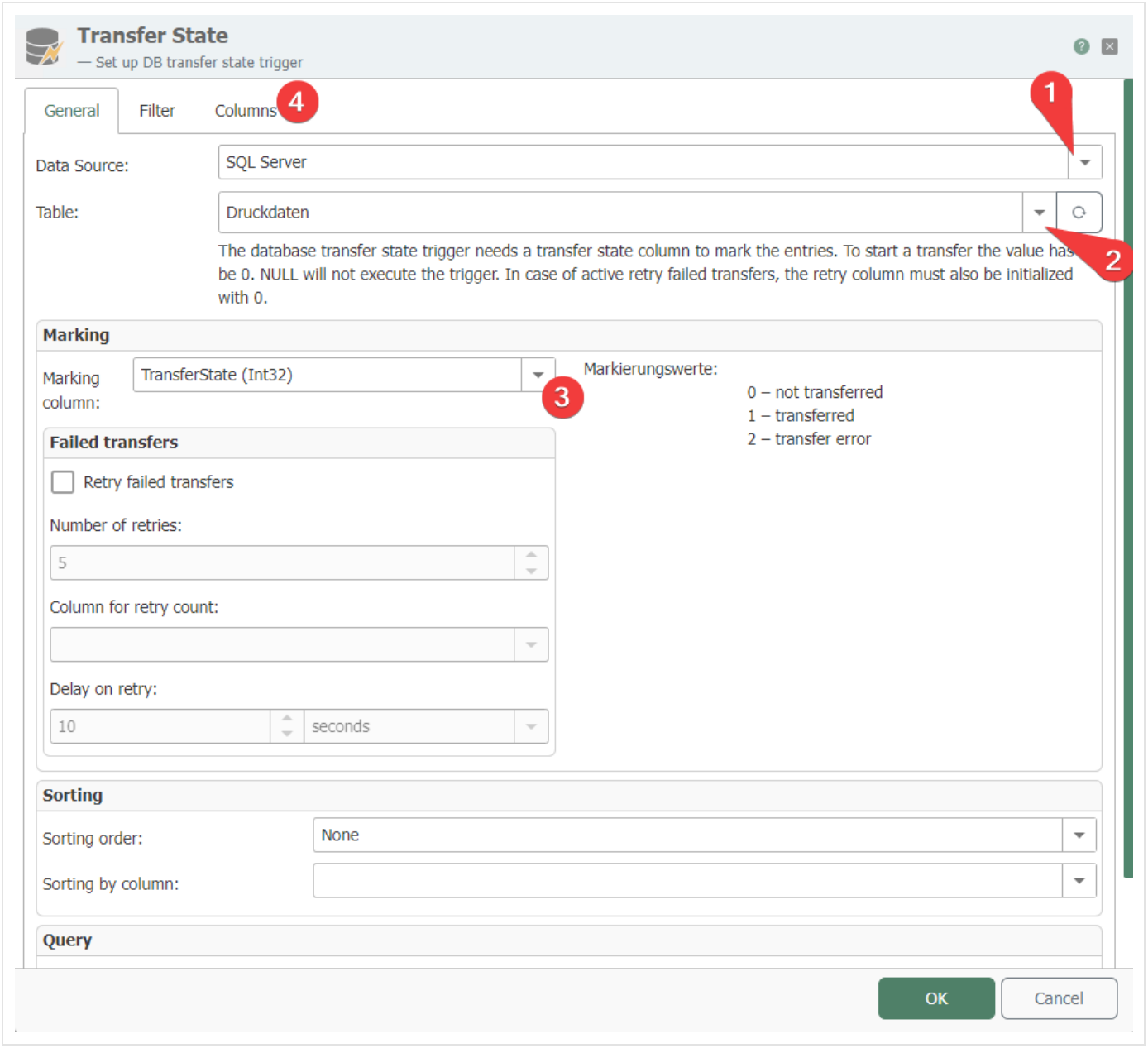
It's also possible to enable an alerting mechanism for erroneous transfer states by using an additional OPC Router connection.

Step-by-Step Configuration of a Transfer Column Handshake

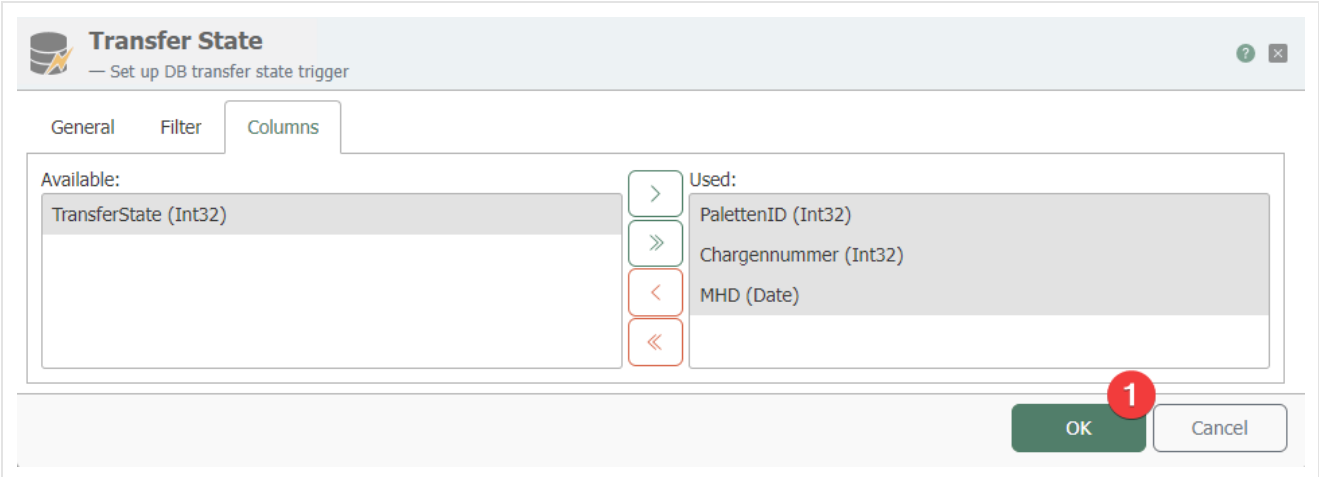
1. Add an additional 'TransferState' column to your table and set the default value to 0 (2), disallowing null values (1).\

```
<figure><img src="E:/Doku/5.5finaleEN/articles/assets/Database_to_other_systems_image (3) (1).png" alt=""
/><figcaption></figcaption></figure>
```

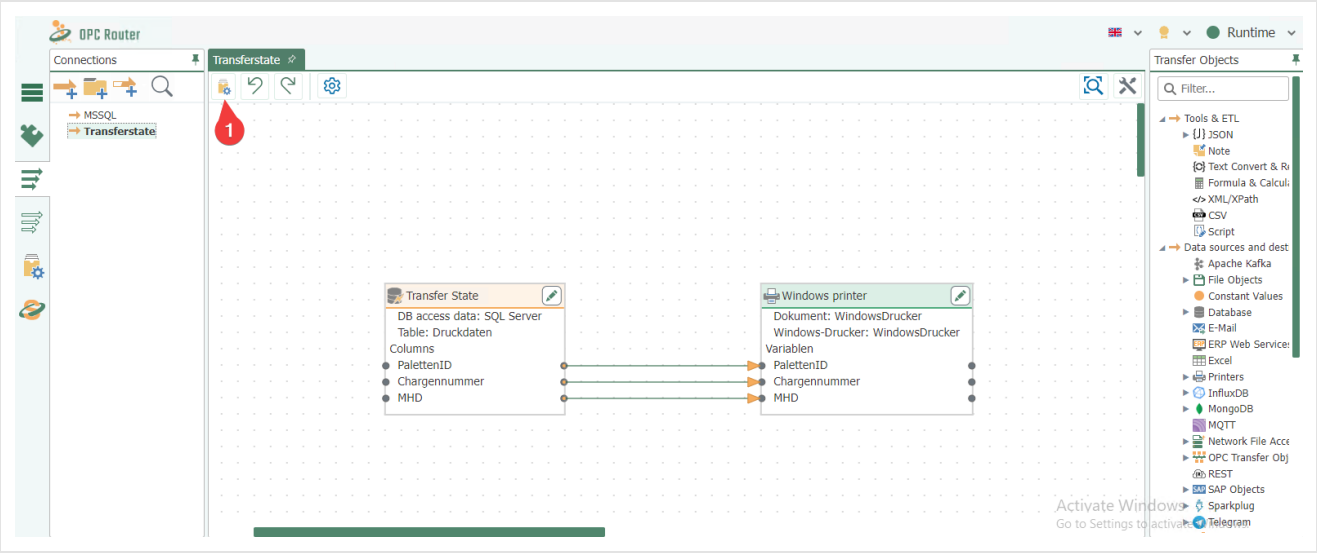
1. Add the Transfer State Trigger to your OPC Router connection and open it by double-clicking for editing. Select your database (1) and your table (2). Then, choose your recently created 'TransferState' column for marking (3) and switch to the 'Columns' tab (4).



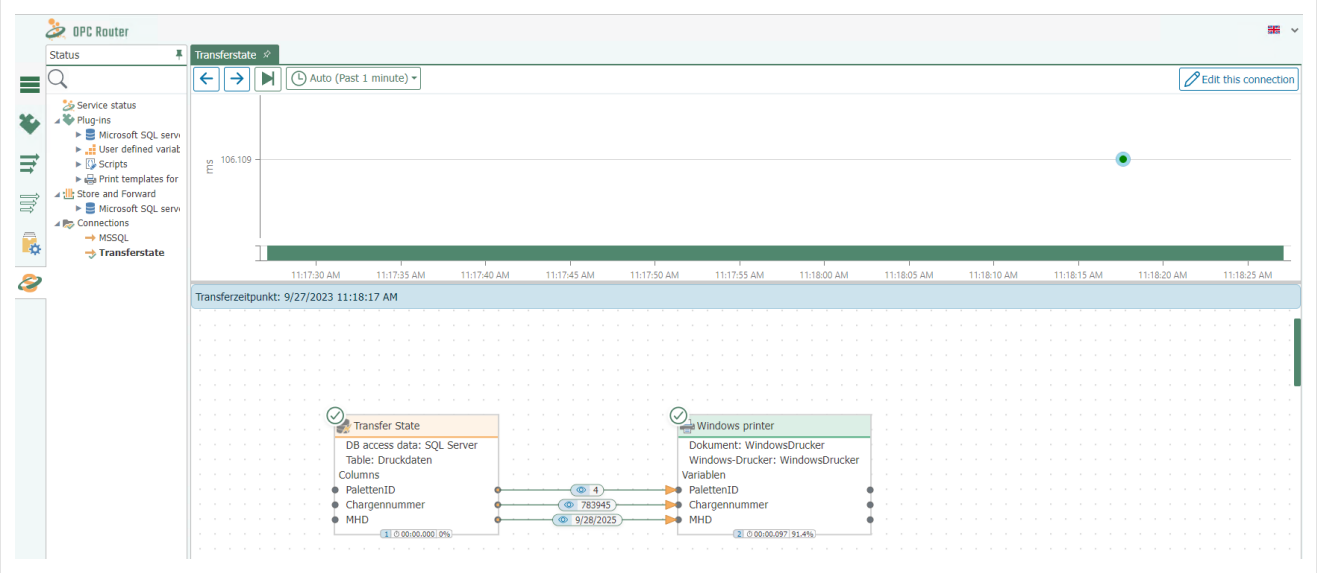
1. Select the columns you want to use and click OK to save the trigger with the settings (1).



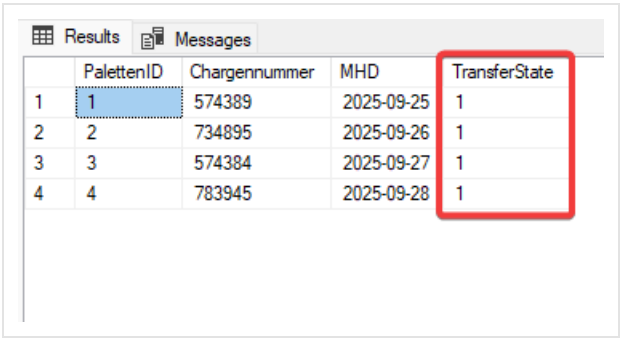
1. As an example, we send our print data to our Windows printer and activate production (1).



1. When we write data to the table, and the Transfer State is set to 0, the connection is triggered, and the data is successfully sent to our printer.



1. We can now also see this in the Microsoft Management Studio because our data has been marked with a Transfer State of 1, meaning they have been successfully transferred.



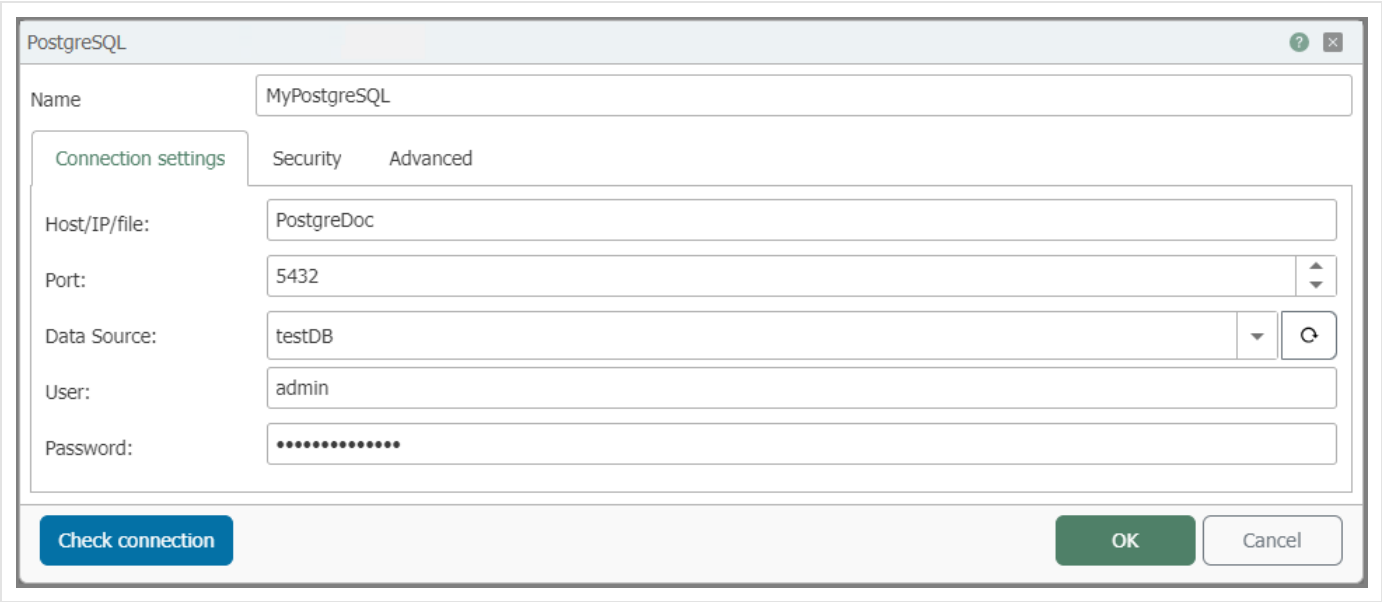
Results | Messages

	PalettenID	Chargennummer	MHD	TransferState
1	1	574389	2025-09-25	1
2	2	734895	2025-09-26	1
3	3	574384	2025-09-27	1
4	4	783945	2025-09-28	1

PostgreSQL Plug-in

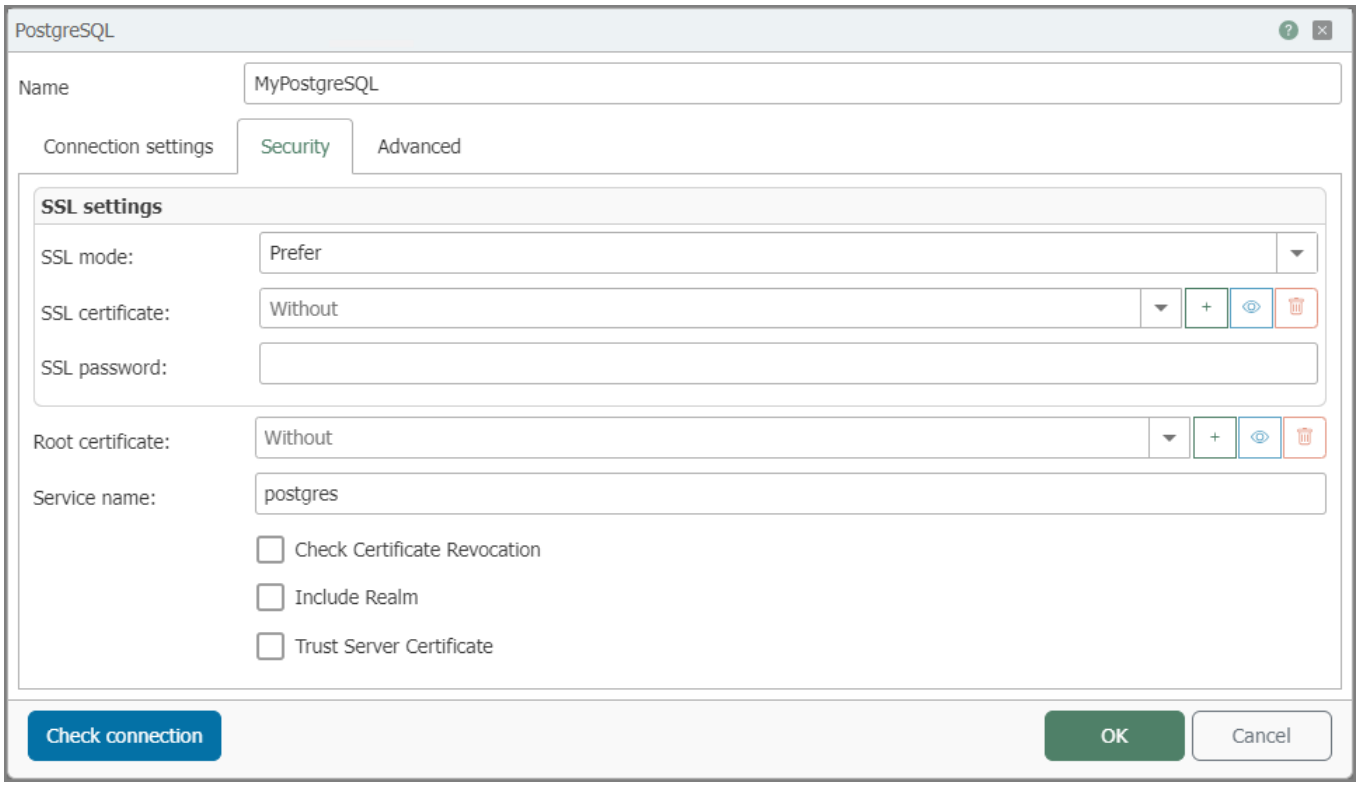
The PostgreSQL plug-in does not support custom data types or JSON as a data type.

Connection settings:



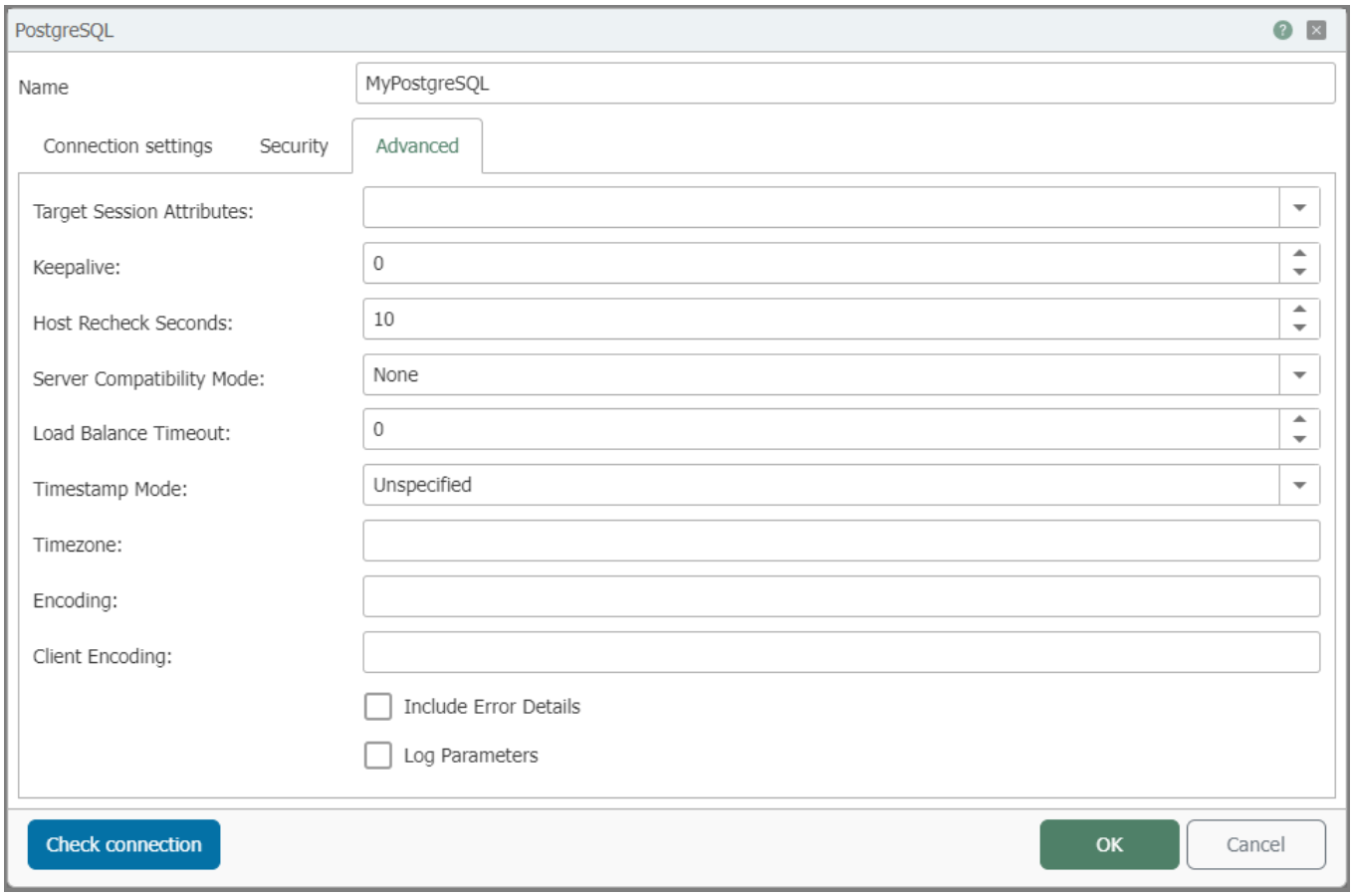
NAME	ANY NAME UNDER WHICH THIS DATABASE SERVER APPEARS IN THE CONNECTION SETUP. CHOOSE A NAME THAT IS UNIQUE TO YOU DURING THE CONNECTION SETUP.
Host/IP/file	Address of the PostgreSQL server.
Port	Selection of the port for database connection. If necessary, the port can be changed.
Data Source	Name of the database to be used.
User	Username for logging into the PostgreSQL server.
Password	Password for logging into the PostgreSQL server.

Security:



SSL MODE	PROPERTY DETERMINING HOW THE SSL CONNECTION IS MANAGEDHERE FUNCTIONALITY IS EXPLAINED.
SSL certificate	Selection of an SSL client certificate from the drop-down list, which the OPC Router sends along with.
SSL password	Password to access the private key of the client certificate sent during the SSL handshake to the server.
Root certificate	Selection of a root certificate (CA) from the drop-down list, which the OPC Router sends along with.
Service name	Kerberos service name of the PostgreSQL server.
Check Certificate Revocation	Enable if the certificate revocation list should be checked during authentication. (Default is False).
Include Realm	Enable if Kerberos realm should be used for authentication.
Trust Server Certificate	Enable if the server certificate is to be trusted without validation.

Advanced:



TARGET SESSION ATTRIBUTES	**DETERMINES THE PREFERRED POSTGRESQL TARGET SERVER TYPE.**
Keepalive	The interval in seconds at which consecutive keepalive packets are sent when no acknowledgment is received.
Host Recheck Seconds	Specifies how long the host's cache state is considered valid.
Server Compatibility Mode	Property describing the server compatibility mode. Here functionality is explained.
Load Balance Timeout	The maximum overall lifespan of connections (in seconds). Connections exceeding this value are destroyed instead of being returned to the pool. Useful in cluster configurations to force load balancing between a running server and one that has just been brought online.
Timestamp Mode	Property defining how the timestamp is treated. Here functionality is explained.
Timezone	The timezone of the PostgreSQL session in Olson/IANA database format.
Encoding	Specifies the encoding for encoding/decoding PostgreSQL string data.
Client Encoding	Specifies the PostgreSQL character set used by the client.
Include Error Details	When this option is enabled, PostgreSQL error details are included. These may contain sensitive data.

TARGET SESSION ATTRIBUTES	**DETERMINES THE PREFERRED POSTGRESQL TARGET SERVER TYPE.**
Log Parameters	When this option is enabled, parameter values are logged during command execution.

SSL Mode

The SSL Mode property determines how the SSL connection is managed:

- **Disable**: SSL is disabled. If the server requires SSL, the connection will fail.
- **Allow**: Prefers non-SSL connections if the server allows them but also allows SSL connections.
- **Prefer** (Default): Prefers SSL connections if the server allows them but also allows connections without SSL.
- **Require**: The connection fails if the server does not support SSL.
- **VerifyCA**: The connection fails if the server does not support SSL. Also verifies the server certificate.
- **VerifyFull**: The connection fails if the server does not support SSL. Also verifies the server certificate with the hostname.

Server Compatibility Mode

- **None**: No specific server compatibility mode is active.
- **NoTypeLoading**: The server does not support full type loading from the PostgreSQL catalogs.
- **Redshift**: The server is an Amazon Redshift instance.

Timestamp Mode

- **Unspecified** (Default): Date format without timezone
- **Timestamp with timezone**: Date format UTC
- **Timestamp without timezone**: Date format without timezone

File based databases

Overview over the plug-ins:

MS Access Plug-in

Transfer Object

Trigger

MS Access Plug-in

Options for MS Access

Attention: MS Access is not supported in Docker.

The Microsoft Access plug-in additionally requires the Microsoft Access Database Engine, which you can download from Microsoft.

Name	Name of the configuration
File	Choose Access file via button “...” In addition to the .mdb format .accdb files are also supported.
DB data source	Name of “database” to be used
User	Username (if file is protected)
Password	Password (if file is protected)

Transfer Object

The properties of the transfer object are explained in detail in relational databases.

Database Transfer Object

Trigger

The properties of the trigger are explained in detail in relational databases.

Database trigger

Timeseries databases

With this database you can record time series. Each entry has a time stamp. It can be used, for example, to record sensor data or RFID devices.

InfluxDB

Defining access data:

1. Open plug-in by clicking “InfluxDB” in the project configuration’s plugin list.
2. Chose “Add” or double-click an existing entry to edit.
3. Specify access data. (chapter “Properties of InfluxDB access data”).
4. Confirm with “OK”.

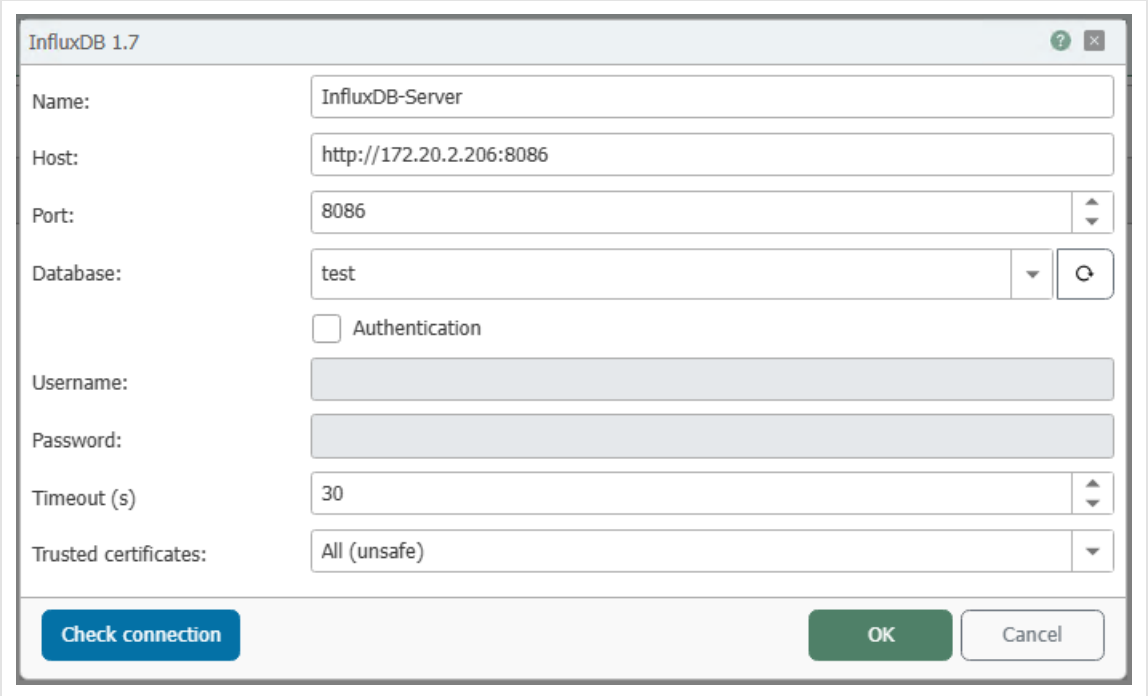
The InfluxDB server can now be used to define a connection.

InfluxDB Plug-in

The following properties are set in the configuration of the InfluxDB plug-in:

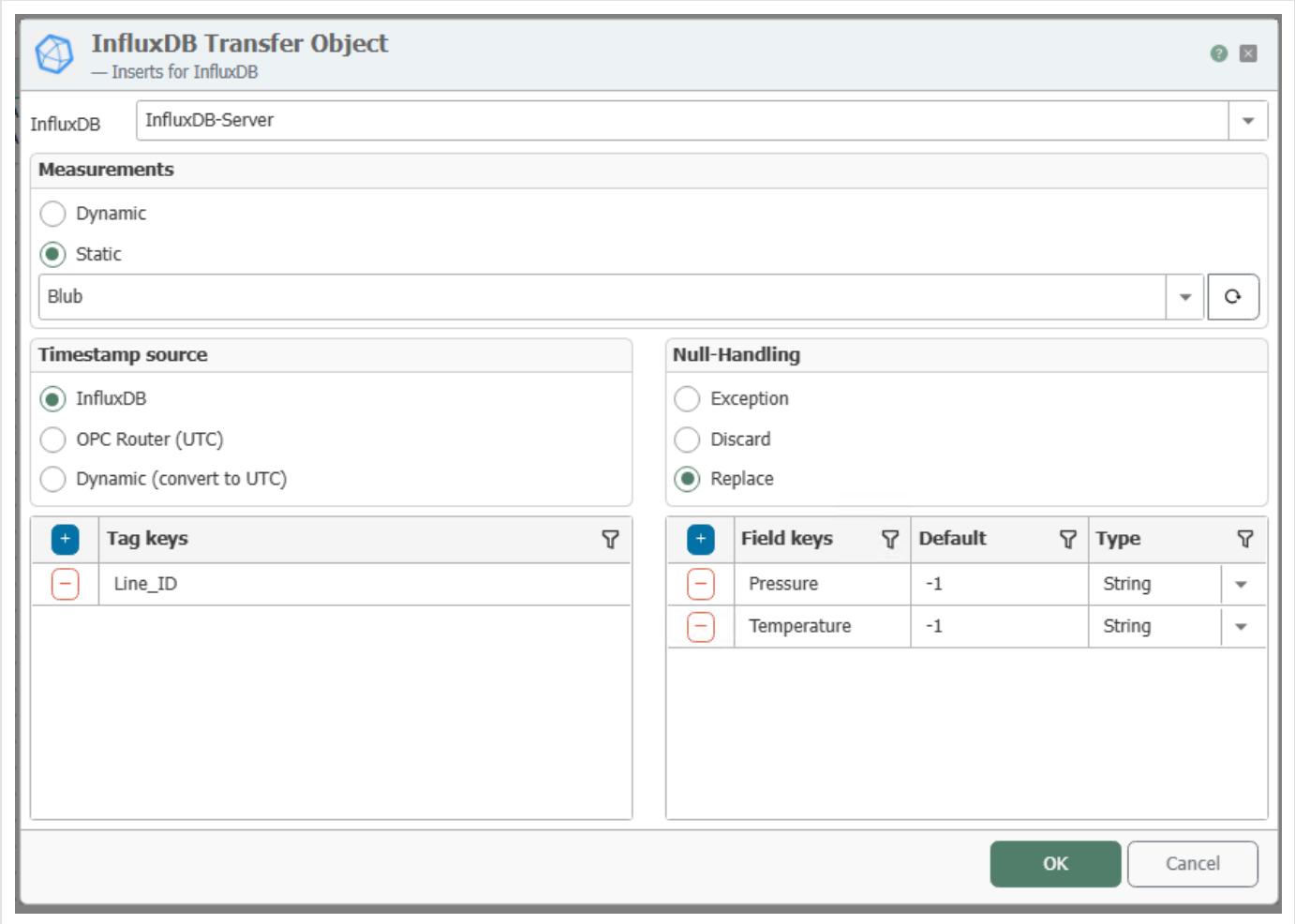
Name	Any name where this connection appears in the connection project planning. Choose a name that you can recognize uniquely!
Host	Host address of the InfluxDB server with http:// prefix and port.
Port	Specification of the port
Database	Name of the database to use.
Authentication needed	Activate this option if you need username and password for your InfluxDB server
Timeout(s)	Provide the timeout in seconds
Trusted certificates	Select the certificate store.

You can use the button “Check connection” to test the plug-in settings.



InfluxDB 1.7 Plug-in

Transfer Object



InfluxDB Transfer Object

Double click the transfer object, configure the options and confirm by clicking “OK”:

InfluxDB connection	Select a InfluxDB instance from the dropdown menu.
Measurement	Select dynamic or a static. You can change a dynamic measurement by a input arrow through the runtime. A static measurement can only be configured in this dialog.
Timestamp	Select the source of the timestamp. If you select InfluxDB, the timestamps of the InfluxDB server will be used, if you select OPC-Router the timestamp of the server where the OPC Router is installed will be used. If you select Dynamic you can define the timestamp by your own through an input arrow. All of the times will be stored in UTC.
Null-Handling	Specify what should happen to entries that are null. **Error:** The transfer is aborted with an error. **Discard:** The entry is discarded. **Replace:** Null is replaced with a value that is specified.

Tag keys	Tags for the value **Note:** The column is only available if “Replace” is selected.
Field keys	Field of the value **Note:** The column is only available if “Replace” is selected.

Note: Measurements, tags and fields which doesn't exist will be created automatically.

Note: If possible, inserts are made in batches. Incorrect data records do not prevent the insertion of valid data that has been transmitted at the same time. The transfer object shows a corresponding error

InfluxDB2-Server

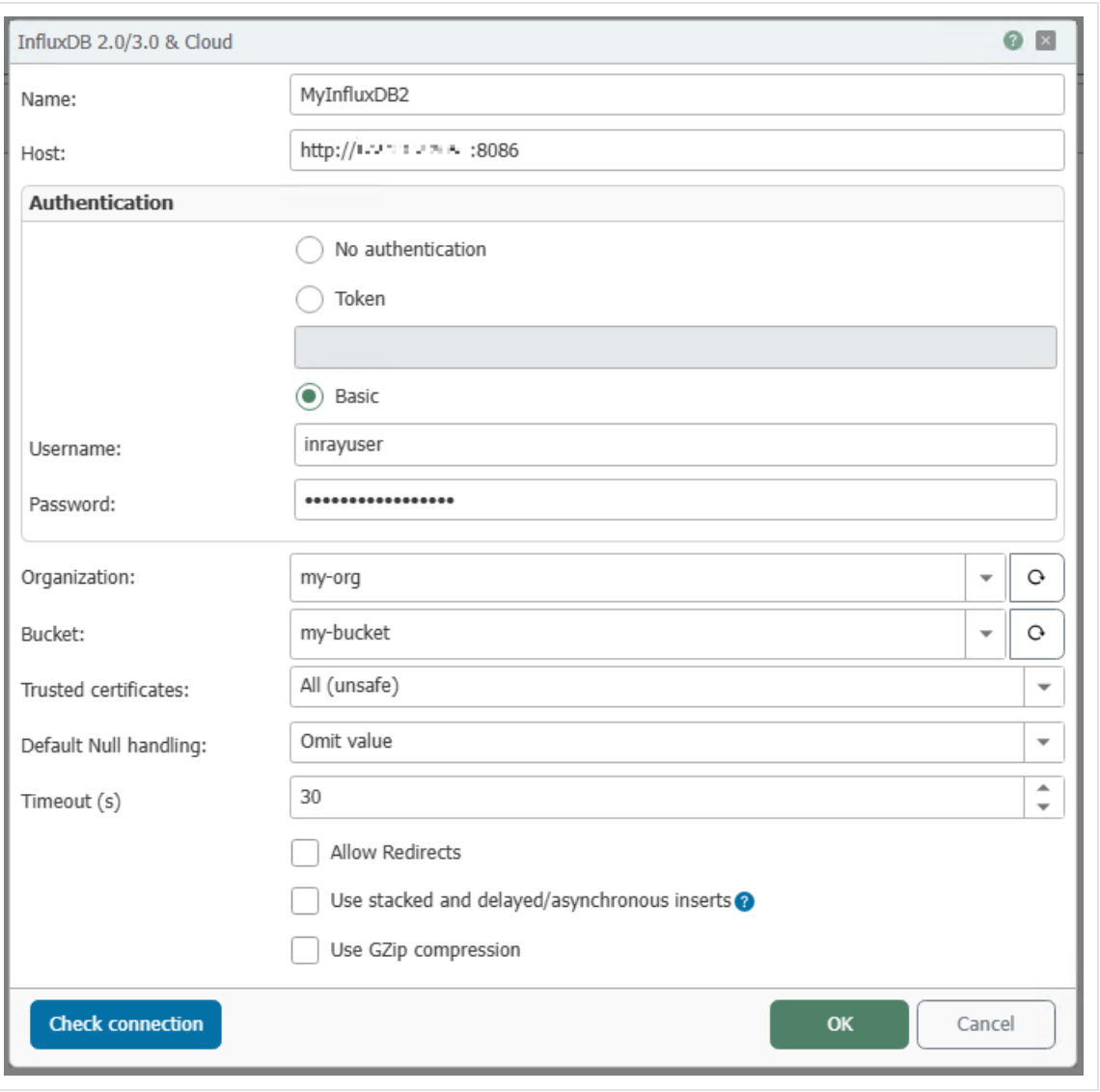
Overview

InfluxDB2 Plug-in

InfluxDB2 Transfer object

InfluxDB2 Plug-in

Open the plug-in setting for the InfluxDB2-Server.

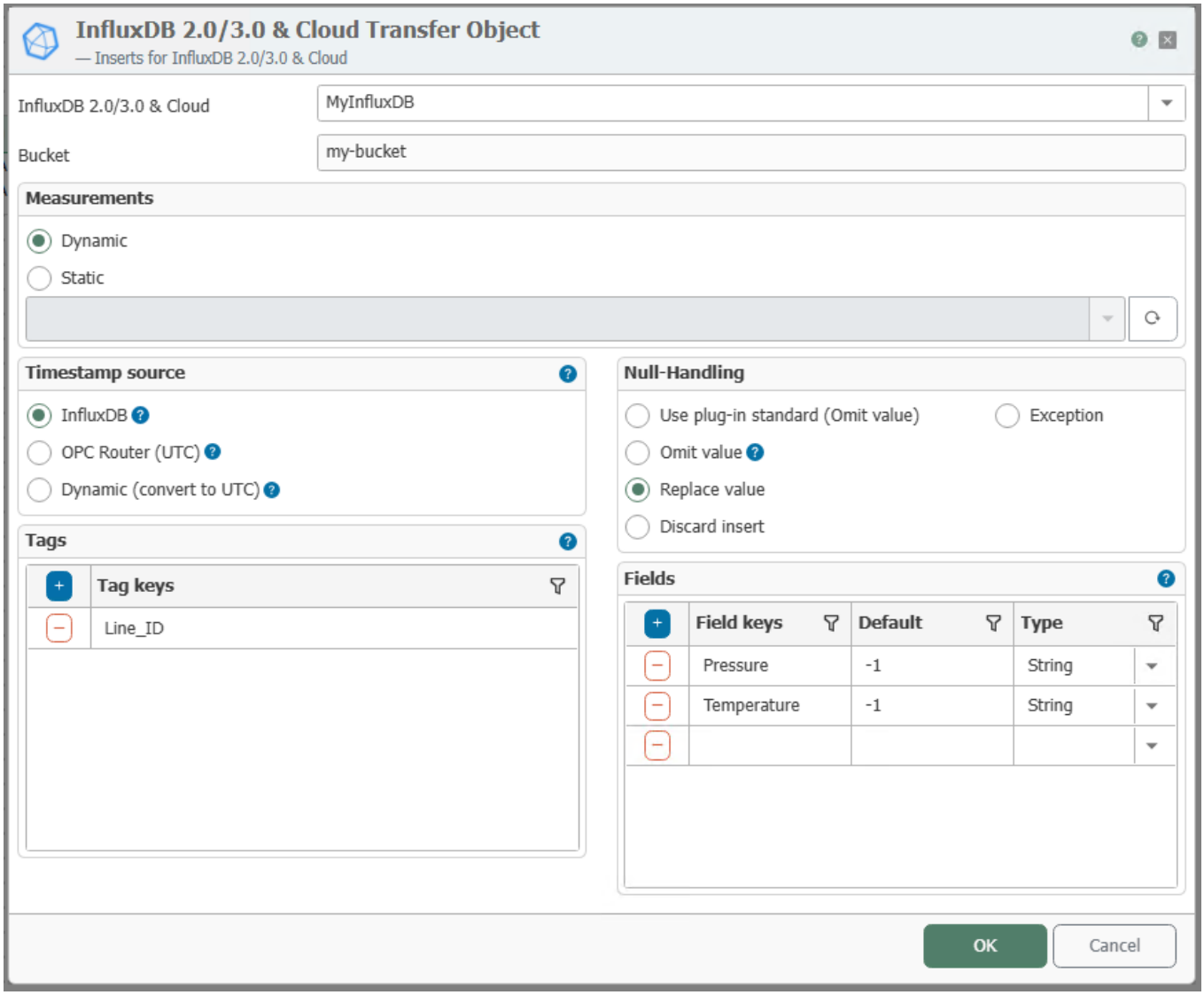


InfluxDB 2.0/3.0 & Cloud

Name	Name how the connection should be called in the OPC Router
Host	Enter the URL of the InfluxDB Cloud here.
No authentication	Select this option if you don't have any security measures in the Influx Cloud.
Token	Enter her the token, which was created in the InfluxDB Cloud. Click in the InfluxDB Cloud on the corresponding token and copy it.
Basic	• Username: Name from the user in the InfluxDB Cloud • Password: Password with which you log in
Organization	Select here the organization which was created in the InfluxDB Cloud.
Bucket	The data is stored in the bucket. Select here your created bucket.

Trusted Certificates	Select here the corresponding certificate if you need one.
Timeout	Enter here the timeout to the InfluxDB Cloud.
Timeout R/W	Enter here the timeout for read and write.

InfluxDB2 Transfer object



InfluxDB 2.0 / 3.0 & Cloud Transfer Object

InfluxDB connection	Select a InfluxDB instance from the dropdown menu.
Measurement	Select dynamic or a static. You can change a dynamic measurement by a input arrow through the runtime. A static measurement can only be configured in this dialog.
Timestamp	Select the source of the timestamp. If you select InfluxDB, the timestamps of the InfluxDB server will be used, if you select OPC-Router the timestamp of the server where the OPC Router is installed will be used. If you select Dynamic you can define the timestamp by your own through an input arrow. All of the times will be stored in UTC.
Null-Handling	Specify what should happen to entries that are null. **Error:** The transfer is aborted with an error. **Discard:** The entry is discarded. **Replace:** Null is replaced with a value that is specified.

Tag keys	Tags for the value
Field keys	Field of the value

Document oriented databases

MongoDB stores data in the form of documents, which are in JSON format.

MongoDB

The regular handling:

1. Open the plug-in via doubleclicking on “MongoDB” at the plug-in-menu.
2. Choose “Add new set of access data” or double click on the existing entry.
3. Do the needed settings (see part “Plug-in”).
4. Confirm with “OK”.

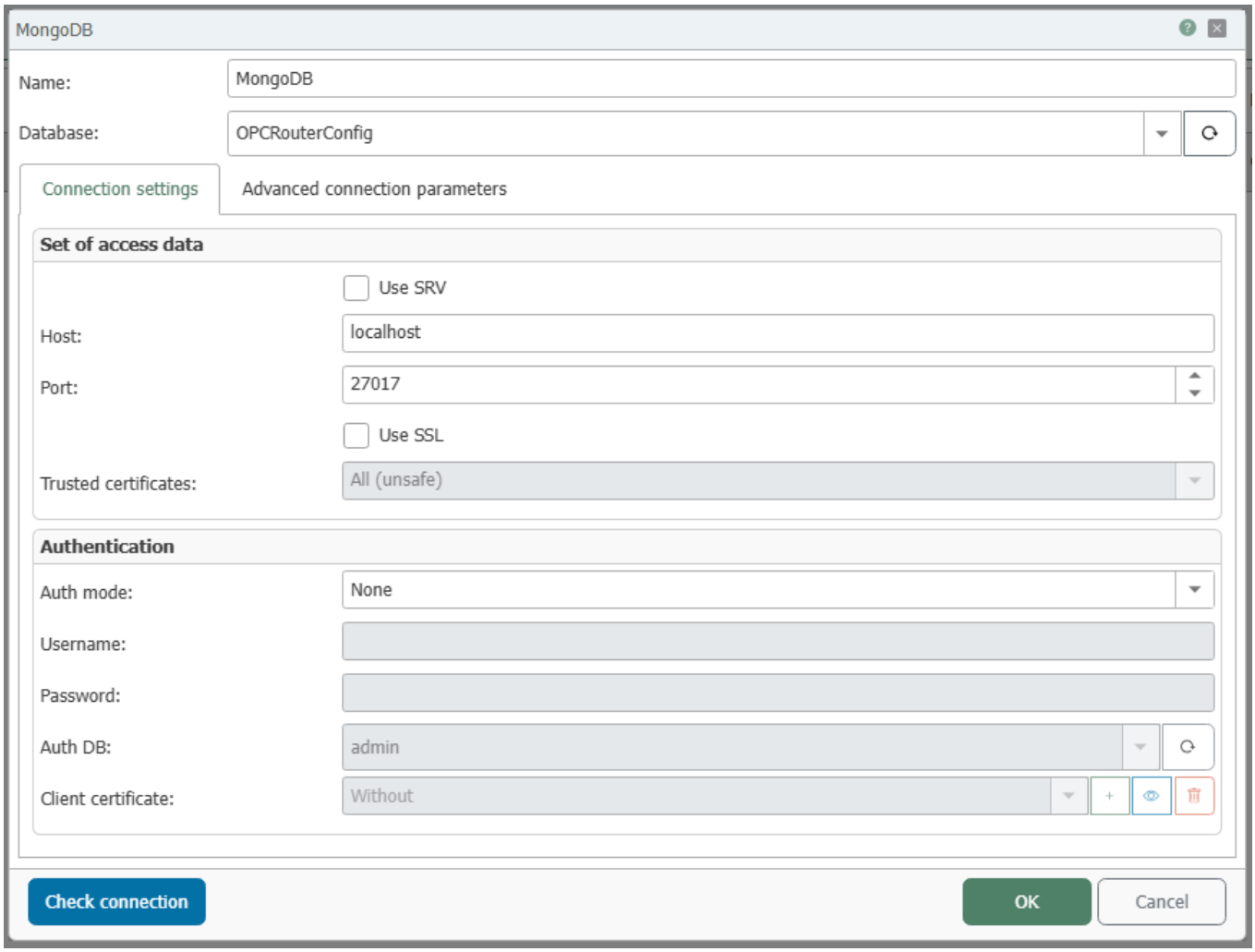
MongoDB Plug-in

You can set the following properties in the configuration of the MongoDB plug-in:

Name	Enter a unique name for you MongoDB plug-in
Use SRV	Activate to use the databases SRV mode
Host	Enter the IP or DNS name of the MongoDB server
Use SSL	Activate to encrypt the network connection to the server
Auth mode	Select the authentication method you need
Username	Enter a username
Password	Enter the user password
AuthDB	Select a user database for your MongoDB
Client certificate	Select a client certificate to authenticate
Database	Select a Mongo database

Tab: Advanced

Timeout(s)	Select a timeout in seconds
Use Replica-Set	Select if you want to use a replica set
Replica-Set name:	Name of the replica set
Read Preference	Select your read preference settings



MongoDB Plug-In

MongoDB Transfer Objects

Overview over transfer objekts:

MongoDB Insert Transfer Object

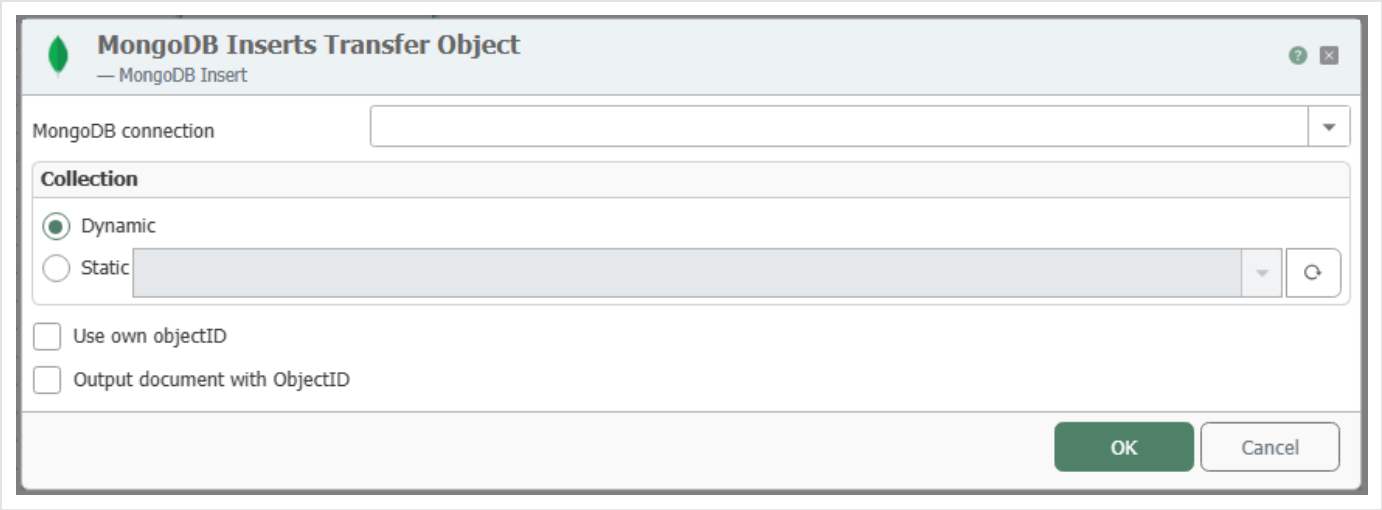
MongoDB Select Transfer Object

MongoDB Update Transfer Object

MongoDB Delete Transfer Object

MongoDB Insert Transfer Object

You can directly configure the transfer object within the connection as described.



MongoDB Inserts Transfer Object

Configure the object's properties by double click:

MongoDB connection	Select one of the configured sets of access data.
Collection Dynamic	Activate this option if you want to change the MongoDB collection by an arrow.
Collection Static	Activate this option if you want to use a static MongoDB collection.
Use own objectID	Use this option if you want to set the object ID. If not checked the OPC Router will create a random number.
Readout document with objectID	Returns the inserted document and object ID (JSON)

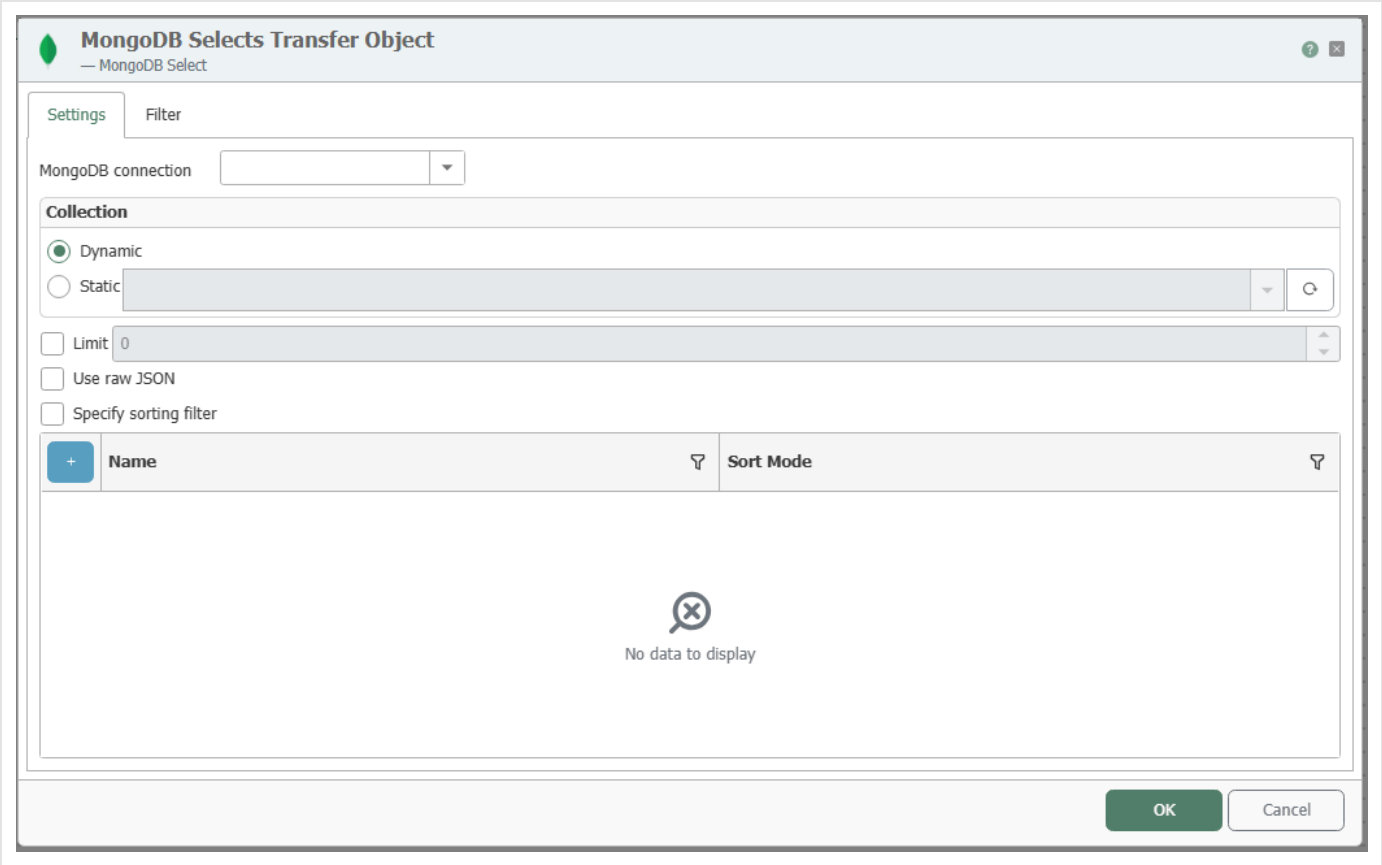
Apply your settings by clicking "OK".

The runtime needs the data in JSON.

If the data is not send in JSON, the data will get inserted with additional "data" tag.

MongoDB Select Transfer Object

You can directly configure the transfer object within the connection as described.



MongoDB Selects Transfer Object

Configure the object's properties by double click:

MongoDB connection	Select one of the configured sets of access data.
Collection Dynamic	Activate this option if you want to change the MongoDB collection by an arrow.
Collection Static	Activate this option if you want to use a static MongoDB collection.
Limit	Set maximum of selected documents.
Use raw JSON	Activate this otion if you want to use a raw JSON string as input. In this case the filter tab is not availabile.
Specify sorting filter	Here you can define sorting filters.Select the data field and descending or ascending.

Tab: Filter (Select, Update, Delete)

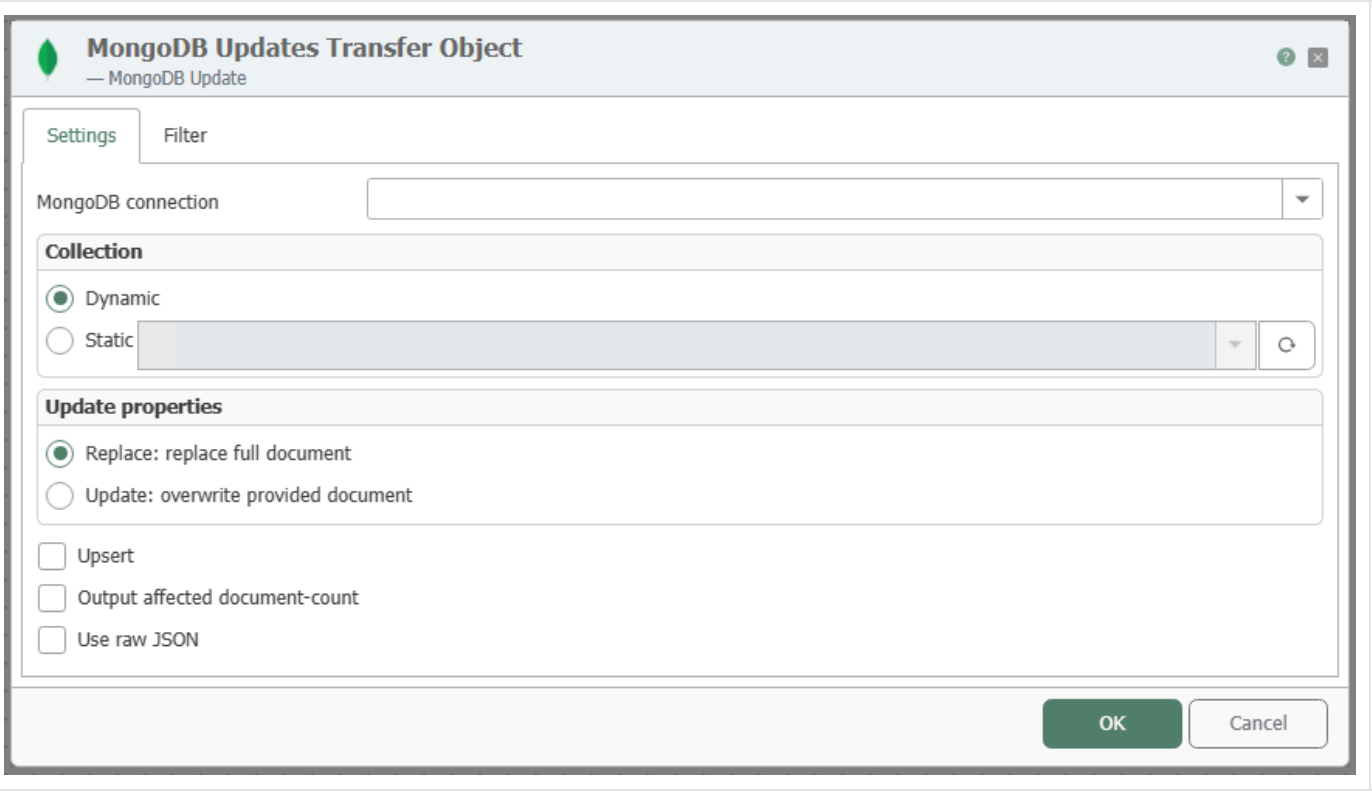
Filter	Shows filters and filter groups. Filter values can be inserted dynamically during runtime. Use Variables/Constant values to insert constant filter criteria. Use Add-buttons to add filters or groups. Delete will remove the selected entry.

Edit Filter/Filter group	Entries are shown dependent on selected entry. **Junction:** Defines whether filters are linked with AND or OR. **Column:** Column to be filtered **Relational operator** to compare values
Preview	Preview of the JSON-String, eg `{"Date": {$gte: "?"}` In this case all documents which date is greater than the timestamp you have used are selected. Values for the filters can be changed during runtime.

Apply your settings by clicking “OK”.

MongoDB Update Transfer Object

You can directly configure the transfer object within the connection as described.



MongoDB Updates Transfer Object

Configure the object's properties by double click:

MongoDB connection	Select one of the configured sets of access data.
Collection Dynamic	Activate this option if you want to change the MongoDB collection by an arrow.
Collection Static	Activate this option if you want to use a static MongoDB collection.
Upsert	If Upsert is activated, the OPC Router will insert a document if there is no document to update
Output affected document-count	Returns the number of updated documents.
Use raw JSON	Activate this option if you want to use a raw JSON string as input. In this case the filter tab is not available.

Tab: Filter (Select, Update, Delete)

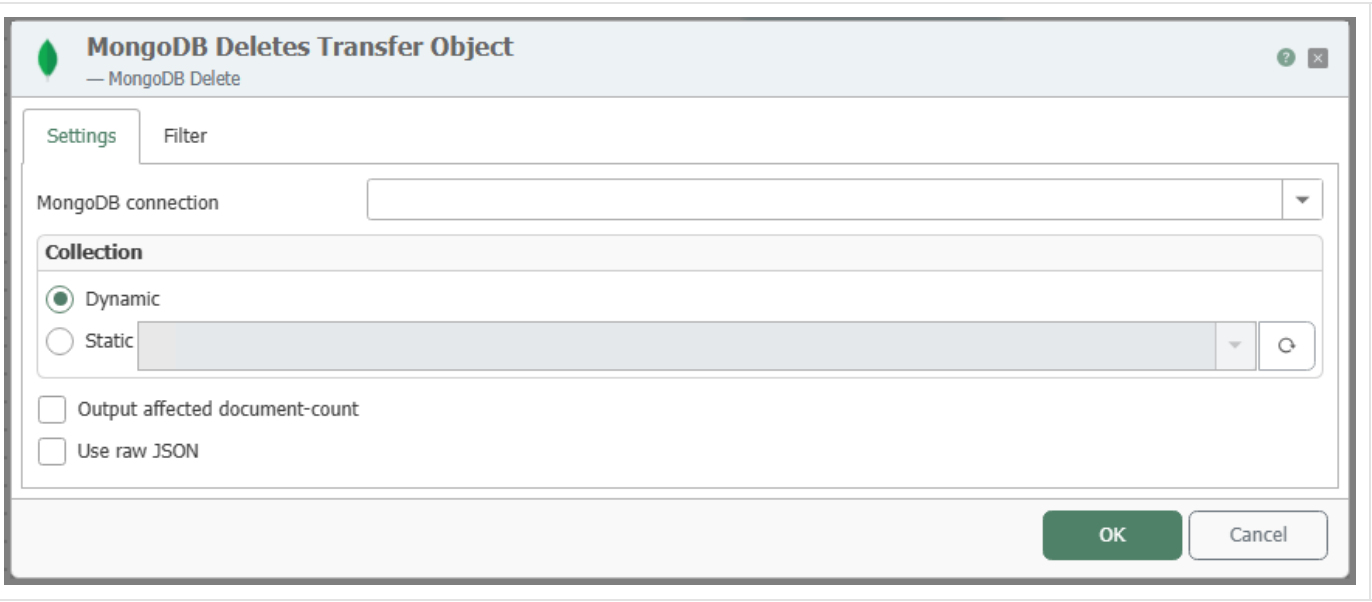
Filter	Shows filters and filter groups. Filter values can be inserted dynamically during runtime. Use Variables/Constant values to insert constant filter criteria. Use Add-buttons to add filters or groups. Delete will remove the selected entry.

Edit Filter/Filter group	Entries are shown dependent on selected entry. **Junction:** Defines whether filters are linked with AND or OR. **Column:** Column to be filtered **Relational operator** to compare values
Preview	Preview of the JSON-String, eg `{"Date": {$gte: "?"}` In this case all documents which date is greater than the timestamp you have used are updated. Values for the filters can be changed during runtime.

Apply your settings by clicking “OK”.

MongoDB Delete Transfer Object

You can directly configure the transfer object within the connection as described.



MongoDB Deletes Transfer Object

Configure the object's properties by double click:

MongoDB connection	Select one of the configured sets of access data.
Collection Dynamic	Activate this option if you want to change the MongoDB collection by an arrow.
Collection Static	Activate this option if you want to use a static MongoDB collection.
Output affected document-count	Returns the number of the deleted documents.
Use raw JSON	Activate this otion if you want to use a raw JSON string as input. In this case the filter tab is not availabile.

Tab: Filter (Select, Update, Delete)

Filter	Shows filters and filter groups. Filter values can be inserted dynamically during runtime. Use Variables/Constant values to insert constant filter criteria. Use Add-buttons to add filters or groups. Delete will remove the selected entry.
Edit Filter/Filter group	Entries are shown dependent on selected entry. **Junction:** Defines whether filters are linked with AND or OR. **Column:** Column to be filtered **Relational operator** to compare values

Preview	Preview of the JSON-String, eg `{"Date": {$gte: "?"}` In this case all documents which date is greater than the timestamp you have used are deleted. Values for the filters can be changed during runtime.

Apply your settings by clicking “OK”.

Network File Access

Chapter overview:

SMB

FTP

Network File Access Transfer Objects

Network File Access trigger

FTP

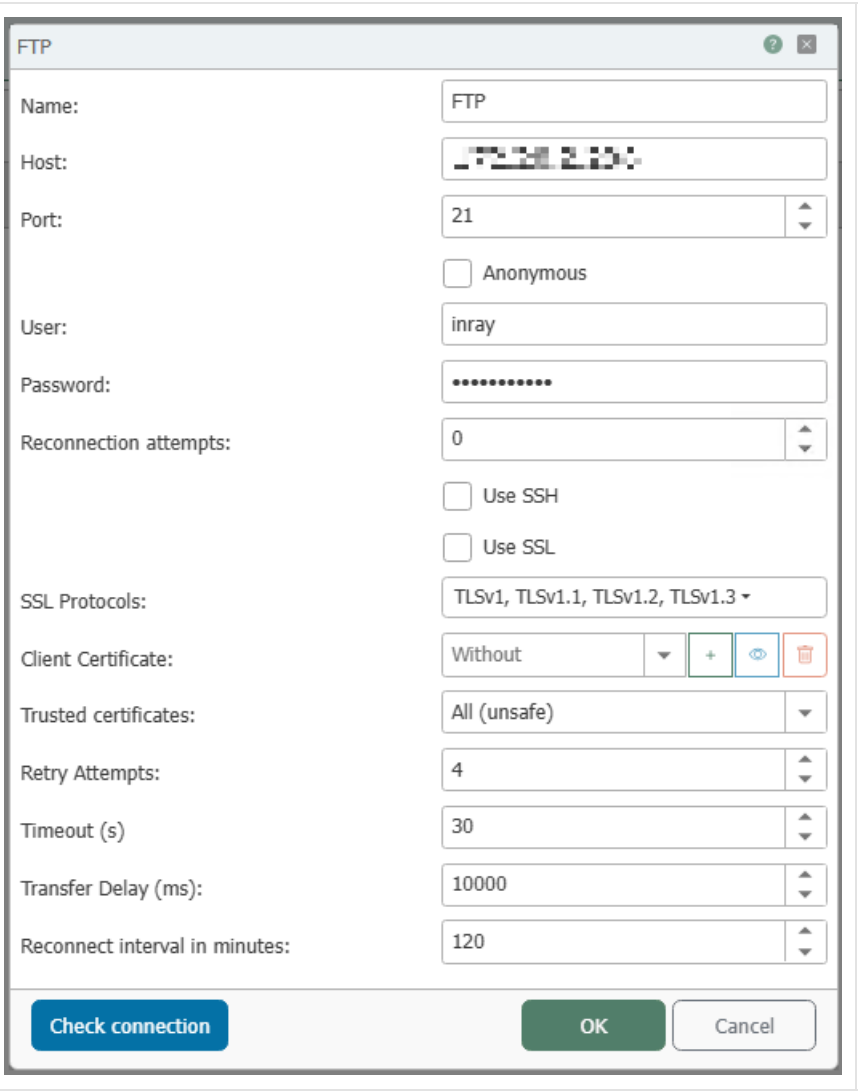
Operation is standard:

1. Open plug-in by double-click “FTP”. The list of created FTP connections is displayed.
2. Chose “Add new set as access data” or double-click an existing entry to edit.
3. Specify access data. (chapter “Properties of FTP”).
4. Use the “Check connection” button to initialize a connection.
5. Confirm with “OK”.

The FTP server can now be used to define a connection.

FTP-Plug-in

Set up following properties to connect to FTP servers:



FTP Plug-in

Name	This set's name. The name will be shown in projection. Select a name you will recognize!
Hostname	IP or DNS of the FTP server.
Port	TCP/IP port of the FTP server.
Anonymous	Activate if you want to connect without user data.
User	Enter a username for the FTP server.
Password	Enter a password for the FTP server.
Reconnection attempts	Number of attempts to reconnect. When the number of attempts is reached, an error is returned.
Use SSH	Activate to use the SSH protocol.
Use SSL	Activate to use SSL protocols.

SSL Protocols	Select SSL protocols you want to use. It need to match with the server protocols.
Client certificate	Select a client certificate if needed.
Trusted certificates	Select server certificates you want to trust.
Retry attempts	Retries before a connection returned an error.
Timeout (s)	Timeout for your FTP-server plug-in.
Transfer Delay (ms)	Delay between the time of finding the file and the start of downloading.

Use the “Check connection” button to check if your connection is successful.

SMB

Attention: SMB is not supported in Docker.

Operation is standard:

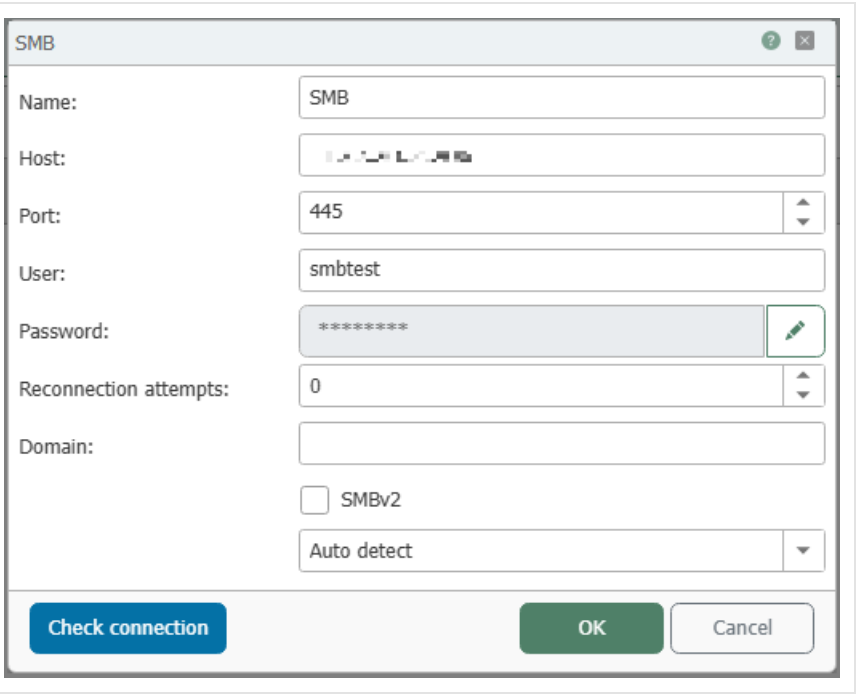
1. Open plug-in by double-click “SMB”. The list of created SMB connections is displayed.
2. Chose “Add new set as access data” or double-click an existing entry to edit.
3. Specify access data. (chapter “Properties of SMB”).
4. Use the “Check connection” button to initialize a connection.
5. Confirm with “OK”.

The SMB server can now be used to define a connection.

SMB-Plug-in

Set up following properties to connect to SMB servers:

Name	This set's name. The name will be shown in projection. Select a name you will recognize!
Hostname	IP or DNS of the SMB server.
Port	TCP/IP port of the SMB server.
User	Enter a username for the SMB server.
Password	Enter a password for the SMB server.
Reconnection attempts	Number of attempts to reconnect. When the number of attempts is reached, an error is returned.
Windows domain	Enter a windows domain.



SMB Plug-in

Transfer objects

Overview of transfer objects:

Network File Access-File Delete Transfer Object

Network File Access-File Write Transfer Object

Network File Access-File Read Transfer Object

Network File Access-File Read

Network File Access-File Read transfer object can read files. File paths can be selected from a database or as a constant. You can directly configure the transfer object within the connection as explained .

After double-clicking on the transfer object, set the following properties:

Tab: Source

Dynamic or static	File source static or dynamic path. Dynamic paths can get defined by an connection arrow and it can change through the runtime.
Abort if file not found	Activate to abort the transfer if the file was not found.
Read binary	Select this option if you want to read a binary file like *.doc, *.jpg, * .xlsx, ...
Read Text, Encoding	Select if you want to read a simple text file and select the files encoding.Select if you want to read a simple text file and select the files encoding.

Network File Access-File Delete

Network File Access-File Delete transfer object can delete files. File paths can be selected from a database or as a constant. You can directly configure the transfer object within the connection as explained .

After double-clicking on the transfer object, set the following properties:

Path	Select the path where the OPC Router should delete files
Name pattern	Select a static or dynamic file name pattern. Dynamic file name pattern can get defined by a connection arrow and it can change through the runtime. You can use placeholders like *.
Include subdirectories	Activate if the OPC Router also have to delete files inside the subdirectories.
Delete unrestricted	Deletes all the files for the name pattern.
Files older than	Only delete files older than selected timespan.

Network File Access-File Write

Network File Access-File Write transfer object can read and manipulate files. File paths can be selected from a database or as a constant. You can directly configure the transfer object within the connection as explained .

After double-clicking on the transfer object, set the following properties:

Tab: Output path

Dynamic or static	File target static or dynamic path. Dynamic paths can get defined by a connection arrow and it can change through the runtime.

Tab: Existing files

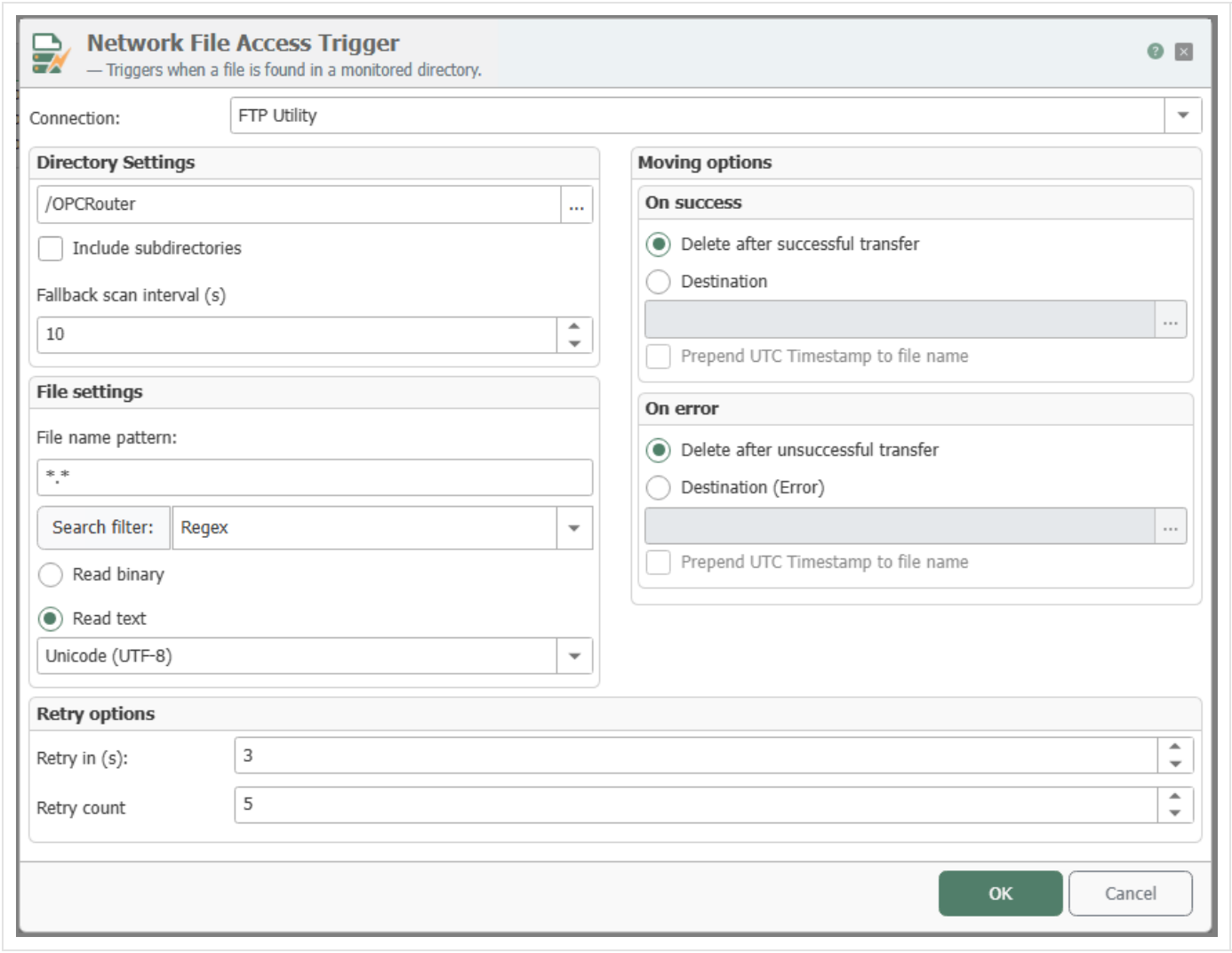
If file already exists	Overwrite: Will overwrite existing files. Throw error: Creates an error if the file is existing. Ignore: No action if file already exists.

Tab: File settings

Dynamic or static filename	Select a static or dynamic file name. Dynamic file names can get defined by a connection arrow and it can change through the runtime.
Prepend UTC Timestamp	Activate if you want to prepend the UTC timestamp to the file. For example the file „test.txt“ gets renamend to „20191204_095646_test.txt“.
Write binary	Select this option if you want to write a binary file like *.doc, *.jpg, * .xlsx, ...
Write text	Select if you want to write a simple text file and select the files encoding.

Network File Access trigger

The Network File Access-File Trigger executes a transfer if a file gets created.



Network File Access Trigger

After double-clicking on the transfer object, set the following properties:

Path	Select the path to observe.
Scan interval	Observes the directory in this interval.
Filename pattern	Define a filename pattern. You can use placeholders like *.
Read binary	Select this option if you want to read a binary file like *.doc, *.jpg, * .xlsx, ...
Read Text, Encoding	Select if you want to read a simple text file and select the files encoding.

Target options:

Delete after successful transfer	Deletes the file which triggered the trigger after a successful transfer..

Success destination	Moves the file which triggered the trigger in a special folder.
Prepend UTC Timestamp to filename	Activate if you want to prepend the UTC timestamp to the file. For example the file “test.txt” gets renamed to “20191204_095646_test.txt”
Delete after error transfer	Deletes the file which triggered the trigger if a transfer aborts for some reason
Error destination	Moves the file which triggered the trigger in a special folder if the transfer gets an error
Prepend UTC Timestamp to filename	Activate if you want to prepend the UTC timestamp to the file. For example the file “test.txt” gets renamed to “20191204_095646_test.txt”
Seconds between tries	Select the time between the retries.
Number of retries	If the OPC Router can't access the selected file, it can retry for n times.

Microsoft Excel

The spreadsheet program Excel is the most widely used tool of its kind. A lot of information is administered, stored and calculated manually in tabular form. The OPC Router can access the files with the Excel Plug-in, with access to all tables and cells to read and write data. You can find a dedicated video about this topic by visiting our tutorial stream

Excel Transfer Object

Excel transfer objects are available as sources and destinations. You can write data to or read data from defined cells. You can also write multiple data records from the database (SQL SELECT) to multiple rows in the Excel file. A plug-in configuration is not necessary, you can configure the transfer object directly within the connection as described.

The following properties can be configured:

"File" tab

File	**Dynamic:** The file path can be passed dynamically as a parameter at runtime. This is helpful if different files are to be written or edited. **Static:** A static file path must be configured in the transfer object. This setting should be used if only one file is to be written or edited.
Action	**Edit file:** An existing file is being modified. **Create file:** A new file will be created.
If file does not exist (available when "Edit file" is selected)	**Create new file:** If the specified file does not exist, it will be created. **Abort transfer:** If the specified file does not exist, the transfer will be canceled. Unlike "Clear error", the error is not written to the log file here. **Throw error:** If the specified file does not exist, an error is triggered. This cancels the transfer.
Options (available when "Create file" is selected)	**Create empty file:** An empty file is being created. **Create from template:** The template from which the file is to be created can be chosen.
Template (available if "Create file" is selected)	**Dynamic:** The file path to the template can be passed dynamically as a parameter at runtime. This is helpful if different files are to be written or edited. **Static:** A static file path must be configured in the transfer object. This setting should be used if only one file is to be written or edited.

"Cells" tab

SPREAD SHEET	**DYNAMIC:** **THE NAME OF THE SPREADSHEET CAN BE PASSED DYNAMICALLY AS A PARAMETER AT RUNTIME. THIS IS HELPFUL IF DIFFERENT SPREADSHEETS ARE TO BE EDITED.** **STATIC:** **A STATIC NAME MUST BE CONFIGURED IN THE TRANSFER OBJECT. THIS SETTING SHOULD BE USED IF ONLY ONE SPREADSHEET IS TO BE EDITED.**
Static cells	Cells can be specified here in the usual "ColumnRow" notation, for example A1, A2, B3 etc. The following write modes are available: **Update:** Overwrites the specified cell with the new value. **Insert:** A new cell is being inserted at the specified position and all cells below it will be moved down. **Append:** The content is being inserted into the first free column below the specified cell. This is useful if several data records, for example the results of an SQL select, are to be written to the table.

"Archiving" tab

Archiving	If this option is activated, the Excel file is saved in the selected folder after the changes have been made. The file name is preceded by the timestamp of the creation time. The name of an archived file then looks like this, for example: `"2024-07-05_13-29-19_{filename}"` **Attention:** If transfers are triggered every second, a new archive file is being created every second too.
Archive folder	**Dynamic:** The folder path can be passed dynamically as a parameter at runtime. This is helpful if archive files are to be written to different folders. **Static:** A static folder path must be configured in the transfer object. This setting should be used if archive files are only to be written to one folder.

Cloud File Access

Overview of available plug-ins:

Cloud File Access Transfer Objects

Overview of available transfer objects:

CFA Read

Cloud File Access file read transfer objects can move files and provide file content from the cloud. In doing so, container/bucket names from the cloud and files from the cloud or as a constant can be selected. You can directly configure the transfer object within the connection as described.

Set the following properties by double-clicking the transfer object:

OPTION	DESCRIPTION
Dynamic or static container/bucket name	Either a dynamic or static name can be chosen for the container or bucket name in the cloud. If a static name is chosen, it can simply be selected from a drop-down menu.
Dynamic or static file name	Dynamic or static file name located in the selected container/bucket. If static, this can simply be selected from a drop-down list.
Read binary data	Select this option if you want to read a binary file. (e.g. *.doc, *.jpg, *.xlsx).
Encoding:	Character encoding can be selected. The default is UTF8.
Abort because file not found:	Activate to abort the transfer process without an error message if no file is found.

Cloud File Access- File Write-Transferobject

Cloud file access file write transfer objects can upload files to the cloud and modify existing files in the cloud. In doing so, container/bucket names from the cloud and filenames from the cloud or as a constant can be selected. You can directly configure the transfer object within the connection as described.

After double-clicking on the transfer object, set the following properties:

Section: Output Container/Bucket

OPTION	DESCRIPTION
Output Container/Bucket	Target container/bucket dynamic or static. If the target container is dynamic, it can be specified at runtime using an arrow.

Section: Existing files

OPTION	DESCRIPTION
Transfer Options	Overwrite: Target file is completely replaced by new file. Create error: If the target file exists, the transfer is aborted with an error. Ignore: File is not written.

Section: File Settings

OPTION	DESCRIPTION
Dynamic or static file name	Select a static file name or use an arrow to enter a file name at runtime.
Prepend UTC timestamp	When enabled, the current UTC timestamp is prepended to the file name. For example, “test.txt” becomes “20191204_095646_test.txt”.
Write Binary Data	Select this option if you want to write a binary file. (E.g. *.doc, *.jpg, *.xlsx).
Write Text	Select this option if you want to write a simple text file. In addition, the file encoding can be selected.

CFA Delete

Cloud File Access – Delete Transfer Objects can delete files from the cloud. To do this, you can select container/bucket names from the cloud and filenames from the cloud or as a constant. You can configure the transfer object directly within the connection as described.

After double-clicking on the transfer object, set the following properties:

OPTION	DESCRIPTION
Container/Bucket Name	Static container/bucket name that can be selected from the cloud in a dropdown.
Dynamic or static file name	Dynamic or static file name located in the selected container/bucket. If static, it can simply be selected from a drop-down list.
Delete without restriction	Deletes all files that match the file name.
Delete files older than	Deletes only files that are older than the specified period.

Azure Blob Storage

You can set the following properties in the configuration of the Azure Blob Storage plug-in:

OPTION	DESCRIPTION
Name	Any name under which this Azure Blob Storage appears in the connection configuration. Choose a name that is unique for you during the connection configuration.
Authorization	Select the connection type for the Azure Blob Storage. Here you will find more information about the different types.
Retry Count	Number of reconnect attempts. If the number of attempts is reached, an error is returned.

Use the “Test Connection” button to check whether your connection is successful.

Authorization

There are three ways to connect to the Azure Blob Storage in the Cloud File Access plug-in:

- **Account Key**: To connect using the account key, you must enter the name of your storage account and the key. You can find the key in the Azure Portal under 'Access keys' (Step 1). The name can be found under Step 2 and the key under Point 3. Copy the key and insert it into the OPC Router. The following figure shows where the information can be found.
- **Connection string**: To establish a connection simply copy the connection string from the Azure Portal under 'Access Key' and paste it into the OPC Router.
- **SAS token**: Generate a SAS token in the Azure Portal under 'Shared Access Signature (SAS)' and enter it together with the storage account name in the OPC Router.

Cloud File Access Trigger

The Cloud File Access trigger initiates a transfer as soon as a file is added to or modified in a container/bucket in the cloud.

The individual properties are:

Container/Bucket Name	Enter the name of the container/bucket to be monitored. You can also select it from the drop-down menu.
Scan Interval	The container/bucket is checked for changes at the interval set here.
Read Binary Data	Select this option if you want to read a binary file. (e.g. *.doc, *.jpg, *.xlsx).
Read text	Read simple text files and select the encoding.
Delete after successful transfer	Delete the file from the container/bucket after a transfer has been successfully completed.
Target container/bucket	Move the file to any container/bucket after a transfer has been successfully completed.
Prepend UTC timestamp to filename	If the file is to be moved, this option can be used to add the current timestamp (UTC) to the source name. For example, the file “test.txt” becomes “20191204_095646_test.txt”.
Delete after failed transfer	Deletes the file from the container/bucket after a transfer has not been successfully completed.
Destination container/bucket (error)	Moves the file to any container/bucket after a transfer has not been successfully completed.
Prepend UTC timestamp to filename	If the file is to be moved, this option can be used to add the current timestamp (UTC) to the source name. For example, the file test.txt becomes 20191204_095646_test.txt.
Time between retries	The time to wait between retries in seconds.
Retries	Can be set to repeatedly try to access a file.

S3 Storage

You can set the following properties in the configuration of the S3 storage plug-in:

OPTION	DESCRIPTION
Name	Any name under which this Azure Blob Storage appears in the connection configuration. Choose a name that is unique for you during the connection configuration.
Access Key	Enter the access key for S3 storage here

Secret Key: Enter the secret key for the S3 storage here Region: If required by your S3 storage provider, you can select the region of your storage here Reconnection attempts: Number of reconnection attempts. If the number of attempts is reached, an error is returned.

Use the “Test connection” button to check whether your connection is successful.

Advanced

Overview:

Variable/Static Plug-in

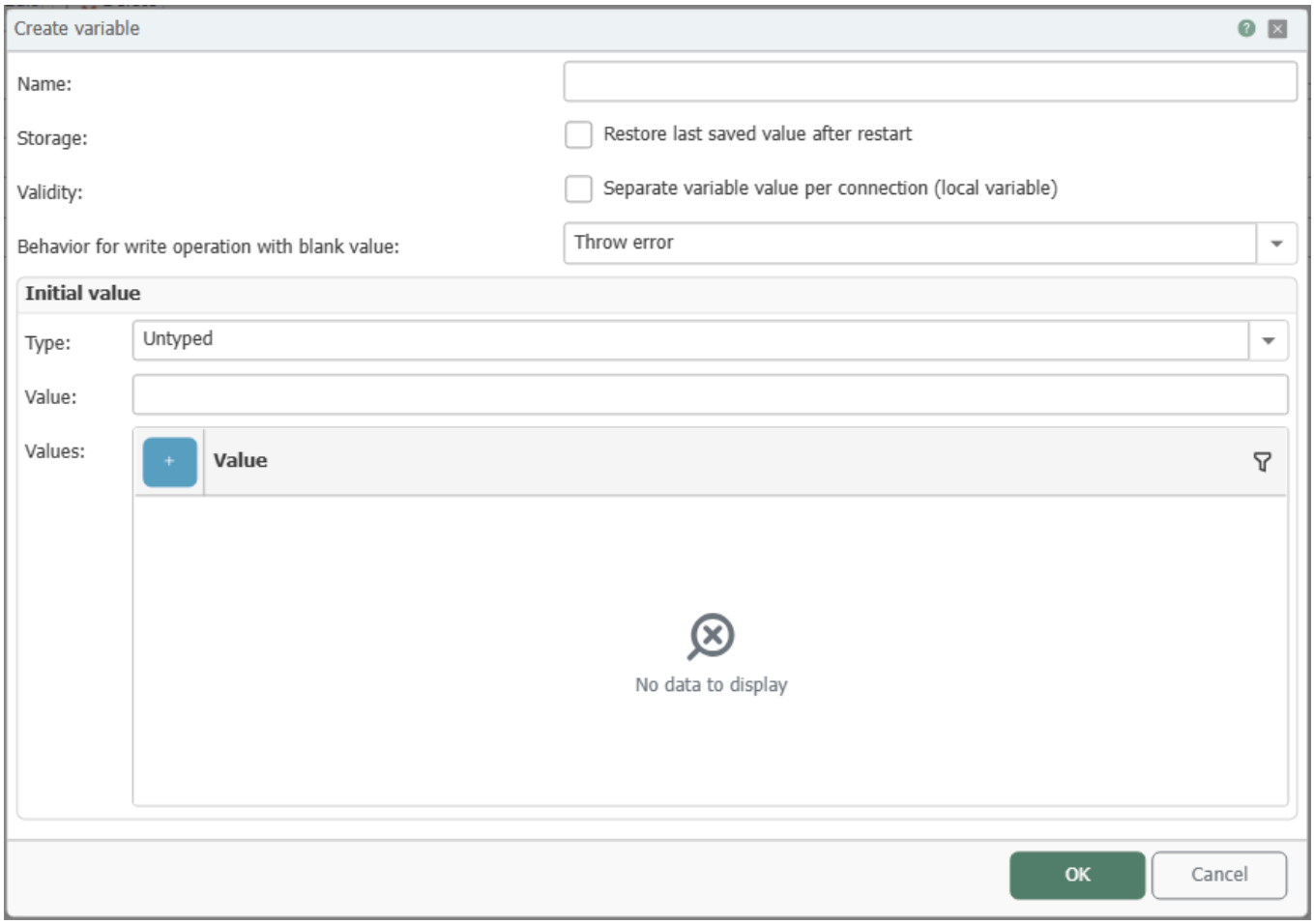
Transfer Objects

Script Plug-in

Variable/Static plug-in

With the variable plug-in you can define any number of variables with name, type, and initial value. Of course, at runtime you can write values in the variables and trigger them again. In addition, some system variables are available that you can use in your configuration.

The use of variables thus lends itself to fixed, repeatedly required values that are not stored in a database or also to cache values that can not or should not be stored in a database. In complex projects, variables can also be used to provide values in multiple connections.



Create variable

The operation is standard:

- Open the plug-in by double-clicking on “Variables” at the plug-in menu
- Choose between global variables and local variables:
  - Global variables: The variable value is always identical in all connections in which it is used.
  - Local variables: The variable is declared individually in each connection. Each value assignment that has been made is only valid within the respective connection.
- Select “Add new variable” or double-click an existing entry.
- Make the necessary settings in the dialog:

Name	Name of the variable, appears in the configuration. The variable name must be unique.
Storage	If enabled, the last value of the variable is reloaded after restarting the OPC Router. If disabled, the set initial value is loaded. Every variable change is stored in the database, local variables are excluded from this.

Validity	Activate if you want to use a local variable. Local variables are only available in one connection. Other connections or the integrated OPC UA Server cannot access the value of local variables.
Type	Select the appropriate data type. The data type is important to know the type beyond doubt, e.g. to distinguish a 1 as a number from a 1 as a byte. The data type Double is also selectable. Typeless is also possible, the variable can then be filled arbitrarily.
Value	Initial value of the variable. In connection configuration, you can read out this value and of course change it. You can also enter the value DB-NULL to output NULL values in the transfer object.

4. Confirm with “OK”.

The variable is now available for configuring a connection.

transfer objects

Overview over transfer objects:

Transfer static

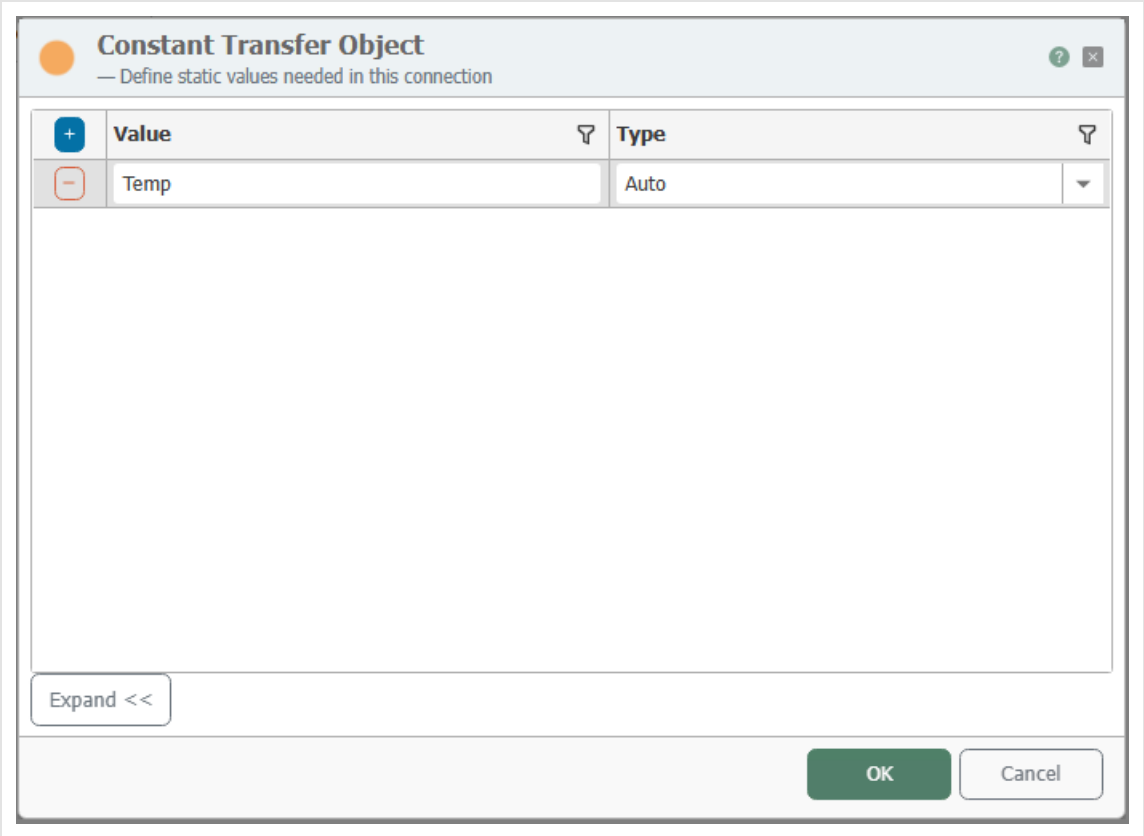
Transfer variable

Transfer static

Constants are available as a data source for fast configuration of “static” values. In contrast to variables, constant objects only apply within the connection in which they are configured. Constant objects can only be set at design time, not changed at runtime.

If a value is too long, you will be asked if you want to cut off this value. If you click “Yes”, the value is truncated to the maximum length of 255 characters. If you click “No”, the dialog remains open and you can select a new value.

Create a constant transfer object within the connection as explained.



Constant Transfer Object

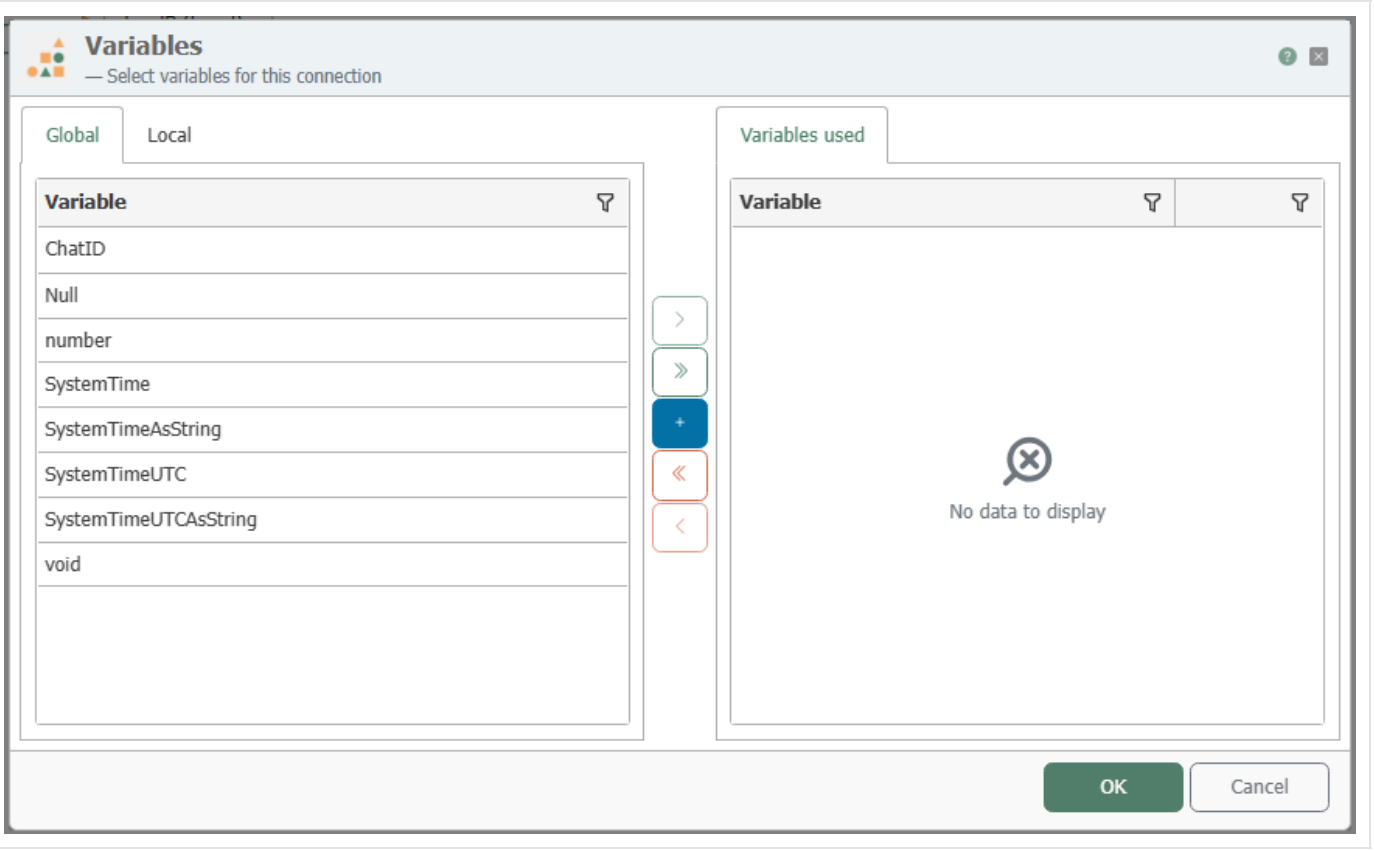
After double-clicking on the transfer object, set the following properties:

Value	Any value
Type (after click on “Expand”)	(optional) Data type

Transfer variable

Self-defined variables are available both as sources and targets, system variables only as sources. The required variables must be configured as plug-in instances. You can create new variables in the transfer object directly. To do this, click on the "Plus" icon. A new window will open as shown in the screenshot. The variable can either be type-less, or you can assign a fixed data type. All common data types, including arrays, can be selected. The newly created variables are then immediately moved to the **Used variables** area. Double-click on the variable or use the < and > buttons to move it. For this purpose, the respective variable must be selected. Use << and >> to move all variables.

Create a variable transfer object within the connection as ecplained.



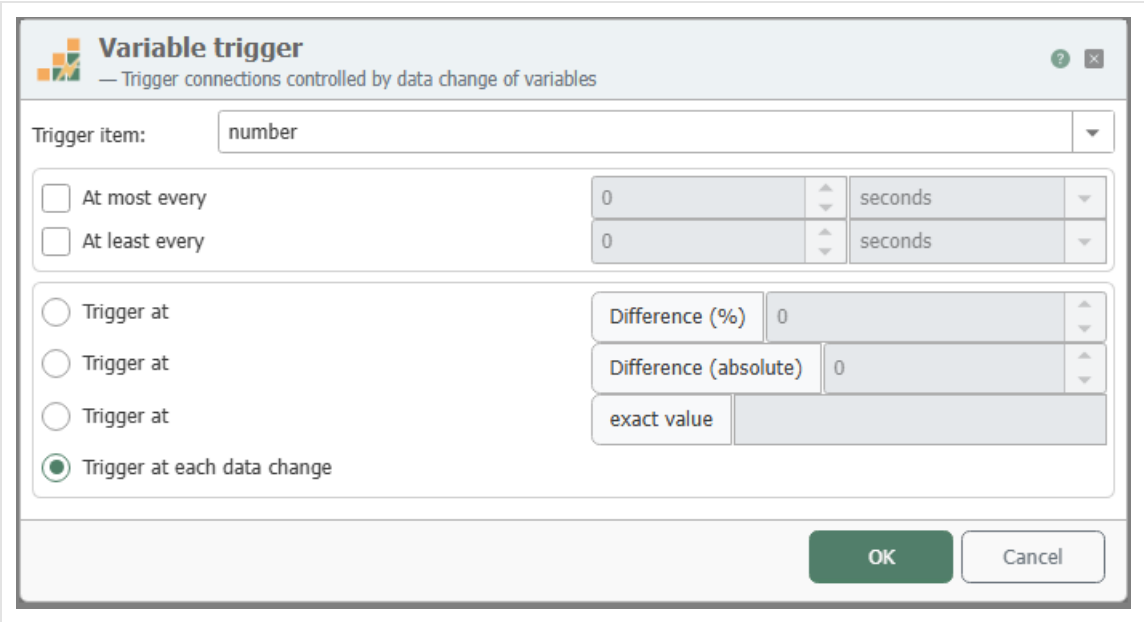
Variables

After double-clicking on the transfer object, set the following properties:

Available variables	Existing, not yet used variables. **Global variables:** The value of the variable is available in each connection. **Local variables:** The value of the variable is only available in the respective connection.
Variables used	Variables that are displayed as elements (items) of the transfer object and are therefore available for transfers.

Name	Name of the variable
Type	Selection of the data type or Untyped
Value	Value of the variable

Variable trigger



Variable trigger

Trigger item	Trigger variable whose change triggers the transfer.
At most every	If the value of the data point changes frequently, the transfer number can be limited here via an interval. The entry thus means: "It is transferred at most e.g. every second, even if the value changes more often". Default always as integer, no decimal places. Therefore, milliseconds, seconds, minutes or hours are to be selected as units.
At least every	Time interval in which data is transmitted in any case, even if the value of the data point has not changed. Default always as integer, no decimal places. Therefore, milliseconds, seconds, minutes or hours are to be selected as units.
Trigger at ... % difference	The transfer is only triggered if the value of the data point changes by this entry (relative).
Trigger at total difference	The transfer is only triggered if the value of the data point changes by this entry (absolute).
Trigger at exact value	Executes the connection if exactly one value is present.
Trigger on each data change	Default setting, overrides triggering at ... difference.

Script Plug-in

Editing scripts

Attention: Within a Docker container, only assemblies compiled for .net Core are supported.

The script plug-in allows you to freely define transfer and trigger objects with your own C # code. In this respect, the plug-in configuration differs from the other plug-ins: After assigning the name and type, open a code editor to edit the script.

At code-editor

- create assembly references
- define input and output parameters
- you already find the basic methods you can program
- there are tools for OPC access and writing log messages

The script plug-in allows you to describe complex issues in the transfer object and triggers; For example, to perform conversions or to trigger complex database queries.

At the top of the menu bar you will find the “Rename script” button (blank sheet with pen) to change the previous name of your script.

Note: In larger application scenarios, the router will be part of an overall strategy, eg. For example, where code carrying business logic is deposited. Even with smaller projects, it makes sense to consider whether z. B. Conversions in the router or stored procedures in the database are made.

IDisposable

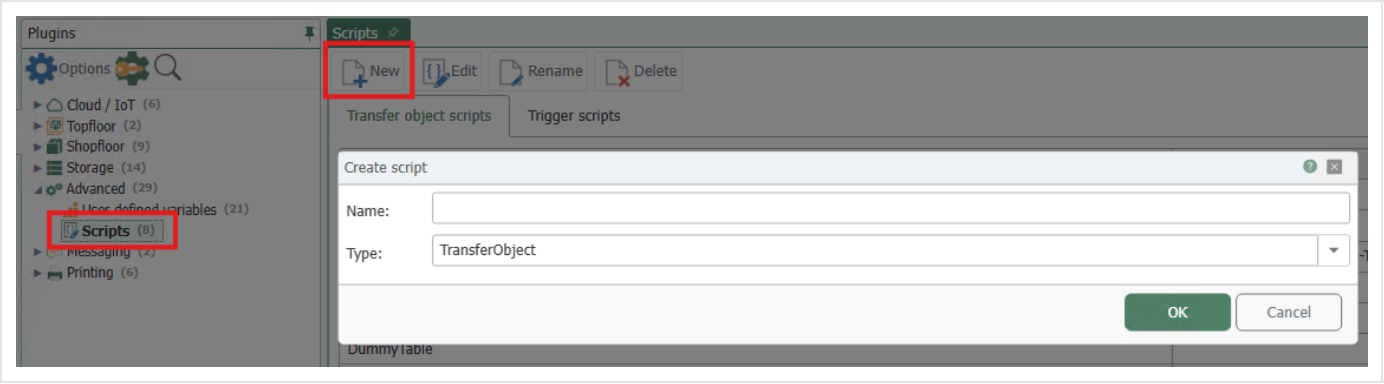
We recommend that you include the IDisposable interface in the script. This allows you to free memory, e.g. when you open ports or start threads in the script. Open ports or running threads will be closed or terminated at the end of the script. Write the required code in the corresponding method.

```
public void Dispose()
{
   // your code
}
```

The Dispose method is called when:

- you publish the script after changes
- Sie das Script ohne Veränderungen veröffentlichen
- after deleting a connection, you publish it
- the plug-in is stopped (this always happens when publishing)

Creating scripts



Script Plug-In

1. Open the plug-in by double-clicking “Script” at the plug-in menu. You receive the list of transfer object scripts and trigger scripts on two tabs.
2. Choose “Add script” to create a new script and specify name and type. When you create a new script, the script opens automatically in the script editor. Or double-click on an existing entry to open the script in the script editor.
3. Program your script (see below). Remember to compile and save.

Transfer objects and triggers are then available for connection configuration.

Transfer object script

When you create a new script of type transfer object, there are three predefined methods available:

Methods in the transfer object

Method	Description
------	----------
Initialize	The method is called once when the OPC Router is started.
Write	The method is called after another transfer object has written data to this transfer object. So it is the code that is executed when this object serves as the destination of a data transfer.
Read	The method is called before another transfer object reads data from this transfer object. So it's the code that gets executed when that object serves as the source of a data transfer.

You use the “Settings - Edit parameters” menu to create the parameters and variables. You can set the following properties for the parameters:

Properties of parameters

PARAMETER	DESCRIPTION
Name	Name of parameter
Array	Yes / No — Determines if it's an array. Arrays can be used as output or SQL sources.
Type of parameter	Data type
Direction of parameter	Output: used as source Both: input and output None: used only inside script, not visible.

When you create a new script, the editor automatically opens for editing.

Trigger Script

When you create a new script of type “Trigger”, there are three predefined methods available:

Methods in the trigger object

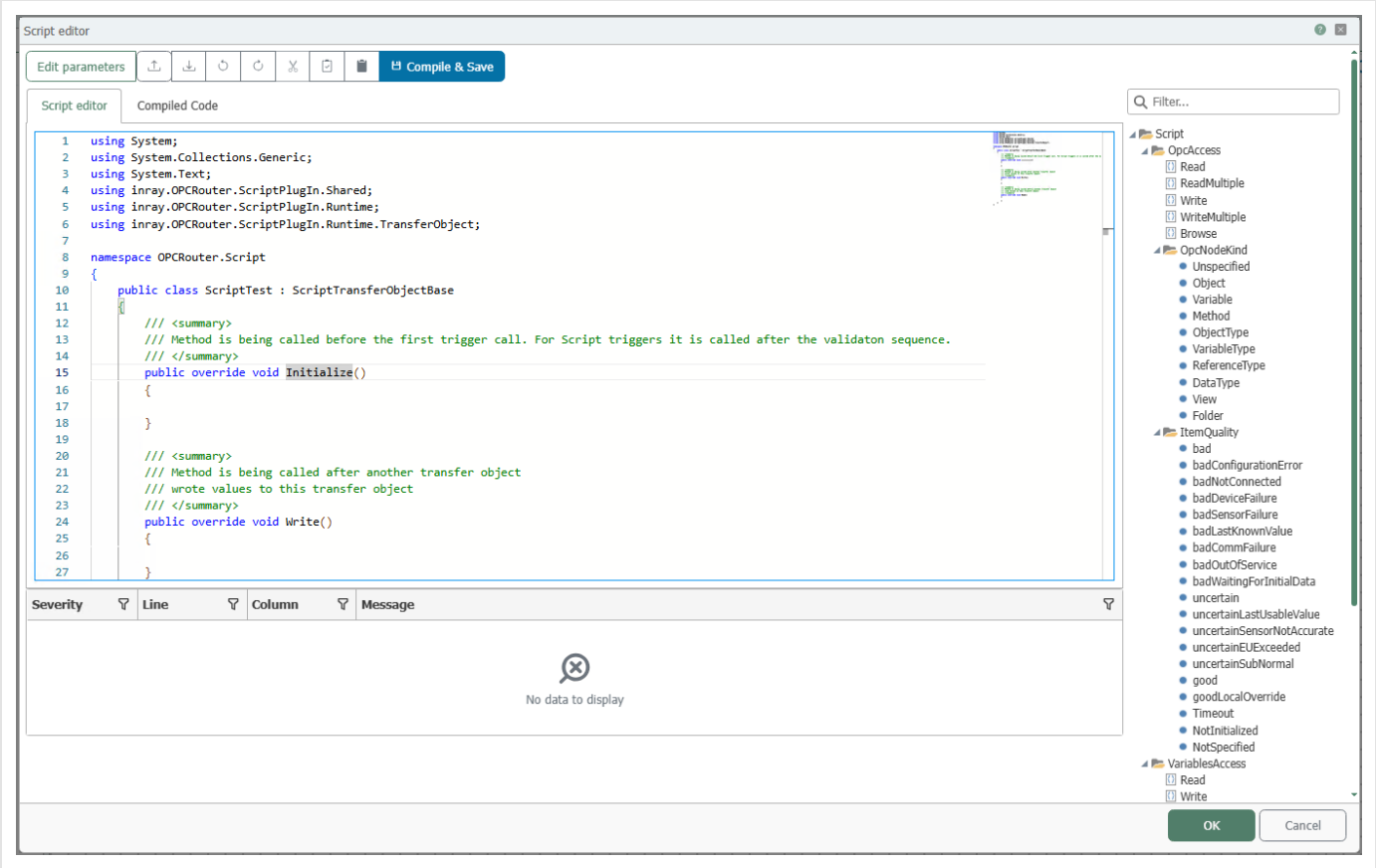
Method	Description
------	----------
Initialize	The method is called once when the OPC Router is started. Here you specify at least the “query interval”, the interval in which it is checked whether the trigger condition is correct.
CheckTrigger	Method in which the trigger condition is stored. It is called cyclically in the “query interval” (see above). The method returns back `true` or `false`. At `true` the data transfer is triggered.
ResetTrigger	The method is called after successful data transfer. The parameter is automatically transferred `(bool transferError)` (`true` = transfer error). In this method, an error handling can also be programmed.

Use the “Settings - Edit parameters” menu to set the parameters. The script trigger only knows input parameters, so you can not specify a direction here.

The parameter value is defined in the connection configuration. It is not dynamic at runtime. If you need actual values at runtime, you would need to program corresponding OPC accesses or database queries in the trigger. The input parameters make it possible to use the same script in different connections, which basically require different properties (table names, access paths, etc.).

When you create a new script, the editor automatically opens for editing.

Editor



Script editor

In addition to standard operating functions (copy, paste, cut, undo, save), the editor offers some special functions:

- Menu “File – Save externally”: Saves the script as an XML file with header data, code, and parameter definitions. Opening such XML files and thus importing scripts is also possible.

Note: If you want to use the assemblies in the scripts, they must still be included in the code with using.

- Menu “Settings – Edit parameter”: Here you create required parameters and variables (s. Transfer object script und Trigger).
- Menu “Settings – Edit comment”: The comment is displayed in the script list of the plug-in.
- “F6” key compiles and tests the program for errors.

The right-hand part of the window contains a graphic programming aid for OPC access, variables from the OPC Router configuration, log messages and parameters.

- Double-click an entry and the prepared line of code will be inserted. Entries can be written as a string (in quotation marks) or as a predefined variable.

OpcAccess.Read

Reads the value of an OPC item.

```
Object <myValue\> = OpcAccess.Read("<ItemPath\>", "<OPCServerName\>")
```

- Return: Object
- ItemPath: Access path and name of the item
- OPCServerName: Name of the OPC connection in the router project

OpcAccess.ReadMultiple

Reads out the value of several OPC items.

```
List<string> <lItemNames\> = new List<string> ();

<lItemNames\>.Add ("<ItemPath.ItemName1\>");
<lItemNames\>.Add ("<ItemPath.ItemName2\>");
...

Dictionary<string,object> <MyValues\> = OpcAccess.ReadMultiple(<lItemNames\>, "<OPCServerName\>");
```

- Return: Dictionary <*string*>
- lItemNames: Access paths and names of the items as a list
- OPCServerName: Name of the OPC connection in the router project

OpcAccess.Write

Writes a value to an OPC item.

```
OpcAccess.Write ("<ItemPath\>", <objValue\>, "<OPCServerName\>")
```

- Return: none
- ItemPath: Access path and name of the item
- objValue: value to be written
- OPCServerName: Name of the OPC connection in the router project

OpcAccess.WriteMultiple

Writes several values into several OPC items.

```
Dictionary<string,object> <dicItemsToWrite\> = new Dictionary<string,object> ();
<dicItemsToWrite\>.Add("<ItemPath.ItemName1\>", <Value1\>);
<dicItemsToWrite\>.Add("<ItemPath.ItemName2\>", <Value2\>);
...
OpcAccess.WriteMultiple(<dicItemsToWrite\>, "<OPCServerName\>")
```

- Return: none
- dicItemsToWrite: Dictionary with items and values
- OPCServerName: Name of the OPC connection in the router project

OpcAccess.Browse

Offers the possibility to perform a browse (object search) for OPC DA and UA connections.

ItemPath: Access path and name of the item

ServerName: Name of the OPC connection in the router project

Various queries are at your disposal:

```
OpcAccess.Browse(ItemPath, ServerName).Select(e => e.DisplayName).ToArray()
```

- Return: Specifies the names of the tags from this OPC path.

```
OpcAccess.Browse(ItemPath, ServerName).Select(e => e.NodeKind.ToString()).ToArray()
```

- Return: The type of tags from this OPC path.

```
OpcAccess.Browse(ItemPath, ServerName).Select(e => e.Identifier).ToArray()
```

- Return: The name of the tags with full path.

```
OpcAccess.Browse(ItemPath, ServerName).Select(e => e.HasChildren).ToArray()
```

- Return: If the tags still have child objects.

VariablesAccess.Read

Reads out the value of a variable of the OPC Router project. What is meant here are only variables created via the variable plug-in, ie no placeholders in templates, mail objects or similar. If the variable does not exist, NULL is returned.

```
Object <myValue\> = VariablesAccess.Read("<VariableName\>")
```

- Return: Object
- VariableName: Variable name in the OPC Router project

VariablesAccess.Write

Assigns a value to a tag of the OPC Router project. What is meant here are only variables created via the variable plug-in, ie no placeholders in templates, mail objects or similar. If the variable does not exist, it will be recreated.

```
VariablesAccess.Write("<VariableName\>", Value)
```

- Return: none
- VariableName: Variable name in the OPC Router project
- Value: Value assigned, Object

Log

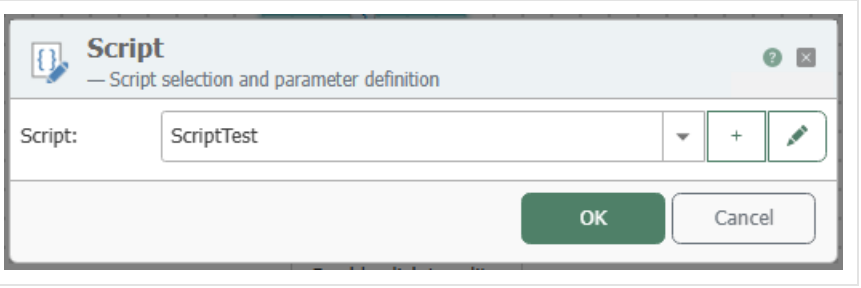
Schreibt eine Nachricht in die Logdatei

```
Log("<strMessage\>", MessageType.<LogLevel\>)
```

- Return: none
- strMessage: message to be written
- LogLevel: to be written log level. If the corresponding log level is deactivated, it will not be recorded either. `MessageType.All` logs in any case, regardless of the router configuration.

Script Transfer Object

You use the self-programmed scripts in the data transfer with the transfer object “script”. The scripts must be of the transfer object type in the Plug-in-configuration.



Script Transfer Object

Create a script transfer object within the connection as explained.

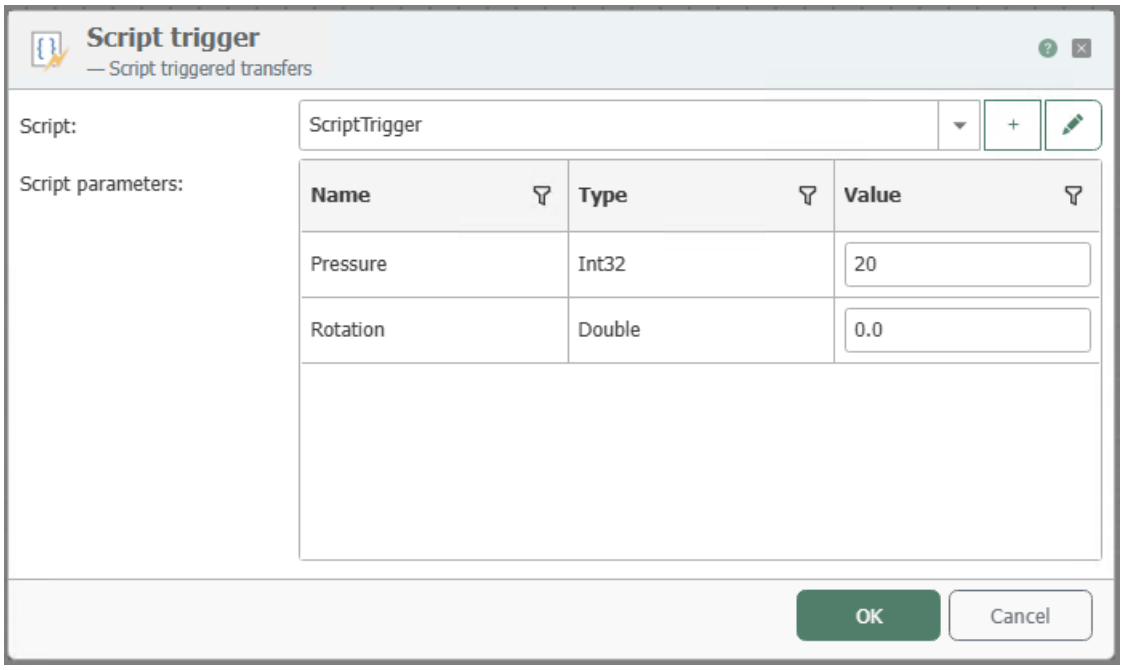
After double-clicking on the transfer object, set the following properties:

Data Source	Available scripts to choose from.

The input and output parameters of the script are now available to you as elements of the transfer object.

Script trigger

The script trigger triggers a data transfer if the CheckTrigger method returns `true`. In the script trigger, the programmer can define input parameters that must be set here, in the connection configuration (with value or template variable). They can not be filled dynamically at runtime. If the script trigger requires dynamic values, these must be determined in the trigger itself (e.g. with `OpcAccess.Read`). Script triggers can also be used with variables in templates and transferred to instances. Attention: No values can be output in the initialization area. The parameters are reset there before the transfer. Conversion from the OPC Router 3.x can be carried out including variables and values.



Script trigger

The individual properties are:

Script	Trigger script being executed.
Script parameters	Input parameters defined in the script. All parameters must be provided with a value or a template variable.

Messaging

Here all plug-ins are listed, which are used for communication or for sending messages.

Telegram

E-Mail

E-Mail

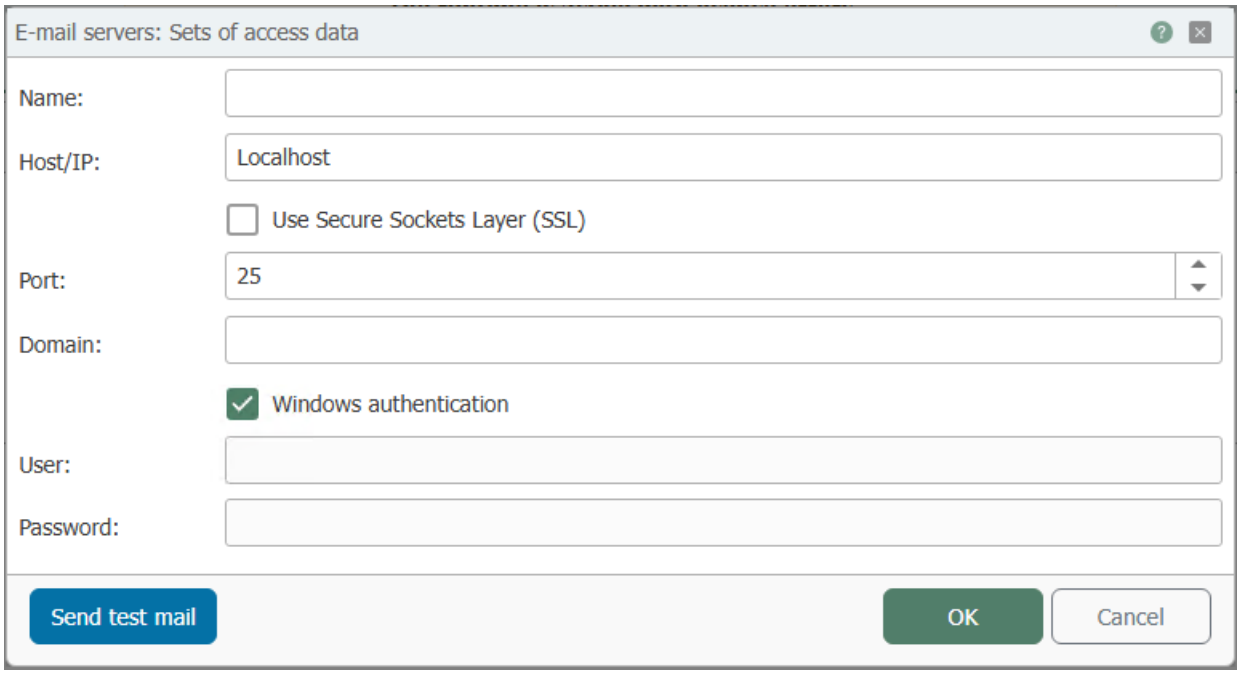
The email plug-in connects the OPC Router to a random number of accessible mail server. You just have to state the login information for the mail (outbox) server. Details to mailing are projected later in the connection.

As far as a firewall is between the OPC Router and the mail server, please check the authorization.

The mail server plug-in regularly supports SMTP- and NT athentification, SSL is optional.

The regular handling:

1. Open the Plug-in by clicking on “E-mail servers” in the plug-in menu.
2. Click the plus button or double click on the existing entry.



1. Make the necessary settings (see part “Plug-in”).
2. With “Send test mail” you can send a mail, to test the settings. Another window will pop up. Enter an email address and click “Test”. The process log will be displayed in the dialog, if the sending fails, you can get more information from the "Error Message".\

```
![Sending Testmail](/assets/TestMailEN.png)
```

1. Confirm with “OK”.

Note: The test mail is sent in the language that was set in the management.

The Mail server is ready to be used in connection projection.

E-Mail Plug-in

The following properties can be set in the configuration of the email plug-in:

Name	Any name under which this mail server appears in the connection configuration.
Host/IP	IP-address or hostname of the mail server
Use Secure Sockets Layer (SSL)	Uses SSL encryption. The mail server must support this and the appropriate port must be set. Explicit SSL (also known as STARTTLS) is used; on the server side, this is usually port 587.
Port	Port on the server that is used for mail traffic (default: 25, for SSL: 587).
Domain	Domain where the user is registered (for login to a mail server with NT authentication).
Windows authentication	Uses alternatively the user, under which the router service runs (Standard: “System”).
User	Username for logging on to the mail server (SMTP or NT authentication).
Password	Password used to log in to the mail server (SMTP or NT authentication).

With the button “Send test mail” you can check your entries.

E-Mail Transfer Object

An email object is a possible transfer destination. The needed email (SMTP) server has to be configured as plug in instance. Create a mail transfer object as explained within the connection.

![](</assets/EmailTOEN (1).png>)

Placeholders can be used with the syntax [%placeholder name%] in all text fields.

EMAIL CONNECTION	**THE SMTP SERVERS ARE AVAILABLE HERE AS CONFIGURED INSTANCES OF THE OPC PLUG-IN.**
Number of recipients	Enter the number of recipients here. The transfer object will provide this number of elements, which you fill with e-mail addresses at runtime, e.g. from variables or a database table. You can use a dynamic database filter to flexibly design the group of recipients.
Number of CC	Like number of recipients. If CC is specified, all CC recipients are added to each email.
Sender	The sender of the email must have the format of an email address (name@domain.xx).
Display name	This name is displayed as the sender.
Reply-To	Enter an e-mail address if a reply e-mail from the recipient should not be sent to you but to another recipient.
On behalf	Enter the e-mail address of the person on whose behalf you have written the e-mail.
Subject	Subject line of the e-mail.
Number of file attachments	Provides a corresponding number of elements for file paths/names in the transfer object, which you can fill either via a constant object or dynamically at runtime. These file(s) are then sent as attachments. The path of the file is then dragged into the corresponding line of the transfer object.
Attachments as binary data	The attachments are converted into binary data. A line for the file name and one for the file content is then displayed in the transfer object in the workspace.
Text as HTML	Delivers the e-mail in HTML format.
Send emails individually	An email is sent to each recipient individually so that the individual recipient cannot see who else the email has been sent to. Even if an e-mail address does not work, all other recipients can still receive their message.
Text content	Text of the e-mail.
Placeholder	Define any variables here that you fill at runtime, e.g. with process values or messages. Create a new placeholder with the plus button. Select a placeholder and insert it into the text using "Insert in subject line" or "Insert".

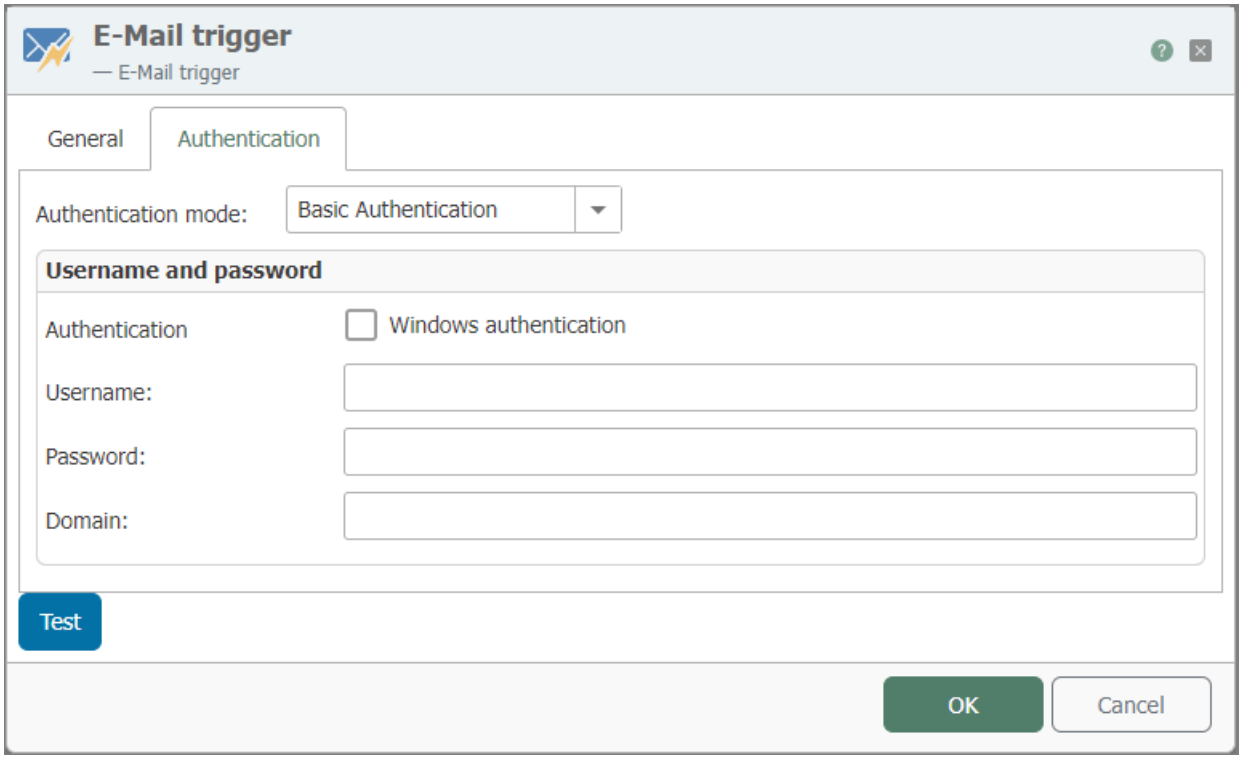
E-mail trigger

The email trigger, starts a transfer after receiving an email. Emails in MIME format are also processed.

The followind properties can be configured:

Server	The address of the email server
Protocol	Select your preferred protocol
Port	Enter the TCP port of the mailserver
Encryption	Select: • Auto • SSL/TLS • STARTTLS • STARTTLS(Optional) • None
Trusted certificates	Set which certificates should be trusted: • Router • Windows • All(unsafe)
Mail folder	A "Mail folder" in an email trigger is a specific location within an email application where the trigger is set to monitor for new email. When setting up an automated workflow, such as with Microsoft Power Automate, you can specify which mail folder (such as Inbox, Sent Items, or a custom folder) the trigger should monitor. When a new email arrives in the specified folder, the trigger can initiate a predefined action or workflow.
Polling interval (s)	Select the interval to check for new mails
Idle timeout	The connection is kept open by the router for a set period of time. If no emails are received, the time restarts. If an email is received during this period, the timeout is terminated immediately. **Attention:** This function must be supported by the email server.
No-Op interval	Refers to a time period during which the trigger checks for new emails but takes no operation (no-op) if no new emails are found.
Delay (ms)	Refers to the amount of time, in milliseconds, that the system waits before executing the next action after the trigger conditions are met. This setting is useful for ensuring that actions do not overlap or execute too quickly, which can be important when dealing with email processing where timing is crucial.
Retries	Count of retries before a transfer gets cancelled
Processing order	Select which emails are processed first

Action	Select whats happening to the emails after the transfer



Authentication mode	Refers to the method by which the email system authenticates a user or service before allowing access to trigger an email-based workflow. This setting is crucial for security and ensures that only authorized entities can initiate the email trigger.
Windows authentication	Uses alternatively the user, under which the router service runs (Standard: “System”).
Username	This is the username of the account that is being used to authenticate.
Password	The secret key associated with the user account.
Domain	This property is relevant when using Windows Authentication. The domain refers to the group of computers, users, and devices that are managed together and operate under the same set of rules in a network. Providing the domain name is essential for authentication when the user account is part of a Windows domain network.

Telegram

The operation is standard:

1. Open the plug-in by double-clicking on “Telegram Bot” in the “Plug-ins” menu.
2. Choose “Add” or double-click an existing entry.
3. Make the necessary settings (see section “Plug-in").
4. Confirm with “OK”.

You can now set up telegram bots.

Telegram Plug-in

You set the following properties in the configuration of the Telegram Bot plug-in:

Name	Assign a free unique name for your connection.

Part: Plug in

Bot access token	The Access Token is provided by the “BotFather”:1. Go to the Telegram program and select BotFather from the contacts. 1. Go to the Telegram program and select BotFather from the contacts. 2. Ask for a new bot with the command `/newbot`. 3. The BotFather will now ask you for a name for your bot. This name can be defined by you.. 4. The BotFather then asks for a username. This must always end with `_bot`. e. g. `Temperature_bot` 5. If the name is not yet occupied, the BotFather will send you a confirmation and the AccessToken, which you can copy into the properties at the Plug-in area. 6. Enter the other properties and confirm your entries with “OK”.
Maximum file size	The maximum file size indicates (in MB) how large received files may be before the transfer is aborted automatically.

The BotFather was made to administrate your Telegram bots. With the BotFather you can create, edit or delete bots. If you already created a bot, the BotFather can tell you the AccessToken. You can ask the BotFather for your AccessToken whenever you want. The account, which has created the bot, is the inalienable owner of the bot. This section is for information, you do not need to enter anything in this area.

Part: BotFather

Help text	If you want to show users automatic suggestions of possible commands, proceed as follows: 1. Send **“/setcommands”** by the botholder to the BotFather. 2. Send **@BotUsername** to the BotFather. 3. Copy the help text from the “BotFather” section in the Telegram plug-in of your OPC Router and send it to the BotFather. 4. Repeat the process after adding, deleting or changing telegram triggers.

Telegram Bots are always displayed publicly and can be accessed by any Telegram user. Internal authentication filters out unauthenticated users before triggers are initiated.

Users can be authenticated manually after writing to the bot (/start) or by authenticating themselves with a previously generated confirmation code.

Authentication method	Select the desired authentication method: • Authentication method “None”: Telegram bots are always public and everyone can send request to the bot. No further settings need to be made. • Authentication method “Intern”: The internal user authentication filters unauthorized users before they uses any trigger. Users can get authorized manually after a request (/start) or they can send an authorization code which was already generated. **Note:** Messages can only be sent to users who have already sent at least one message to the bot. Messages to be sent to all will be sent to all registered users (even those who are not authenticated). However, a maximum of 30 messages per second may be sent (Limit Telegram API).

Section: Authentication

Authentication method	• **None:** None authentication (not recommended) • **Internal:** user can be managed in the plug-in • **FAS Login:** Sign in via the FAS-Login

For Authentication method Internal

Section: Known users

Known users	List of known users. You can remove a user from the list using the “Delete user” button. You can use the “Deny permission” or “Allow permission” buttons to deny or allow permissions to known users.

Section Registration key

Registration key	Displays the known registration keys. Use the “Delete registration code” and “Create validation code” buttons to remove registration keys from the list or add new ones.

For authentication method FAS-Login

Section: Database

Provide the database instance where the FAS login database is deployed.

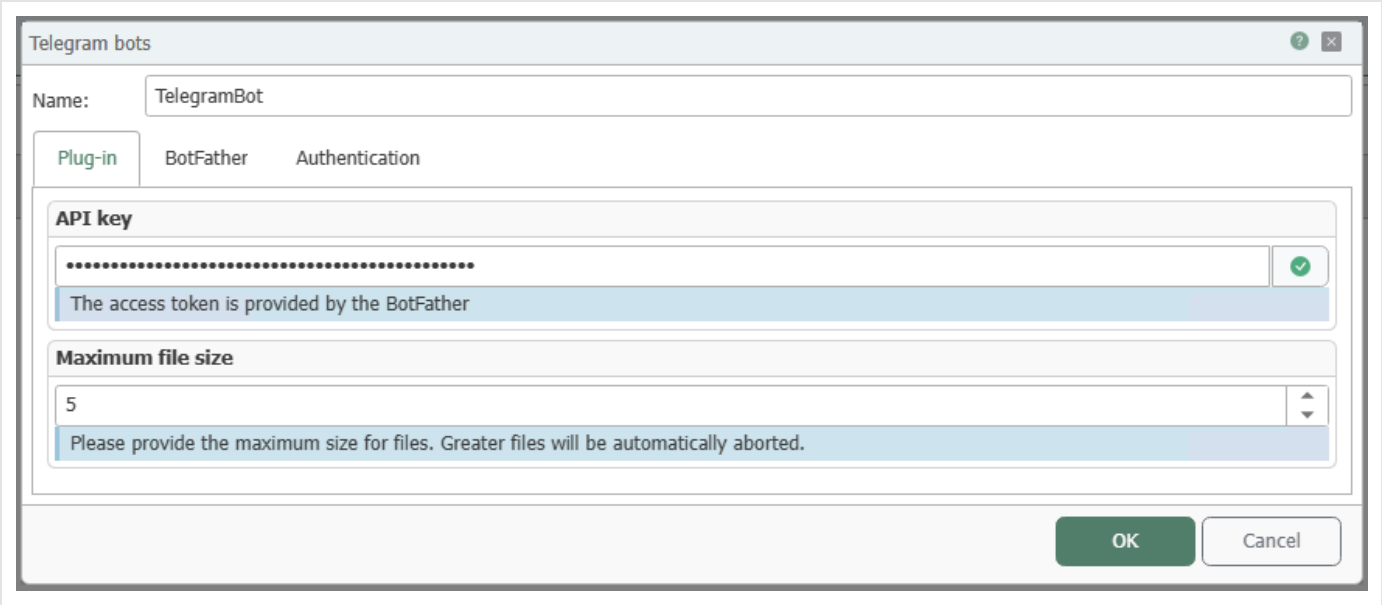
Section: E-mail

Provide the email instance which should be used to send the authorization emails.

Enter here the sender address.

Edit the E-mail text of the Registration email.

Confirm your entries with “OK”.



Telgram bots Plug-in

The following edit bots are helpful:

- /setname - change a bot's name
- /setdescription - change bot description
- /setabouttext - change bot about info
- /setuserpic - change bot profile photo
- /setcommands - change the list of commands
- /deletebot - delete a bot
- /setjoingroups - the bot may be added in groups
- /setprivacy - the bot may see/read messages in groups

Transfer Objects

Overview

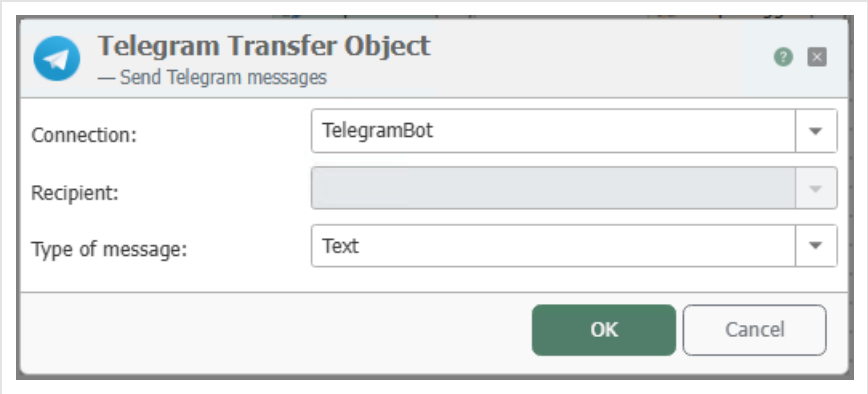
Telegram Transfer Object (worktitle)

Telegram Edit Transfer Object (worktitle)

Telegram Delete Transfer Object (worktitle)

Telegram Transfer Object (worktitle)

Create a telegram object within the connection as described.



Telegram Transfer object

Set the following properties after double-clicking the object:

Connection	Select your telegram connection from the drop-down menu.
Receiver	Select the receiver for receiving the telegram from the drop-down menu.
Message type	Select the message type with which the message of the telegram is to be transmitted, e.g. text message or voice message.

Telegram Edit Transfer Object (worktitle)

You can use the Telegram Edit transfer object to customize messages. To do this, specify the ChatID and the MessageID. Additionally, specify the text that should be displayed after the change.

Connection	Select your telegram connection from the drop-down menu.
Message type	Select the message type with which the message of the telegram is to be transmitted: • Text • Photo • Audio • Voice • Document

Telegram Delete Transfer Object (work title)

With the Delete transfer object you can delete messages. To do this, pass the ChatID and the MessageID. If the button in the keyboard is pressed, the message will be removed.

Connection	Select your telegram connection from the drop-down menu.

Printer

With the printer plug-in you can compile document templates within the OPC Router, which are filled with values during runtime and which can be printed with a printer installed on the local computer or in network.

The following printer plug-ins are available next to the Windows printer plug-in:

- Domino printer plug-in
- Plug-in for generic printers
- Wolke printer plug-in
- Zebra printer plug-in
- Videojet printer plug-in

The following steps are to be done:

- Create the document template in the plug-in configuration
- Selection of the printer and filling in the values to the template variables in the connection projection

Creating the document template

1. Open the corresponding printer plug-in (Domino, Zebra, Wolke, generic printer) in the area “Print templates” in the plug-in-menu.
2. Choose “Add new label” or double click on the existing headword, to change a template.
3. Enter a name for the print template and enter a comment if necessary. To submit a cloud printer label, you must also enter the label name.
4. Determine the data for the print template.
5. Confirm your entries with “OK”.

The document template can be used for projecting a connection now.

Domino and Zebra Printer Plug-in: The fields replace the strings in the {} before the label is sent to the printer. Either the default value from the table is specified or a connection e.g. read out via a script.

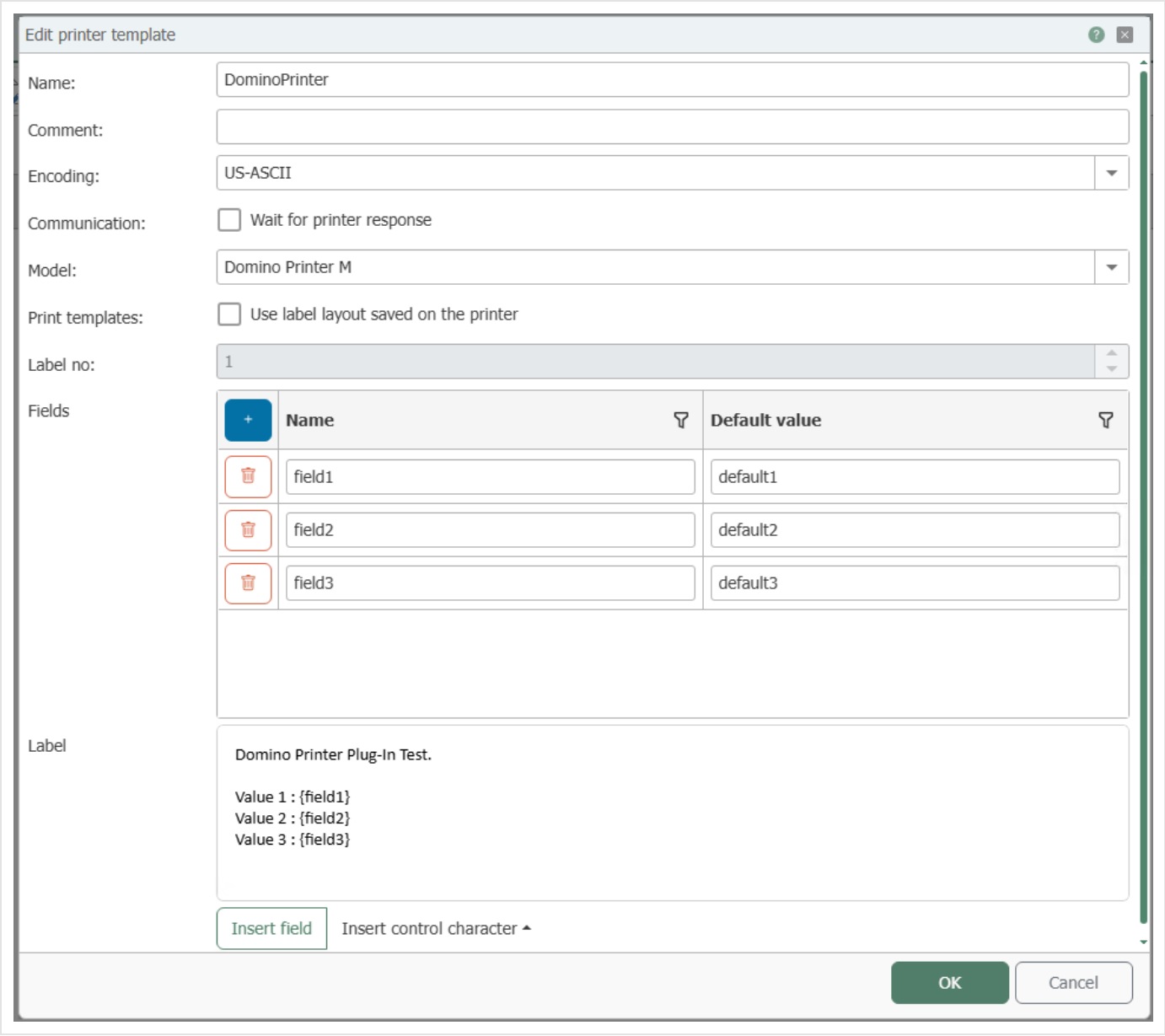
Plug-in generic printer: When configuring, either the default value from the table is specified or a connection is made. E.g. read out via a script

Plug-in Videojet printer: No print template is defined here, only the default values for various labels that can be sent to the print template available in the printer. In the “Videojet Update Printer” transfer object, you can select a label from the drop-down menu in the “Label” corresponding field.

Define printer templates

Template variables become dynamic or static at run-time with values, e.g. from the PLC or batch and order data from the database filled. At this point, you define the required placeholders and, if desired, assign default values.

1. Assign names in the table for the label fields. Simply click on the next free line. To delete fields, go to the corresponding line and right-click to delete a line.
2. If necessary, give your label fields a default value.
3. Define your label printing by the fields in the print template. Here you have the option to drag & drop individual fields into the print template. You can also right-click the boxes and select “Add label field” or select the field, place your cursor in the correct place in your print template and click the “Insert Field” button.
4. Confirm your entry with “OK”.



Domino Printer Plug-In - Editing printer template

To delete a print template, select it in the list and go to the button “Delete Label” at the top.

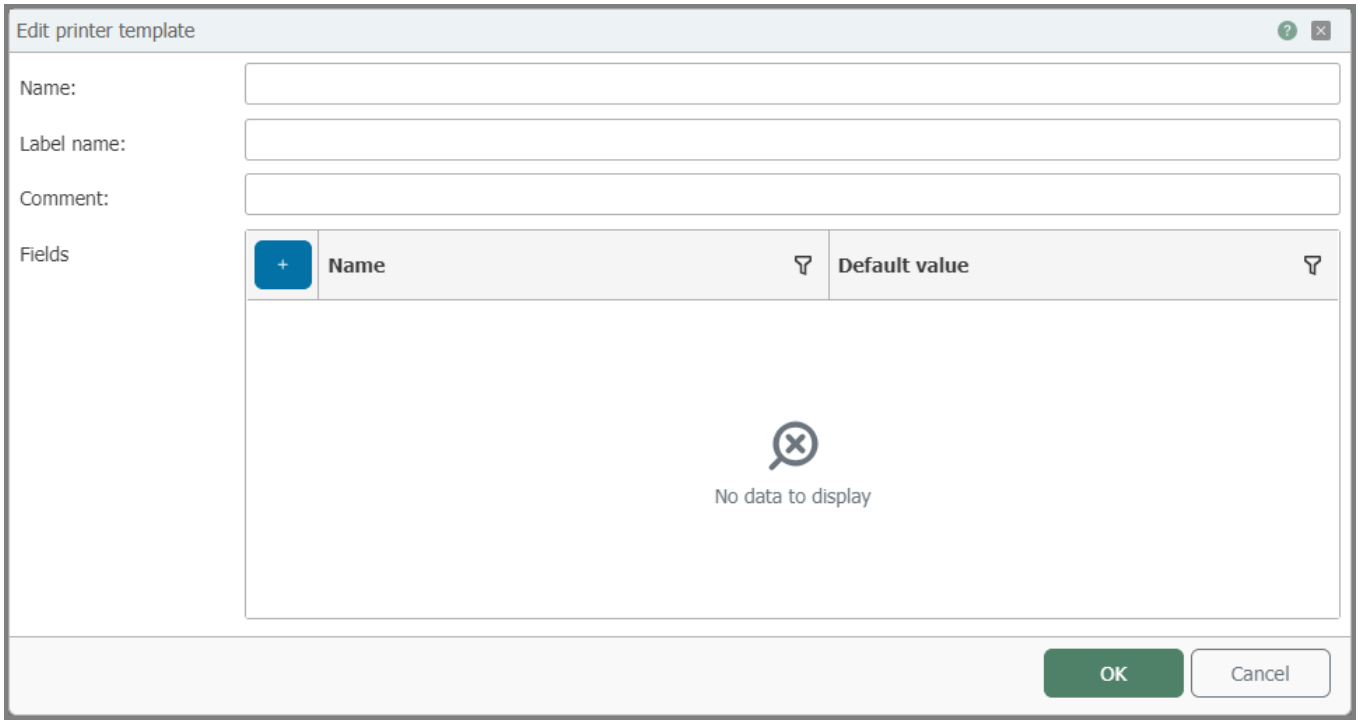
In order to give a new print name a new name, mark it in the list and click on the “Rename Label” button at the top.

With button “Insert control character” you can also insert control characters for Domino-, Zebra and generic printers in the label description. The control characters are highlighted in black.

Transfer object

The print starts with triggering. The template is set up in the transfer object; printer and variable content are to be set up during runtime. Create a printer transfer object within the connection as explained.

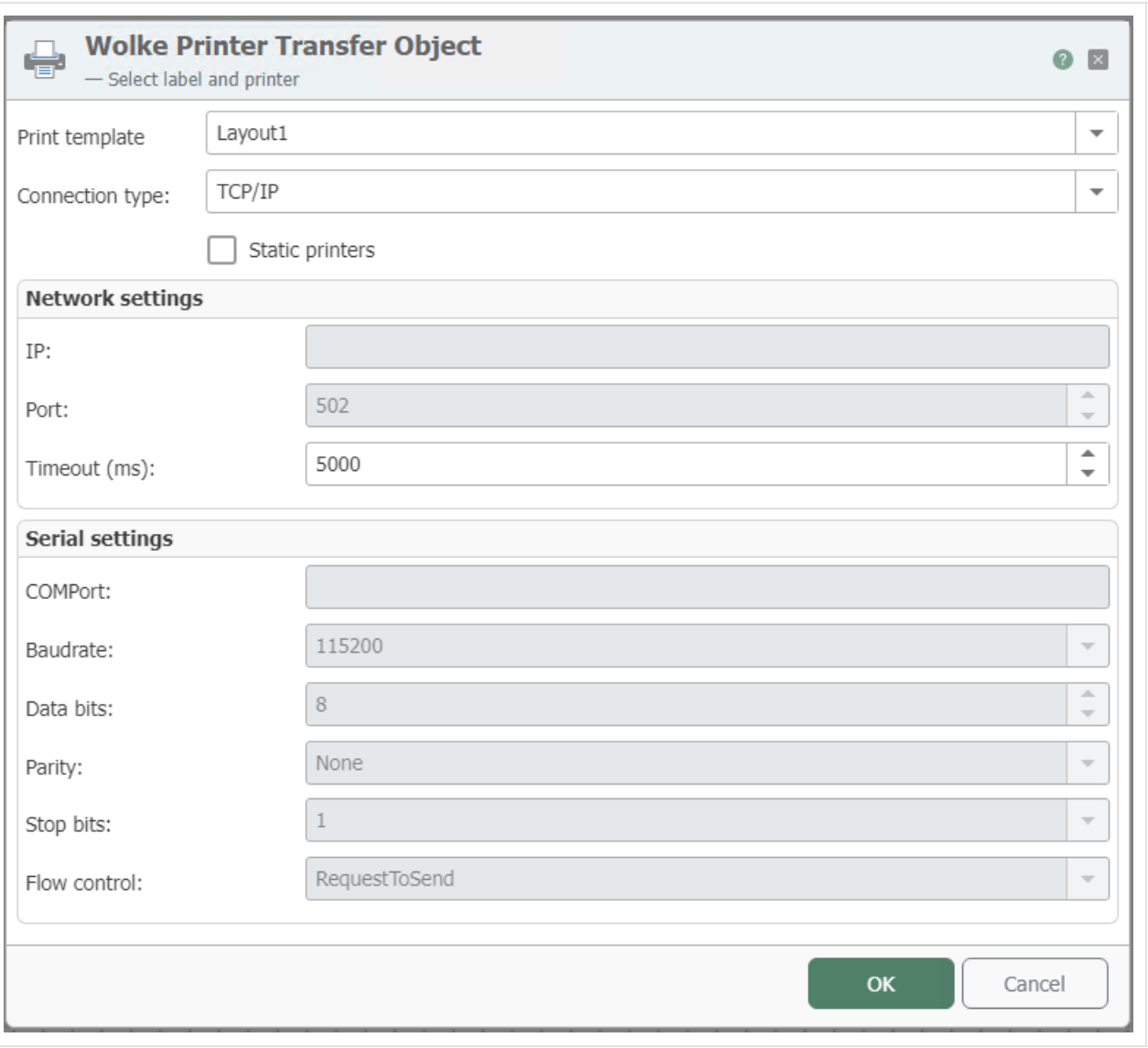
Wolke Printer Plug-in



Name	Name of the print template.
Label name	Name of the label. **Note:** The label name must match the label name assigned by the printer.
Comment	Assign a comment to the layout.
Name	Assign a name to the field. **Note:** The field name must match the field name in the printer. Incorrectly written field names will be ignored by the printer.
Default value	Assign a default value to the field.

Wolke printer transfer object

After double-clicking on the Wolke printer transfer object, set the following properties:



Wolke Drucker Transfer object

Print template	Print template created in the plug-in configuration.
Static printer	Activate the checkbox if you always want to control the same printer. Enter the IP and port for this. If the checkbox is not activated and you want to control a dynamic printer, then the printer name must be defined at runtime (from a constant, variable, database or similar)
IP	Enter the IP address of your static printer.
Port	Enter the port of your static printer.
Timeout	Time (in ms) that the connection establishment may take before a connection error is returned

The transfer object provides you with all placeholders from the document template. If these template variables are not filled in the transfer object, the router uses the default values stored in the template.

Note: You can also enter an external host in the IP address field. To do this, create a constant with the IP of the printer, for example, and drag an arrow to the IP address field.

Windows Printer Plug-in

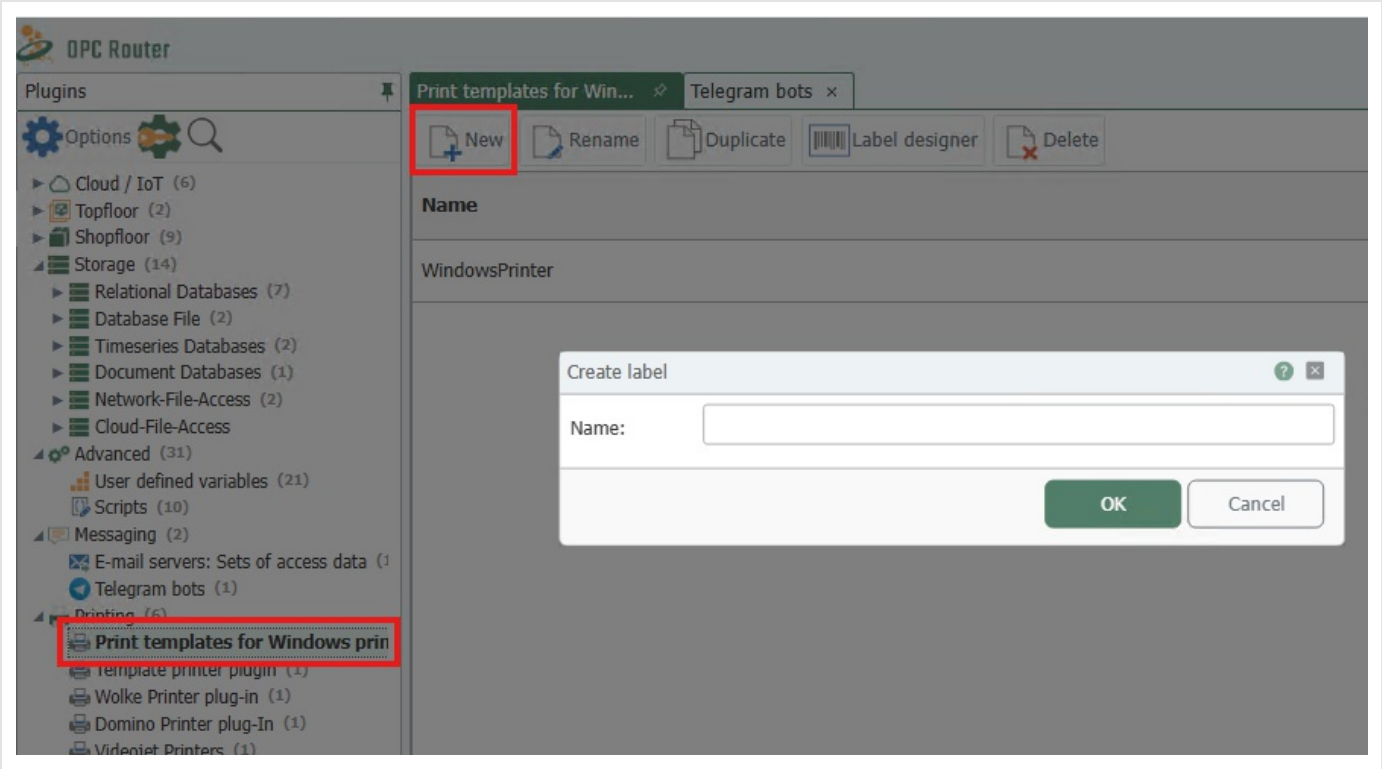
With the printer plug-in you can compile document templates within the OPC Router, which are filled with values during run time and which can be printed with a printer installed on the local compter or in network. The following steps are to be done: ,

- Create the document template in the plug-in configuration
- Selection of the printer and filling in the values to the template variables in the connection projection

Creating the document template

1. Open the plug-in via double cklicking on “Print templates” at the plug-in-menu.
2. Choose “Add document template” or double click on the existing headword, to change a template.
3. Fill in a name for the document template.
4. Create the template with the integrated designer (see part “Designer”).
5. Close the editor with the button “Close Window” on the windows title bar. Confirm the save request.

The document template can be used for projecting a connection now.



Windows - Create label

Interactive Printers

The OPC Router runs ? as Windows application non stop in the background of the server. An interaction with the user is not planed. That means, that printers, which need user feed (e. g. PDF output, XPS output with “Save As... -Dialouge” can´t be used: In this case the user would have to make the feed with every print initialised by the router.

Depending on the Windows version state displays of the printer (status of ink/ toner) react differently as well, when called by a service. Test your projecting carefully. If necessary allow interaction between service and desktop in the service administration, to make user feeds possible. This is explicit recommended just for testing, not for producing:

1. Open the service administration with “System Control - Administration - Services”.
2. Open the OPC Router via double click.
3. Activate the property “ Allow Data Transfer Between Service And Desktop” on the tab “Log In”.
4. Confirm with “OK”.

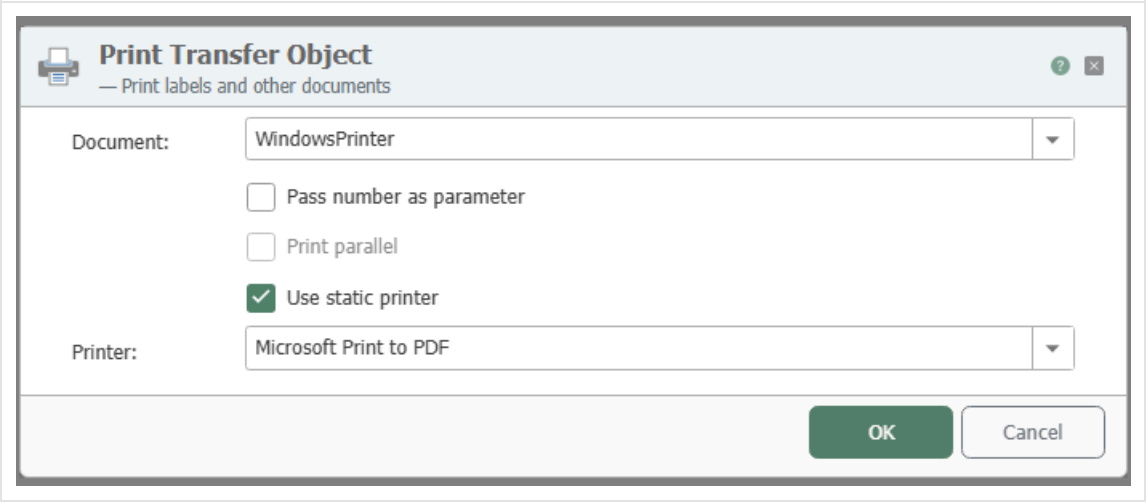
5. Restart the service.

If you should test your label design e. g. with Windows XPS Writer, you shouldn´t save the XPS data in the suggested directory, but choose one directly on the root (e. g. C:\Label Testing), this avoids access conflicts.

Transfer object

Windows printer

Set the following properties after double clicking the transfer object for generic printer:\



Windows Printer Transfer Object

Document	Document template, created in the Plug in Configuration
Specify number of copies	It can now be set to request multiple copies for one-sided documents.Only 1 (!) print job is sent for this! Whether the document is printed single-sided or double-sided must be specified in the printer driver. So labels can be printed on the front and back side if necessary.
Static printer	- activated: The printer is set up during creating time and is not be changed during run time. • activated: The printer is set up during creating time and is not be changed during run time. • deactivated: The printers name has to be defined during run time (from variable, data base or similar).
Parallel printing	Activate this option if you want to print on different printers parallel.
Printer	Enter the port of your static printer.

The transfer object gives you all placeholders from the document template. Are these template variables not be filled in the transfer object, the router uses the standard value from the template.

Should you call the printer dynamically, please pay attention to the correct spelling of the printers name- it has to be equal to name given in Windows.

Designer

You will find the Template Designer at the top of the printer plug-in menu next to the buttons for creating and editing new labels with the template designer

- you define the size of a document,
- you define the template variables with standard values,
- you place the elements “Text”, “Image”, “Rectangle” and “Barcode”. Elements from the type “Text” and “Barcode” can be seized with template variables.

Define document size

You can define the document size in the menu at “Settings - Document Size” or select the sizes available on the printer of your choice.. The second option is especially useful, when you allready know what printer you`ll use. On the other hand, you don´t fix on a printer by choosing the second option, to e. g. set up a DIN standard with Microsofts XPS printer. Stating the size is in millimeters.

1. So choose the option “User Defined Size” or “Supported Size”.
    - **User-defined size:** Fill in the document size.
    - **Supported size:** Choose a printer and document size.
2. Confirm with “OK”.
3. If necessary, select “Landscape” by “Settings”.

Define Template Variables

Template variables are filled during run time, e. g. from the controlling or batch- and task data from the data base. At this point, you define the needed placeholder and state standard values, if wanted. The values can be printed as text or barcodes.

1. Choose the button “New placeholder”.
2. Assign a name and confirm with “OK”.
3. The template variable is shown in the list. Assign a standard value, if wanted.
4. Repeat the steps 1 to 3 for all needed placeholders.
5. If you want to remove a variable, click on the name in the list and delete or rename it.

Place Elements

From the following elements can be chosen:

- Text: Free text and placeholder in combination. Size and position of the frame, font -type, -size and -colour are to be setted.
- Rectangle: A frame of free size and position. Frame colour, -type and -width are to be setted as well as the fill in colour.
- Barcode: Free definable Barcode. Size and position, encryption, content (text, numbers and placeholders) are to be setted. The alignment can also be configured for a barcode and/or a barcode text.
- Image: Image data in free position and size, to be chosen from the data system.
- Choose the button “New Element” and the type in the opening list.
- The element will be placed in the upper left corner. Slide it to the desired position with the mouse or state the coordinates exactly in the property menu or simply by drag and drop.
- Define the size of the element. To do this, click on the element frame and drag it to the desired size, or state the size exactly in the property menu.
- Define the other properties.

Tip: With “Insert placeholder” (Text and Barcode) you fill in a defined template variable to the relevant place. Don´t change the syntax.

Tip: In the colour definition you can assign besides the value for red, green and blue, the value “alpha”, to set up lucency (0 =

invisible, 255 = not translucent, for text and frames).

Test print, save and close

- Use the button “Print (test)”, to print a test on a configured printer.
- A memory query appears if you want to close your input window via the window button.
- Use the button “Close”, to close the designer and go back to the router configuration.

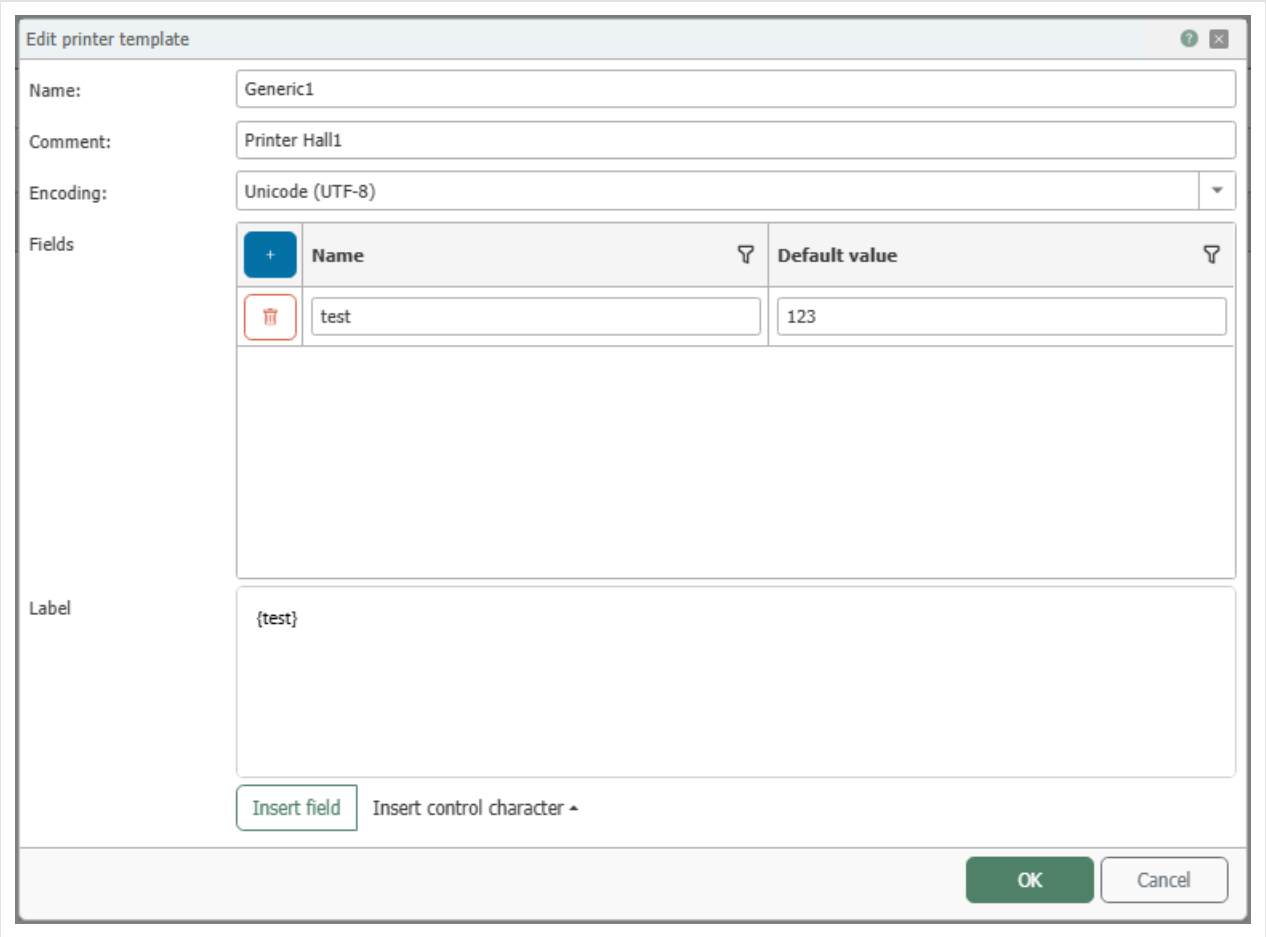
Changing existing templates

Via double clicking an entry in the plug-in configuration you can change the chosen template at all times.

Attention: Don´t change variable names and don´t delete template variables, which allready are used in the connection projection. The router can´t use placeholders manipulated like that and gives out a defective label (e. g. with standard value).

Generic Printer Plug-in

To define a print template, create a new plug-in for the generic printer.



Generic Printer Plug-in

Name	Assign a name for the print template.
Comment	Write a comment for the print template.
Encoding	Set the encoding for the print template.

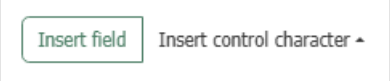
Fields

Name	Assign a name for the field.
Default value	Set the value of the field.

Label

To add a field to the label, click “Insert field”.

If you want to add a new field, click on “Insert field” and then on “Add new field”.



You can add control characters with “Insert control character”.

Transfer object

Generic printer

Set the following properties after double clicking the transfer object for generic printer:

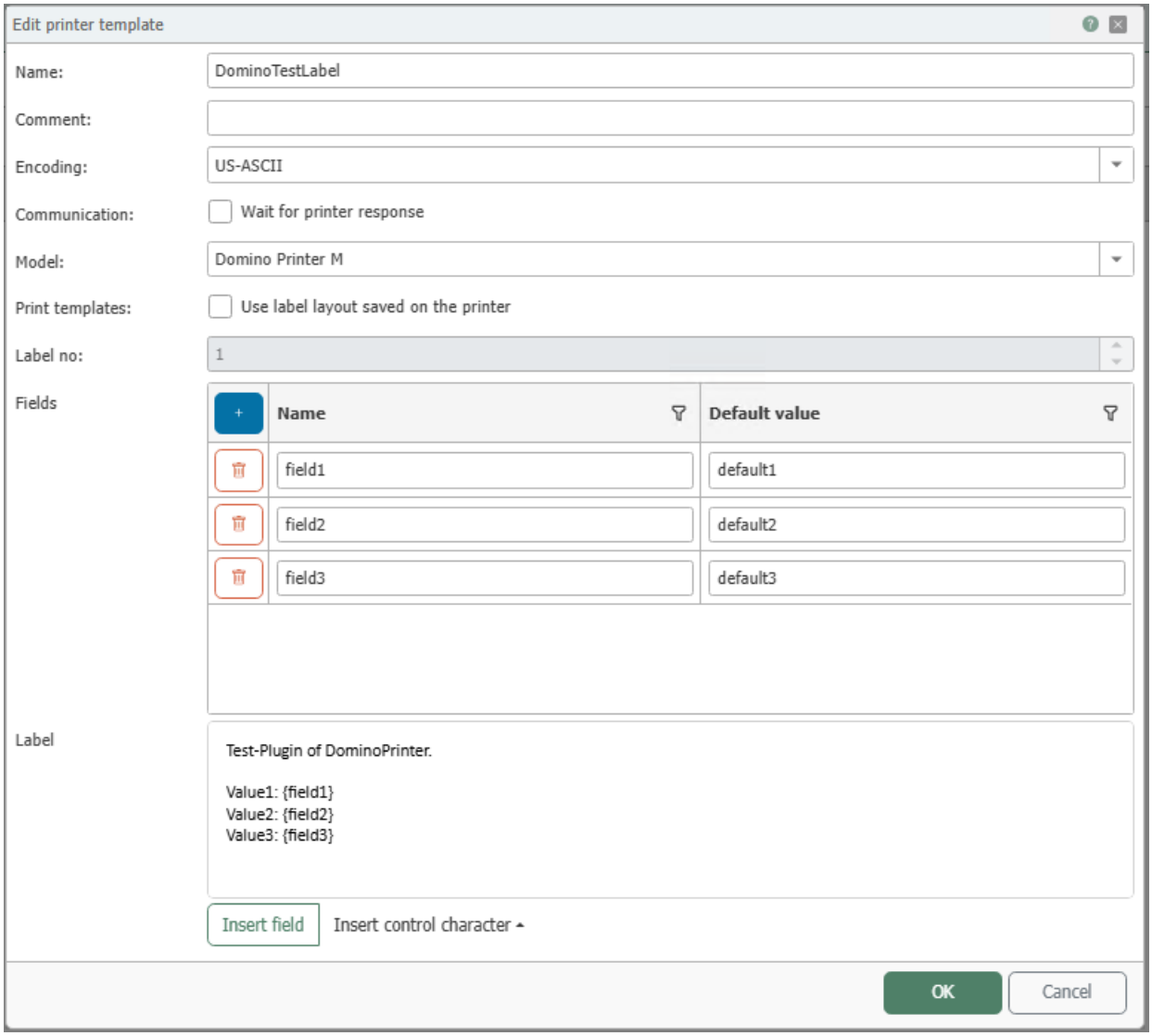
Label	Document template, created in the Plug-in Configuration
Static printer selection	• activated: The printer is set up during creating time and will not be changed during run time. • deactivated: The printers name has to be defined during runtime (from variable, data base or similar).
IP	Enter the IP address of your static printer.
Port	Enter the port of your static printer.
Timeout	Time (in ms) that the connection setup may take before a connection error is returned.

The transfer object provides all placeholders from the document template. If these template variables are not filled in the transfer object, the OPC Router uses the default values stored in the template.

If you bind the printer dynamically, make sure to enter it's exact name as it is used in Windows too.

Note: You can also enter a host from outside into the IP address field. To do this, create a constant with the IP of the printer, for example, and drag an arrow to the IP address field.

Domino Printer Plug-in



Domino Printer Plug-In

Name	Name of the print template
Comment	Enter a comment here.
Encoding	Choose between the different encodings here. Note: The encoding of the OPC Router should match the encoding of the printer. Please contact the manufacturer if you have questions about the printer encoding.
Wait for Response from the printer	Please do not check this box if you have a Domino printer model that does not send a response that printing was successful.
Model	Choose between the following models: Domino Printer A Domino Printer M The difference lies in the communication protocols. If you have any questions about the protocols, please contact the manufacturer.

Use label layout saved on the printer	Check this box if you have already created a layout for the printer.
Label no.	If you have already designed a layout, enter the label no. of the layout here.

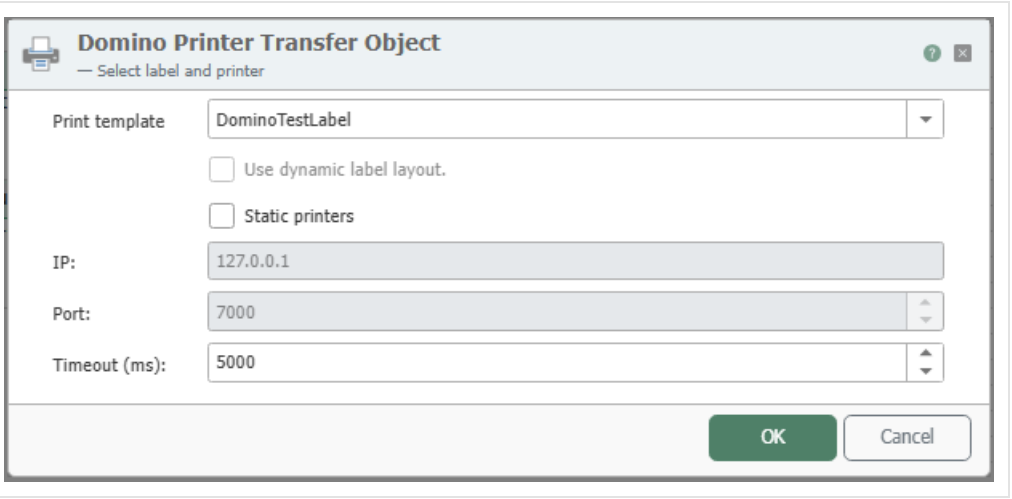
Fields

Name	Name of the field
Default value	Enter the default value for the field here.
Field length	Specify the length of the field here.

Label

Uncheck “Use label layout saved on printer” if you do not have a layout yet. Enter the appropriate commands for the desired layout here if you have not yet created one. Use the “Control characters” button to add new control characters to the label. This The commands in the `<>` are converted to special characters.

Printer Transfer object



Domino Printer Transfer Object

Set the following properties after double clicking the transfer object for Domino printer:

Label	Document template, created in the Plug in Configuration
Static printer selection	• activated: The printer is set up during creating time and is not be changed during run time. • deactivated: The printers name has to be defined during run time (from variable, data base or similar).
IP (available if static printer is checked)	Enter the IP address of your static printer.
Port (available if static printer is checked	Enter the port of your static printer.
Timeout	Time (in ms) that the connection setup may take before a connection error is returned.

The transfer object gives you all placeholders from the document template. Are these template variables not be filled in the transfer object, the router uses the standard value from the template.

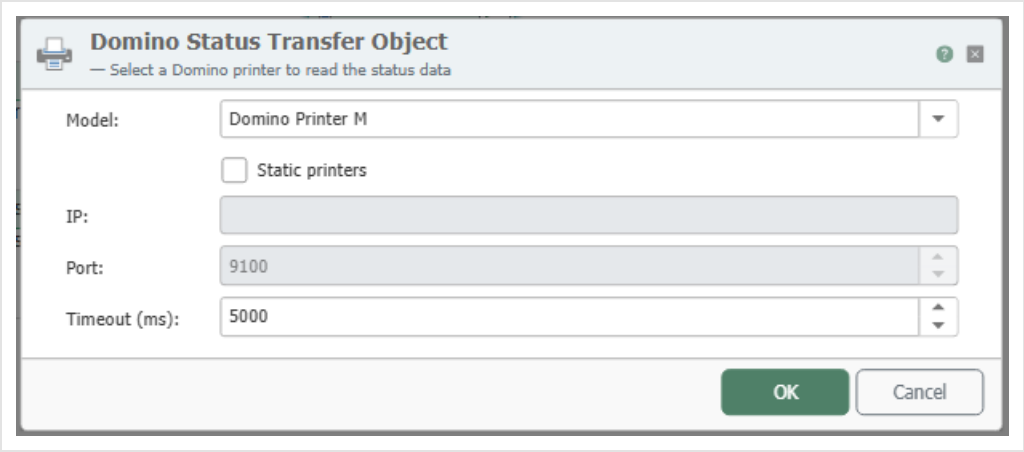
Should you call the printer dynamically, please pay attention to the correct spelling of the printers name- it has to be equal to name given in Windows.

Status Transfer object

Note: You can also enter a host from outside into the IP address field. To do this, create a constant with the IP of the printer, for example, and drag an arrow to the IP address field.

Domino status

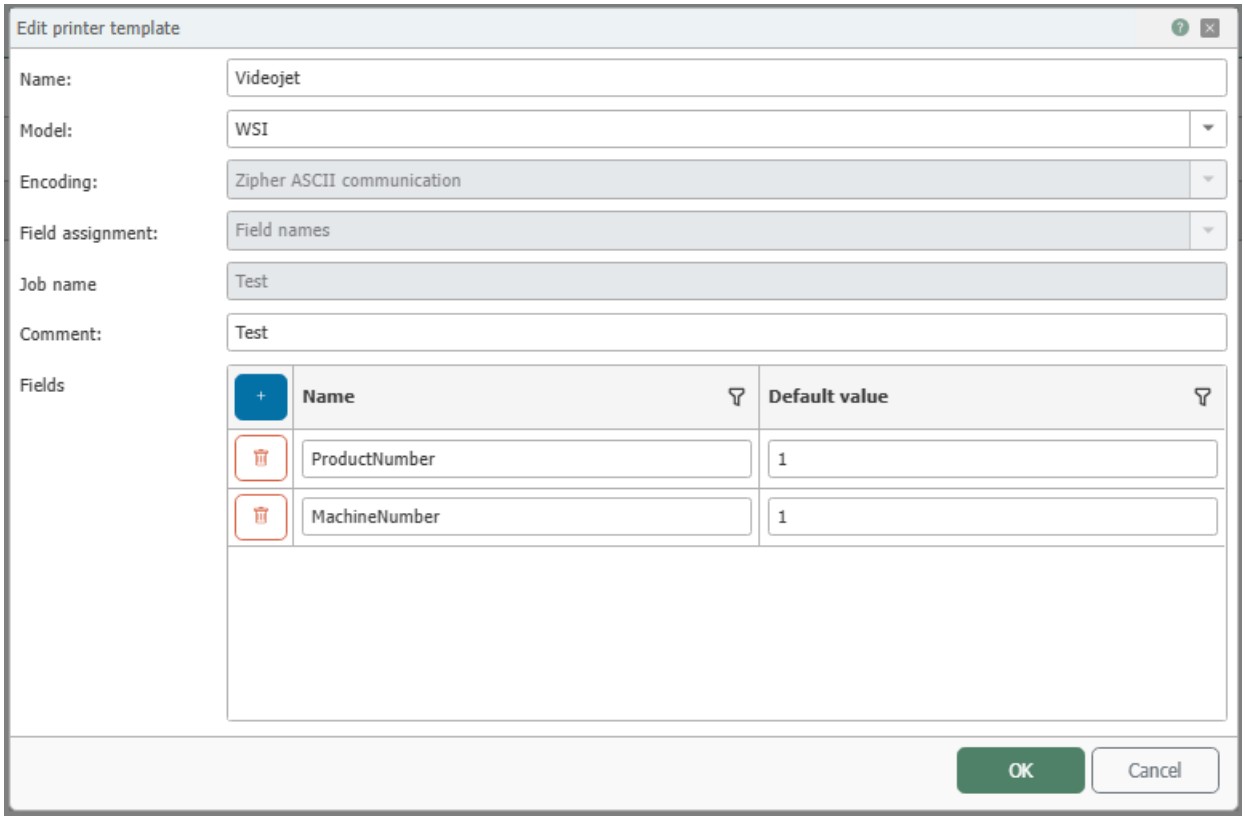
To query the status of the printer you can set the Domino status transfer object.



Domino Status Transfer Object

Model	Select the printer to query the status.
Static printer	Set this checkbox if you want to define a printer directly in the transfer object.
IP	Enter the IP of the Domino printer here. **Note:** Only available if Static printer is checked.
Port	Enter the port of the printer here. **Note:** Only available if Static printer is checked.
Timeout	Define here from when a timeout should be set if the printer is not reachable.

Videojet Printer Plug-in



Videojet Printer Plug-in

Name	Name of the template
Model	Select one of these models: • WSI • Zipher
Encoding	Following allocations are possible: • Zipher ASCII communication • Zipher Unicode communication **Note:** Only available if you select Zipher as model.
Field assignment	Following allocations are possible: • No field names • Field names • Line and field names **Note:** Only available if you select Zipher as model.
Job name	Enter her the name of the job **Note:** Only available if you select Zipher as model.

Comment	Enter a comment for the template.

Fields

Name	Name of the field
Default value	Default value for the field

Transfer Objects

Overview of the transfer objects:

Videojet Status Transfer Object

Videojet Printer Transfer Object

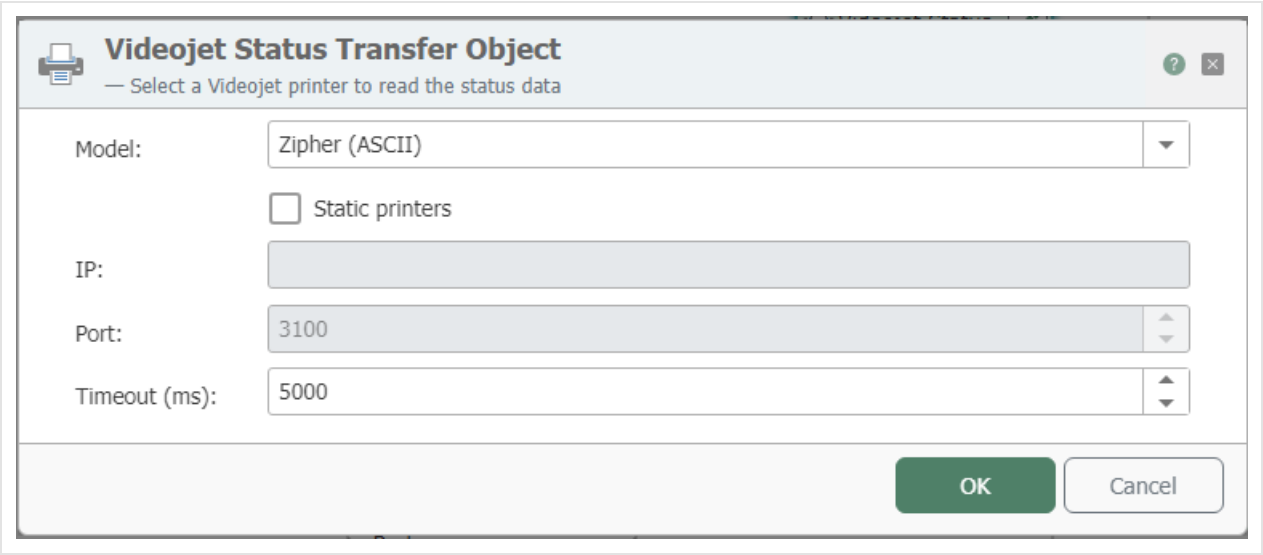
Videojet Message Printer Transfer Object

Videojet Update Printer Transfer Object

Videojet Status Transfer Object

Videojet Status

You can read out the status of the printer via the status transfer object.\



Videojet Status Transfer Object

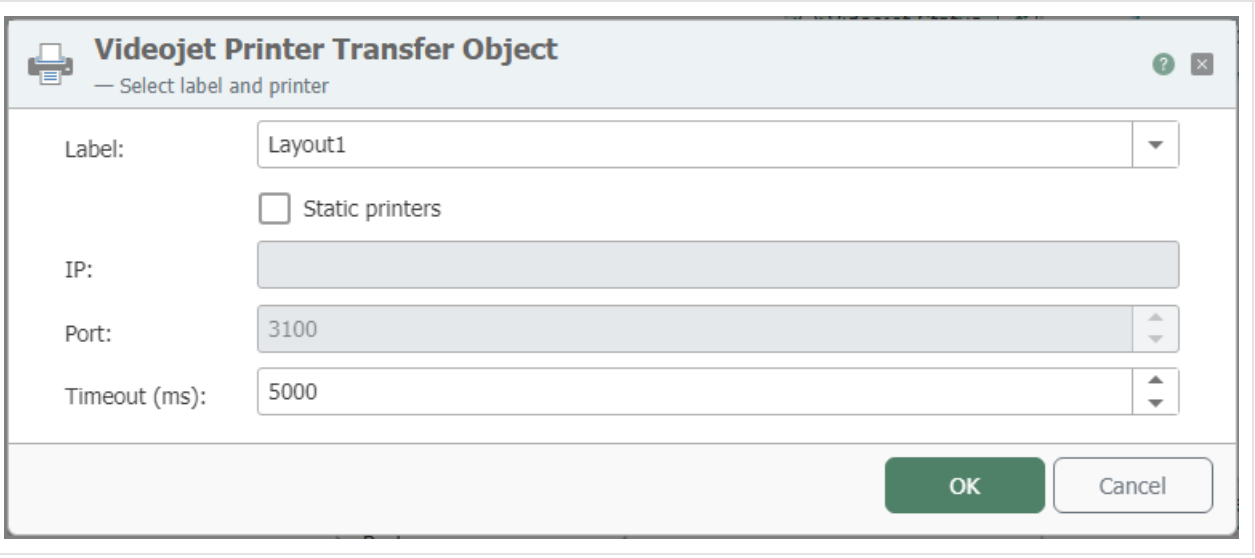
SETTINGS NAME	DESCRIPTION
Model	Select the printer model here.
Static printer	• **activated:** The printer controlled will always be the same, for this enter the IP and the Port. • **deactivated:** The printer is dynamic, the name of the printer will have to be defined during the runtime (by a constant, a variable, a database or other.)
IP	Define the static address of the printer here. **Note:** This field is only available if you have selected the "Static printers" field.
Port	Define the static port of the printer here. **Note:** This field is only available if you have selected the "Static printers" field.
Timeout	Amount of time in milliseconds after which the printer will be considered unreachable.

Videojet Printer Transfer Object

Videojet Printer

The printer transfer object can be used to select a print template that is defined in the plug-in. You can connect the printer via a static IP address.

Note: You can also enter a host from outside into the IP address field. To do this, create a constant with the IP of the printer, for example, and drag down an arrow to the IP address field.



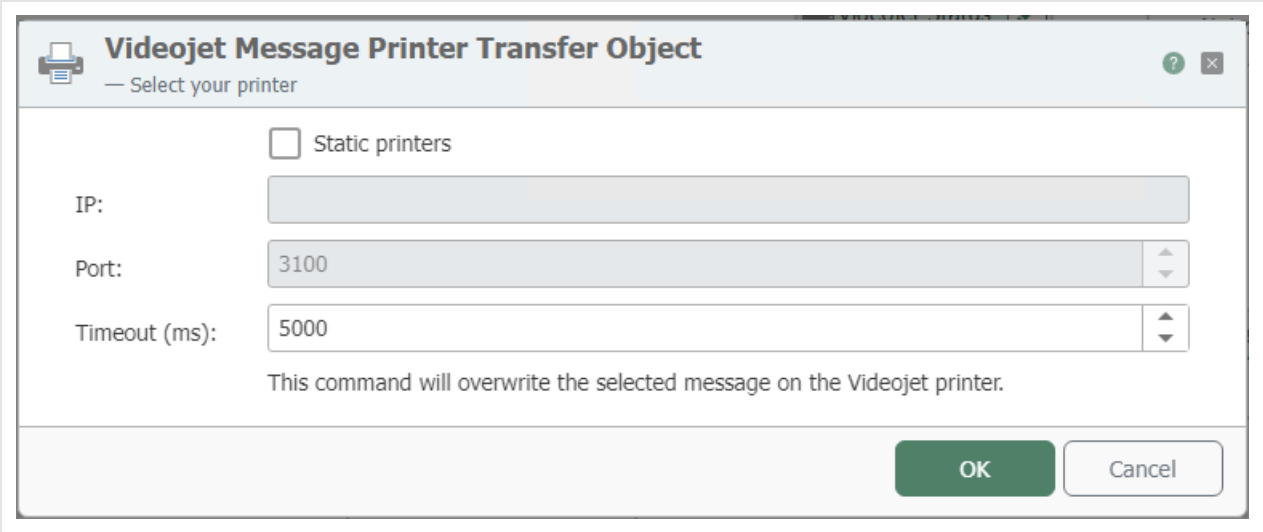
Videojet Printer Transfer Object

SETTINGS NAME	DESCRIPTION
Label	Select the label here.
Static printers	• **activated:** The printer controlled will always be the same, for this enter the IP and the Port. • **deactivated:** The printer is dynamic, the name of the printer will have to be defined during the runtime (by a constant, a variable, a database or other.)
IP	Enter the IP of the Videojet Printer here. **Note:** This field is only available if you have selected the "Static printers" field.
Port	Enter the port of the Videojet printer here. **Note**: This field is only available if you have selected the "Static printers" field.
Timeout	Amount of time in milliseconds after which the printer will be considered unreachable.

Videojet Message Printer Transfer Object

Videojet Message Printer

The Videojet Message Printer transfer object overwrites the currently selected Message and refills it with the transferred text elements.



Videojet Message Printer Transfer Object

After double-clicking the Videojet Message Printer Transfer Object, set the following properties:

SETTINGS NAME	DESCRIPTION
Static printer	• **activated:** The printer controlled will always be the same, for this enter the IP and the Port. • **deactivated:** The printer is dynamic, the name of the printer will have to be defined during the runtime (by a constant, a variable, a database or other.)
IP	Enter the IP address of your static printer. **Note:** This field is only available if you have selected the "Static printers" field.
Port	Enter the port of your static printer. **Note:** This field is only available if you have selected the "Static printers" field.
Timeout	Amount of time in milliseconds after which the printer will be considered unreachable.

This command will overwrite the selected message on the Videojet printer.

In this transfer object, you can enter values corresponding to the predefined fields :

- **FONT NUM** to specifiy the font size of the text element.
- **HORC** for horizontal alignement of the text element.
- **VERC** for vertical alignement of the text element.
- **ATTRIB** for selecting various printer-specific attributes.
- **MESSAGE TEXT** to enter your freely selectable text on the label.
- Label = Name of the label to be overwritten.

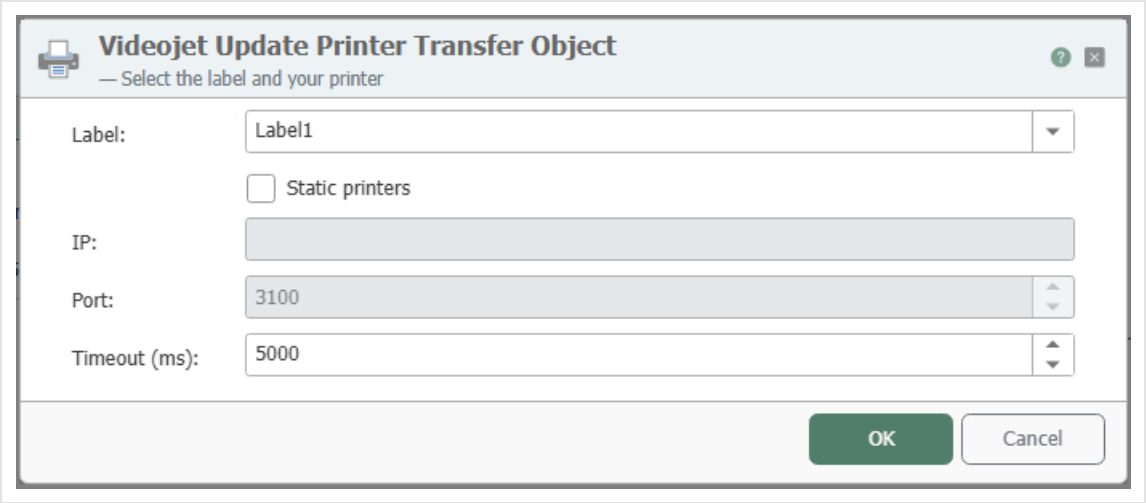
How the individual entries must be made is described in the documentation of your Videojet printer in the **MESSAGE DATA** Field

Format section. \ \ Array input also allows you to create several text elements.

Videojet Update Printer Transfer Object

Videojet Update Printer

The Videojet Update Printer transfer object selects an existing label in the printer and fills the specified printer variables with values.



Videojet Update Printer Transfer Object

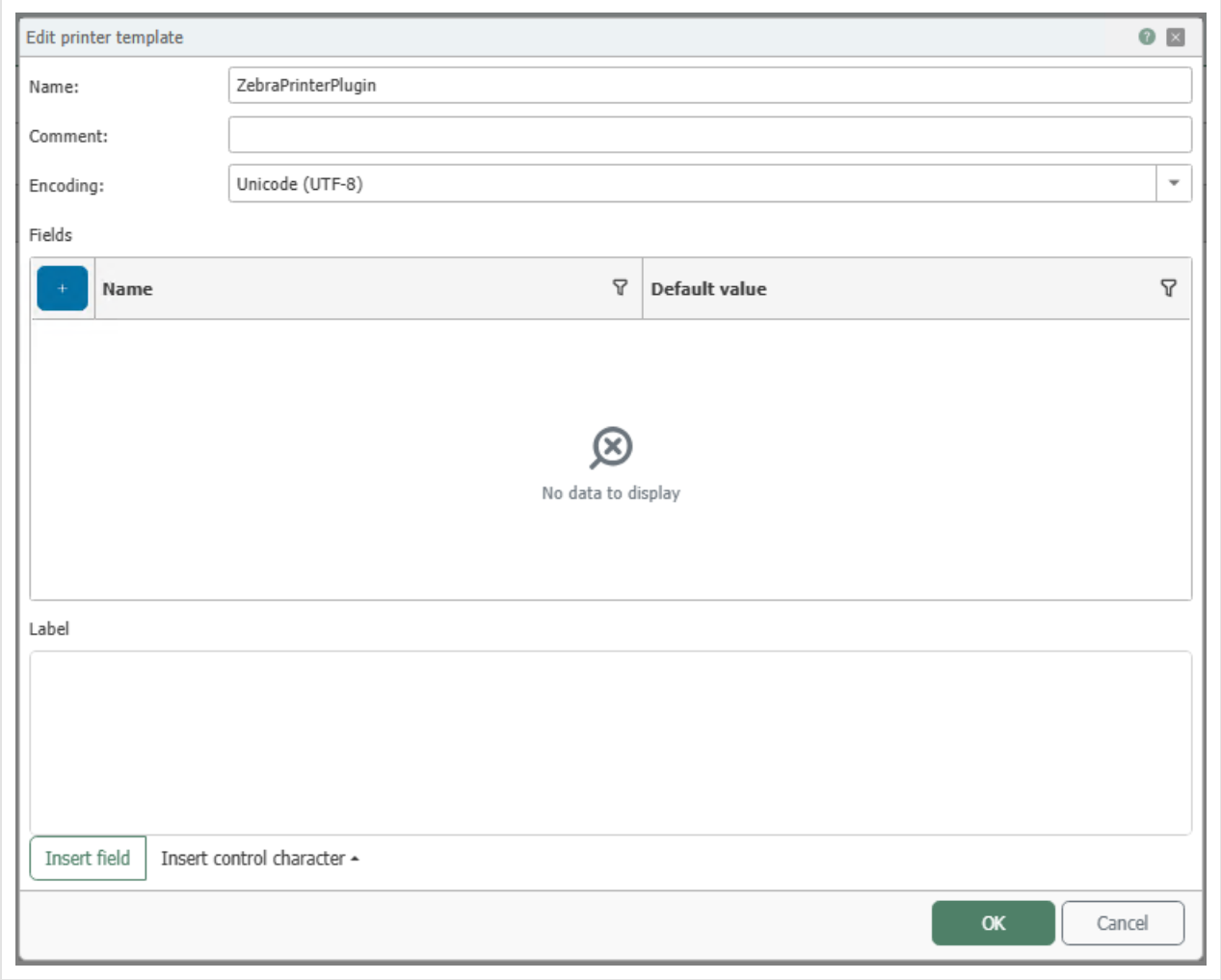
After double-clicking the Videojet Update Printer transfer Object, set the following properties:

SETTINGS NAME	DESCRIPTION
Label	Select a label from the drop-down menu. If you do not assign any values to the transfer object, the default values that you specified for this label in the Videojet Printer plug-in will be used.
Static printers	• **activated:** The printer controlled will always be the same, for this enter the IP and the Port. • **deactivated:** The printer is dynamic, the name of the printer will have to be defined during the runtime (by a constant, a variable a database or others.)
IP	Enter the IP address of your static printer. **Note:** This field is only available if you have selected the "Static printers" field.
Port	Enter the port of your static printer. **Note:** This field is only available if you have selected the "Static printers" field.
Timeout	Amount of time in milliseconds after which the printer will be considered unreachable.

You can assign values to the fields of the transfer object and trigger the printer using the “Label” area.

Zebra Printer Plug-in

Zebra Printer Plug-in

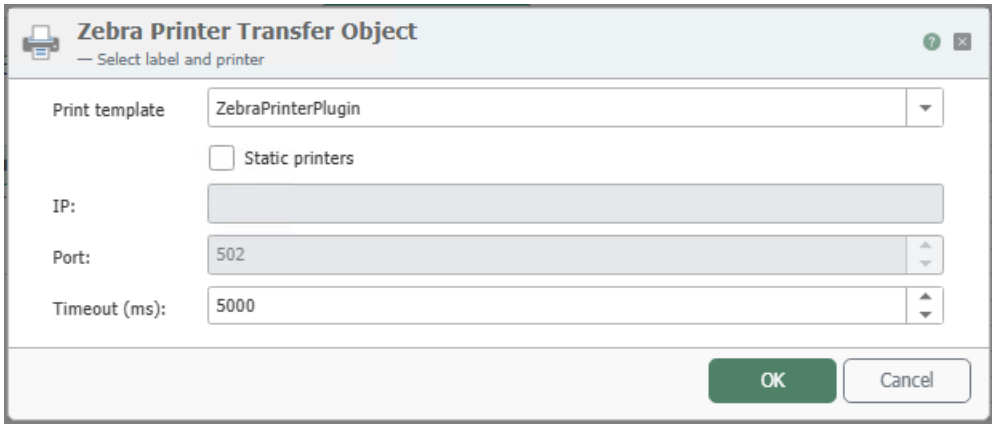


Zebra Printer Plug-in

Name	Assign a name for the template.
Comment	Enter a comment.
Encoding	Select the encoding. **Note:** The selected encoding must match that of the printer. Please contact the manufacturer of the printer for more information.
Name	Assign a name for the field.
Default value	Assign a default value for the field.
Label	The commands for the label are displayed here.
Insert field	Add new fields.

Insert control characters	Add commands which will be displayed in the label.

Printer transfer object



Zebra Printer Transfer Object

Set the following properties after double clicking the transfer object for Zebra printer:

Label	Document template, created in the Plug-in Configuration
Static printer selection	• activated: The printer is set up during creating time and is not be changed during run time. • deactivated: The printers name has to be defined during run time (from variable, data base or similar).
IP	Enter the IP address of your static printer.
Port	Enter the port of your static printer
Timeout	Time (in ms) that the connection setup may take before a connection error is returned.

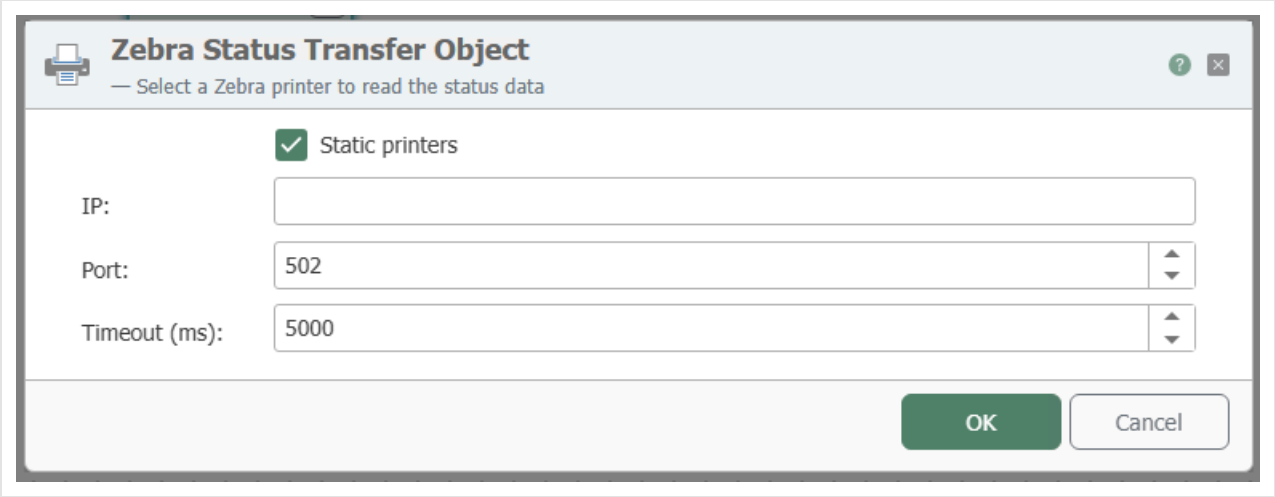
The transfer object gives you all placeholders from the document template. Are these template variables not be filled in the transfer object, the router uses the standard value from the template.

Should you call the printer dynamically, please pay attention to the correct spelling of the printers name- it has to be equal to name given in Windows.

Zebra Transfer Objects

Note: You can also enter a host from outside into the IP address field. To do this, create a constant with the IP of the printer, for example, and drag an arrow to the IP address field.

The transfer object can be used to read various printer states such as the state of the connection or the presence of the paper roll.



Zebra Status Transfer Object

Set the following properties after double clicking the transfer object:

Static printers	May be checked if the statuses of a specific printer are to be retrieved.
IP	IP address of the printer
Port	Port of the printer
Timeout	“If the printer cannot be reached within the specified time period, an error will be returned.”

IP and port are only available if “static printer” is checked.

Additional interfaces

Overview over additional interfaces:

XML/XPath Transfer Object

CSV Transfer Object

Excel Transfer Object

JSON Transfer Objects

File Objects

XML/XPath

XML (Extensible Markup Language) has been the stadard for transmitting structured data of the Internet for many years. XML is therefore often used as an output or input for parameters in systems.\ XPath (XML Path Language) is the language that enables easy access to the data in the XML package so that the OPC Router can extract and pass on the relevant information.

The XML/XPath functionnality is already included in the basic version of the OPC Router.

xml_xpath_transfer_object.md

xml-read-data.md

xml-create-data.md

XML/XPath Transfer Object

XML transfer objects are available as sources and destinations. XPath queries allow elements to be extracted from XML files or written to XML files. A plug-in configuration is not necessary, you can directly configure the transfer object within the connection as described.

After double-clicking on the transfer object, set the following properties:

Section: X-Path Elements

Name	Name where the query appears in the transfer object.
X-Path Expression	Valid X-Path query, in the simplest case the path to the node (see below).
XML Mode	Indicates if only the content of the query result or the enclosing tag will be output (e.g. `Heinrich Mann` or `<author>Heinrich Mann</author>`. When writing to the object (Create XML), only content must always be specified because the structure is already established. By writing at the object (create XML), only content must always be specified because the structure is already established.

X-Path Placeholder

Name of parameter	Name of the parameter
Default value	Default value of the placeholder

Section: Namespaces and Prefixes

Prefixes	Unique prefix
Namespace	Namespace

When importing the XML code at design time, all existing namespaces are entered. Use the checkbox to mark the namespace that should be used as the default for the XPath queries. If a standard is defined, formulate the X-Path queries without a prefix, this will be added automatically. In the source code, you have to specify the prefixes always,.

Section: XML Code

	The multiline field is for entering XML code. At design time you can use it to check your X-Path expressions. Then you can delete the code if you dynamically pass the XML code at runtime (transfer object as source) or first generate (transfer object as destination). Of course, you can also leave static code here if that meets your requirement.

XML Read Data

XML documents as source

As a transfer source, the XML object extracts content from XML code based on the configured X-Path queries.

- The XML code is either set statically at design time or submit at run time to the element “XML”.
- The results are read from the X-Path query elements.

For example, the following code is supplied from a file:

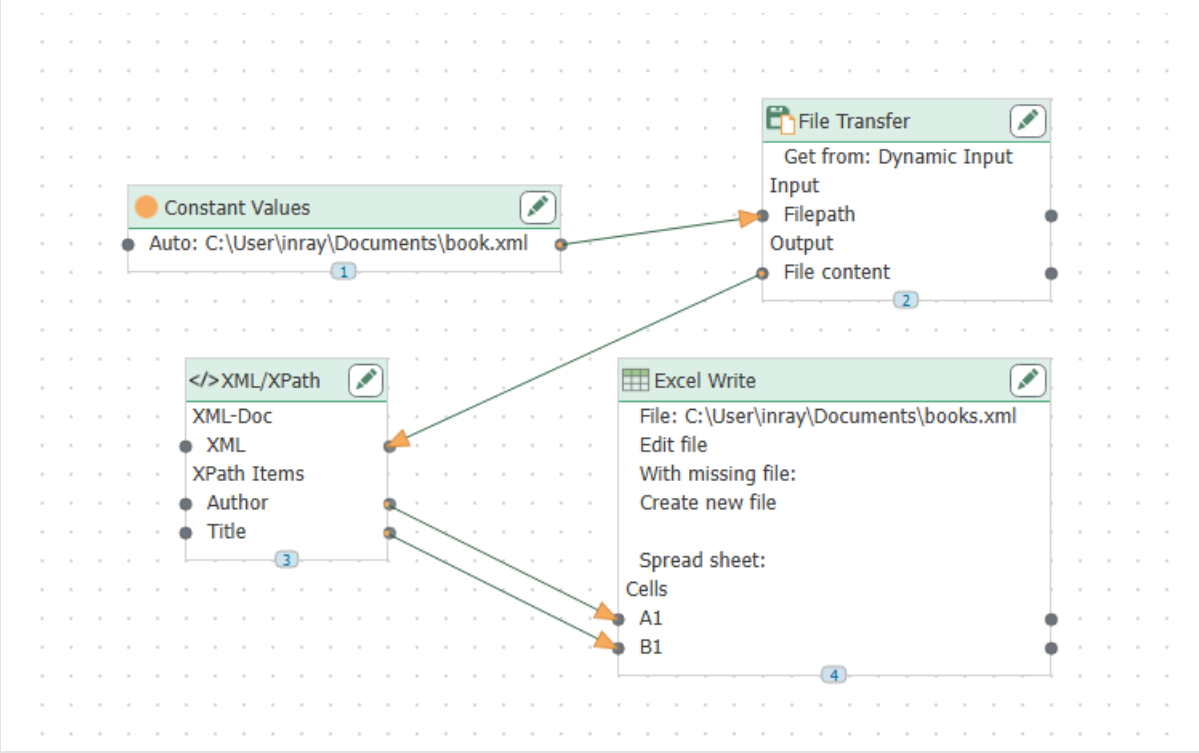
```
<?xml version="1.0"?>
<catalog>
  <book id="bk101">
    <author>J.R.R. Tolkien</author>
    <title>The Hobbit</title>
  </book>
  <book id="bk102">
    <author>Joanne K. Rowling</author>
    <title>Harry Potter and the Philosopher's Stone</title>
  </book>
 </catalog>
```

Then two X-Path expressions would supply the author and title of the books:

- Name: “Author”, X-Path expression “/catalog/book/author”
- Name: “Title”, X-Path expression “/catalog/book/title”

On the Internet you will find different documentations about the possibilities of X-Path, these are just the simplest (Restriction: Any nodes must be initiated with “//”).

For this example, the configuration looks like this:



- The path and name of the file are transferred from a constant object to the file transfer object. The name could also be determined dynamically from a database.
- The file transfer object transfers the contents of the XML file to the XML element in the XML object. At design time no code

was deposited here, only the two described X-Path expressions.

- The query results are written to an Excel file. Every other data destination is just conceivable (e.g. database INSERTS). Since several lines are expected, the write mode is “attache rows” in the Excel object.

XML Create Data

XML document as destination

As a transfer target, the XML object creates XML code based on the configured X-Path queries.

- The XML code can be transferred further, e.g. to a database. To create an XML file, a script would be required.

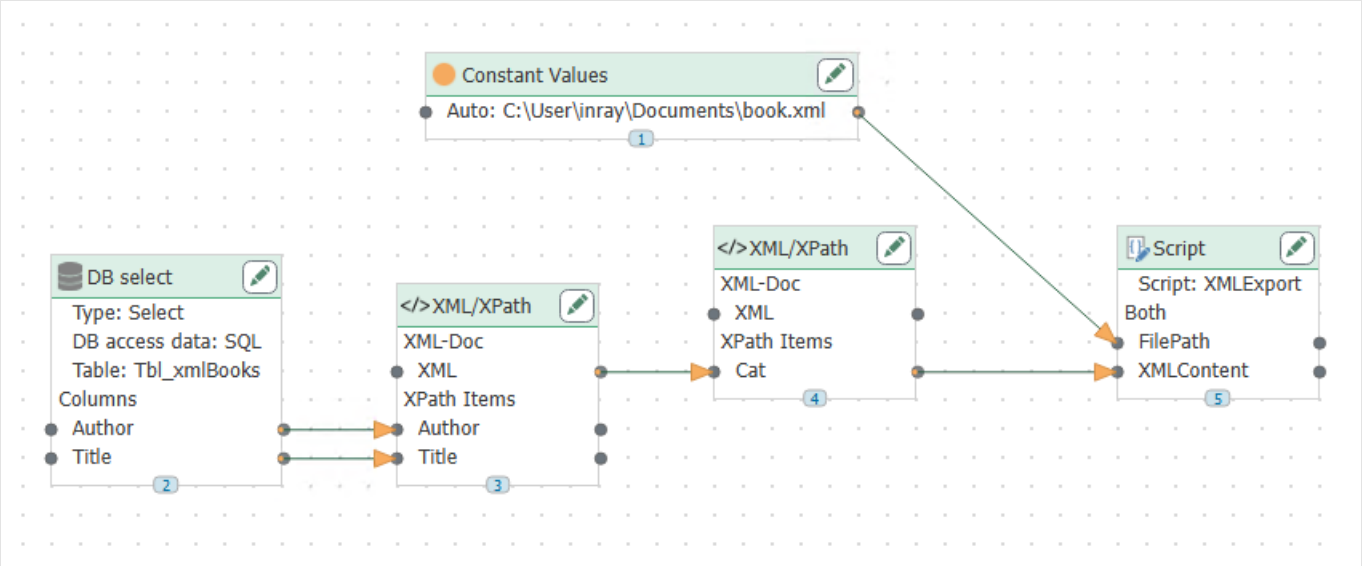
For example, it comes from a table:

AUTHOR	TITLE
-----	-----
J.R.R. Tolkien	The Hobbit
Joanne K. Rowling	Harry Potter and the Philosopher's Stone

From this, an XML file with the known structure is to be generated.

- Name: “Author”, X-Path expression “/catalog/book/author”
- Name: “Title”, X-Path expression “/catalog/book/title”

For this example, you need two staggered XML objects. The first one will transform the result set into individual XML elements, the second will group the elements into one document:



1. The data records have to be written by query from a database and transferred to the X-Path elements of the first XML transfer object.

```
<book>
  <author></author>
  <title></title>
</book>
```

2. The XML object generates the XML code. The XML object contains the well-known X-Path expressions:
    - Name: “Author”, X-Path expression “book/author”
    - Name: “Title”, X-Path expression “book/title”

as well as the basic structure

```
<?xml version="1.0"?>
<catalog>
</catalog>
```

\ Important is the property XML output = “Multiple XML documents”, so that each record from the database becomes its own element.

The structure could also be passed dynamically at runtime. E.g. from a text file or from a database

3. The code is passed from the XML element to the X-Path element of the second XML transfer object. This object contains only the XML declaration and the root element:\ With the X-Path query / catalog and the property XML output“One XML document”

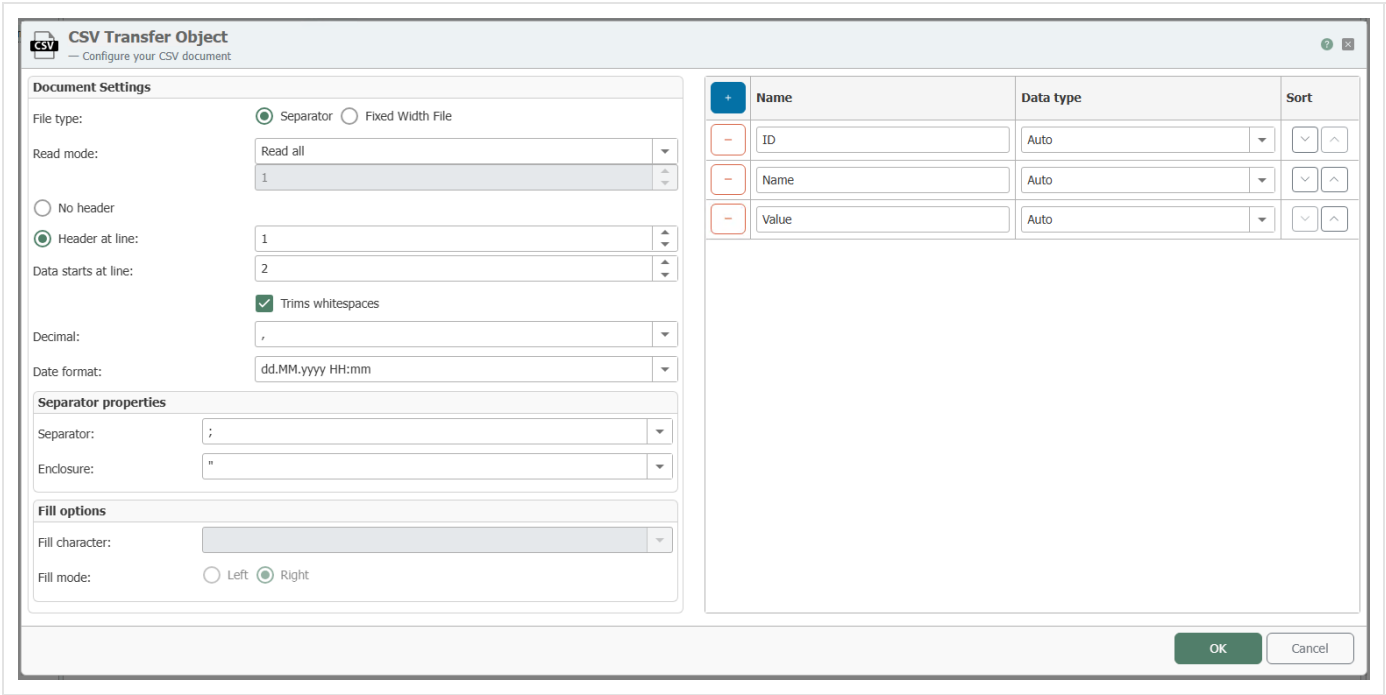
4. The code can be transferred further. Here, a script transfer object is used to create a file. File path (type: string) and XML content (type: array, string) are passed as input parameters:

```
public override void Write()
  {
    StringBuilder stb = new StringBuilder();
    stb.AppendLine("<?xml version=\"1.0\"?>");
    foreach(string xml in XMLContent)
    {
      stb.AppendLine(xml.Substring(xml.IndexOf("?>")+2));
    }
    System.IO.File.WriteAllText(FilePath, stb.ToString());
  }
```

CSV Transfer Object

CSV transfer objects can read, analyse and write strings. There is no plug-in configuration required. You can directly configure the transfer object within the connection as described.

After double-clicking on the transfer object, set the following properties:



CSV Transfer Object

Attention: The CSV transfer object can only read and write CSV strings. To read and write CSV files you need to use the File Read and File Write transfer object additionally to the CSV transfer object.

File type	Select Delimiter if your data is seperated by a defined delimiter like a semicolon. Select Fixed Width File if the columns has a defined, fixed width.
Read mode	Select which part of the CSV string you want to read Read all: Reads the whole document Read first n: Reads the first rows of the CSV document. (Select the amount of rows) Read last n: Reads the last rows of the CSV document. (Select the amount of rows)
Header	Select if the document has got a header. Also select the line number of the header
Data starts at	Select the number of the row where the data starts.
Decimal	Select the sign which is used in decimal digits.
Date format	Select the date format. You can also create your own format.

Delimiter	Select the delimiter of you CSV string.
Enclosure	Select the enclosure of your CSV document.
Fill character	Select the sill character. This character is user if you have selected Fixed Width File and it is the character which is used for unused space.
Fill mode	Select where the fill character should fill up the empty spaces.

On the right side you can define the data fields of the CSV string.

The names of the data fields need to match to the header names. You don't need to define every value of the header. If you have a CSV string with fixed file width, the field size need to match the field size of the csv string.

JSON transfer objects

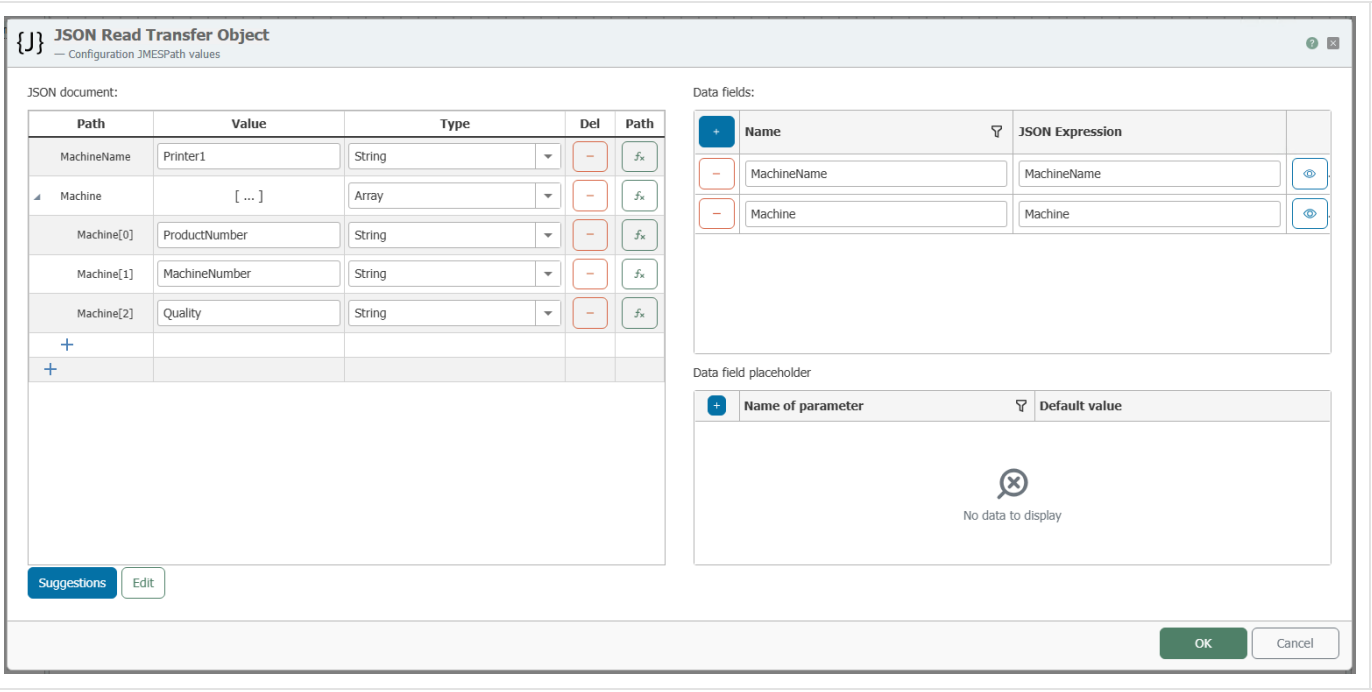
Overview over transfer objects:

json_write_transfer_object.md

json_read_transfer_object.md

JSON Read Transfer Object

With the JSON transfer object as a simple data exchange format, you can evaluate, change or create formatted texts or data in JSON. Create a JSON transfer object within the connection as described. A plug-in configuration is not necessary. You can configure the transfer object directly within the connection as described. The JSON expression is “JMESPath”.



JSON Read Transfer Object

After double-clicking on the transfer object, set the following properties:

JSON document	In this area you can define the JSON to read. The button “Edit” is for editing the raw/source data of the JSON. The button “suggestions” searches the connection for already executed sessions and JSON files. If a JSON document was found, you can take it as template.
Data fields	You can add data fields to read data of the JSON elements through the runtime. Click a value of the JSON document and use the “right arrow” or the “F+” button to add a field from the JSON.
JSON placeholders	JSON placeholders are made to add variables inside the JSON document. Inside of the JMES path expression, the variables are written in curly brackets. For example: `{MyPlaceholder}`

A tree structure is defined in the editor of the JSON tree as follows:

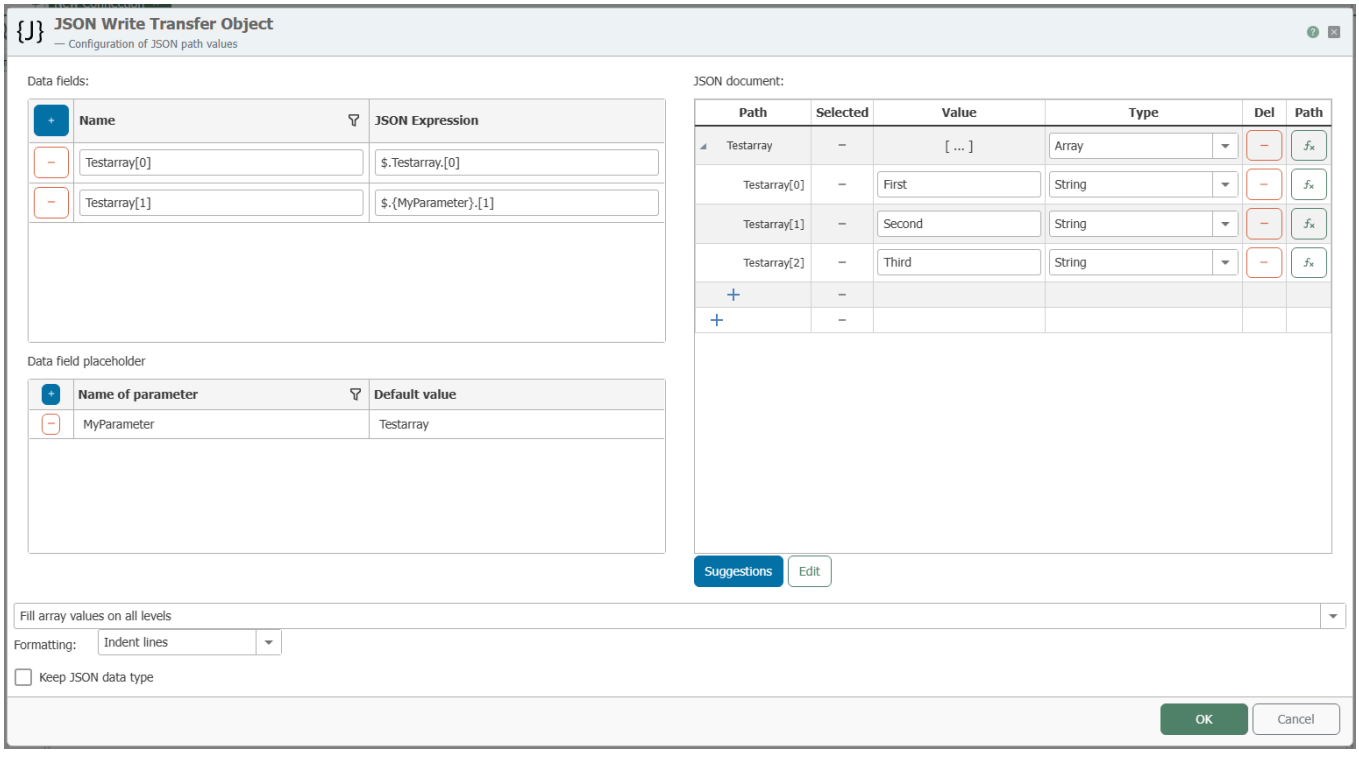
```
{
"key":"value",
"some":[
{"name":"one","value":1},
{"name":"two","value":2},
{"name":"three","value":3}
]
}
```

JSON Write Transfer Object

With the JSON transfer object as a simple data exchange format, you can evaluate, change or create formatted texts or data in JSON. Create a JSON transfer object within the connection as described. A plug-in configuration is not necessary. You can configure the transfer object directly within the connection as described. The JSON expression is “JSONPath”.

Here you will find an overview of the most important JSONPath syntax elements compared to the XPath elements:

XPath	**JSONPath**	**Description**
/	$	The parent object/element
.	@	Element in current context
/	.or []	Specification for child elements
//	..	Recursive descent
*	*	Placeholder. All objects/elements regardless of their name are selected.
[]	[]	Subscript operator for element collections.
[]	?()	Executes a filter (script) expression.



JSON Write Transfer Object

After double-clicking on the transfer object, set the following properties:

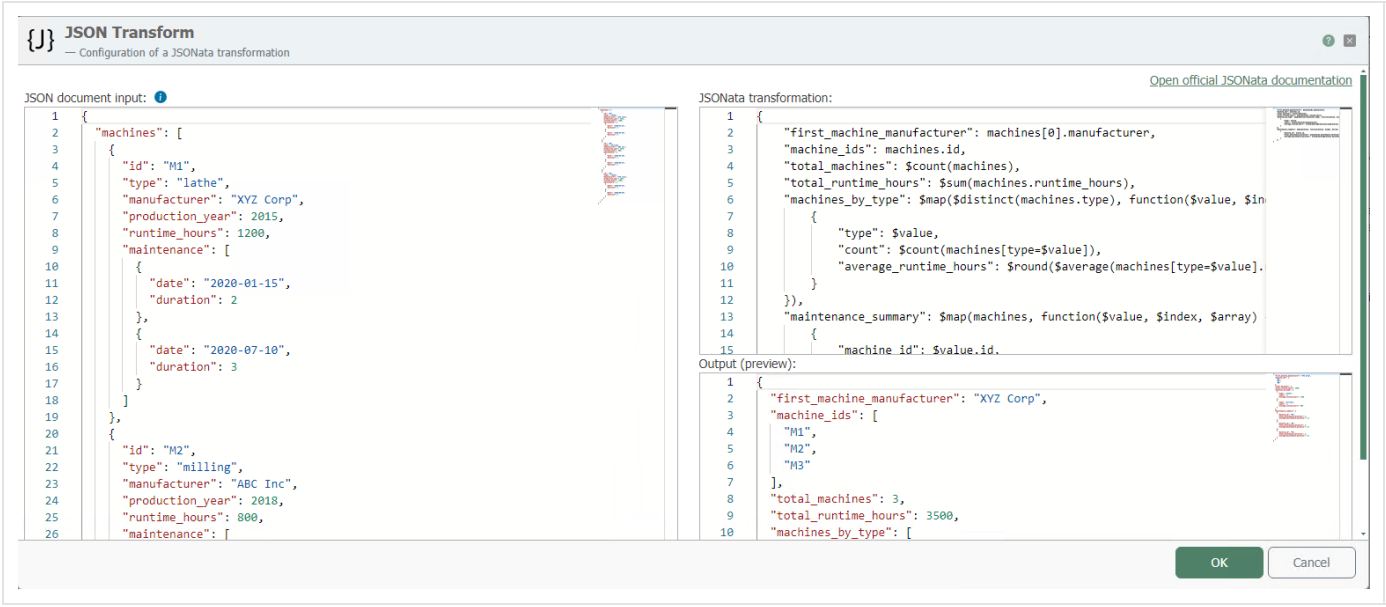
Data fields	Enter the JSON items directly or select entries from the JSON tree and use the “arrow up” or “F+” button to add. The names of the items can be freely assigned and serve as help during project planning. The items can be written and read during a transfer.
JSON Parameters	If necessary, enter additional parameters for default values here. During the transfer, the placeholder “{parameter}” is replaced by the given value ($.some[0].value). If nothing was passed, the default value is used.
JSON	The JSON Tree is the standard JSON document during a transfer. This can be read and written to. You can click on the “Edit” button and enter your tree structure or you can create the JSON structure through the UI. The button “suggestions” searches the connection for already executed sessions and JSON files. If a JSON document was found, you can take it as template.
Fill values	select between different modi • Fill array values on all levels • Fill array values only within the same array. • Duplicate document when an array is written to the data field

A tree structure is defined in the editor of the JSON tree as follows:

```
{
"key":"value",
"some":[
{"name":"one","value":1},
{"name":"two","value":2},
{"name":"three","value":3}
]
}
```

JSON Transformation

The JSON Transformation transfer object allows queries and transformations to be applied to JSON documents. The input and output format is JSON. JSONata is used as the transformation language. Create a JSON transfer object within the connection, as described. The transfer object can be projected without additional configuration; a plug-in configuration is not necessary.



JSON Document	The input document in JSON format, to which queries and transformations are applied, is passed at runtime. If no documents are passed, the document configured here is used.
JSONata Transformation	The JSONata transformation applied to the input documents is generated. Applying the transformation results in a new JSON document that can be further processed with the JSON Read transfer object. If no transformations are passed at runtime, the transformation configured here is used.
Output	The expected result. The view updates automatically when changes are made to the JSON document or the JSONata transformation.

JSON Schema Plug-in

The JSON Schema Plug-in extends OPC Router with functions for validating and processing structured JSON data according to JSON Schema definitions. Typical use cases:

- Validation of incoming JSON data (e.g., via REST or MQTT)
- Extraction and conversion of structured content
- Plausibility check before further processing ## Structure of the documentation The documentation consists of the following sections:
- `json-schema-plugin`: Overview of plug-in, configuration, and supported features
- `json-schema-transfer-object`: Use in transfer object (reading, writing, validating)
- `json-schema-validation`: Validation behavior, error diagnosis, and examples

> □□ The samples provided are for illustrative purposes only. Full support for external specifications (e.g., OPC UA, Siemens IECTL, Portainer) cannot be guaranteed.

Further information can be found directly in the respective section.

JSON Schema Plug-In

Introduction

The JSON Schema Plug-In allows you to centrally manage JSON schemas in OPC Router and use them in your connections. The plug-in enables you to formally define data structures and ensure that incoming and outgoing JSON data conforms to a predefined schema. This reduces integration errors—for example, incoming JSON messages can be validated before processing, or outgoing JSON payloads can be created according to a fixed structure. The JSON Schema Plug-In is intended for use cases in the IIoT and integration environment where **JSON** serves as the exchange format and a consistent data structure (e.g., according to the **JSON Schema** standard) must be ensured.

Schema management

The JSON Schema Plug-In is configured via schema management. Here, you can either store schemas **embedded** in the OPC Router project or link a **schema folder**. In schema management, you also have the option of generating JSON schemas from existing JSON documents or displaying sample data for a schema. **Embedded:** In this mode, schema files are stored directly in the OPC Router project. You can add new schemas using the editor or import existing ones. Embedded schemas are saved when the project is exported, making them available on other systems without separate file exchange. **Linked Folder:** In this mode, the plug-in references an external folder in the file system that contains JSON schema files. All `.json` and supported `.yaml` files in this folder (and any subfolders) are loaded automatically. Changes to the files affect the OPC Router project (after reloading/restarting). This mode is useful if schemas are to be used by multiple projects or versioned externally. Note that when exporting a project, only the path is saved, **not** the files themselves. **AsyncAPI/OpenAPI YAML support:** In addition to pure JSON schema files, the JSON Schema plug-in can also load schema definitions in YAML format, such as those used in **AsyncAPI** or **OpenAPI** specifications. This means you can include an AsyncAPI or OpenAPI definition (`.yaml` / `.yml`); the plug-in extracts the JSON Schema structures (e.g., defined message payloads) and treats them like regular schemas. This makes it easier to reuse existing API definitions to validate incoming JSON messages or format outgoing messages accordingly.

Schema management in the JSON Schema Plug-In: Choose between embedded schemas and linked folders, with loaded sample schemas. In the plug-in's schema editor, you can edit schemas manually or use convenient functions: **"Create schema from JSON"** automatically generates a schema proposal from a sample JSON. Conversely, you can use **"Show sample JSON"** to generate a valid JSON document based on the current schema—useful for illustrating or testing the structure. The editor supports the current draft versions of the JSON Schema standard (e.g., Draft 7 to Draft 2020-12) and validates the schema syntax so that errors are detected early on during editing.

Editing a JSON schema: In the schema editor, you can assign names, import or generate schemas, and validate the structure live.

Use in connections (transfer object)

The JSON Schema plug-in provides a transfer object that can be used to read, write, and validate structured JSON data based on a selected schema. The schema is selected from the schema management. Either the latest version is used automatically or a specific version is set manually. The transfer object uses **JsonPointer** internally to address individual fields within the JSON document. This allows values to be read, written, or modified via patch. For complex structures, such as arrays with objects, the JsonPointer may need to be supplemented manually, as dynamic indexing is not supported.

> **Note:** If a pointer points directly to an array or object, it is output completely as a JSON string. Dynamic indexing with JsonPointer (e.g., [0], [i]) is not possible. Pointers may need to be set manually. Depending on the schema and data context, these features may only be validated to a limited extent or in a simplified manner. For reliable results, complex constructs should be checked separately. The transfer object can:
>
> - Generate JSON data based on the schema
> - Validate incoming JSON data
> - Return a structured error list or abort the transfer in case of errors Optionally, a **format culture** can be set, which is used to convert between JSON schema types and .NET data types. ## Supported JSON Schema Features | JSON Schema

Feature | Supported | | ---------- | ---------- | | `type` (object, array, ...) | ✔□ | | `properties`, `required` | ✔□ | | `enum`, `const`, `default` | ✔□ | | `pattern`, `format` | □ (partially)[1] | | `items` (same schema) | ✔□ | | `anyOf`, `oneOf`, `allOf` | □ (partially)[2] | | `if` / `then` / `else` | □ (partially)[2] | | Array of Value (String, Number, ...) | ✔□ | | Array of Object | □ (partially)[3] | | `$ref` local | ✔□ | | `$ref` external (HTTP) | □ | | `patternProperties` | □ |

Use manual pointers if necessary.

Notes:

[1] `format`: Only common formats such as `date-time`, `email`, etc. are taken into account. No strict validation of complex formats.

[2] `anyOf`, `oneOf`, `allOf`, and `if` / `then` / `else`: Validation is simplified. Combinations with nested conditions or optional types may lead to unexpected behavior.

[3] `Array of Object`: JsonPointer cannot process dynamic addressing such as `ArrayProp/*/ItemProp`. If necessary, use manual pointers with direct assignment such as `ArrayProp/0/ItemProp`, `ArrayProp/1/ItemProp`, ...

It is recommended to use `$ref` to define a separate schema for the object in the array and to project two individual transfer objects. The JSON is processed in two stages (e.g., the outer object creates the array structure, the inner object processes the objects individually).

Example Schemas & Samples

The installation provides sample schemas under `Samples/`, grouped into folders such as `asset`, `event`, `measurement`, `production`, and `refs-demo`. The latter show referencing between schemas with `$ref`. These templates serve as an aid to project planning and simplify project planning tests. They are not intended as production-ready examples, but rather to show what is fundamentally possible with JSON Schema in OPC Router. Example:

- `equipment.schema.json`
- `location.schema.json`
- `machine_state_change.schema.json`
- `telemetry-packet.schema.json` In addition, “ready-made” specifications are provided, e.g.:
- OPC UA WebAPI specification from the OPC Foundation
- Siemens Industrial Edge Common Payload Format
- Portainer REST API These templates serve as an aid to project planning. However, we do not assume any liability or warranty for their completeness or timeliness. It cannot be guaranteed that all functions of these specifications are fully supported or correctly interpreted. ## Best Practices
- Embedded schemas for portable projects
- Linked schemas for external versioning (e.g., Git)
- Clear naming (including version), e.g., `order-v1.schema.json`
- Use of `$ref` for reusability
- Regularly test validity with sample data ## Note / Disclaimer **Attention:** JSON Schema is a dynamic, evolving standard. The OPC Router plug-in deliberately supports only a subset of it. Complex features such as `patternProperties`, tuple arrays, or external `$ref` references are not implemented.

The samples provided show which features are supported.

JSON Schema Transfer Object

Purpose and Function

The JSON Schema Transfer Object is used for structured processing of JSON data based on a previously selected JSON schema. It can **read**, **write**, **patch**, and **validate** JSON data against the assigned schema. The object is integrated into a connection in OPC Router on the project side and uses a schema from the central schema management.

Schema Selection

When configuring the transfer object, select a schema from the management. The following options are available:

- **Use latest version**: The transfer object always uses the highest available version of this schema (e.g., for planned migration).
- **Use fixed version**: A defined version is permanently assigned to ensure stable processing. ## Access with JsonPointer The standard **JsonPointer** (RFC 6901) is used to address individual fields within the JSON document. This allows you to address specific fields and read, overwrite, or patch their values. Example: `json { “equipment”: { “id”: “EQ-01”, “status”: ‘active’ } }` JsonPointer `equipment/status` addresses the field `“active”`. > **Note:** If a pointer points directly to an array or object, it is output in its entirety as a JSON string. ### Restrictions on arrays JsonPointer does not allow dynamic addressing of elements in arrays (e.g., `array/*/property`). If necessary, you must either:
- work directly with fixed indexes (e.g., `array/0/property`), or
- recommended: schematize the object in the array separately and process it with its own transfer object. ## Operations ### Reading The transfer object extracts values from JSON data using the configured pointers and makes them available as data fields. ### Writing At runtime, data values are taken from other objects and written to a new JSON document according to the schema. ### Patching Existing JSON data is updated at specific points without regenerating the entire document. ## Validation The transfer object can check incoming or generated JSON data against the assigned schema. The behavior in the event of a schema violation can be configured as follows:
- **Generate error list**: Validation errors are collected and provided as a list of items.
- **Abort on error**: The transfer is stopped and an error message is generated when the first violation occurs. ## Format culture and type conversion A **format culture** can be selected in the transfer object for type conversions between .NET types (such as `DateTime`, `int`, `double`) and JSON schema data types (such as `string`, `number`). This determines, among other things:
- Date and time formatting (e.g., `dd.MM.yyyy` vs. ISO format)
- Decimal separator (e.g., comma vs. period)
- Language-dependent format conventions ## Manual JsonPointer extension In cases where automatic assignment of JsonPointer entries is not possible (e.g., for complex or nested structures), additional pointers can be added manually. This applies in particular to:
- Arrays with multiple objects of the same type
- Referenced sub-schemas via `$ref` ## Notes
- The transfer object only works with valid, fully loaded schemas.
- All pointers used must point uniquely to a data field in the JSON schema.
- Optionally defined fields can remain empty, but mandatory fields (according to `required`) must be filled in at runtime. ## Example usage A transfer object reads a temperature message that is validated against the schema `temperature.schema.json`. Fields such as `value`, `unit`, and `timestamp` are then extracted and passed on to an SQL connection. Alternatively, another transfer object generates a machine status message according to `machine_state_change.schema.json`, in which all required fields are filled with data and optional fields are added.

JSON Schema Validation

Purpose of Validation

Validation ensures that a JSON document corresponds to the expected structure and meets the requirements defined in the schema. The check is performed either when **reading** JSON data or when **creating** or **patching** in the transfer object.

Supported Validation Functions

The supported features and restrictions for validation are documented centrally in: `json-schema-plugin.md` There you will also find information on:

- Partially supported constructs (`allOf`, `oneOf`, `anyOf`)
- Special features for arrays and `$ref`
- Recommendations for project planning ## Error diagnosis and testing options Validation in the JSON Schema transfer object offers two operating modes:
- **Mode 1 – Output errors as result items:**
- All validation errors are collected.
- The fields `IsValid` and `ValidationErrors` contain the status and a structured list of violations.
    - Processing runs through completely.
- The errors can be transmitted to external systems or logged in a database, for example, depending on the use case.
- **Mode 2 – Abort in case of violation:**
- The data transfer is aborted if validation is unsuccessful.
- The error message appears in the status field of the transfer object and in the log file.
- Triggers with retry logic can automatically restart the transfer. Further information on error diagnosis:
- Validation errors include, among others:
- Missing required fields (`required`)
- Incorrect data types (`type`)
- Invalid value ranges (`enum`, `minimum`, `format`, etc.)
- Unresolved `$ref` references
- Validation errors are documented in the log with the JsonPointer path and plain text.
- Ensure that all referenced sub-schemas (`$ref`) are loaded correctly and accessible. ## Examples of more complex validations
- `uns-machine-context.schema` uses `oneOf`, `anyOf`, `allOf` to model alternative configurations.
- `job_status.schema` uses `enum`, `pattern`, `format` to validate fields such as timestamps, status codes, and ID formats. These examples are installed with the OPC Router, but are **not intended as productive templates**; rather, they serve as examples of how JSON Schema can be used. > **Legal notice:** Included schemas such as OPC UA (opc.ua.openapi.allservices), Siemens IECTL, or Portainer are for illustrative purposes only. No claim is made as to their completeness or functionality.

File

Overview:

Transfer Objects

File trigger

Transfer objects

Overview over transfer objects:

File Write Transfer Object

File Delete Transfer Object

File Read Transfer Object

File Read

File Read transfer objects can move files and make file contents available. File paths can be selected from a database or as a constant. You can directly configure the transfer object within the connection as explained .

After double-clicking on the transfer object, set the following properties:

Tab: Source

Dynamic or static	File source static or dynamic path. Dynamic paths can get defined by an connection arrow and it can change through the runtime.
Abort if file not found	Activate to abort the transfer if the file was not found.
Read binary	Select this option if you want to read a binary file like *.doc, *.jpg, * .xlsx, ...
Read Text, Encoding	Select if you want to read a simple text file and select the files encoding.

Tab: Target

Move file after transfer	Activate this property and select a folder to save the file in the state before the transfer operation. Attention: If transfers are triggered every second, a new archive file is created every second. This option is therefore more suitable if, for example, daily statistics are created. The file name of the archive file contains the date and time of archiving (not the creation date of the original application file). Format: yyMMdd_hhhmmss_fileName.
Dynamic or static	Dynamic or static target path. For static target path, select the file name to read the file content.

File Write

File Write transfer object can read and manipulate files. File paths can be selected from a database or as a constant. You can directly configure the transfer object within the connection as explained .

After double-clicking on the transfer object, set the following properties:

Tab: Output path

Dynamisch oder Statisch	File target static or dynamic path. Dynamic paths can get defined by an connection arrow and it can change through the runtime.

Tab: Existing file

Action	Overwrite: Will overwrite existing files. Throw error: Creates an error if the file is existing. Append: For simple text files, the contend can get appended to the existing contend. Move existing: Moves existing files to a target folder.

Tab: File settings

Dynamic or static filename	Select a static or dynamic file name. Dynamic file names can get defined by a connection arrow and it can change through the runtime.
Prepend UTC Timestamp	Activate if you want to prepend the UTC timestamp to the file. For example the file „test.txt“ gets renamed to „20191204_095646_test.txt“.
Write binary	Select this option if you want to write a binary file like *.doc, *.jpg, * .xlsx, ...
Write text	Select if you want to write a simple text file and select the files encoding.

File Delete

File Delete transfer object can delete files. File name patterns can be selected from a database or as a constant. You can directly configure the transfer object within the connection as explained .

After double-clicking on the transfer object, set the following properties:

Path	Select the path where the OPC Router should delete files
Name pattern	Select a static or dynamic file name pattern. Dynamic file name pattern can get defined by a connection arrow and it can change through the runtime. You can use placeholders like *.
Include subdirectories	Activate if the OPC Router also have to delete files inside the subdirectories.
Delete unrestricted	Deletes all the files for the name pattern.
Files older tha	Only delete files older than selected timespan.

File trigger

The File Trigger executes a transfer if a file gets changed or created.

After double-clicking on the transfer object, set the following properties:

Path	Select the path to observe.
Fallback scan interval	The OPC Router gets notifications if files get changed or being created. If it misses a notification, it will observe the directory additionally in this interval.
Filename pattern	Define a filename pattern. You can use placeholders like *.
Read binary	Select this option if you want to read a binary file like *.doc, *.jpg, * .xlsx, ...
Read Text, Encoding	Select if you want to read a simple text file and select the files encoding.

Target options

Delete after successful transfer	Deletes the file which triggered the trigger after a successful transfer.
Success destination	Moves the file which triggered the trigger in a special folder
Prepend UTC Timestamp to filename	Activate if you want to prepend the UTC timestamp to the file. For example the file “test.txt” gets renamed to “20191204_095646_test.txt”.
Delete after error transfer	Deletes the file which triggered the trigger if a transfer aborts for some reason
Error destination	Moves the file which triggered the trigger in a special folder if the transfer gets an error
Prepend UTC Timestamp to filename	Activate if you want to prepend the UTC timestamp to the file. For example the file “test.txt” gets renamed to “20191204_095646_test.txt”.
Number of retries	If the OPC Router can't access the selected file, it can retry for n times.
Seconds between tries	Select the time between the retries.

.NET C# Script

No system covers 100% of every requirement, which is why a certain degree of flexibility is necessary. The script plug-in offers the possibility to create transfer objects and triggers in the OPC Router with little effort. Basic functionalities can be user directly in the script, but external .NET assemblies can also be connected as required.

Programming in the script plug-in works according to the low code principle and can therefore be implemented even without above-average programming knowledge.

You can find a detailed video on this topic here in our Tutorial Stream.

General .NET Script instructions

Edition of Scripts

Attention: Within a Docker container, only assemblies compiled for .net Core are supported.

The Script plug-in enables you to define freely with your own C# code transfer objects and triggers.\ The configuration of the plug-in is therefore different as well from the other plug-ins : after giving a name and a typ, open the code editor to edit edit the script.

In the Code-Editor

- You create assembly references.
- You define input and output parameters.
- You will already find the basic method that you can program.
- You have tools for OPC access and writing log messages at your disposal.

The script plug-in allows you to describe complex issues in transfer objects and triggers, for example to perform conversions or trigger complex database queries.

At the top of the menu bar you will find the "Rename script" button (blank sheet with pen) to change the previous name of your script.\

Note: In larger application scenarios, the router will be part of an overall strategy, for example with regard to where code that carries business logic is stored. Even in smaller projects, it makes sense to consider wheter, for example, conversions are carried out in the router or via stored procedures in the database.

IDisposable

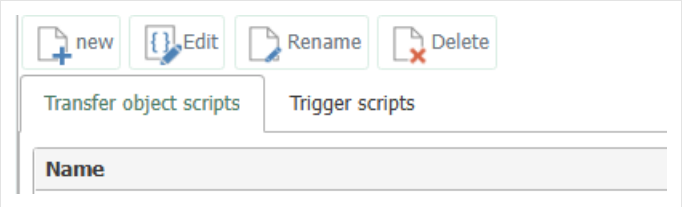
We recommend integrating the IDisposable interface into the script. This allows you to release memory again, for example when you open ports or start threads in the script. Open ports or running threads are thus closed or terminated at the end of the script. Write the required code in the corresponding method.

```
public void Dispose()
{
   // Your Code
}
```

The Dispose method is called if:

- You publish the script after changes.
- You publish the script without changes.
- You publish a connection after deleting it.
- The plug-in is stopped (this always happens when publishing).

Creation of Scripts



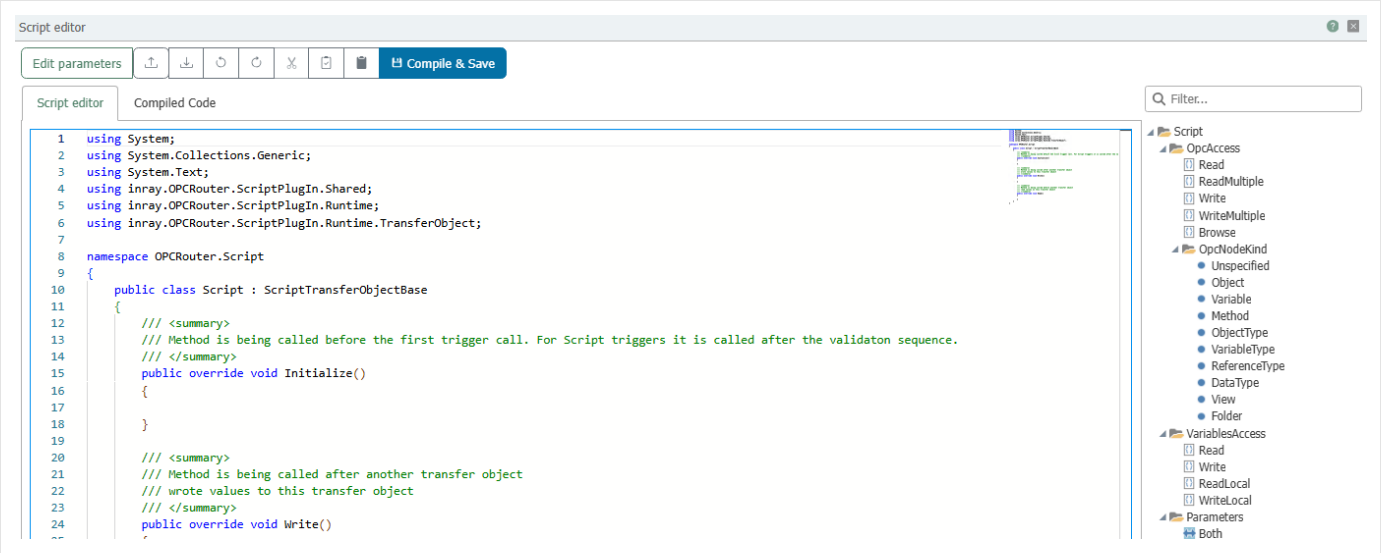
1. Open the plug-in by double-clicking on "Scripts" under the "Plug-ins" menu. You will see a list of the transfer object scripts and trigger scripts created on two tabs.
2. Select "Add new connection" to create a new script and enter the name and ype. When you create a new script, the scripts

opens automatically in the script editor. Double-clicking on an existing enty also opens the script in the script editor.

3. Programme your script, (see below). Remember to compile and save.

Transfer objects and triggers are then available for connection configuration.

.NET C# Script Editor



In addition to standard operating functions (copy, paste, cut, undo, save), the editor offers some special functions:

- **"File - Save externally" Menu:** Saves the script as an XML file with header data, code and parameter definitions. It is also possible to open such XML files and thus import scripts.
- **"Settings - Edit parameters" Menu:** This is where you create the required parameters and variables (see Transfer object scripts and trigger scripts).
- **"Settings - Edit comment" Menu:** The comment is displayed in the script list of the plug-in.
- **"F6" Key:** Compiles and tests the programme for errors.

On the right-hand side of the window you will find a graphical programming aid for OPC accesses, variables, from the OPC Router configuration, log messages and parameters.

- Double-click on an entry and the corresponding prepared line of code will be inserted. Entries can be written as string (in inverted commas) or as a pre-assigned variable.

Indications on OPC zero values

If you have ticked "Transfer OPC zero values" in the OPC UA plug-in, the corresponding values of the data points are always transferred if they are zero.

If the box is not ticked and the "Quality" of the data point is not "Good", the value is not transferred again and the connection runs on error.

If the tick is not set and the "Quality" of the data point is "Good" and the value is zero, then zero is transferred.

OpcAccess.Read

Reads out the value of an OPC item.

```
Object <myValue\> = OpcAccess.Read("<ItemPath\>", "<OPCServerName\>")
```

- **Return:** Object
- **ItemPath:** Access path and name of the item.
- **OPCServerName:** Name of the OPC connection in the router project.

OpcAccess.ReadMultiple

Reads out the value of multiple OPC items.

```
List<string> <lItemNames\> = new List<string> ();

<lItemNames\>.Add ("<ItemPath.ItemName1\>");
<lItemNames\>.Add ("<ItemPath.ItemName2\>");
...

Dictionary<string,object> <MyValues\> = OpcAccess.ReadMultiple(<lItemNames\>, "<OPCServerName\>");
```

- **Return:** Dictionnary <*string*>
- **ItemPath:** Access paths and item names as a list.
- **OPCServerName:** Name of the OPC connection in the router project.

OpcAccess.Write

Writes a value to an OPC item.

```
OpcAccess.Write ("<ItemPath\>", <objValue\>, "<OPCServerName\>")
```

- **Return:** None
- **ItemPath:** Access path and name of the item.
- **objValue:** Value to write.
- **OPCServerName:** Name of the OPC connection in the router project.

OpcAccess.WriteMultiple

Writes multiple values to multiple OPC items.

```
Dictionary<string,object> <dicItemsToWrite\> = new Dictionary<string,object> ();
<dicItemsToWrite\>.Add("<ItemPath.ItemName1\>", <Value1\>);
<dicItemsToWrite\>.Add("<ItemPath.ItemName2\>", <Value2\>);
...
OpcAccess.WriteMultiple(<dicItemsToWrite\>, "<OPCServerName\>")
```

- **Return:** None.
- **dicItemsToWrite:** Dictionnary with items and values.
- **OPCServerName:** Name of the OPC connection in the router project.

OpcAccess.Browse

Provides the option of performing a browse (object search) for OPC DA and UA connections.

ItemPath: Access path and name of the item.

OPCServerName: Name of the OPC connection in the router project.

Various queries are available to you :

```
OpcAccess.Browse(ItemPath, ServerName).Select(e => e.DisplayName).ToArray()
```

- **Return:** Specifies the names of the tags from this OPC path.

```
OpcAccess.Browse(ItemPath, ServerName).Select(e => e.NodeKind.ToString()).ToArray()
```

- **Return:** Returns the type of tags from this OPC path.

```
OpcAccess.Browse(ItemPath, ServerName).Select(e => e.Identifier).ToArray()
```

- **Return:** Returns the name of the tags with full path.

```
OpcAccess.Browse(ItemPath, ServerName).Select(e => e.HasChildren).ToArray()
```

- **Return:** Returns wether the tags still have sub-objects (childs).

Important: To search for OPC objects, the usings "Using System.Linq;" and the assembly reference "System.Core.dll" must be specified.

VariableAccess.Read

Reads the value of a variable from the OPC Router project. This only refers to variables that were created via the variable plug-in, that means no placeholders in templates, mail objects or similar. If the variable does not exist, NULL is returned.

```
Object <myValue\> = VariablesAccess.Read("<VariableName\>")
```

- **Return:** Object.
- **VariableName:** Variable name in the OPC Router project.

VariableAccess.Write

Assigns a value to a variable of the OPC Router project. This only refers to variables that have been created via the variable plug-in, that means no placeholders in templates, mail objects or similar. If the variable does not exist it is created again.

```
VariablesAccess.Write("<VariableName\>", Value)
```

- **Return:** None.
- **VariableName:** Variable name in the OPC Router project.
- **Value:** Value that is assigned, Object.

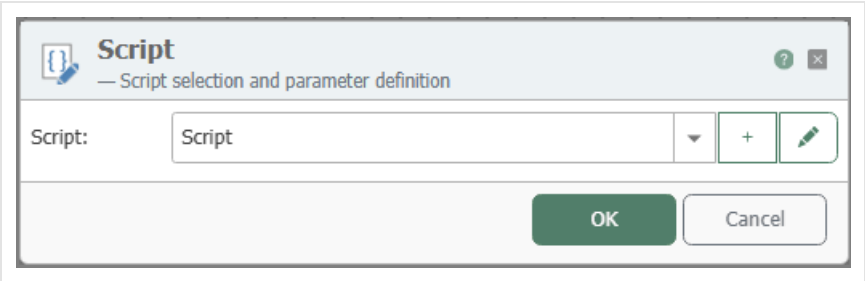
Log

Write a message in the log file.

```
Log("<strMessage\>", MessageType.<LogLevel\>)
```

- **Return:** None.
- **strMessage:** Message to write.
- **LogLevel:** Log level to write. If the corresponding log level is deactivated, no logging takes place. MessageType.All logs in any case, regardless of the router configuration.

Script Transfer Object



Script Transfer Object

When you create a new script of the "Transfer object" type, three predefined methods are available:

Methods in the transfer object

METHOD	DESCRIPTION
Initialize	The method is called once when the OPC Router is started.
Write	The method is called after another transfer object has written data to this transfer object. It is therefore the code that is executed when this object is the target of a data transfer.
Read	The method is called before another transfer object reads data from this transfer object. It is therefore the code that is executed when this object serves as the source of a data transfer.

Use the "Settings - Manage parameters" menu to create the parameters and variables. You can define the following properties for the parameters:

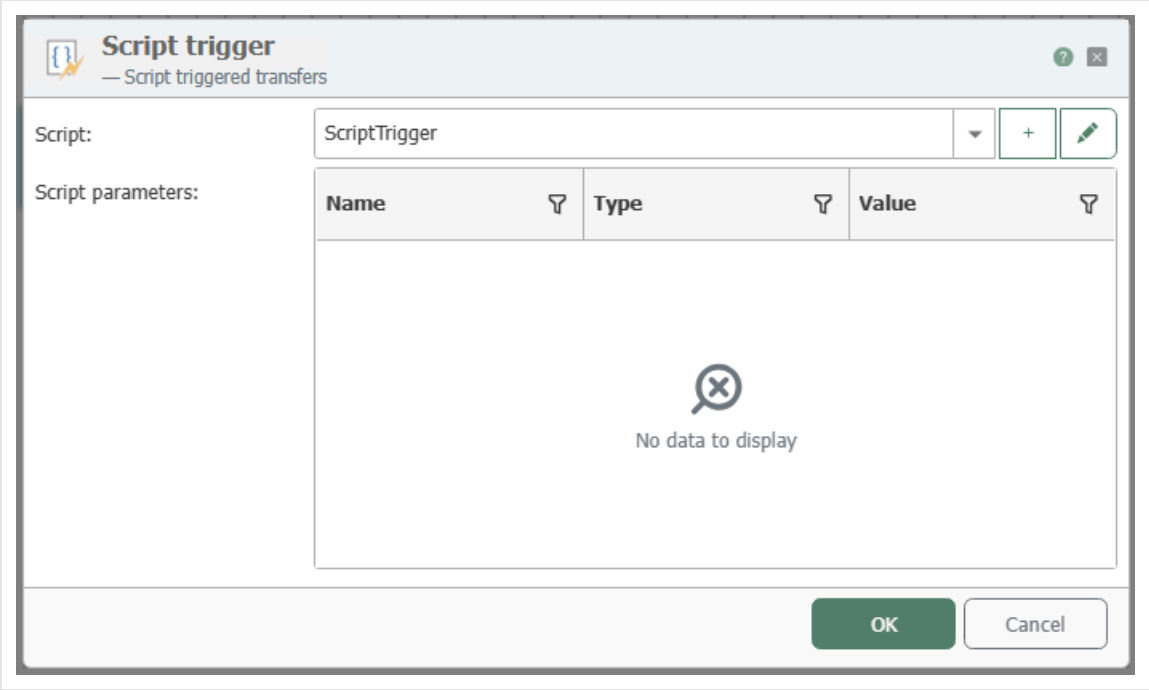
Properties of parameters

NAME	PARAMETER NAME
Array	Yes

PARAMETER	DATA TYPE
Parameters direction	• ***Input:*** Input parameter. In connection configuration, this parameter can be used as the target of a data transfer. • ***Output:*** Output parameter. In connection configuration, this parameter can be used as the source of a data transfer. • ***Both:*** The parameter is both an input and output parameter. • ***None:*** Variable, that is not visible to the connection projector and is only used within the script.

When you create a new script, the script editor opens automatically.

Script trigger



Script Trigger

When you create a new script of the "Trigger" type, three predefined methods are available to you.

Methods in the trigger object

METHOD	DESCRIPTION
Initialize	The method is called once when the OPC Router is started. Here you define at least the "Query interval", that means the interval at which the system checks wheter the trigger condition applies.
CheckTrigger	Method in which the trigger condition is stored. It is called cyclically in the "Query Interval" (see above). The method returns `true` or `false`. If `true`, the data transfer is triggered.
ResetTrigger	The method is called after the data transfer has been completed. The parameter `(bool transferError)` is automatically passed (`true` = Transfer error). Error handling can therefore also be programmed in this method.

You create the parameters via the "Settings - Edit parameters" menu. The script trigger only recognises input parameters, so you cannot specify a direction here.\ \ The parameter value is defined in the connection configuration. It cannot be determined dynamically at runtime. If you require actual values at runtime, you would have to programme corresponding OPC accesses or database queries in the trigger. The input parameters make it possible to use the same script in different ocnnections that require fundamentally different properties (table names, access paths, etc.).\

When you create a new script, the script editor opens automatically.

Additional functions

Overview over additional functions:

Cron trigger

Time trigger

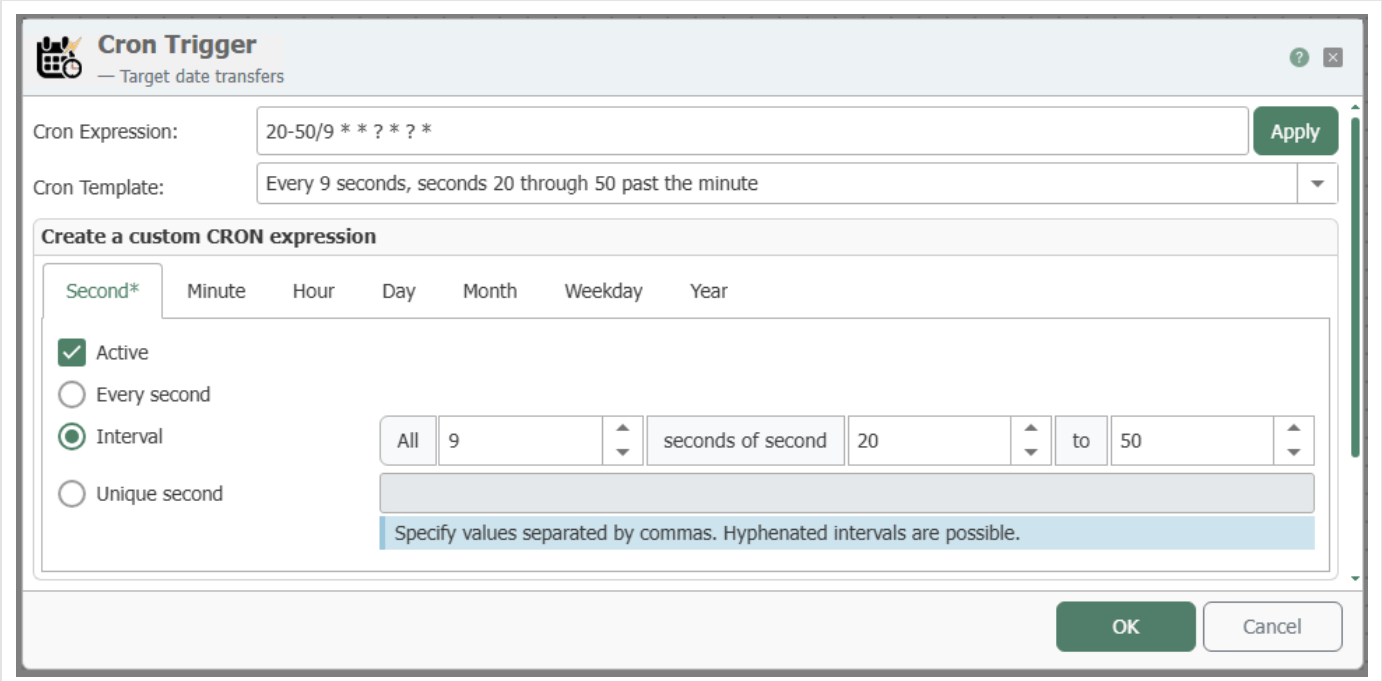
Note Transfer Object

Text Replace Transfer Object

Calculator Transfer Object

Cron trigger

The cron trigger triggers scheduled transfers. In the property mask, you can arrange the appointments as needed. For advanced users, however, a manual input is also possible.



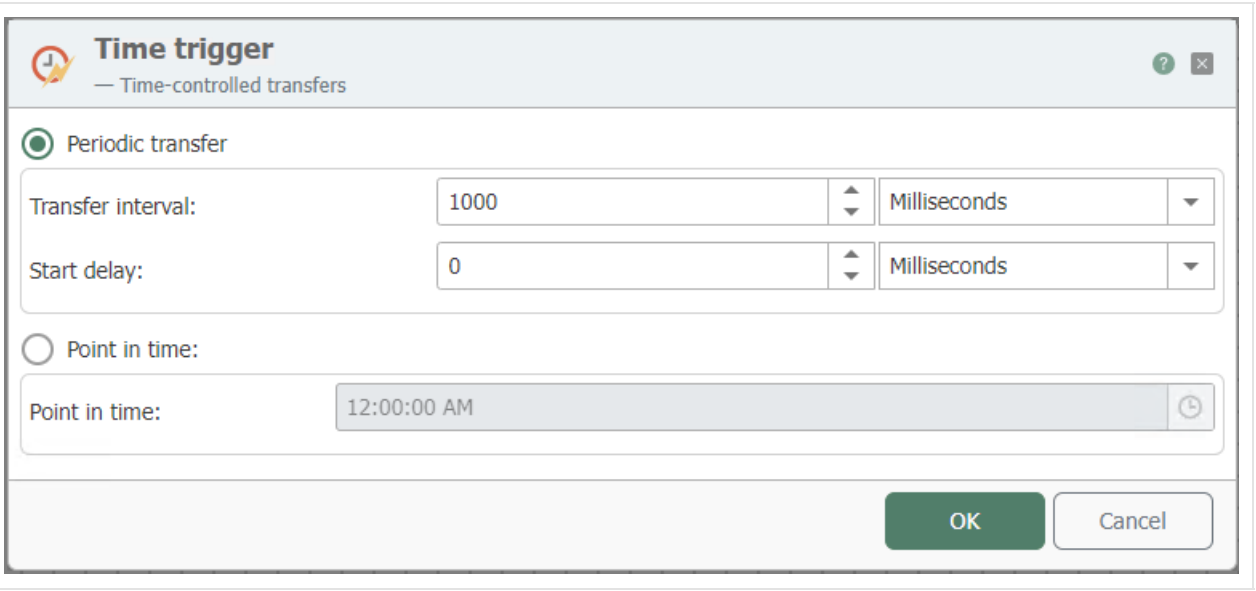
Cron Trigger

The individual properties are:

Expression	Display of timed transfers after selection of properties. Here is also the possibility of manual input. After entering, confirm with the “Apply” button.
Description	Display of the expression in text form. The display changes automatically when the selection and entries are changed. Here you can choose between different expression proposals.
Active	After selecting an area (second, minute, hour, day, month, weekday, year), activate the checkbox to activate the trigger. Which areas are activated is shown in the tab by an asterisk *. You can define specific details by simply entering a number. For example, if 5 is entered for specific hours, then the transfer will only be triggered directly at 5 o'clock. Several specific inputs can be separated by a comma.

Time trigger

The time trigger triggers data transfers at variably intervals in a time-controlled interval. If a transfer is to be carried out once a day, you can also specify a fixed time.



The individual properties are:

Periodic transfer	The transfer will be triggered at the specified interval.
Transfer interval	The interval at which the transfer is triggered. If start delay = 0, the first transfer after router start will not be triggered until the interval has elapsed. Default always as an integer, no decimal places. As units therefore milliseconds, seconds, minutes, hours or days are to be selected. The router always calculates the new interval after the end of the last transfer, so that at setting “1 s” a transfer will not be triggered every second, but 1 second after the end of the previous transfer.
Start delay	Start delay after which the interval count starts after the router has started. The first transfer thus takes place after elapse of start delay + transmission interval. Default always as an integer, no decimal places. As units therefore milliseconds, seconds, minutes, hours or days are to be selected.
Point in time	The transfer will be triggered once a day at the set time.
Point in time	Time at which the transfer is triggered.

Note Transfer Object

In extensive projects, especially when several colleagues work together, the comment on the connection is quickly no longer sufficient to keep track.

Use the object “Note” to attach one or more detailed comments to a connection, e.g. a documentation for the configuration. To view or edit the note, double-click on your note transfer object. In this way, you also provide extensive comments clearly.

Times are given in the e-mail and in the PDF file as system time, but in CSV as UTC.

The note can not be used as a data source or destination.

Create a Note object within the connection as explained.

Note

After double-clicking on the object, set the following properties:

Enter note	any text

Text Replace Transfer Object

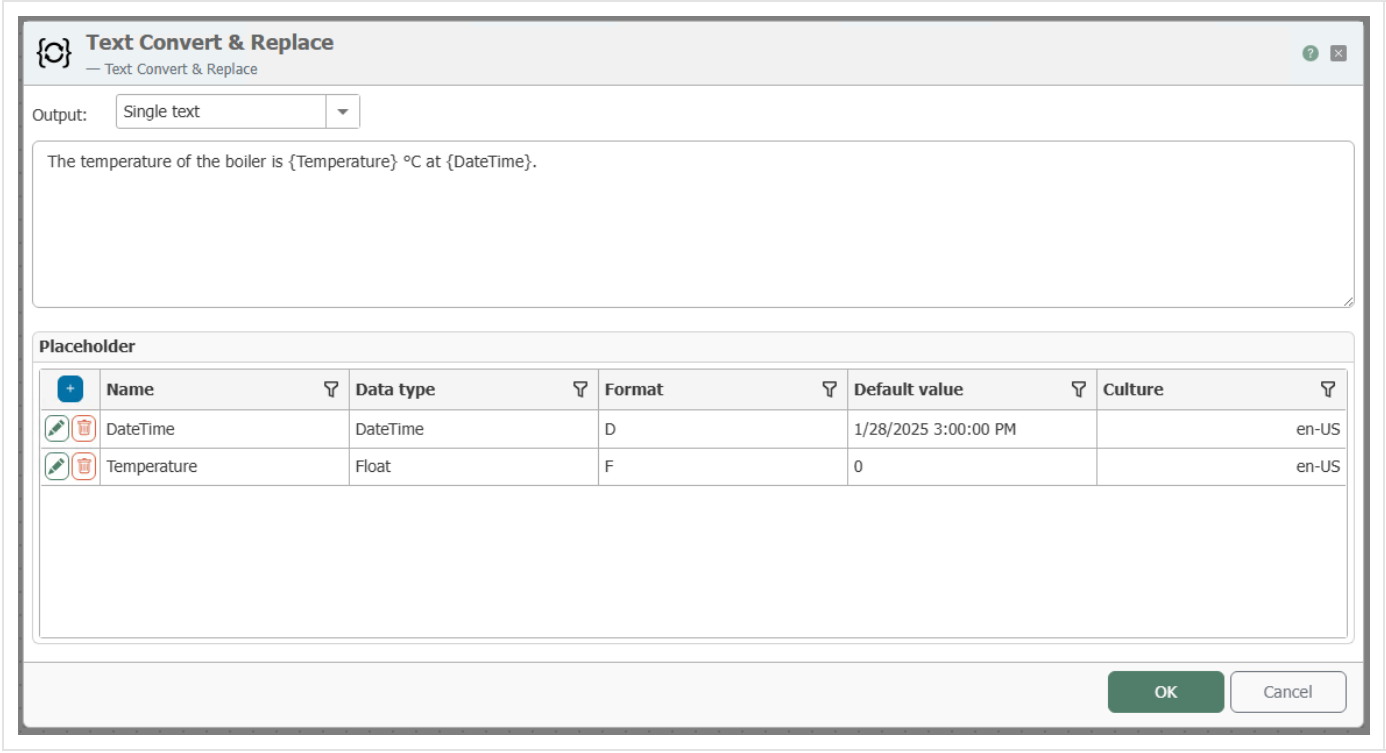
The “Text replace” transfer object is used to transfer texts using connections in order to combine values and texts.

After double-clicking the transfer object, set the following properties:

Output	Choose between SingleText and Array: • Output “Array”: Depending on the length of the largest input array, the same number of strings is created in the output array. • Output “SingleText”: Regardless of the amount of data entered, only 1 string is output at a time. The data is summarized accordingly.
Text	Enter your text here. You can enter placeholders by double-clicking on the corresponding entry in the placeholder list to the right.
Placeholder	At this point, enter a new placeholder using the plus sign or edit a selected placeholder using the corresponding button. Use the red cross to remove a placeholder.

Define placeholder

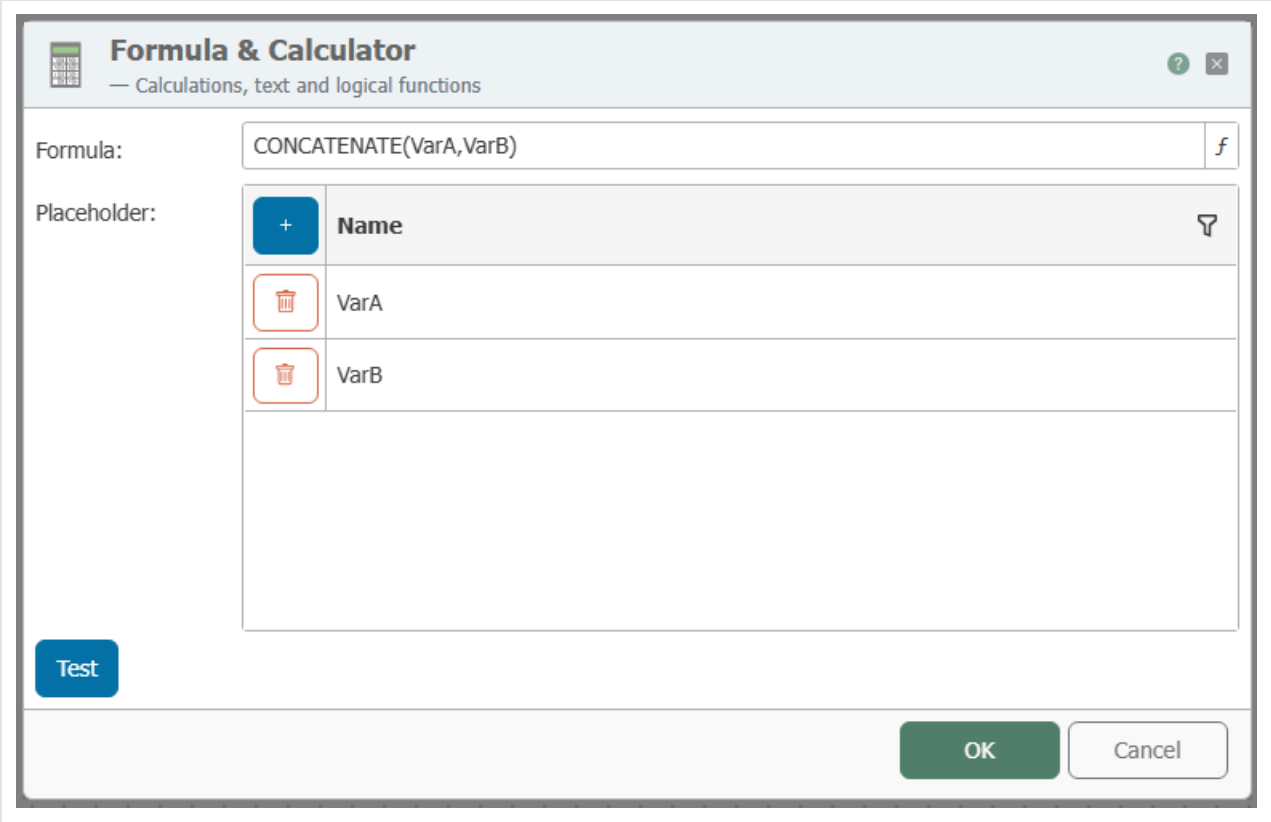
Name	Enter a unique name for the placeholder.
Data type	Specify the data type of the placeholder, for example, String, DateTime.
Default Value	Enter a default value if required.
Preview	The preview shows you the form in which a value will be displayed.
Format	Selection of a format for the data types Int64, Float or DateTime,
Culture	Select the culture to get the corresponding spelling. For example, commas or dots are set differently depending on the selected culture.



Text Convert & Replace Transfer Object

Calculator Transfer Object

You can use all available (numerical) values as input for calculations. You then use the results as a data source in the further transfer. Create a computer transfer object within the connection as explained.



Formula & Calculator Transfer Object

After double-clicking on the transfer object, set the following properties:

1. Use the button “New” to add as many placeholders as your formula requires. These placeholders are filled later with values of OPC items, from database tables, from variables ...
2. Compose the formula.
   - The placeholders are simply entered or accepted by double-clicking.
   - Use the normal mathematical symbols for connection (`+-*/()`).
   - Functions apply to the immediately following placeholder, otherwise please set brackets (e.g. `sqrt(A+B)` = Square root of the sum of A and B).
   - Rounding functions: Via the “Round up” or “Round down” button (rounds up to whole numbers)
   - Concatenate function = concatenate strings: The values must be marked with "" in the formula e.g. CONCATENATE("A","B").
   - Additional functions via the button “Further”.
3. Use the “Test” button to open another dialog. Enter values for the placeholders here and check the formula.
4. If all right, confirm both dialogues with “OK”.

The computer transfer object is now available in the connection. Connect variables, OPC items, database fields, etc. with the placeholders and use the result as a data source.

FAQ

Overview of chapter sections:

Error Diagnostics

Connections

How can I...

Best Practice

Escalation levels

The OPC Router recognizes when too much status data is being recorded or the database storage is almost full. To prevent a full storage and RAM, the OPC Router automatically initiates increasingly stronger countermeasures. The transmission of user data is NOT impaired by these measures! The measures only affect the status data recording of the OPC Router. The recording of payload data should be ensured by these measures. In the status-storage-options.md of the OPC Router, you set the extent to which the OPC Router can fill the status databases.

There are the following escalation levels:

- Escalation level 0:
  - Normal behavior
- Escalation level 1:
  - Ring buffer batch size exceeded
  - Status database 60% full
  - Action: No more logging of transfer values from successful transfers for fast firing connections
- Escalation level 2:
  - Ring buffer batch size exceeded by a factor of three
  - Status database 80% full
  - Action: No more logging of transfer values from successful transfers
- Escalation level 3:
  - Ring buffer batch size exceeded by a factor of six
  - Status database 90% full
  - Action: No more logging of transfer values
- Escalation level 4:
  - Ring buffer batch size exceeded by a factor of ten
  - Status database 100% full
  - Action: Deactivate status recording of transfers completely

The current escalation level can best be seen in the status storage options.

See:

status-storage-options.md

Error Diagnostics

There is – even with all coverage – always the possibility, that a connection doesn´t “work”. In most cases, these are not real errors, but only trivia, which can show while connecting foreign systems. In the following we collected several snares – and are always thankful for additions and hints.

Hardly any problems show, if router, OPC- and data base server run on the same computer. In the distributed system the right DCOM configuration is important for the OPC server, data base server allow or forbid even the remote access.

Note: Use in any case the tests offered in the management: “Test Connection” for the data base access data and “Test” for the mail server access. The OPC link works, if you can browse the OPC server in the management (see below). Please note, that configuration and service may run under a different user context and act differently.

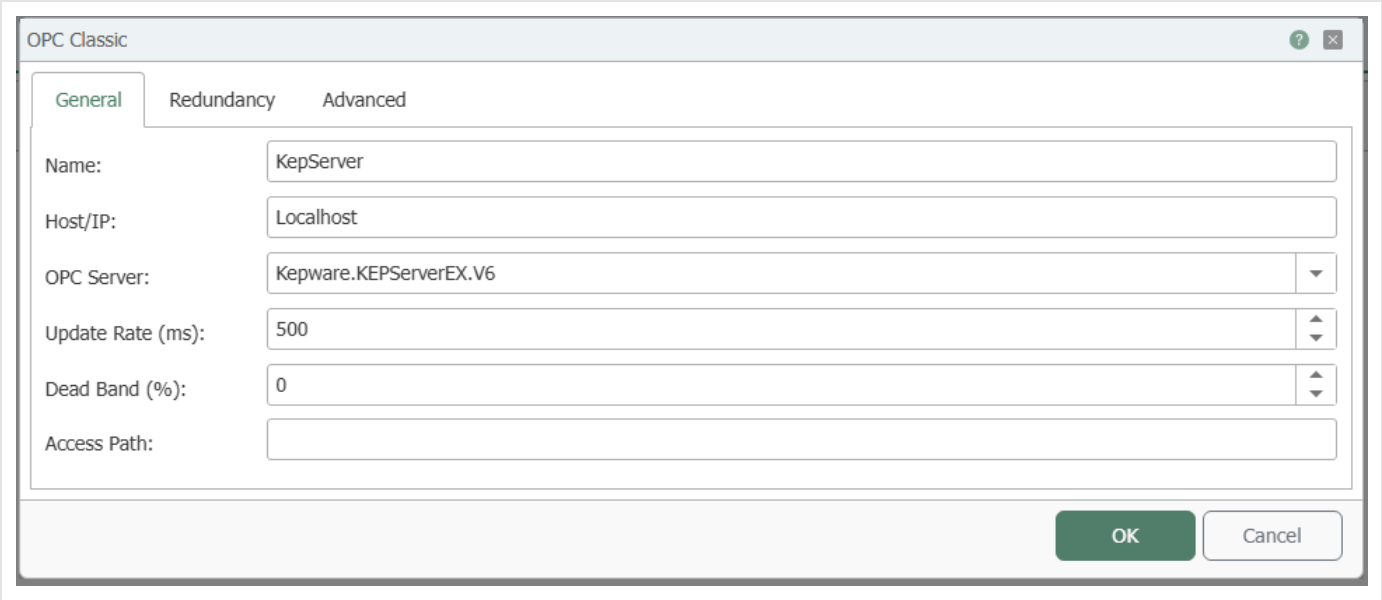
In the Basic Set Ups the log levels “Error” and “Warning” should be activated as recommended. In case of a runtime error, call the log data with the tray icon and analyze the error message.

OPC Links

Sources for errors, which limit the access on local OPC server, are not known yet. However accessing distributed OPC server can be tricky.

Connection can´t be set up – remote only

The property “OPC-Server” gives out *<error browsing opc-servers>*.



OPC Classic Transfer Object

Check if the DCOM Configuration is correct:

- Does the configuration environment run under the same user, who has DCOM access to the OPC server?
- Are the components OPC Enum and OPC server on the destinated computer cleared for the user?
- Is DCOM in general activated on the destinated computer?
- Are the local firewalls shut down on all computers involved?
- Of course the destinated computer is switched on and accessable!

Connection is set up, but the OPC link doesn´t show in the OPC transfer object.

The OPC server could be linked in the configuration, but the OPC link doesn´t show up within the connection projecting in the OPC transfer object, or an error is given out.

- You have to save the OPC link with the “OK”button, before you can use it.

Tag browser isn´t able to connect with the OPC server

As far as the OPC link was configurated without problems, check if the OPC server has just been switched off manually, or has been initialised.

- Is the computer with the OPC server switched on an accessable?

No data transfer can be done during run time

The connection could be projected easily during concept time. During run time there is no transfer. Call the log data with the tray icon. Search for the error message at the end of the data. Does it look like this?

```
16.07.10 14:49:06.194 [Error:OPCConnectionManager]
  Wasn´t able to  connect<Item\> (E_INVALID_ITEM_NAME)
```

- Most likely you´ve written the item name incorrectly, or your OPC server interprets the property Access Path wrong, which is deposited in the OPC link. Choose the item in the tag browser and/or configurate a link without Access Path for testing. Check if your OPC server runs without problems.

Error message in log data:

```
16.07.10 15:19:34.886 [Error:NoChannel] Timeout connect OPC Server!
```

- In distributed systems: Is the computer with the OPC server switched on and accessable?

Error message in log data:

```
16.07.10 15:16:24.694 [Error:OPCConnectionManager] Wasn´t able to connect to OPC server
  <OPC-Server\> on <IP\>.
 (Message: CoCreateInstanceEx: Access denied)
```

- If the connections runs during concept time, most likely the router configuration runs under the right user, but the router service doesn´t. Check if the router service runs under the user, who has DCOM rights on the OPC computer.

Database connections

Not all database systems can be handled here. But maybe one or the other clue will come up with the solution for “your” system.

Connection does not materialize – local and remote

The “Test Connection” button returns an error. The error message often gives the correct hint.

- Is the server named correctly? For example, for MySQL or MS SQL the specification of the IP is sufficient, but for MS SQL Express the instance must be named, e.g..: localhost\SQLEXPRESS
- Is the name of the database spelled correctly? Does the user have access rights to the database? For example, in MS SQL, the databases are assigned to each logon. Ask your database administrator!
- MS SQL: Is the correct authentication set for the user (SQL and Windows on access with user / password)?
- Username and password are correct? Spend attention to upper and lower case!
- Integrated security is enabled. The connection test is not meaningful. Integrated security uses the local Windows or domain user where the program was started to log in. In the configuration environment, this is usually the username with which you logged on to the computer, by contrast, the router service runs under system default (sufficient for local access to MS SQL, not for remote access).
- Are the required services started (MS SQL: SQL Server and SQL Server Browser, via Server Configuration Manager or Control Panel - Administrative Tools - Services).

Connection does not materialize – only remote

The “Test Connection” button returns an error. The error message often gives the correct hint. In addition to the above points, you have to check:

- Is the network connection active? Can you ping the database machine (“Start - Run - ping <*IP*>”)? Ask your network administrator!
- Are the firewalls of the involved computers switched off? We recommend deactivating the Windows firewalls of the participating computers. Otherwise, please ensure correct port release. Ask your network administrator to allow unrestricted communication between the router and the database server.
- Is the server named correctly? For example, for MySQL or MS SQL the specification of the IP is sufficient, but for MS SQL Express the instance must be named, e.g.: 192.168.5.184\SQLEXPRESS or Rechnername\SQLEXPRESS
- Allows the database to allocate remote connections (e.g. in MS SQL this must be activated for the respective user)?
- Are the corresponding protocols activated? (MS SQL: Enable Named Pipes or TCP / IP in Network Configuration and Native Client Configuration in Server Configuration Manager depending on connection)?

Connection is established but no tables are visible (MS SQL)

The button “Test connection” returns OK. However, tables can not be selected in the configuration of the transfer object.

- Does the user have the appropriate rights to the database? – “Listing tables” may not be allowed to the user, depending on the role.
    - At “Login Properties” of the corresponding login in the user mapping for the required database assign to “db_datareader” and “db_datawriter”. Attention: “db_owner” is required to run stored procedures.. – Ask your database administrator.
    - Of course, the user can also be assigned administrator rights on the database or on the system (e. g. the role “sysadmin”). But this should only happen for a short time for troubleshooting

Stored Procedures can not be executed. (MS SQL)

- Does the user have the appropriate rights to the database?
    - At “Login Properties” of the corresponding login in the user mapping for the required database “db_owner”

Stored Procedures are canceled.

- Are Command Time-out of the connection **and** Time-out of the corresponding connection set correctly?
  - Changing the time out of the corresponding database in Storage.
  - Increase the value of Time-out of the corresponding database in Storage.

Connection is established, but the database connection does not appear in the database transfer object.

The connection test was successful, but the database connection in the connection configuration in the database transfer object does not appear or an error is returned.

- You must save the database connection via the “OK” button before it is available.

At runtime, no data transfer occurs

The connection could be easily made at design time. At runtime, the status display indicates a connection error. Open the log file via the tray icon. Look for the error message at the bottom of the file. Look for the error message at the bottom of the file.

```
14.07.10 17:33:42.228 [ErrorSqlConstantConnection]
Error while opening the sql connection: System.Data.SqlClient.SqlException:
Error logging in for the user 'NT-AUTORITÄT\ANONYMOUS-ANMELDUNG'
```

- If you access the database server remotely and have “built-in security” enabled, you must not run the service as a system user. Configure the service so, that it runs at the domain user.

SQL-Selects

No data transfer from the database - but also no transfer error

Your connection should read data from the database. You have configured a DB object with SQL Select and Filter for this. The OPC Router status does not show an error, but no data is transmitted at the set time.

- In the DB transfer object, there is the tab “Query”. Set the option here “With empty SELECT - Return read error”. If the cause for the failed transfers is an empty SELECT result, the status will now display an error.

Creating diagnosis

Create a diagnostic report with the router surface

Using the menu “File - Creating diagnosis”, you can select different areas that should be taken into account for the diagnosis and placed in a folder as a complete package.

E-Mail Settings

Save as	Enter the path where the diagnostic report will be saved.
Project files	Activate the checkbox if the project configuration is to be exported.
Runtime database	Activate the checkbox if you want to create a backup at runtime for the diagnostic report.
State database	Activate the checkbox if you want to make a backup of the status of the database for the diagnostic report.
Log files	Activate the checkbox if you want to include the log files for the report.
Include windows event log	Select the checkbox if you want the windows event log to be included in the report.
Include devices	Enable the checkbox if the device data should be included in the report.
Logfile of last ... days	Activate the checkbox if you want to include the log files of the last X days for the report.

The file is packed in the desired directory and can be opened with any zip program.

Connections

After having configured all plug-ins needed, you can start defining connections. As only one Router project can be loaded as runtime configuration, you want to project all connections within one single project file. You may, however, sort the connections into different folders (=connection group) (similar to Windows Explorer). Every single connection or connection group can be activated or deactivated so that it is executed (or not executed) during runtime.

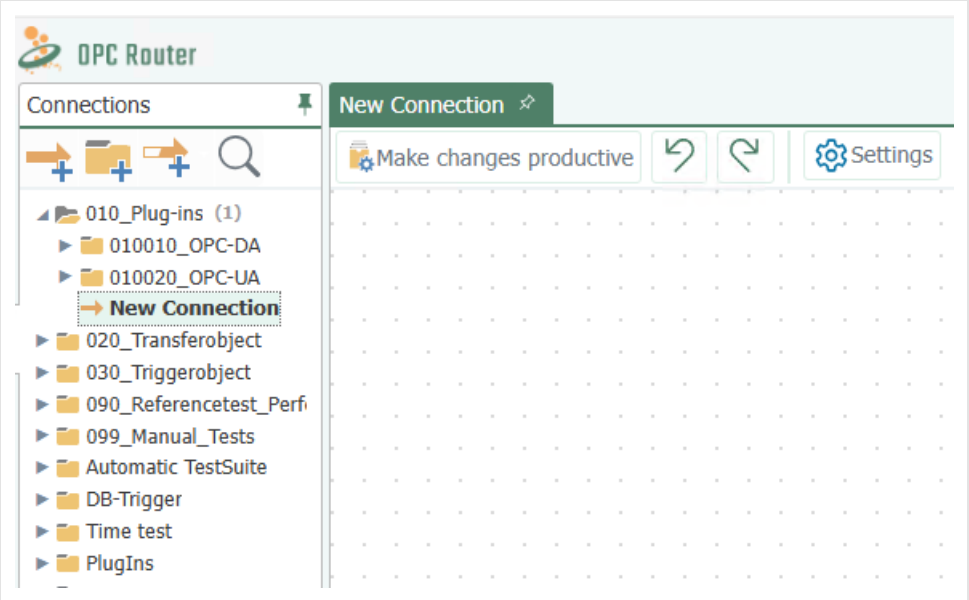
Important: We recommend to backup a productive configuration before editing. Thus, you can revert to a running project should any error occur after editing.

Tip: You can move the drawing area in all areas with the middle mouse button held down.

Creating Connections

To create a connection, go to the Snap-in area and select the connections snap-in:

1. Go to the “New Connections” button.
2. Insert a name. The name has to be unique within the project.
3. Confirm with “Enter”.



Create a connection

The connection has been created and opened in main window. Existing connections can be opened by double-click in the project configuration tree. The currently called element is displayed in bold in the directory tree.

Note: Right-clicking on Local Service will also bring you to the Settings area.

Creating Connection Groups

A connection group combines several connections. Groups can be nested as required, e. g. like folders in Windows Explorer. (De)activation of a group (de)activates every single connection of that particular group. You can, however, e. g. deactivate a group and reactivate single connections within that group afterward.

To create a connection group, go to the connection snap-in

1. Go to the connection button “New folder”.
2. Insert a name. The name has to be unique within the project.
3. Confirm with “Enter”.

Manage Connections and Connection Groups

You can copy, paste, move, rename or delete connections and connection groups, similar to Windows Explorer. However, only one object can be edited at once. Selection of multiple elements is not possible.

1. Select the connection (group) to be edited.
2. Choose the desired action
    - from the menu “Connections”
    - from the connection’s context menu (right mouse click)

3. The following actions are possible:
    - Copy: You can create connection groups, connections, template instances, templates with the right mouse button

“Copy” and “Paste”.

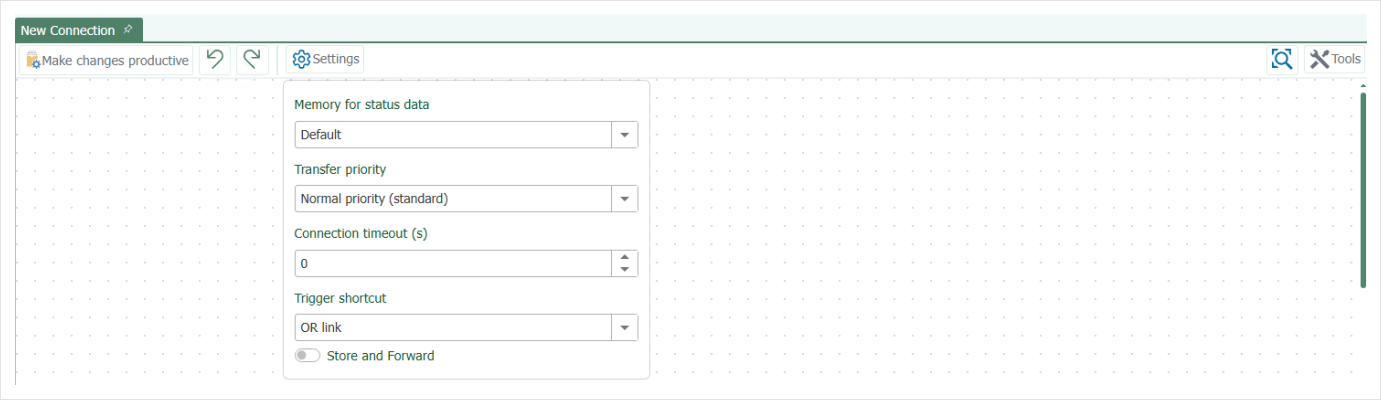
- Delete: After selecting the right mouse click on the connection, go to the action “Delete” and confirm with “Enter”.
- Rename: After having selected “Rename” enter a new name and confirm with “Enter”.
- Move: Move by copying the element and delete the original afterward or drag & drop with mouse.

If you right-click a parent folder in the tree structure and select “Expand All” or “Collapse All” from the menu, all subfolders and their connections are displayed or hidden.

Pop-up-Menu

Above the connections area you will find another pop-up menu:

![Pop-up menue](</assets/menue_verbindungen (5) (9) (9) (2) (1) (4).png>)



- The “Set connection productive” button activates the selected connection.
- Use the “Trigger shortcut” button to select whether the links in your connection should be AND or OR links.
- With the “Connection properties” button you can select the form in which the transferred values are to be recorded. You can also define a timeout: Time (in ms) that the connection setup may take before a connection error is returned.
- With the “Zoom” button you can enlarge or reduce the size of your connection.
- With the “Export View” button, you can export your connection as an image file or copy it to the clipboard as an image or XML.The images can be saved as Bitmap, PNG, JPEG or GIF.
- With the “Connection arrow add mode” button you can create single or multiple connection arrows. Arrows are used to connect the source and target objects. If several arrows are to be created at the same time, an attempt is made to link all data points of the target and source objects using their data point names. The icon changes depending on which mode you have selected.
    1. “Create a single connection”: This is the default selection. You only need to draw a single connecting arrow. If you hold down the Ctrl key in this mode and drag a connection arrow, several connection arrows will be created once.
    2. “Create multiple connection arrows”: In this mode several connection arrows are drawn. While dragging an arrow, only one arrow is displayed. As soon as the arrowhead is placed on a transfer object, the other arrows are created automatically. If the arrowhead points to an Input Transfer Object Item and either the original Transfer Object Item or the target Transfer Object Item does not get an automatic arrow, this arrow is also added, even if the objects do not match. This can be prevented by dragging the arrow onto the header of a transfer object.
    3. “Create multiple connection arrows once”: Behaves like the “Create multiple connection arrows” mode. After dragging an arrow, the system returns to the “Create a single connection” mode.

- With the “Item sorter” button you can activate automatic sorting of the connection arrows. This causes the system try to arrange the connection arrows so that they do not overlap. This is only a display option. The items are not "really" moved, but only displayed in a different position. The option can therefore be activated and deactivated without hesitation. This option has no effect on how the connection is processed.

Activate/Deactivate

It is required to save all connections in one single project file. You can, however, deactivate connections or connection groups within this file.

Activate/Deactivate Connections

1. Go to the Snap-in bar “Go productive”.
2. Select all connections and / or connection groups that you want to activate by selecting a checkmark next to the connections / connection groups.
3. Click on button “Go productive”.

Activate/Deactivate Connection Groups

1. Go to the Snap-in bar “Connections”.
2. Right-click on the connections you want to activate / deactivate.

This command de/activates all connections and subgroups of this group. You can open the connection group and reactivate single connections or subgroups if necessary.

If you right-click on the folder of your connections, you can activate or deactivate all connections in your folder at once via the menu with the properties “Activate all connections” or “Deactivate all connections”.

How to project

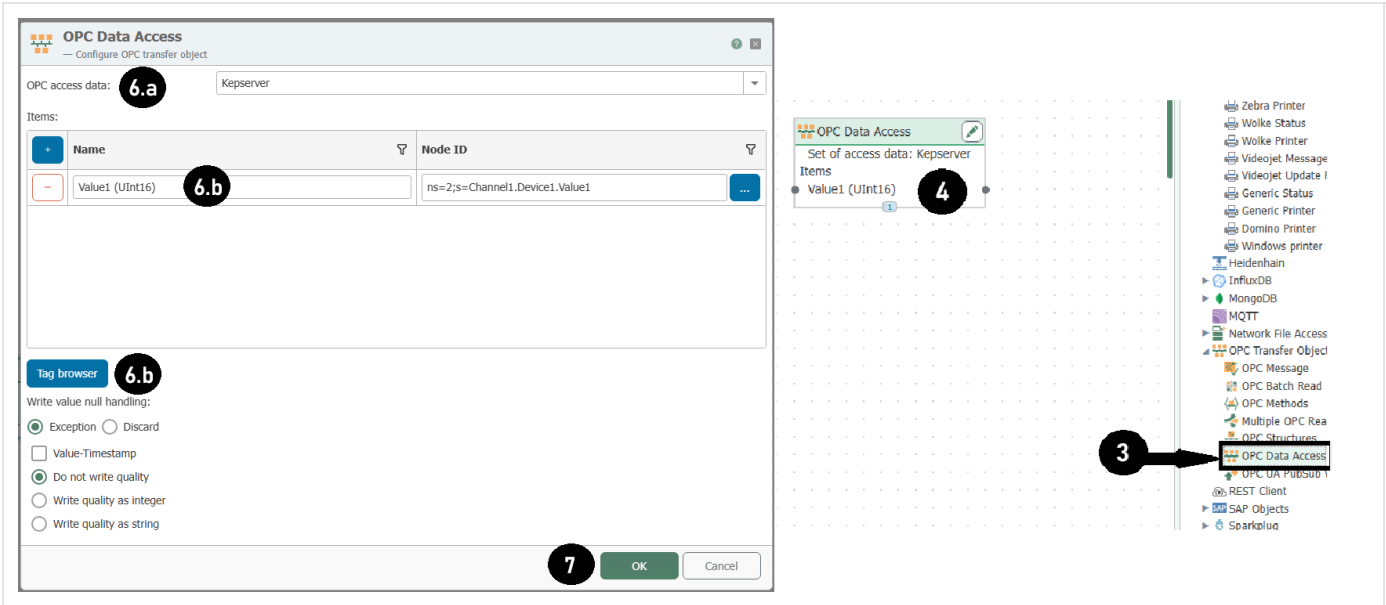
A connection contains three transfer objects at least: a source, a destination and a trigger that triggers the transfer should its conditions occur. You can define a free number of sources and destinations which can be accessed in multiple steps. You can define several triggers as will. Triggers might be linked with AND or OR, triggering either if all conditions occur or if at least one condition occurs.

The transfer objects may be found in the right-hand toolbox.

Right mouse-click on a transfer objects with in project allows editing, copying, pasting, cutting, deleting.

Note: To project a connection, all plug-ins needed have to be configured.

1. Create a new connection.
2. Give your connection a suitable name.
3. Choose the transfer object you want to define as source from the toolbox “Transfer Objects” (e. g. “OPC UA/DA” if you want to read data from an OPC Server).
4. Place the source in the connection (main window) per drag and drop.
5. Open the transfer object with double-click.
6. Set the properties. The properties of the transfer objects are explained in the next chapters.
    1. Selection of the OPC connection.
    2. Alternatively: Enter data points’ name or open the tag browser.
    3. Transfer data points to the transfer object by double-clicking.
7. Confirm with “OK”.



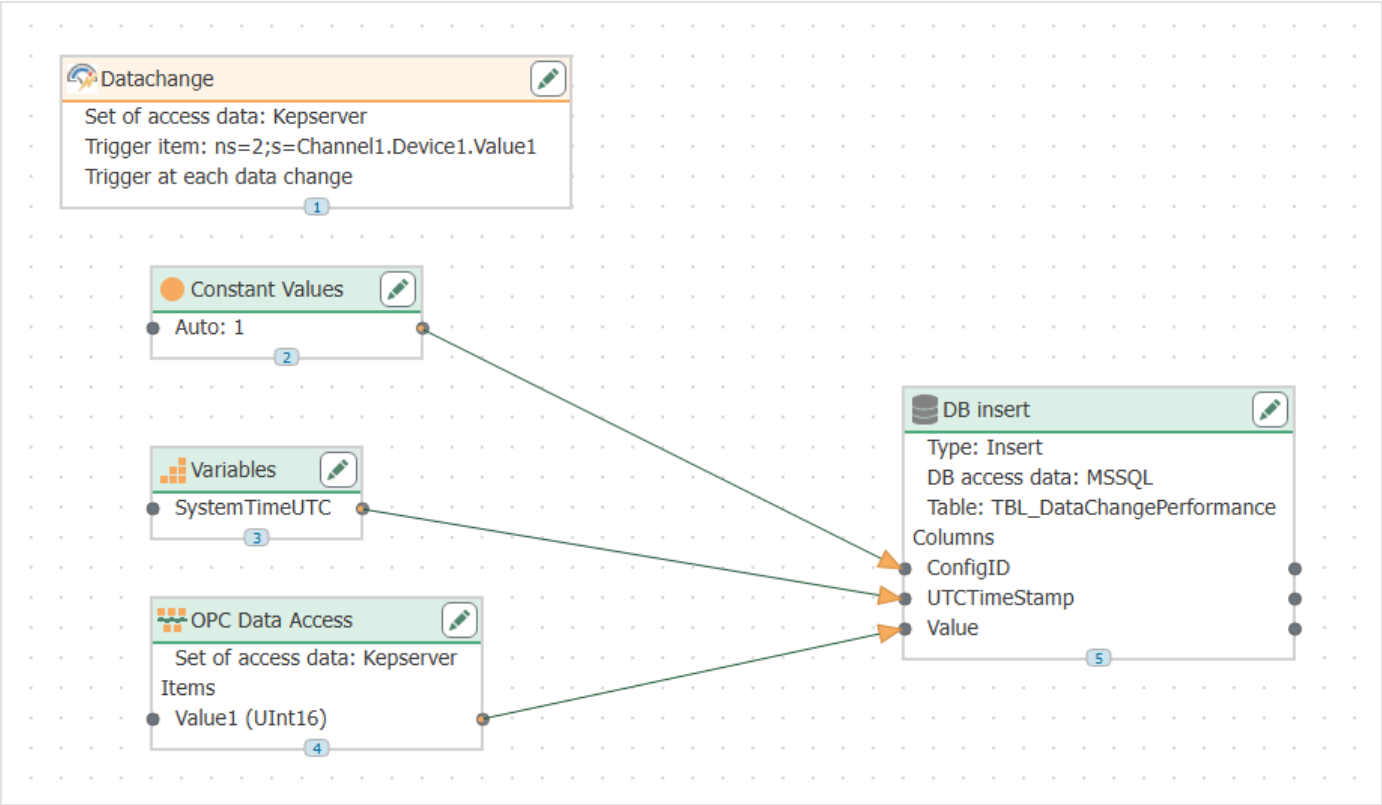
OPC Data Access Transfer Object Configuration

1. Project a destination, e. g. a database transfer object.
    1. Choose the DB transfer object from toolbox “Transfer Objects”.
    2. Place it within projection area.
    3. Open transfer object with double-click.
    4. Choose a set of DB access data.
    5. Choose a type for this transfer object. If database servers as destination, that would be either INSERT or UPDATE. Choose INSERT in this case.
    6. Choose the table you want to write to.
    7. Choose the columns you want to write to. Sort with “Up” and “Down” to keep the overview in the configuration.
    8. Confirm with “OK”.

Database Transfer Object Configuration

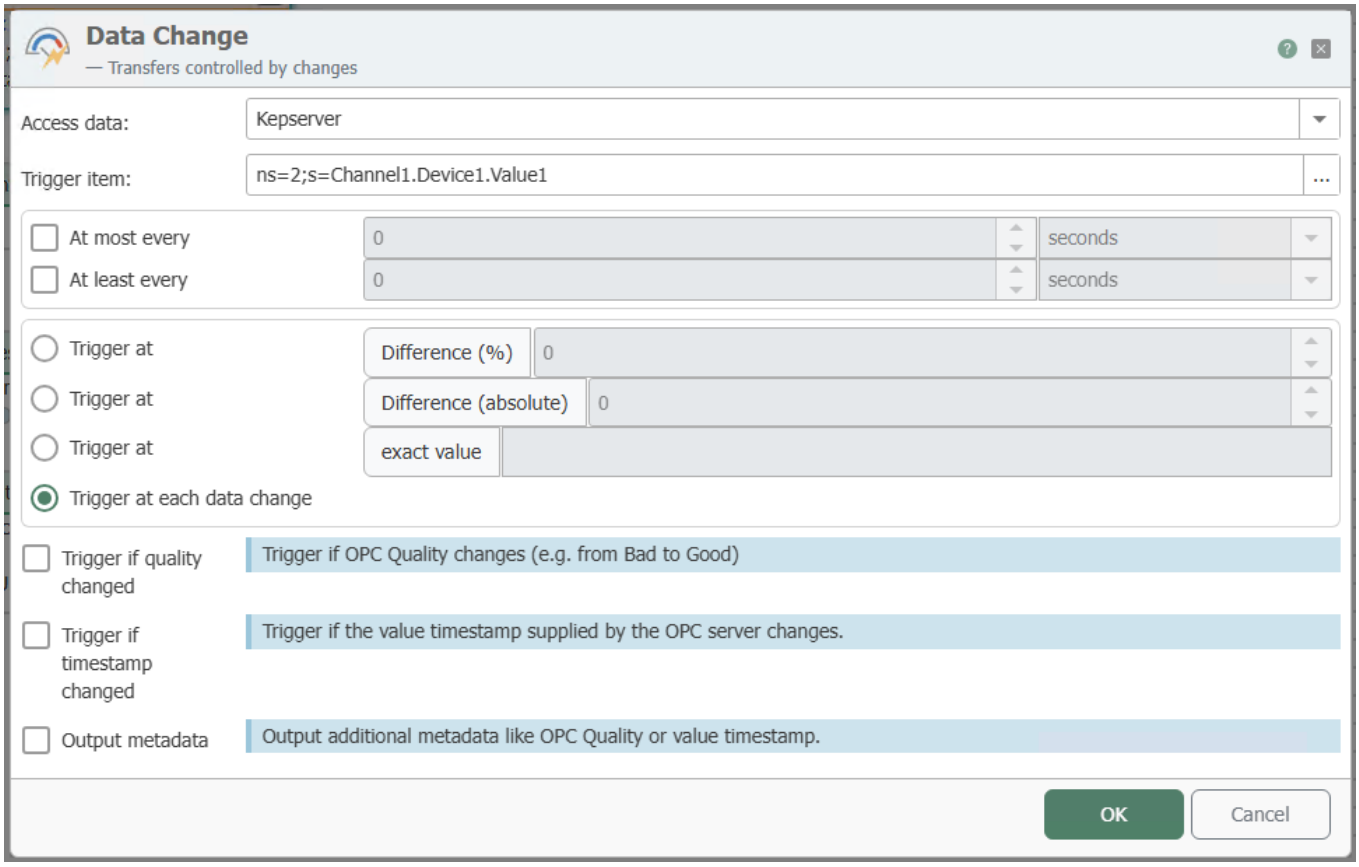
Tip: Copy/Paste is possible in the projection of the transfer objects.

1. Drag connections arrows between transfer object items of the source and of the destination. Click onto the source's item, hold mouse button down, drag over destination item. Drop by releasing mouse button. Finished! You can delete a connection arrow by right-clicking the arrow and clicking "Delete". At this point you also have the possibility to delete several connection arrows at once. To do this, go to the entry "Delete several" and select an option.



1. Configure a trigger:
    1. Choose a trigger from the toolbox.
    2. Place it within projection area.

3. Open with double-click.
4. Configure the trigger.



Data Change Trigger Configuration

Tip: Press “Shift” to drag all columns from a transfer object to a single column of the other transfer object.

Working with templates

Many congenial connections are at best be projected with the template function. A typical example is the collection of dial count in a database, wich e. g. only differ by OPC item and the ID of the collection of data. With templates you work as follows:

1. You create a template using the snap-in “Templates”, in which you configure all immutable parts of the connection. At the adjustable parts define template variables. That can be OPC items, names or columns of tables, names of access data, constants or just parts of these names. The examples in the following parts illustrate your possibilities.

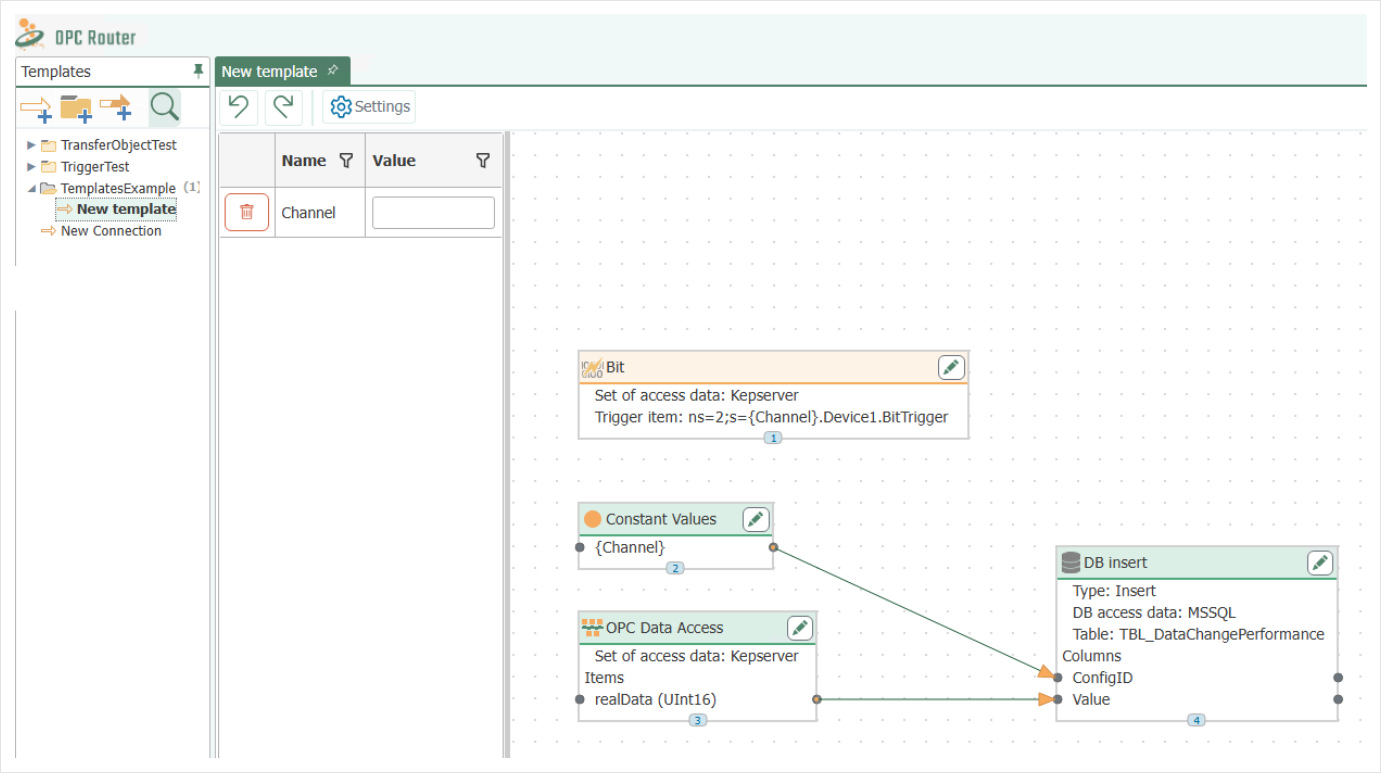
info **Note:** To avoid errors in the configuration, the template variables are type-safe. A combination of multiple template variables or template variables with characters (e.g. `g\_atDBDataBlock\[{Variable})]` is only in type “string” possible.

1. From this template you create connections by going to the snap-in “Connections” and selecting the button “Create instance”.
    - Either you create a single instance after another. You only fill in the value of the template variables and you are done.
    - Or you have a list of template variables and their values created in a CSV data. In that case you only have to import the CSV data. The OPC Router 4 offers a simple table editor to create the list without external program.

2. Make instances productive.

Create Templates

The following example assumes that OPC Server and data base are correctly connected. In this example the tags within the OPC Server are projected custom-fit, so that only one template variable is necessary, to collect the data.



Example of a template

- In the snap-in “Templates”, select the “New template” button or right-click on an existing entry and select there “New template”.
- Name it, e. g. like here “Wago Channel”.
- Define the connection properties for the template. The Properties will be applied for all instances.
- Project the connection. In this example we want to collect data from a Wago controller The OPC Server is projected to name the items (tags) identically and to put them into different access paths. This path is to be created as template variable. So you have to place an OPC transfer object in the template and open it via double click. It opens a window for editing template variables at the same time.
- Create a template variable Channel. To do that, click “Add” in the template variable dialouge and state Name (“Channel”) and Type (“Integer”).
- Fill in the properties. First insert the template variable via drag & drop. It is integrated into the transfer object with the correct syntax. Then add the static string.

OPC Access Data	*<Name of the Access Data from the Plug-In Configuration>*
OPC-Items	g_atDBDataBlock**\[**{Channel}**]**.rRealData

- Close the OPC transfer object.
- Project a data base transfer object for recording. At this point a table TBL_Values is to be used for collecting. The test series is defined by the field ConfigID, the time stamp is given from the data base.

DB-Access Data	*<Name Of the Access Data From the Plug-In Configuration>*

Type	Insert
Table	TBL_DataChangePerformance
Used Columns	ConfigID, Value

- Project a constant value with the value Channel. The constant must be of type Auto, so that at runtime the string can be converted to an integer (the ConfigID column is expected to do so).
- Connect the constant value to the ConfigID and the item with the value.
- Project a Bit Trigger, to start the transfer. Activate either reset bit or insert a response bit, depending on the PLC program.

OPC-Item	`g\_atDBDataBlock\*\*\[**{Channel}**]\*\*.tHandshakeBits.xTrigger`

With this the template is ready. You can sort templates to groups just like normal connections.

Handling template variables in objects

Template variables can be added

- via the “Add” button in the object dialogs.
- via the buttons with the blue plus sign next to the individual fields. This creates and enters a new variable of the correct type.

The button with the red minus sign removes the template variables from the fields.

Existing template variables are drag & drop into the corresponding field.

Template variables with OPC-UA

Please note for the projection:

- If the OPC data access is a template variable itself, the router can´t recognize the specification. It only states the property Item. If you create a connection from the template and the OPC data access is a OPC-UA data access, the Node ID must be stated under Item.
- Are the data accesses from OPC-Triggers (Bit, Datachange, Message, Grenzwert) template variables, the Node ID has to be stated as well, while creating a connection from the template.

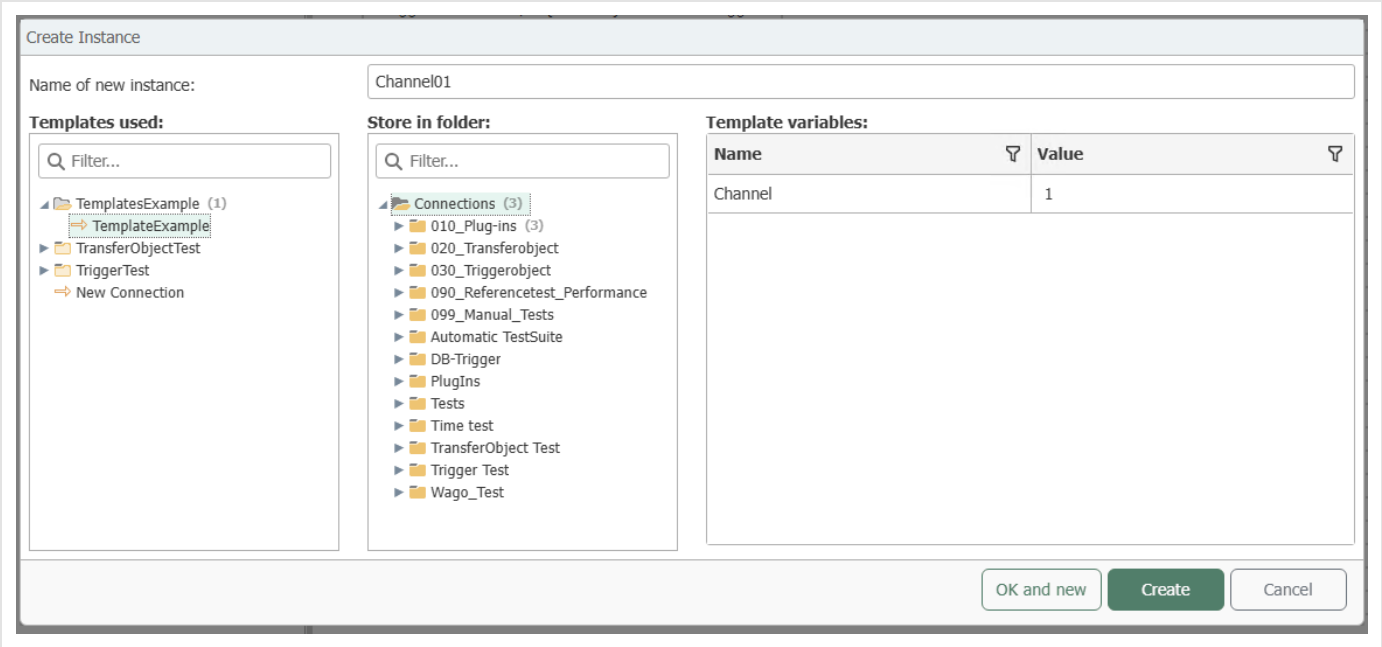
Create templates from connections

You can generate a template from a connection:

1. In the snap-in pane, select a connection.
2. Right-click to select “Create template from connection”.
3. Give the new template a unique name and confirm with “OK”.
4. If you activate the “Open template directly” checkbox at the bottom of the window, the template will be opened after it has been created.

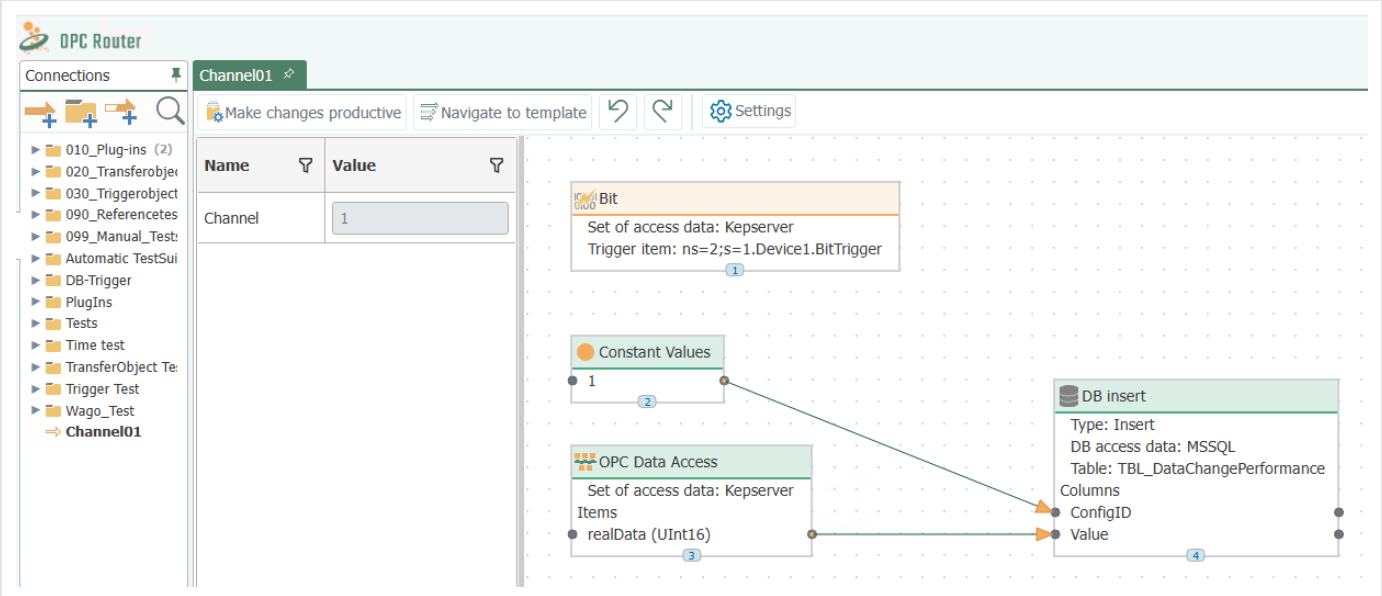
Creating Instances

To use the template for the data transfer, you have to create instances, thus actual connections.



Create instances

1. Select in the snap-in-menu “Templates” and choose “Create Instance”.
2. Name the template, e. g. “Aquisition_Ch1”
3. Select the desired template.
4. Select the desired storage location.
5. Within the connection you can´t change anything, you just have to state the value for the variables. In this case everything is done by entering “1”. So just put in the “1” and close the connection.
6. Create a connection for channel 2, 3, 4, 5 ... the same way.
7. Make the instances productive.



Create instances

The instances have been created, the data points are named correctly. The “Edit” button allows you to subsequently change the values of the template variables. You can also override some connection settings.

Attention: For template variables with relative time specifications (eg intervals in the time trigger), the time unit must also be

written (ms, s, m, h, d, month, year).

Note: If you change the template later, the instances will be adjusted.

If a template has been selected and you select “New instance” the name “New instance” is preset and variables and template displayed.

Creating Instances from Data

If you want to create many links from a template, you can

- import a prepared Excel or CSV file
- or create a corresponding table in the OPC Router itself

Please note, that the names of the template variables have to be the same in CSV data and router template, and that in CSV data columns are seperated with a semicolon “;” . The CSV data is structured as follows:

InstanceName	Variable1	Variable2	...
Connection Group.Connection Name1	Value	Value	Value
Connection Name2	Value	Value	Value
Connection Group.Sub Group1.Connection Name3	Value	Value	Value
...	Value	Value	Value

There are the following time-offsets:

- 1ms for milliseconds
- 1s for seconds
- 1m for minutes
- 1h for hours
- 1d for days

There must be no space between the number and the unit.

Stating a Connection Group is optional. You can spread the connections like this to several groups and sub groups.

1. In the “Templates” or “Connections” snap-in, select the “Create Instance” drop-down button and then “Create Multiple Instances”.
2. Select the template to use
3. Select the destination folder.
4. Enter the values or open your CSV file.
5. Determine whether existing connections will be retained or over-written.
6. Select the “Create instance” button.
7. Make the new instances productive.

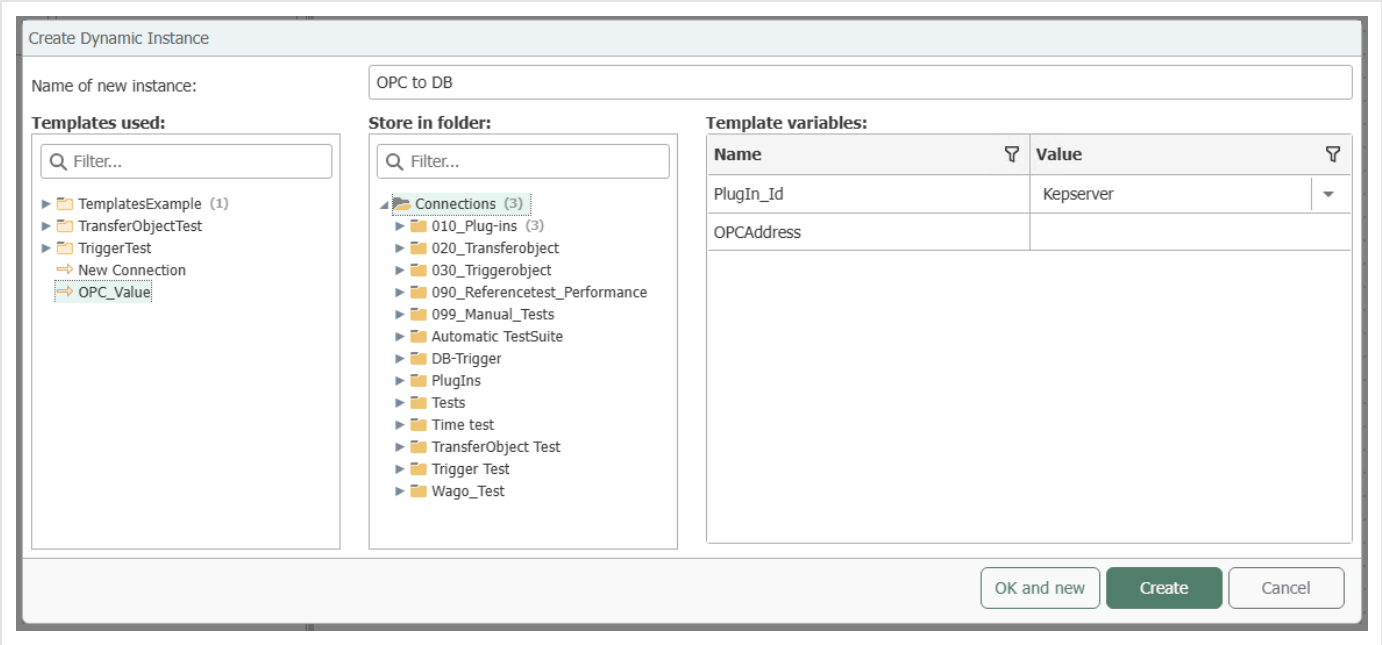
Note: If you change the template later, all instances will be adjusted.

Creating dynamic instances

Attention: Dynamic instances are tables which are created in external MS SQL tables.

If you want to generate dynamic connections please select the “Connections” snap-in and use the option “create instance” point “create dynamic instance”

If you selected like described, you will see a dialog and you can, like you know from normal instances, select a name and default values for the dynamic instance.



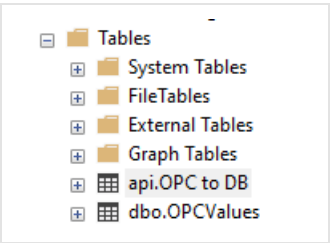
Create Dynamic Instance

If you entered everything you need, confirmed with “Create” and published the dynamic instance you will get a new table in the database. These tables will use the “api” schema. The table is named like the instance you created. Please do not change this name!

Note: We recommend to create a database user which has access to the api tables only. For this you will find the “apiRole” in the database.

![Instanzen erzeugen](</assets/vorlage_dyn_tabelle (9) (6) (10).png>)

![Datenbank](</assets/vorlage_dyn_tabelle_II (9) (1) (7).png>)



Create Dynamic Instance

Within these tables you will find a database field for every variable of your instance. Additional you will find the fields [ID], [OpcR_Changed] and [InstanceName]. To create a new connection you need to insert a new row to this table and fill out the variables and the [InstanceName] field. If you didn't entered a value for a variable, the default value will be selected automatically.

	Id	OpcR_Changed	InstanceName	PlugIn_Id	OPCAddress
▶	1	2025-01-28 11:5...	DynamicInstance1	Kepserver	ns=2;s=Channel1.Device1.Temperature
	2	2025-01-28 11:5...	DynamicInstance2	Kepserver	ns=2;s=Channel1.Device1.Pressure
*	NULL	NULL	NULL	NULL	NULL

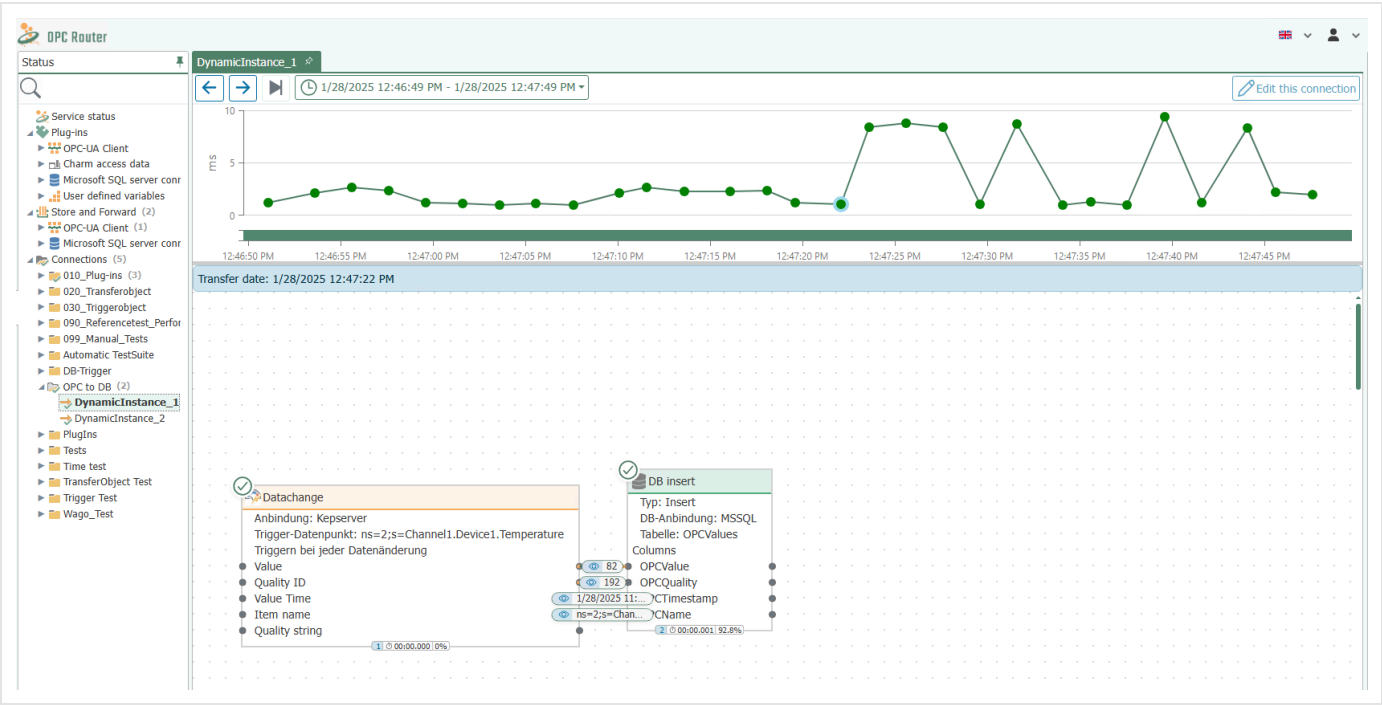
Edit of the tabel "api.OPC to DB"

Note: As you can see in the picture, some connections need a PlugIn_Id. You will find the IDs of your plugins by using the view “api.View_PlugInInstances” of the “OPCRouterRuntime” database.

The columns [ID] and [OpcR_Changed] are filled out by the OPC Router, the value of [InstanceName] needs to be unique and is the name of your connection. You can use backslashes within the instance name to create folders for the runtime.

Attention: The created connections are active immediatley and don't need to get publishes again!

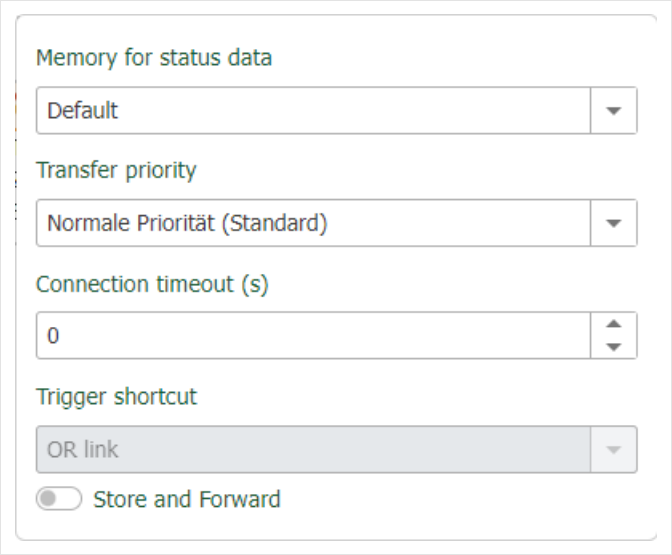
Maybe you need to refresh the State snapin to display the dynamic instances.



Productive Dynamic Instance

Settings for connections and templates

There is a "Settings" button in the editing view for each connection and template. Some important settings for the connection can be made here.



The following properties can be configured:

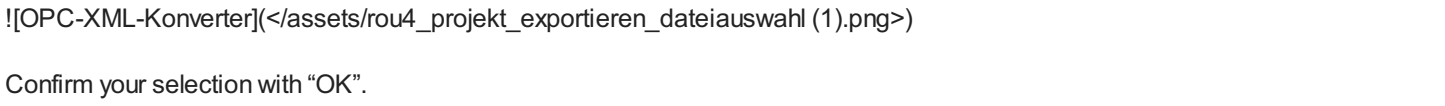
Memory for status data	The diagnostic memory for the connection execution in which this connection will store the transmitted data can be selected here. The default storage is selected by default. Further information about these memories, e.g. how to change the default memory and create your own, can be found here: Create a Connection Execution Diagnostics Storage
Transfer priority	Determine the connection priority. See:
Connection timeout	Default = 0 (300 seconds) The connection timeout determines the maximum duration of a connection.
Trigger link	Interchangeable between AND and OR

Export OPC Router 4 project

You open the “Export OPC Router 4 project” function in the menu bar under “File”.

![OPC-XML-Konverter](</assets/rou4_projekt_importieren (9) (7) (1) (1) (1) (8).png>)

Select a file to export to and select all the connections and connections you want to export. If you select templates and connections, the corresponding structures are automatically selected as well.



Confirm your selection with “OK”.

Import OPC Router 4 project

You open the “Import OPC Router 4 project” function in the menu bar under “File”.

![OPC-XML-Konverter](</assets/rou4_projekt_importieren (9) (7) (1) (1) (1) (8).png>)

Select a file to import and resolve the conflicts of the bindings in the opening mask. Only connections in the conflict list are displayed, connections are automatically renamed and a consecutive number is added. In order to solve conflicts, you have the opportunity to eliminate all conflicts together in the mask that opens. Depending on which button you select, the appropriate selection is made for all conflicts. Of course you can continue to make individual decisions for individual connections, the respective button then "overwrites" each time previously selected option:

- Select “Keep all” if you want to use the application you are currently using.
- Choose “Rename all” if you want to keep your existing connection and want to receive the imported connection in parallel. Select a name for it until the red circle with the cross disappears.
- Choose “Replace all” if your existing connection is to be replaced by the imported connection.

![OPC-XML-Konverter](</assets/rou4_projekt_importieren_konflikte (2) (2).png>)

Confirm your selection with “OK”.

If connections cannot be carried out sensibly during the import, you get a message “Update import project” with the message “You can import this project if you apply following changes”. If you accept the offered changes, confirm with the “Import with changes” button.

How can I ...

Overview of this chapter:

... read out a current data set?

... read out a new ,current data set?

... read out all data sets which have not been transferred?

... read out a current data set?

To read out a current data set from a data base table, sort the table descending from the time stamp field ans transfer just one data set. The data base has to be polled in the necessary range with a time trigger. In this model the current data set is submitted with every call, read more in the Next Chapter.

Of course you can set up optional strain criterions, e. g. B. if the table has several lists of measurement readings.

Model 1: Time Stamp Sorting without Filter

Required fields/columns:

<Time Stamp>	Time stamp from type date/time, from which is sorted
<Values>	Column(s) with relevant values

Project a data base transfer object with the following properties:

Data base access data	*<Your data base access data from the plug in configuration>*
Type	Select

Tab: General

Table	*<Table which includes your data>*
Available Columns/Used Columns	*<All Value-Columns>*

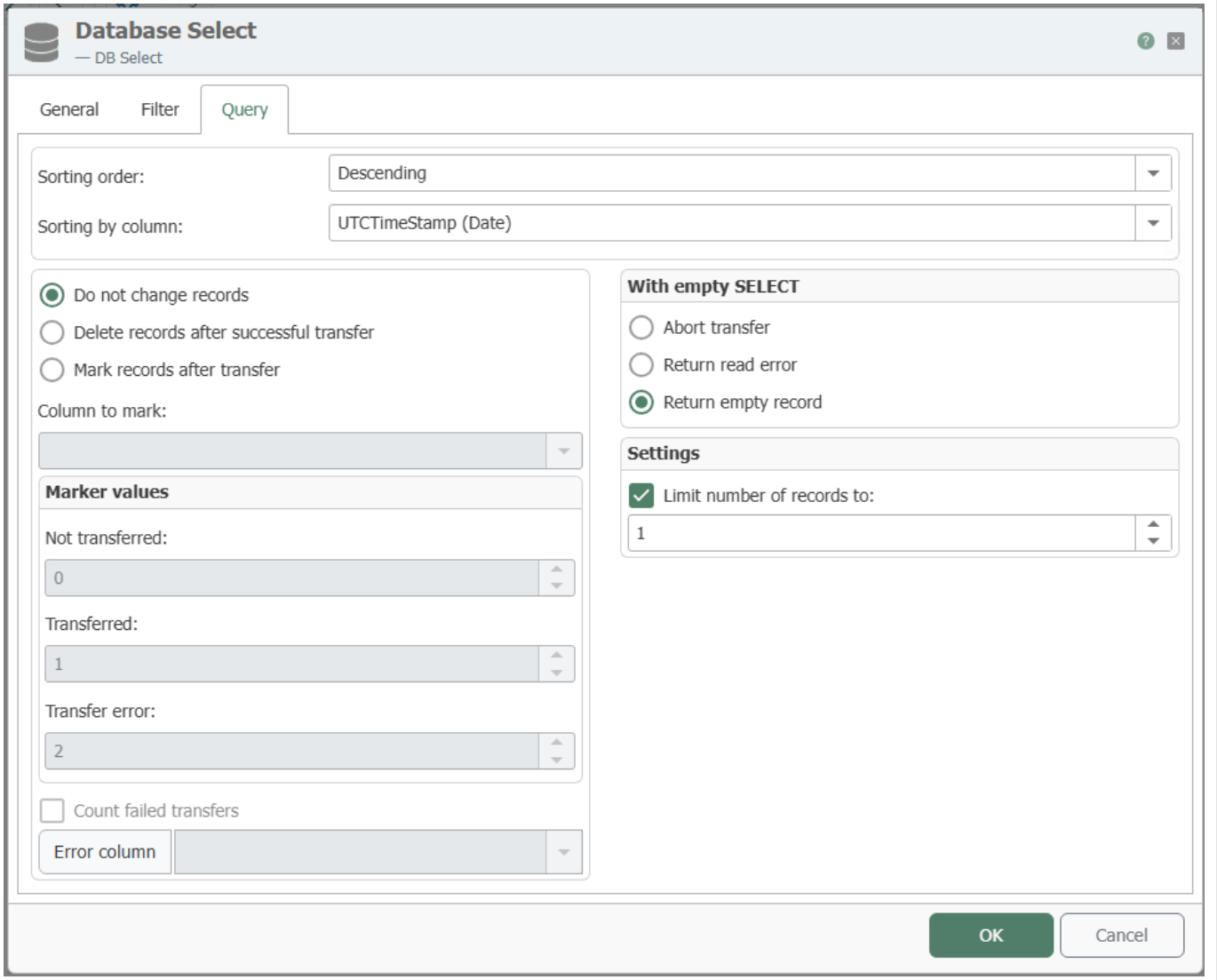
Tab: Filter (Query, Update, Delete)

Filter	No more filters in this model

Tab: Query

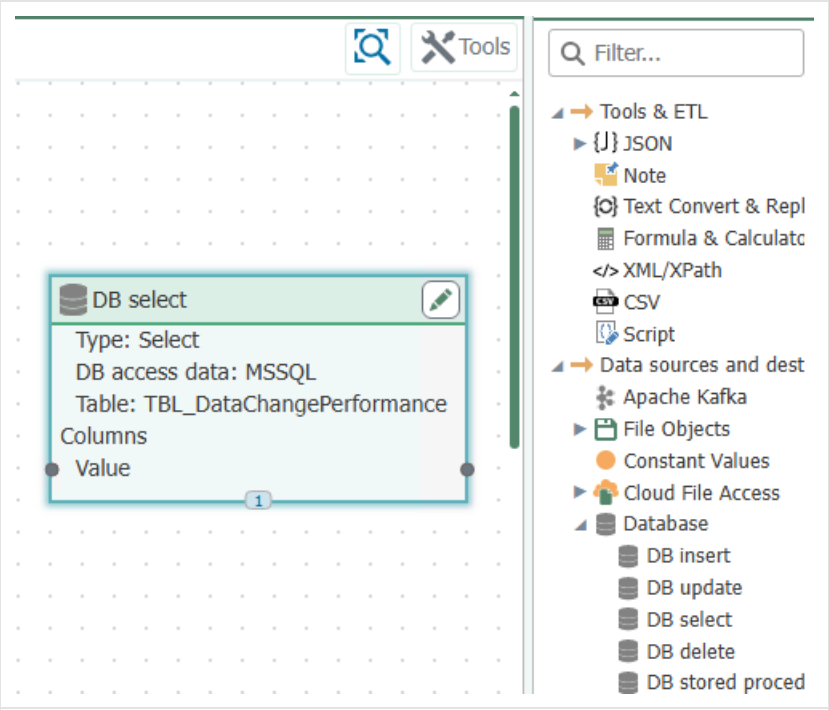
Sorting order	Descending (new value on top)
Sorting by column	*<Time Stamp-Column>*
Do not change records	Activate if the records are not to be changed
Mark transferred records	Activate if transferred datasets are to be used. In the column for marking the corresponding data records can be selected
Delete transferred records	Activate if transferred records should be deleted.

Abort transfer	on demand
Return read error	on demand
Limit number of records	1 (just the first data set = the latest shall be transferred)



Database Select Transfer Object - Query

Your transfer object looks in the connection! like shown in the next image. In the example the column value was used as the value column. It can be used now as data source. The sorting can´t be seen in the connection projecting:



Transfer Object

Project another time trigger with the necessary range, e. g. 10 seconbds. Then every ten seconds the latest data set from the table is transferred, no matter if it was allready transferred before .

Model 2: Time Stamp Sorting with another filter

In this model the table has several test series, which are each defined with a ConfigID. There has to be projected another filter.

Required fields/columns:

<Time Stamp>	Time stamp from type date/time, from which is sorted
ConfigID	ID of the test series
<Values>	Column(s) with relevant values

Project a data base transfer object with the following properties:

Data base access data	<Your data base access data from the plug in configuration;
Type	Select

Tab: General

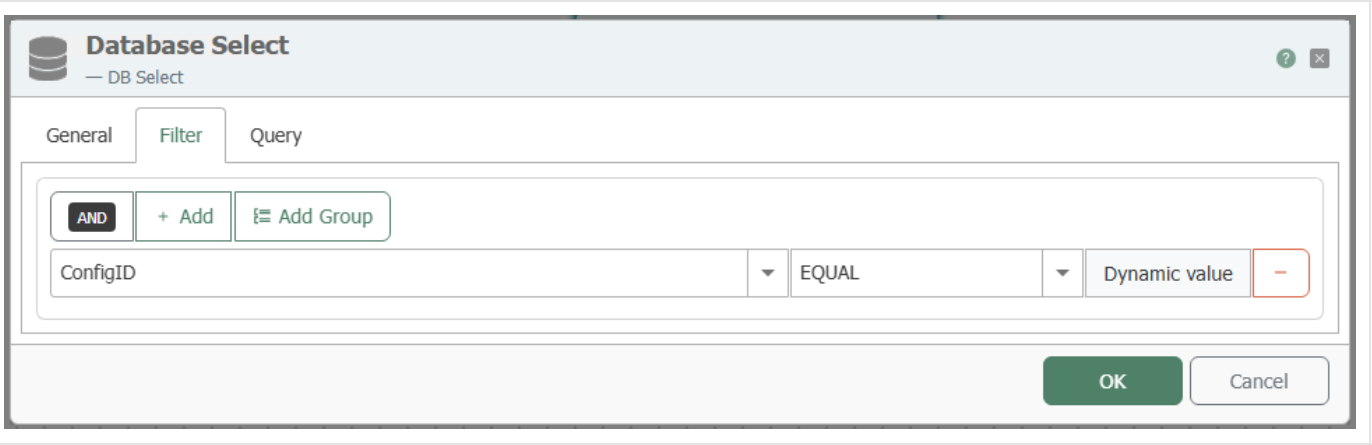
Table	<Table, which contains your data;
Available Columns/Used Columns	*<All Value-Columns>*

Tab: Filter

Filter	“Add Filter” creates a new filter.
Filter/Filter group bearbeiten	**Column:** ConfigID **Relational Operator:** EQUAL
Preview	Here should be stated: WHERE (ConfigID = ?)

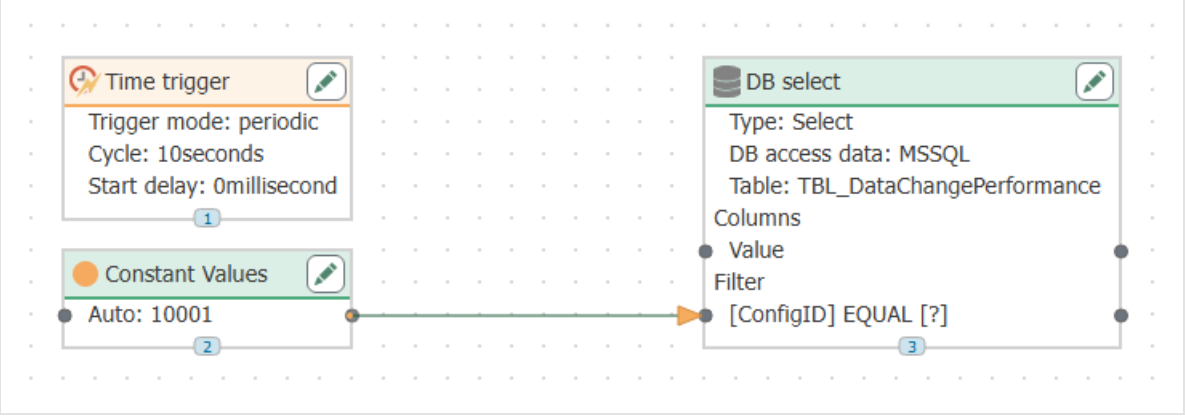
Tab: Query

Sorting by Columns	*<Time Stamp Column>*
Order of Sorting	Descending (new value on top)
Reducing the number of data sets to	1 (just the first data set = the latest shall be transferred)
Error with empty select	on demand
Cancel transfer with empty select	on demand



Database Select Transfer Object - Filter

Your transfer object looks like the one in the image. Remeber to fill in a value on the filter, e. g. with a constant object. The data base transfer object with the filter needs a two step transfer. The first step (blue arrow) defines the filter, the second (green) reads out the value. Because the data destination is variable, it´s not mentioned any further here.



Transfer Objects

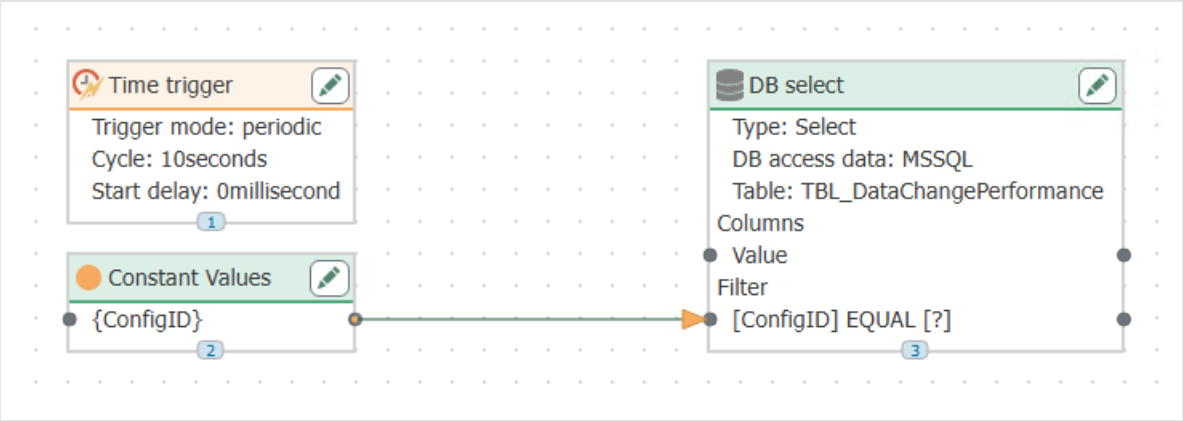
Project another time trigger with the necessary range, e. g. 10 seconds. Then every ten seconds the latest data set from the table will be transferred, no matter if it allready was transferred before or not.

Model 3: Time Stamp Sorting with anonther filter from a Template

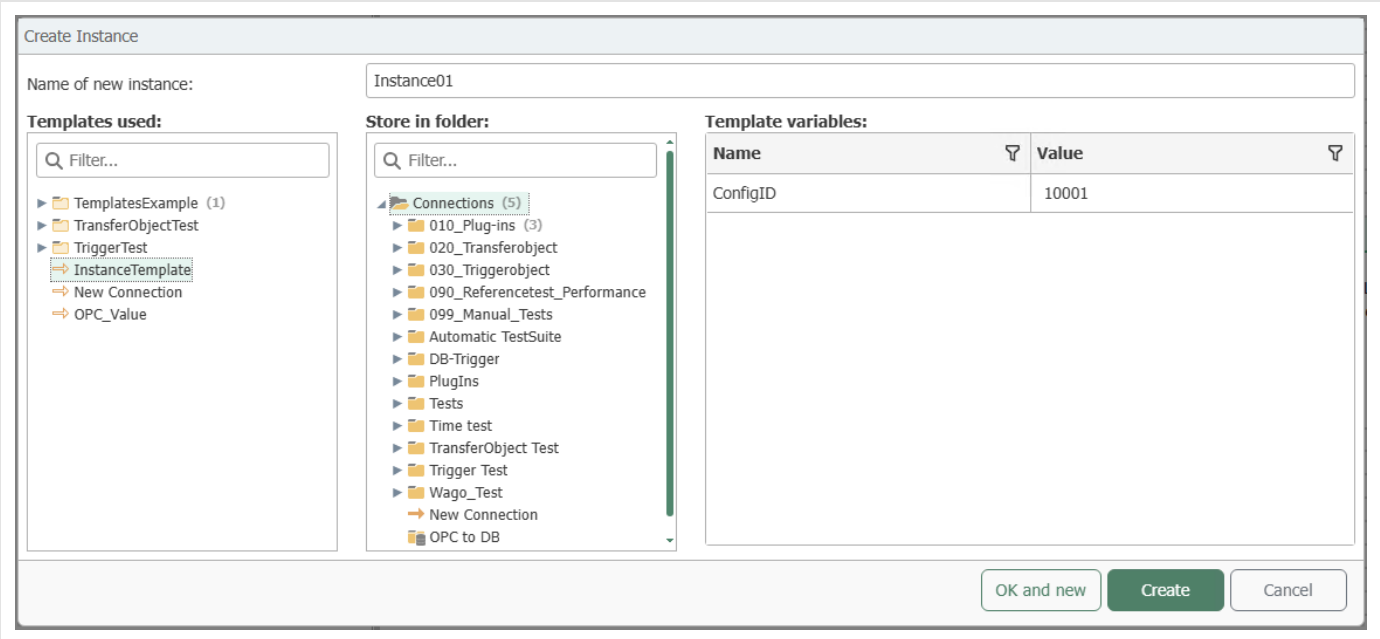
Model 2 can be used as a projection over a connection template.

1. Don´t project a connection, but a connection template with the same settings for the data base transfer object, as you did in model 2.
2. In the constant object you don´t fill in the value, but a template variable [$ConfigID$].
3. Project a time trigger, as you did in model 2.
4. Create an instance from the template and fill in the value for the ConfigID.

This way you create quite simply connections for all existing test series (=ConfigIDs): One instance per ConfigID. Image above: Template; Image below Instance.



Transfer Objects - Template



Create a instance

... Reading Out a New, Current Data Set?

In the preceding example the current data set from the table or the filtered result was transfered cyclicly. The following shows, how the current data set is transfered just when it´s new, thus when it hasn´t been transfered before. For this you need a second column beside the time stamp column, in which a succeeded data transfer is recorded. This example includes an additional filter (ConfigID used as ID of the test series) and a filter on the column Transfer, to filter just data sets which hasn´t been transfered before. Just as the example before this connection could be realised alternatively as a template/ instance. Shown here is just the simple model.

Required fields/columns:

<Time Stamp>	Time stamp from type date/ time, from which is sorted
<Transfer>	Column, in which a successfull transfer is shown (type: int, standard value: 0)
ConfigID	ID of the test series
<Value>	Column(s) with the values of interest

Project a data base transfer object with the following properties:

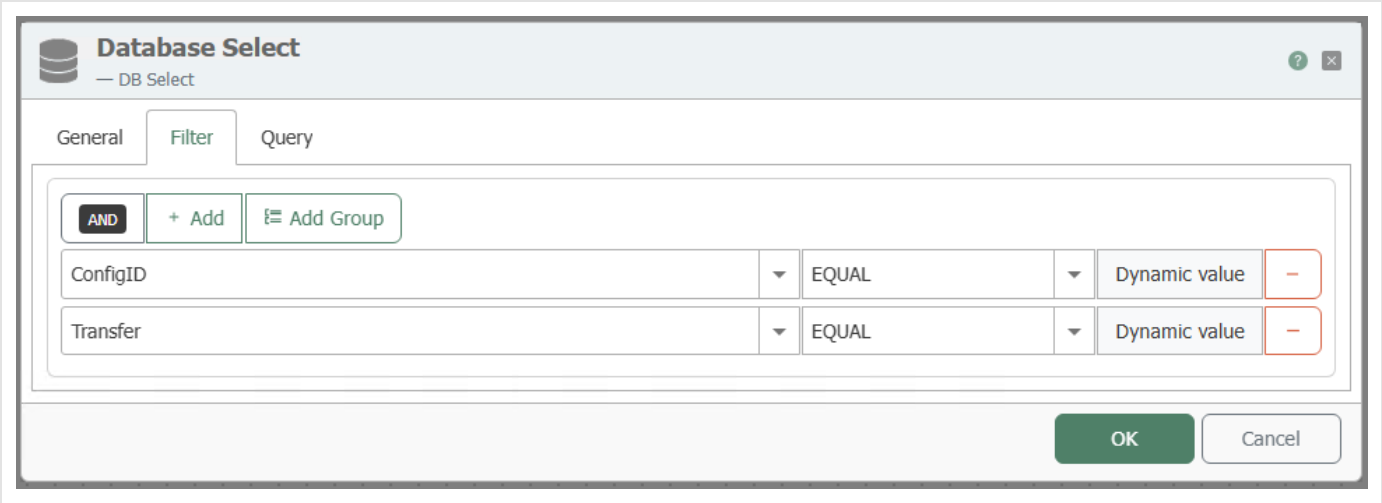
Data Base Access Data	*<Your data base access data from the plug in configuration>*
Type	Select

Tab: General

Table	*<Table, which includes your data>*
Available Columns/ Used Columns	*<All Value Columns>*

Tab: Filter

Filter	“Add Filter” creates a new filter. You need 2 filters. The connections AND is standard.
Edit Filter/ Filter Group	**Column:** ConfigID **Compare Operator:** EQUAL **Column:** Transfer **Compare Operator:** EQUAL
Preview	Here should be given out now: WHERE (ConfigID = ? AND Transfer = ?)



Database Select Transfer Object - Filter

Tab: Query

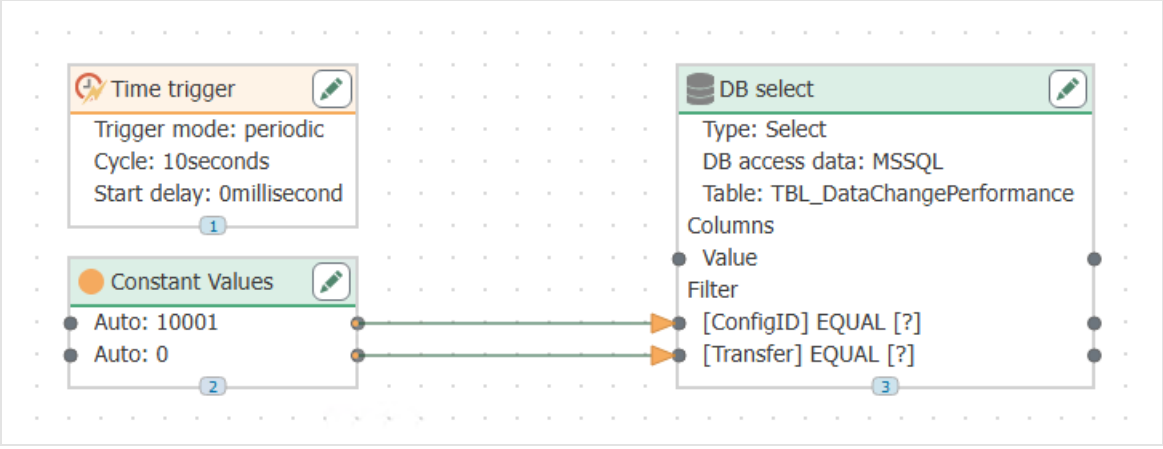
Sorting by Columns	*<Time Stamp Column>*
Sorting Order	Descending (latest value on top)
Reduce number of data sets to	1 (just the first data set = just the latest is to be transfered)
Error With Empty Select	deactivated. If no new, current data set is available, that shouldn´t be counted as an error.
Cancel Transfer with Empty Select	activated. Nothing is transfered, if no new, current data set is available.

This transfer object thus filters the data from the ConfigID to enter and the transfer code. For the transfer code you have to enter “0”. Create a constant object with:

Required constants

<ConfigID>	ID of the test series
0	for not transfered data sets

Connect the ConfigID value to the filter detail ConfigID and the 0 to the filter detail Transfer:



Transfer Objects

Project a time trigger, which polls the data base cyclic.

After transferring the OPC Router gives out “1” the column Transfer, with transfer error “2”. In this projecting only the new, current data set is transfered with Transfer = 0.

Note: If the data comes into the data table faster, than the time trigger polls, it can happen, that the transfered data set is not current any more, because data sets are left out. Scale the needed precision and the needed ressources from the data base server. For data base server using a notification service you can use a script trigger, which starts the transfer as soon as the information comes from the data base server. But there would be another lag because of the cycle time of the OPC Router (max. 50 ms in the standard setting). If you want to have every data set available, you should read out all data sets, which havn´t been transfered.

... Read Out All Data Sets, Which Haven't Been Transferred?

To read out all data sets which havn´t been transfered from a table, you need a column, in which the OPC Router can mark a successfull transfer. You start a request with a filter on this transfer column.

Required fields/columns:

<Transfer>	Column, in which a successfull transfer is shown (Type: int, standard value: 0)
<Values>	Column(s) with values of interest

Project a data base transfer object with the following properties:

Data Base Access Data	*<Your data base access data from the plug in configuration>*
Type	Select

Tab: General

Table	*<Table, which includes your data>*
Available Columns/ Used Columns	*<All value columns>*

Tab: Filter

Filter	“Add Filter” creates a new filter.
Edit Filter/ Filter Group	**Column:** <Transfer> **Compare Operator:** EQUAL
Preview	Here should be stated now: WHERE (Transfer = ?)

Tab: Query

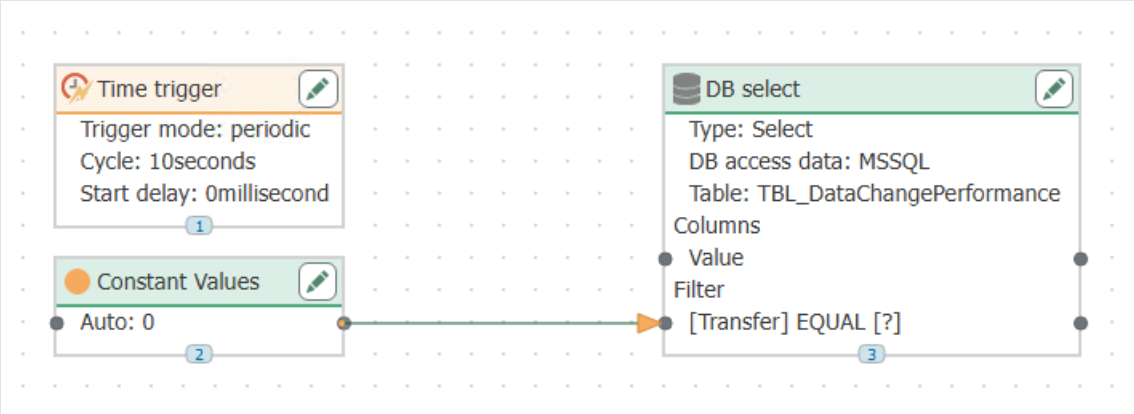
Sorting by Columns	as needed
Order of Sorting	as needed
Reducing the number of data sets to	deactivated. All data sets with <*Transfer*> = 0 shall be transfered.
Error with empty select	deactivated. If no new, current data set is available, it should be no mistake.
Cancel transfer with empty select	activated. Nothing is transfered, if no new data set is available.

This transfer object filters the data by the transfer code. For the transfer code you have to enter “0”. Create a constant object with:

Required constants

0	for data sets which havn´t been transfered

Connect the 0 to the filter element Transfer:



Project a time trigger, which pollst the data base cyclicly.

After the transfer the OPC Router writes “1” the column Transfer, with transfer errors “2”. With this projecting the new data sets with Transfer = 0 are transfered. Of course you can project more filters, as e. g. the ID of the test series from the previous example.

Best Practice

At the most places the projection with OPC Router is intuitive. But there are some things that you should be aware. This will help prevent against data loss and helps to enhance the system's performance

Project development

Do not use more than five transfer steps for a connection

Troubleshooting and further development more easily if the transfers remain clear. In the case of necessary repetitions, you also avoid performance degradation. (e. g. erroneous transfers). With “transfer step”, the number of arrows is not meant: To a single step may well include many individual transfers. “Transfer step” defines the timing of transfers, which you will see by sweeping the arrow with the mouse.

Connection templates avoid labor

If you know by planning a project already that you require many (more than 10) similar connections, which differ e. g. only by an ID and some data point addresses, you should deal with the concept of connection templates. Place the connection template with variables in the snap-in “templates” and you can create similar compounds with button “Create instance”.

OPC transfers

Datachange and threshold trigger for analog values.

Datachange and threshold trigger you have to use if you want to transfer a single analog value. Trigger item and transfer item are identically. The transfer begins immediately after the occurrence of the trigger condition. If the transfer object includes one more to transmitted items, it can happen values will be transfer where the contents do not belong together.

Bit and message trigger for OPC telegrams

Bit and message trigger you have to use for OPC telegrams, if you want to read a number of related items. The OPC Router wait here another read cycle before the items will be read in OPC transfer object to make sure that the PLC can write the values. From PLC side the trigger should be written on the end anyway.

To record values of message trigger

Draw the value of the message trigger on. So you can track, if any records were not transferred or twice. PLC side is to be increased, the data point for the message trigger for new data by the value 1.

Bit trigger always with reply bit

Use the bit trigger always with reply bit. If the data point at the PLC within a read cycle changes twice, the OPC Router does not realize it. The PLC must react to the missing of response from the router. Standard practice: A bit can only be written by one side, that means the PLC sets the trigger data point back after receiving the response. The OPC Router sets the response data point back even after a defined period of time.

Databases

Select always with filter

Use SQL Select only with filters (WHERE clause) to avoid loading the entire table into memory. In conjunction with transfer marks, you filter e.g. after `<Transferflag> <> 1`

Mark transferred records

Mark transferred records. In your tables, keep columns in which the OPC Router can mark transferred data records (“Transfer flag” not transferred, transferred, transfer error). The transfer flag columns should be set to the default value of 0 and should not be allowed to assume NULL, so that a correct assignment to the three statuses is always guaranteed.

Index transfer flag

To improve performance, the transferflag column should be indexed as follows:

- If the OPC Router is to filter for transferflag = 0:

```
CREATE NONCLUSTERED INDEX [IX_TransferTable_Transferflag] ON [dbo].[TransferTable]
(
  [Transferflag] ASC
)
WHERE ([Transferflag]=(0))
GO
```

- If the OPC Router is to filter for transfer flag <> 1:

```
CREATE NONCLUSTERED INDEX [IX_TransferTable_Transferflag] ON [dbo].[TransferTable]
(
  [Transferflag] ASC
)
WHERE ([Transferflag]<>(1))
GO
```

Sort select results

Sort the table in ascending order after the transfer flag. So first the untransferred records are transferred; erroneous individual transfers are only attempted again at the end of the transfer. For performance optimization you can filter for the transfer flag column (transfer flag = 0 or = 2). A correspondingly filtered index can be created in the database for this purpose.

Database design

Database tables should be meaningfully indexed:

- Columns within which you search frequently should be indexed.
- When indexed columns of existing records are updated frequently, the index gradually becomes fragmented and needs to be rebuilt on a regular basis. Otherwise, performance losses would be the result.

Further information, including the use of diagnostic scripts, can be found in the in-commissioning manual, which can be found on the installation source at Additionals/Doku.

Examples and Concepts

Here you will find examples and concepts for various plug-ins :

Database to other systems

Additional settings for logging in via Windows user

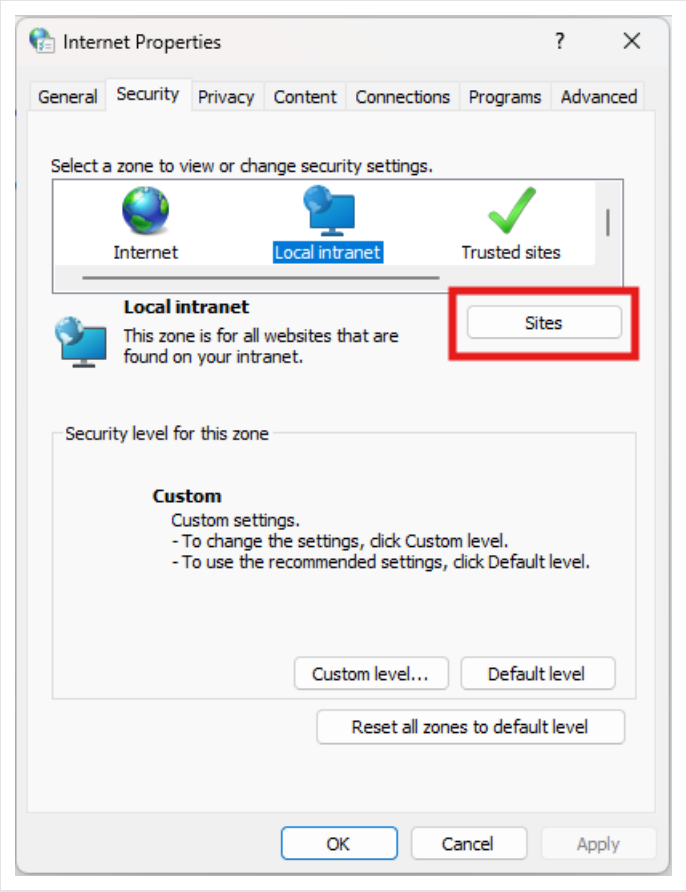
The OPC Router offer you two possibilities of logging, the login by User, created while the installation or system administrators via the OPC Router, and the login by Windows User. For security reasons, in many environments, logging in via Windows user is only possible if HTTPS has been set up.

User HTTPS (Windows)

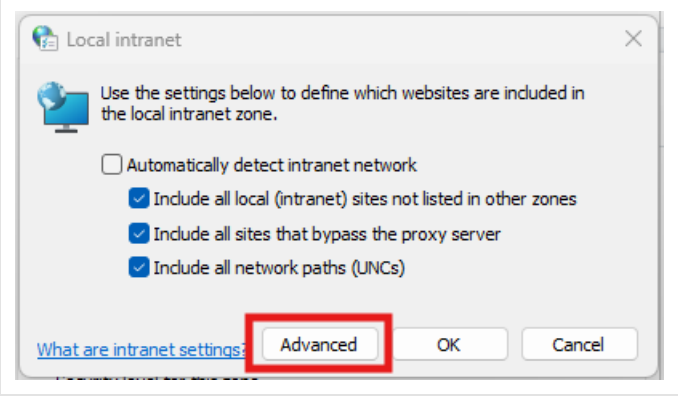
In order, for a login to be possible, the certificate must contain the name used for access via the web browser. For systems that are part of a Windows domain, for example : "computer123.mycompany.local", an alternative name (Subject Alternative Name) like "computer123.mycompany.local" with the full name, including the domain, must be stored in the certificate.\

Certificates created for the HTTPS Installation should be trusted by installing the certificates under Windows.

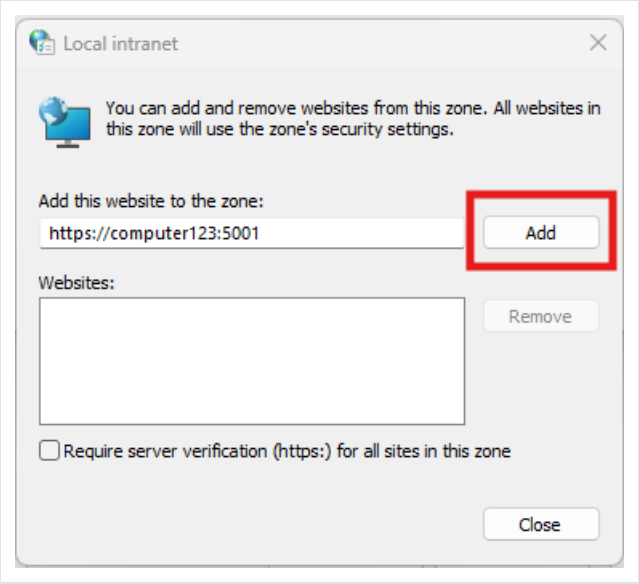
The Windows logging can be limited by security settings in the group policies and via the domain. If the Windows login is only allowed for websites that are located in the local intranet, an exception must be added for all the systems that wish to reach the OPC Router so that the system where the OPC Router was installed is trusted.\



In the Security tab of Internet options, choose Local intranet and edit the sites



Click on the "Advanced" button in the "Site" dialogue



Add the system on which the OPC Router was installed

SSL Certificates

Obtaining an SSL Certificate

An SSL certificate is a digital document that verifies the identity of a website or server and enables an encrypted connection between the browser and server. To obtain an SSL certificate, you must submit a request to a Certificate Authority (CA) that issues and verifies the certificate. The CA checks if you are the legitimate owner of the domain for which you are requesting the certificate and then issues a certificate that you can install on your server. Some CAs offer free SSL certificates, while others charge a fee.

For internal use of OPC Router 5, a certificate from a Certificate Authority is not required; a self-signed certificate is sufficient.

Creating a Self-Signed SSL Certificate

We recommend contacting your network administrator to obtain an SSL certificate with a private key. The certificate generated here allows an encrypted connection but cannot be used to verify the authenticity of the website.

You can easily create a self-signed SSL certificate using the alpine/openssl container.

If you are using a Linux distribution and already have OpenSSL installed on your system, you can use OpenSSL directly and do not need to use the alpine/openssl container with the steps described here.

```
docker run -it --name openssl alpine/openssl
```

This command creates a container named openssl from the alpine/openssl image. The combination of arguments `-it` opens the OpenSSL prompt of the container in the terminal where the command is executed, making the application directly accessible.

In the OpenSSL prompt, you can create an SSL certificate with a private key using the following command:

```
req -newkey rsa:2048 -nodes -keyout https.key -x509 -days 365 -out https.pem
```

When you run the command, you will be prompted for some information to be included in the certificate. After entering all the information, you can exit the prompt using the `exit` command.

The certificate (named `https.pem`) and the private key (named `https.key`) are now at the root level inside the container. You can copy them from the container to your local system (into the folder where you are running your terminal) using these two commands:

```
docker cp openssl:/https.pem ./https.pem
```

```
docker cp openssl:/https.key ./https.key
```

You can now remove the container with the command `docker rm openssl`.

Certificates in OPC Router

On how to import this new certificate in OPC Router read the article below.

Certificate management

Performance Analysis

Disclaimer on performance analysis

The benchmark and test results shown in this documentation were obtained under specific conditions. Actual performance in a production environment can vary significantly depending on numerous factors, including, but not limited to:

- **Server hardware equipment** (processor performance, RAM, memory type and speed, number of cores, cache size, CPU clock speed)
- **network infrastructure** (bandwidth, latency, network topology, firewalls, load balancers)
- **operating system and configuration** (kernel version, process priorities, memory management, background processes)
- **background processes and system-related load**
- **storage systems** (SSD vs. HDD, RAID configuration, read/write access)
- **Number and type of connected third-party systems** (API calls, response times of external systems)
- **and much more.

This performance analysis is therefore only intended as a guide and cannot be interpreted as a general statement about the performance of the OPC Router under any conditions. Customized tests are required to provide a reliable assessment in a specific environment.

OPC data transfer via multi-data change trigger

In our performance benchmark, we were able to simultaneously transfer around **28,500 OPC data points per second** to **MQTT and InfluxDB**. The transfer rate was limited by the processing capacity of the OPC server, which became the limiting factor under higher loads.

Systems

OS: Windows Server 2019 Standard 1809\ RAM: 16GB\ Processor: 64bit 2,6GHz (4 Kerne)\ OPC Router: 5.3.5008.157

OS: Windows Server 2019 Standard 1809\ RAM: 12GB\ Processor: 64bit 2.6GHz (4 Cores)\ KEPServerEx: V6.16.203.0

OS: Windows 10 Pro N 22H2\ RAM: 12GB\ Processor: 64bit 2.6GHz (4 Cores)\ KEPServerEx: V6.16.203.0

Projects

2 KEPServerEx projects

- with one channel each (simulator)
- with one device each (16-bit device)
- with two tag groups each
- with 40 sub-tag groups each
- with 200 tags each (random 1000)

Total 32,000 tags

OPC Router project

- Four OPC UA client connections (two plug-ins each for each KEPServerEx)
- Each plug-in is used in 40 template instances in the multi-data change trigger
- Each multi-data change trigger monitors a tag group (with 200 tags each)
- The values read when the triggers are activated are written to Influx and via MQTT

Summary: Monitoring and reading of 8,000 tags

OPC UA client connection

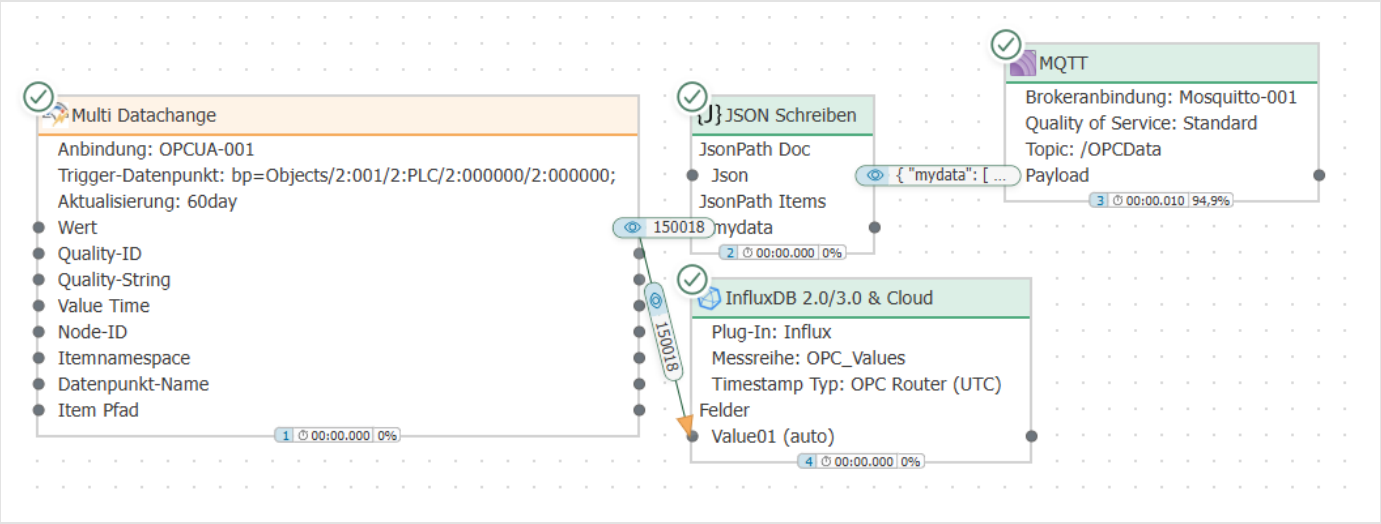
Deviations from the default:

- Subscription – Register OPC tags on start: true
- Advanced – Sample rate (ms): 1000

Multi-Data Change Trigger

Deviations from the default:

- Update items: false



Evaluation

The average time between the executions of the respective multi-data change triggers was considered.

CONNECTIONS	DATA POINTS	PLUG-INS	OPC SERVER	Ø TIME BET. TRIGGERS	PLUG-IN
160	32000	4	2	1100ms	KEPServerEx 1 Connection #1
160	32000	4	2	1027ms	KEPServerEx 1 Connection #2
160	32000	4	2	1204ms	KEPServerEx 2 Connection #1
160	32000	4	2	1101ms	KEPServerEx 2 Connection #2

The OPC server proved to be the limiting factor when the number of data points to be transferred increased. The performance analysis showed that the server's processing capacity was insufficient to handle higher loads efficiently, which limited the overall transfer rate.

Project files

Download Benchmark_KepServer1.opf

Download Benchmark_Kepserver2.opf

Download Benchmark_Multidatachange.rpe

OPC data transfer via BatchRead every 30 seconds

In this data transfer test, we connected **640,000 OPC data points** in an OPC Router instance and transferred them to **MQTT and InfluxDB** every 30 seconds. The average data transfer rate was **21,209 data points per second**.

Systems

OS: Windows Server 2019 Standard 1809\ RAM: 16GB\ Processor: 64bit 2,6GHz (4 Kerne)\ OPC Router: 5.3.5007.156

OS: Windows Server 2019 Standard 1809\ RAM: 12GB\ Processor: 64bit 2.6GHz (4 Cores)\ KEPServerEx: V6.16.203.0

OS: Windows 10 Pro N 22H2\ RAM: 12GB\ Processor: 64bit 2.6GHz (4 Cores)\ KEPServerEx: V6.16.203.0

Projects

Two KEPServerEx projects

- with one channel each (simulator)
- with one device each (16-bit device)
- with 20 tag groups each
- with 100 sub-tag groups each
- with 200 tags each

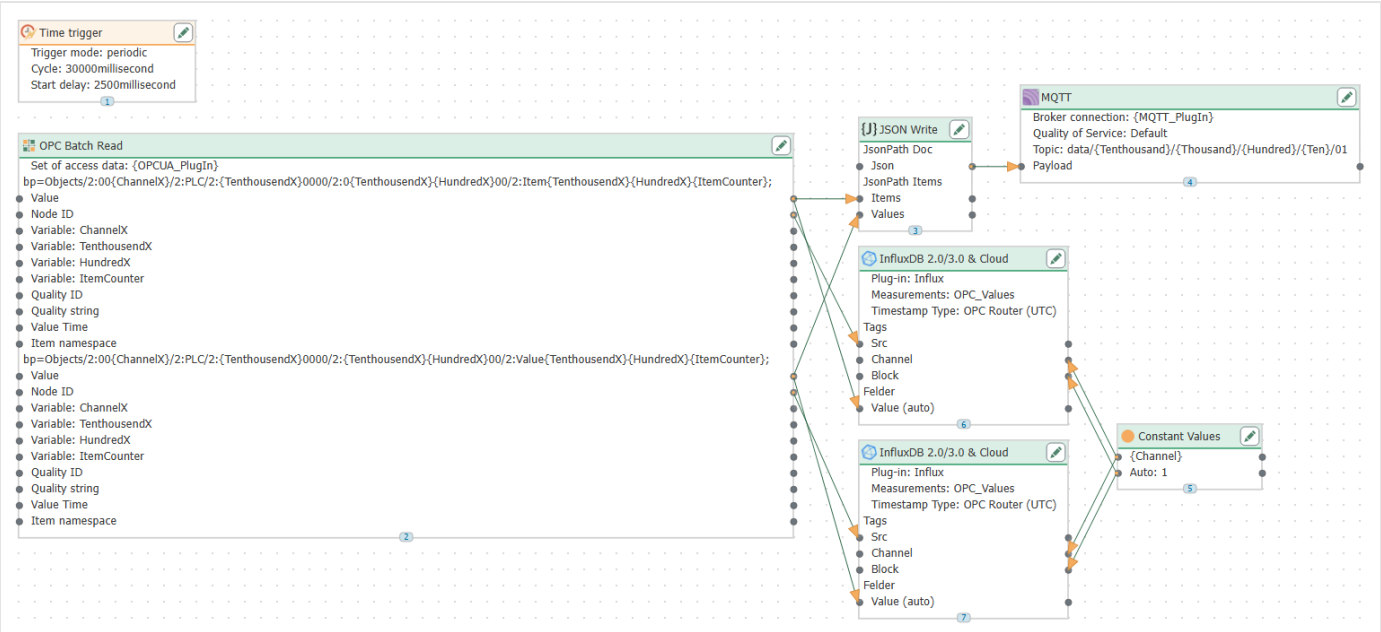
OPC Router projects

For the two test cases, two projects were created with the same structure but different numbers of connections and therefore data points. Description of the simple project:

- Four OPC UA client connections (each with 2 plug-ins per KEPServerEx)
- Each plug-in is used in 400 template instances for each “Batch Read” transfer object
- Each “Batch Read” transfer object polls 200 tags every 30 seconds and writes them to MQTT and Influx-DB

To summarise: In the first test case, 320,000 tags were read every 30 seconds and passed to MQTT and Influx.

Test case 2: Instead of 400 template instances per plugin, there are 800 template instances. This means that 640,000 tags are queried every 30 seconds.



Structure of the template instances (connections)

OPC UA client connection

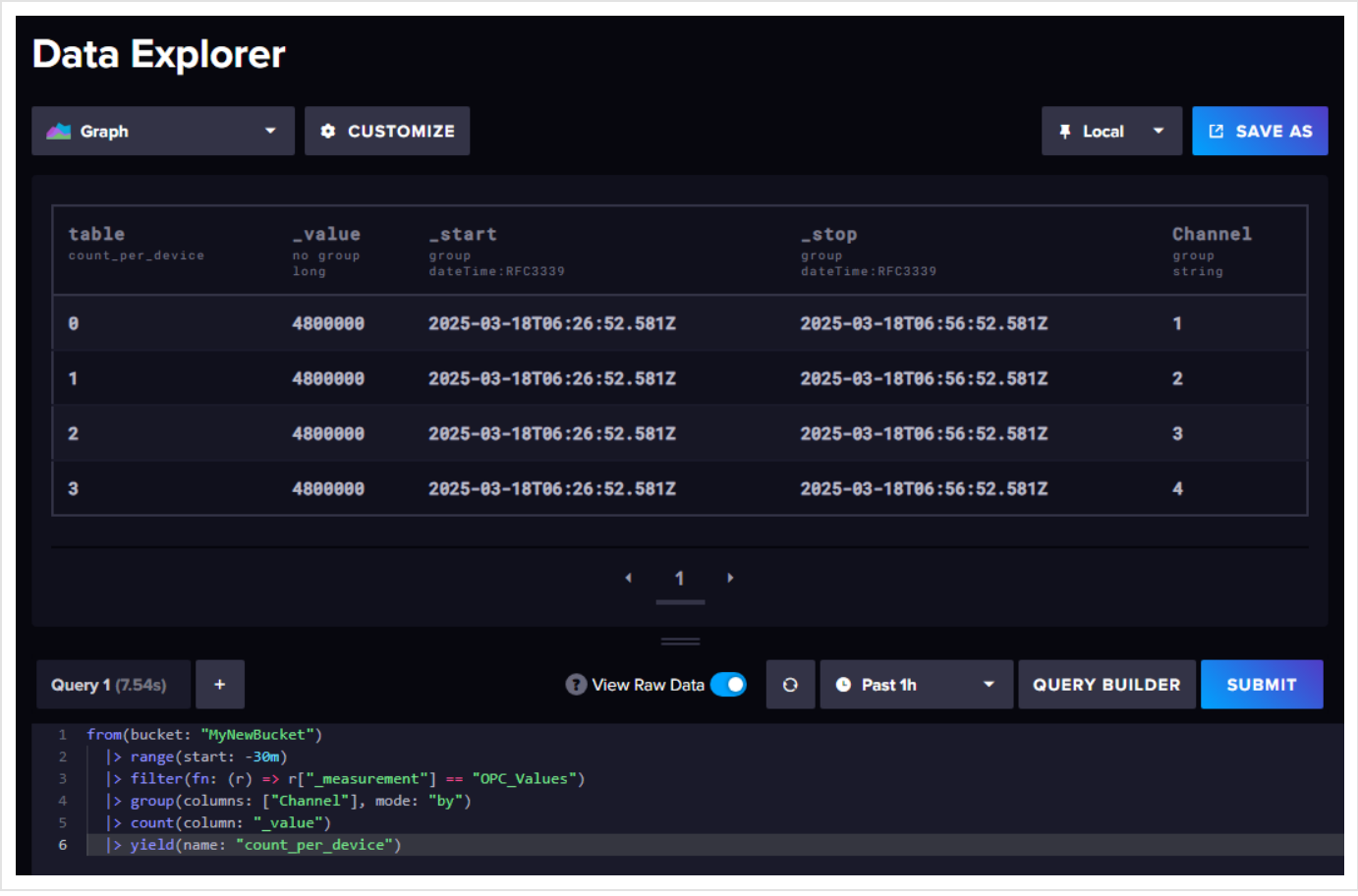
Deviations from the default:

- Quickstart – Timeout (ms): 240000
- Security – Validate server application uri: true
- Subscription – Register OPC tags on startup: true
- Advanced – Timeout (ms): 240000
- Advanced – Enable maximum item count per request: true
- Advanced – Maximum item count per request: 2500
- Advanced – Max Items per Subscription Softlimit: 1500
- Advanced – OPC read mode: Asynchronous

Evaluation

Test case 1

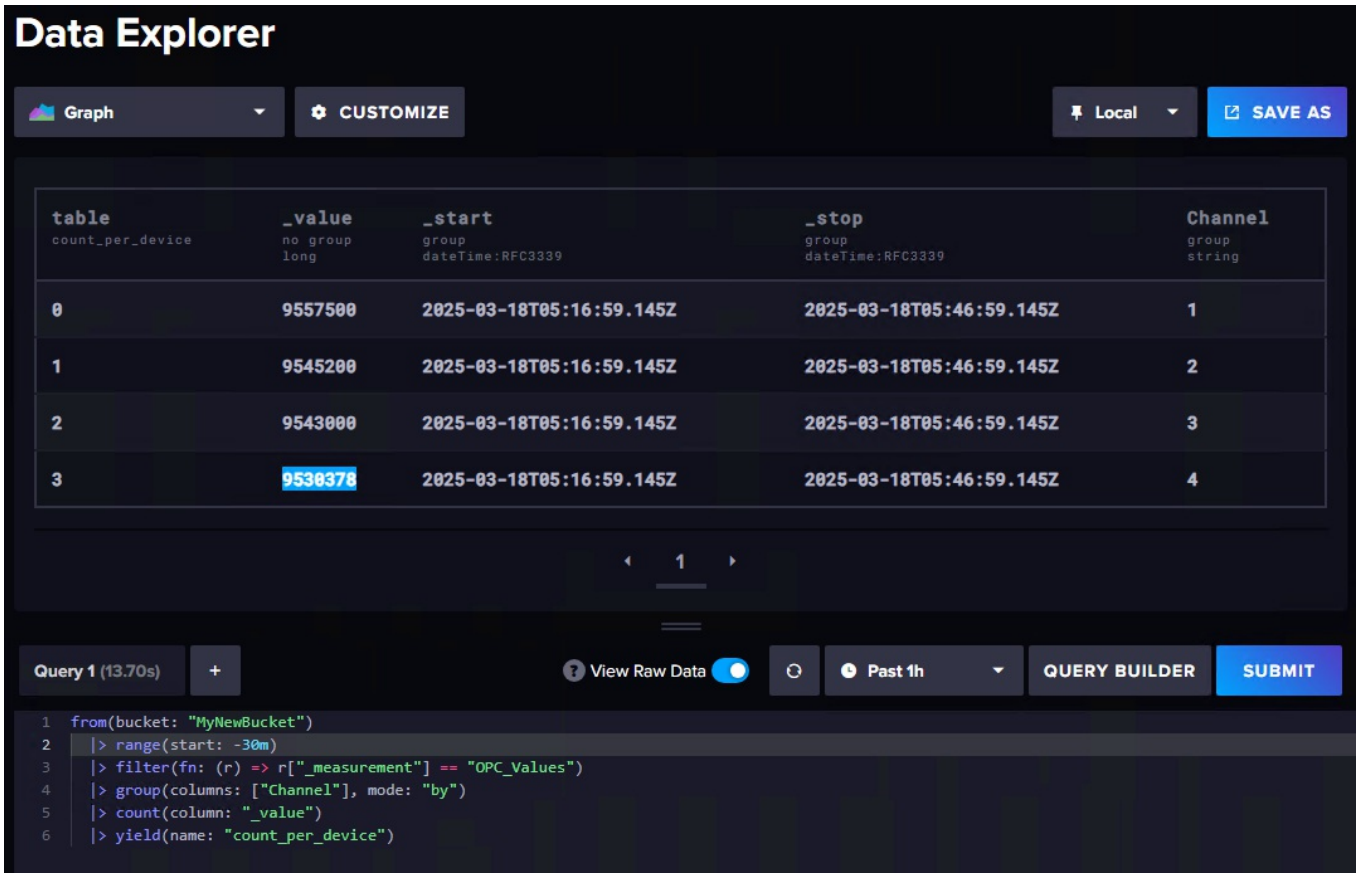
320,000 data points were read and written to MQTT and InfluxDB every 30 seconds across four plug-ins. After running for 30 minutes, a check was made to see how much data had been written to InfluxDB.



At 640,000 data records/minute, 100% of the data was transferred in our test

Test case 2

640,000 data points were read and written to MQTT and InfluxDB every 30 seconds across four plug-ins. After running for 30 minutes, a check was made to see how much data had been written to InfluxDB.



At 1,280,000 data records/minute, more than 99% of the data was transferred in our test

Project files

Download Benchmark_BatchRead_1600.rpe

Download Benchmark_BatchRead_3200.rpe

Download KEPServer Ex – Benchmark Batch Read Server 1.opf

Download KEPServer Ex – Benchmark Batch Read Server 2.opf

OPC data transfer via transfer object in 500ms

Systems

Host

operating system: Windows 11\ RAM: 32 GB\ processor: Intel i7-8700

Virtual machines

OPC Router

operating system: Windows Server 2022\ RAM: dynamic, up to 16GB\ Virtual processors: 8\ OPC Router: 5.3.5008.157 (inraySDK version: 3.32.6002.24)

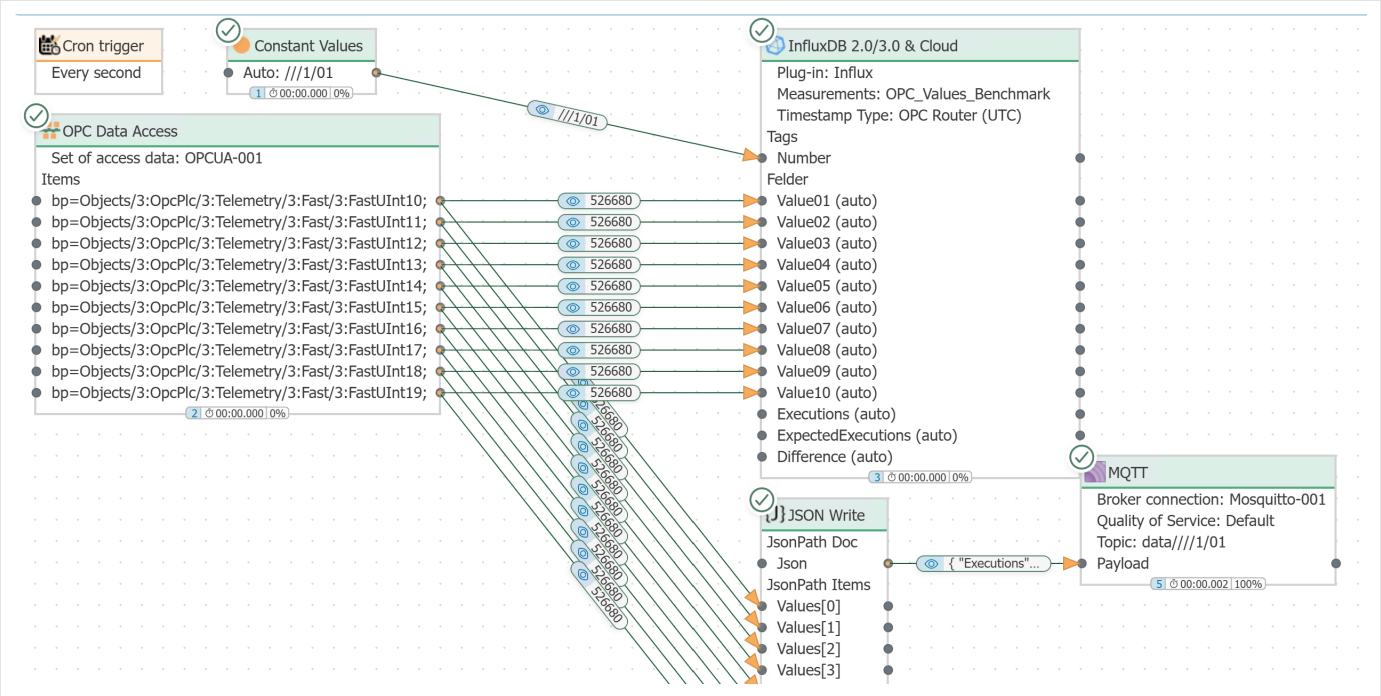
Test server

Operating system: Windows Server 2025\ RAM: 8 GB\ Virtual processors: 6\ OPC server: IoT Edge OPC UA PLC in Docker

Test setup

In the connections, a batch read transfer object or a data access transfer object was used to read 10 data points each. The data is sent to MQTT and also written to an Influx database.

Separate tests were carried out with data change triggers, time triggers and cron triggers. Performance was determined by comparing the expected number of executions with the actual number of executions.



The deviations in the tables were rounded to the nearest five.

In this test, we deliberately pushed the OPC Router and the OPC server to and beyond their performance limits. Data rates of up to 20,000 tags per second were achieved.

Results when using the cron trigger

With the batch read TO

CONNECTIONS	DATA POINTS	TRIGGER INTERVAL	EXPECTED EXECUTIONS	DEVIATION	PLUG-IN COUNT
2000	20000	1 second	500	0 - 10	2
1000	10000	1 second	500	TBD	1

With the OPC Data Access TO

CONNECTIONS	DATA POINTS	TRIGGER INTERVAL	EXPECTED EXECUTIONS	DEVIATION	PLUG-IN COUNT
1000	10000	1 second	500	0 - 5	1

Results when using the data change trigger

With batch read transfer object

At a data change rate of less than 500 ms, the sample rate was adjusted accordingly in the plug-in.

CONNECTIONS	DATA POINTS (TOTAL)	DATA CHANGES	EXPECTED EXECUTIONS	DEVIATION	PLUG-IN NUMBER
100	1000	500 ms	500	0	1
250	2500	500 ms	500	0	1
500	5000	500 ms	500	0 - 20	1
1000	10000	500 ms	500	55 - 80	1
50	500	250 ms	500	TBD	1
100	1000	50 ms	500	TBD	1
250	2500	50 ms	500	TBD	1
500	5000	50 ms	500	TBD	1

With OPC Data Access Transfer Object

CONNECTIONS	DATA POINTS READ	DATA CHANGES	EXPECTED EXECUTIONS	DEVIATION	PLUG-IN NUMBER
100	1000	500 ms	500	0	1
250	2500	500 ms	500	5 - 30	1
500	5000	500 ms	500	25 - 40	1
1000	10000	5 00 ms	500	20 - 60	1
50	500	250 ms	500	TBD	1
100	1000	50 ms	500	TBD	1
250	2500	50 ms	500	TBD	1

CONNECTIONS	DATA POINTS READ	DATA CHANGES	EXPECTED EXECUTIONS	DEVIATION	PLUG-IN NUMBER
500	5000	50 ms	500	TBD	1

Results when using the time trigger

Please note that the time trigger interval is the time that is waited between executions. Since the execution time is not included in the calculation, the time trigger already results in a difference. Therefore, the number of expected executions cannot be achieved exactly here, but only as an approximation.

Using the batch read transfer object

CONNECTIONS	DATA POINTS	TRIGGER INTERVAL	EXPECTED EXECUTIONS	DEVIATION	PLUG-IN NUMBER
100	1000	500 ms	500	TBD	1
250	2500	500 ms	50 0	30 - 35	1
500	5000	500 ms	500	30 - 35	1
1000	10000	500 ms	500	50 - 60	1
50	500	250 ms	500	TBD	1
100	1000	250 ms	null	TBD	1
250	2500	250 ms	500	TBD	1
500	5000	250 ms	500	TBD	1

With the OPC Data Access transfer object

< td>250 ms

CONNECTIONS	DATA POINTS	TRIGGER INTERVAL	EXPECTED EXECUTIONS	DEVIATION	PLUG-IN NUMBER
100	1000	500 ms	500	0	1
250	2500	500 ms	500	30 - 35	1
500	5000	500 ms	500	30	1
1000	10000	500 ms	500	60 - 70	1
50	500	250 ms	500	TBD	1
100	1000	250 ms	500	TBD	1
250	2500	250 ms	500	TBD	1
500	5000	500	TBD	1	

OPC data transfer via transfer object

Systems

Host

Operating system: Windows 11\ RAM: 32 GB\ Processor: Intel i7-8700

virtual machines

OPC Router

Operating system: Windows Server 2022\ RAM: dynamic, up to 16GB\ Virtual processors: 8\ OPC Router: 5.3.5008.157 (inraySDK version: 3.32.6002.24)

Test server

Operating system: Windows Server 2025\ RAM: 8 GB\ Virtual processors: 6\ OPC Server: IoT Edge OPC UA PLC in Docker

Test setup

In the connections, a batch read transfer object or a data access transfer object was used to read 10 data points each. The data is sent to MQTT and also written to an Influx database.

Separate tests were carried out with DataChange triggers, time triggers and cron triggers. Performance was determined by comparing the expected number of executions with the actual number of executions.

OPC UA Client Settings

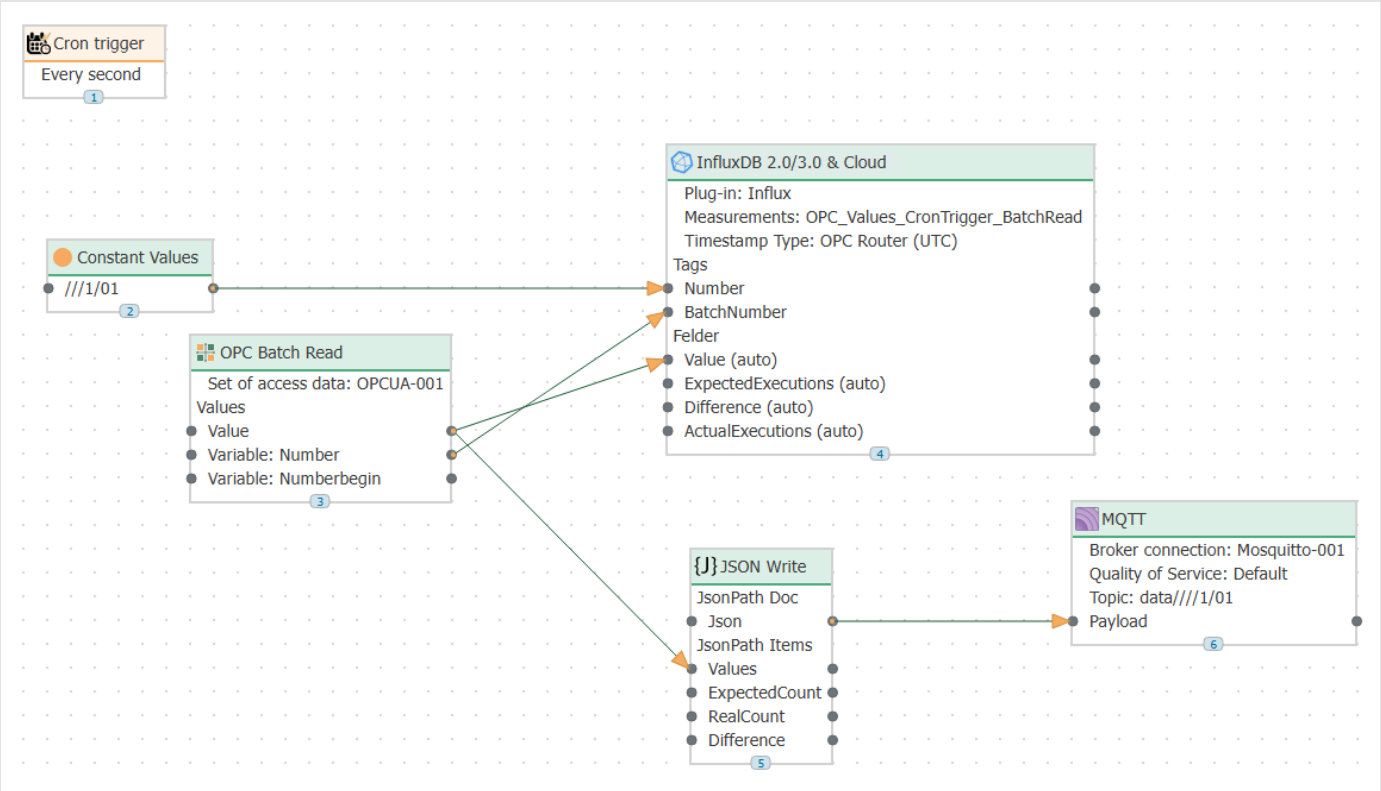
Some settings in the plugin were adjusted for the test.

SETTING	VALUE
Register OPC Tags on Start	Active
OPC Read Mode	Asynchronous

In this test, we deliberately pushed the OPC Router and the OPC server to and beyond their performance limits. Data rates of more than 19,500 tags per second were achieved.

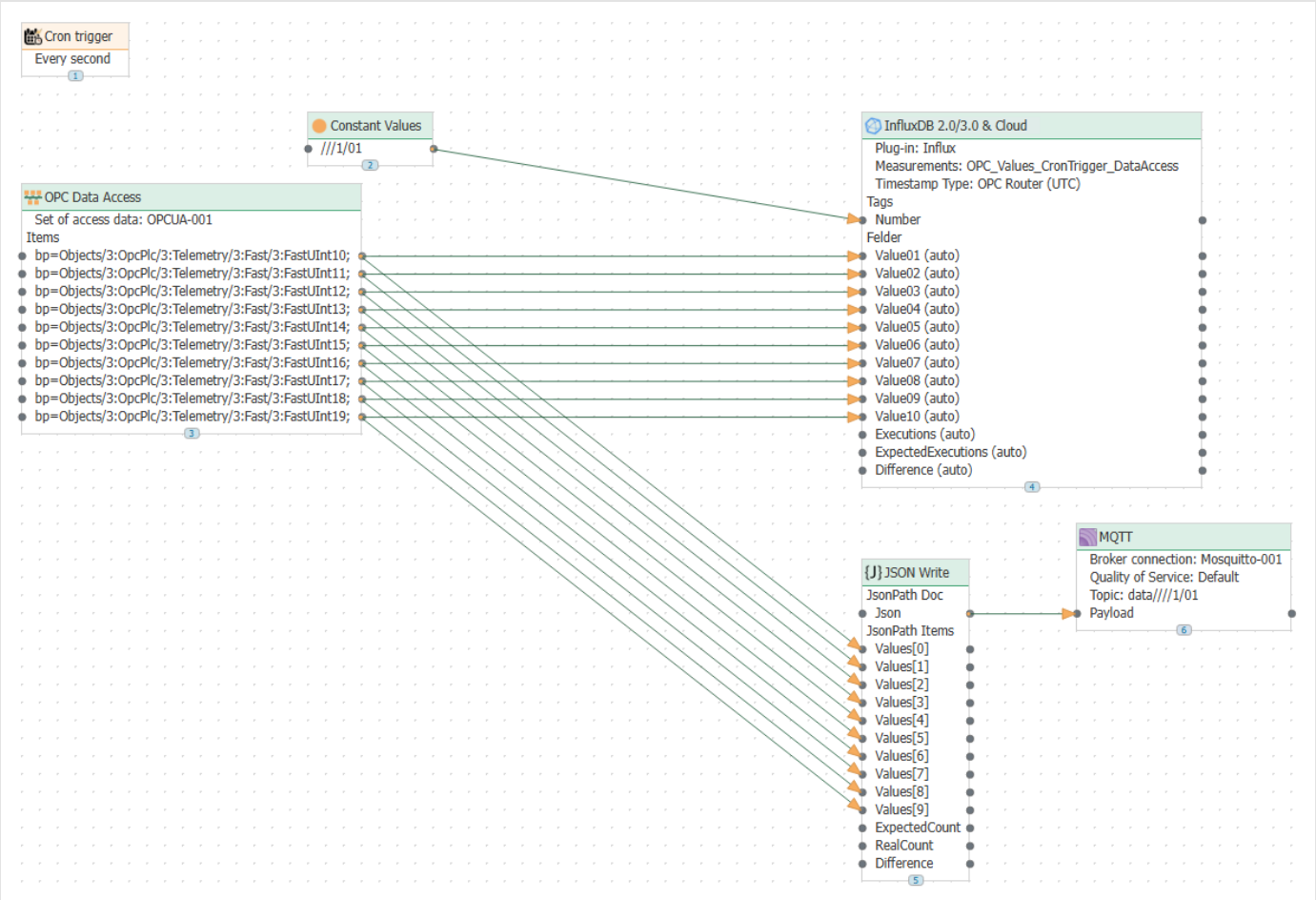
Results when using the Cron Trigger

Using the Batch Read TO



CONNECTIONS	DATA POINTS (TOTAL)	TRIGGER INTERVAL	Ø EXECUTIONS / MIN	EXPECTED
1000	10000	1 second	60	60

With OPC Data Access TO

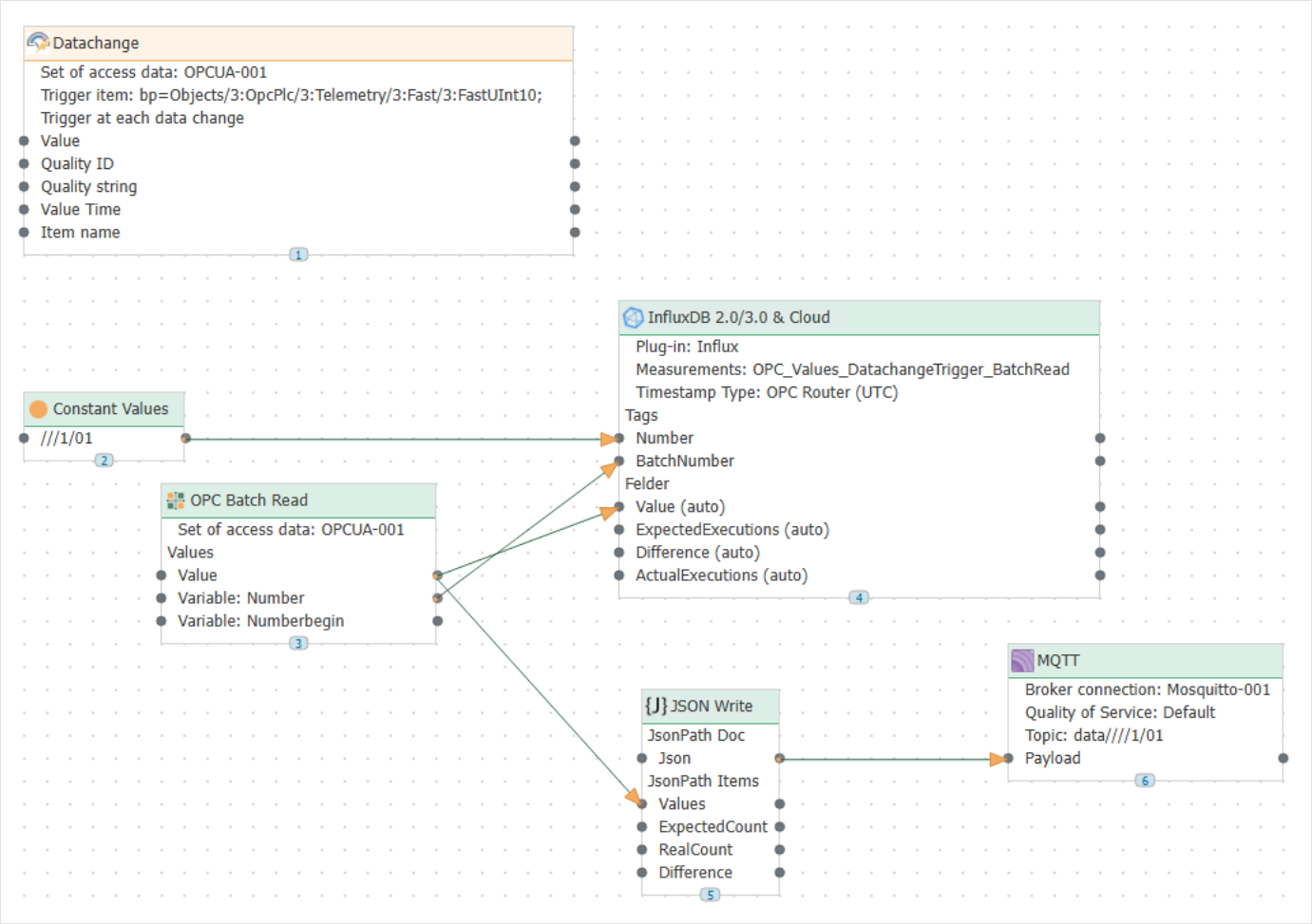


CONNECTIONS	DATA POINTS (TOTAL)	TRIGGER INTERVAL	Ø EXECUTIONS / MIN	EXPECTED
1000	10000	1 second	60	60

Results when using the data change trigger

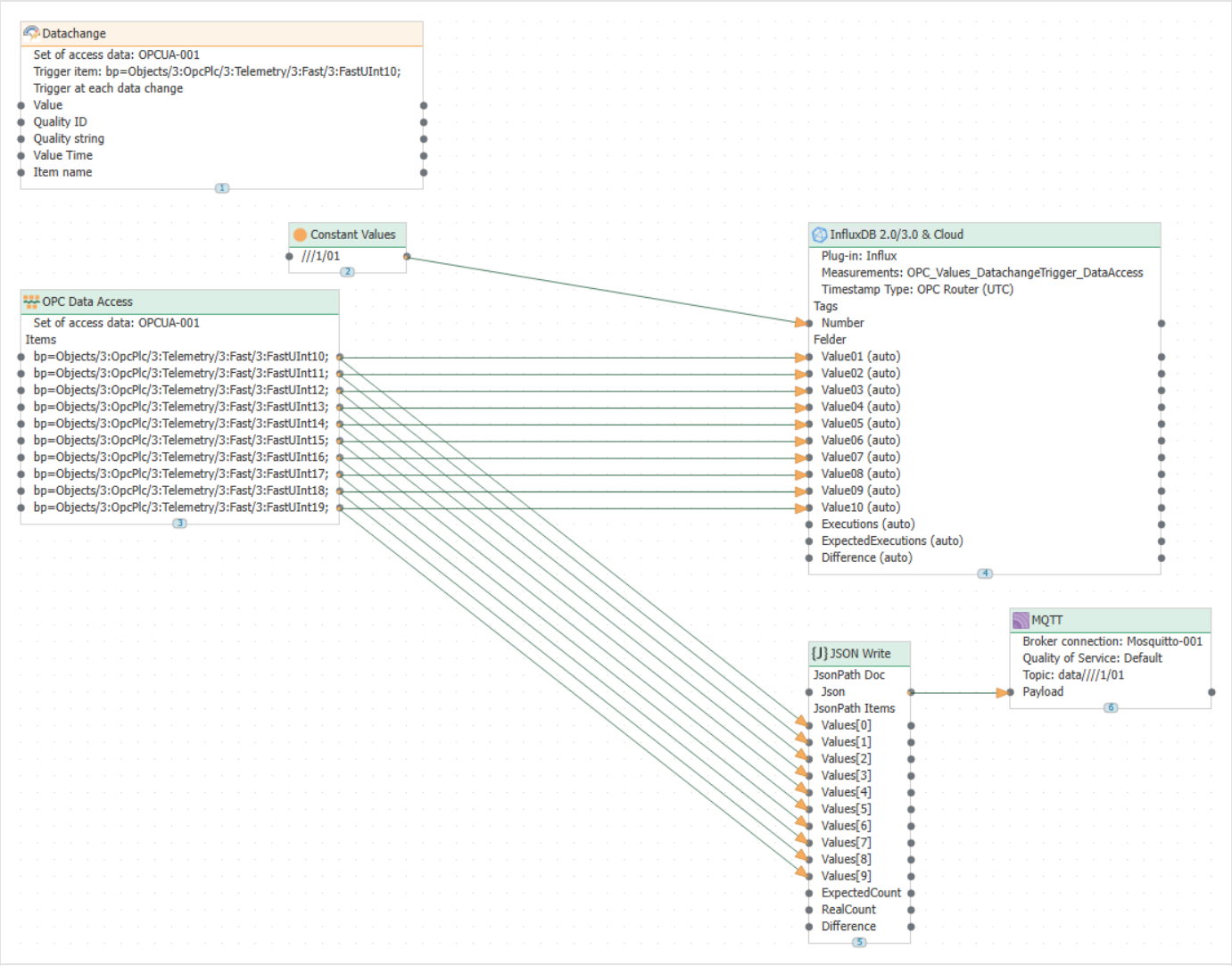
At a data change rate of 250 ms, the sample rate in the plug-in was reduced to 250 ms.

With batch read transfer object



CONNECTIONS	DATA POINTS (TOTAL)	DATA CHANGES	Ø EXECUTIONS / MIN	EXPECTED
1000	10000	500 ms	118.9	120
500	5000	500 ms	120	120
500	5000	250 ms	192.2	240
250	2500	250 ms	198	240
100	1000	250 ms	199.4	240

With OPC Data Access transfer object

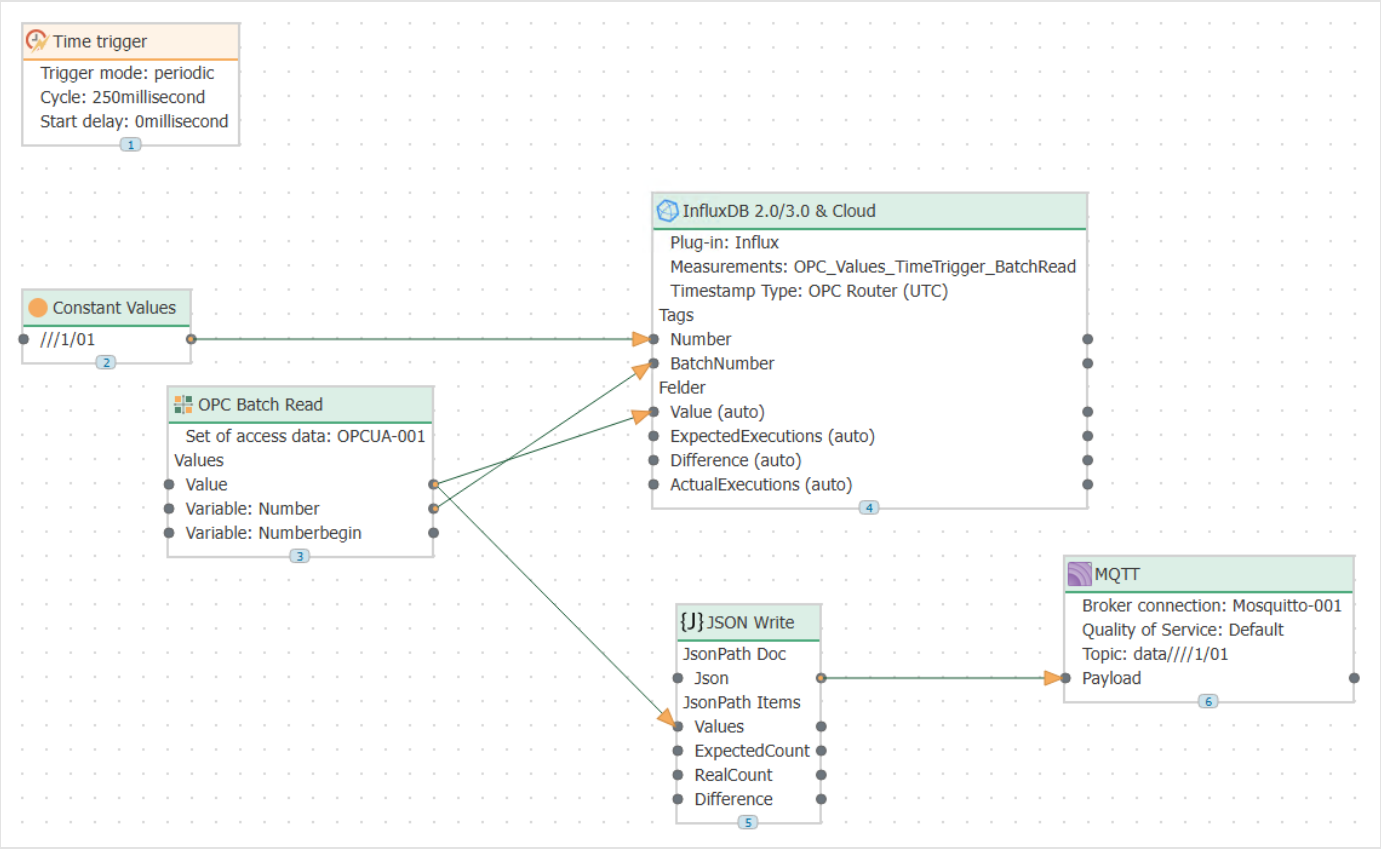


CONNECTIONS	DATA POINTS (TOTAL)	DATA CHANGES	Ø EXECUTIONS / MIN	EXPECTED
1000	10000	500 ms	118,1	120
500	5000	500 ms	120	120

Results when using the time trigger

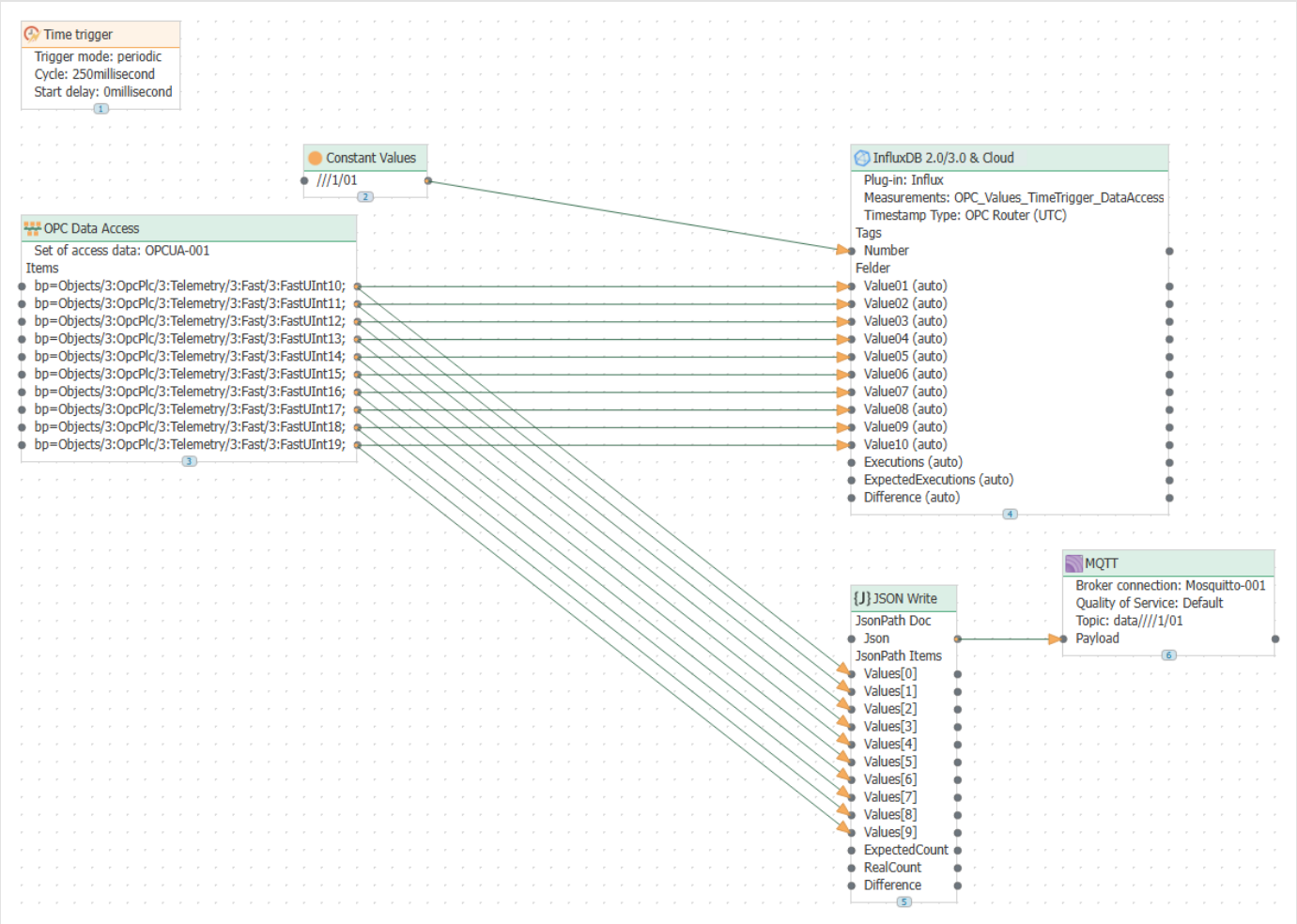
Please note that the time trigger cycle is the time that is waited between executions.

With the batch read transfer object



CONNECTIONS	DATA POINTS (TOTAL)	CYCLE	EXECUTIONS / MIN
1000	10000	500 ms	114
500	5000	500 ms	114
500	5000	250 ms	215

Using the OPC Data Access transfer object



CONNECTIONS	DATA POINTS (TOTAL)	CYCLE	EXECUTIONS / MIN
1000	10000	500 ms	114
500	5000	500 ms	114
500	5000	250 ms	214
250	2500	250 ms	214

Project files

Download Benchmark_CronTrigger_BatchRead_1000.rpe

Download Benchmark_CronTrigger_DataAccess_1000.rpe

Download Benchmark_DatachangeTrigger_BatchRead_1000.rpe

Download Benchmark_DatachangeTrigger_DataAccess_1000.rpe

Download Benchmark_TimeTrigger_BatchRead_1000.rpe

Download Benchmark_TimeTrigger_DataAccess_1000.rpe

Compose file for the OPC Server

MQTT Publish benchmark

Benchmark parameter

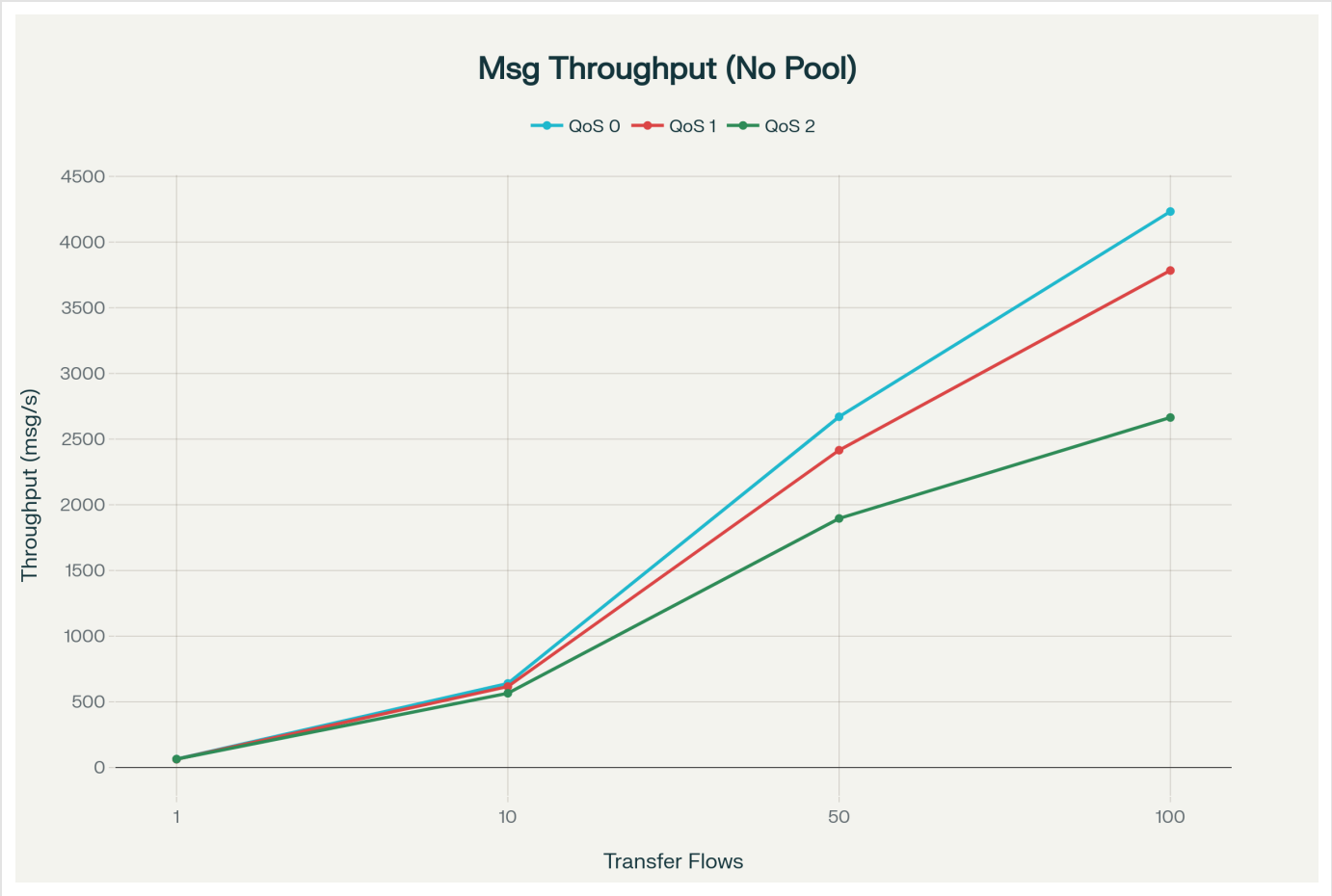
- Broker: mosquitto 2.0.22 (Windows x64)
- OPC Router: 5.5
- Hardware: Windows Sandbox (Intel i7-13700, 4 GB RAM)
- Payload-Größe: 100 Byte
- MQTT Version: 5
- OPC Router Cycletime (expert setting): 5ms

Benchmark-Tool:

https://github.com/dist-sys/mqttloader

- num_subscribers = 1
- num_publishers = 0

Results without Connection Pool



1 OPC Router Connections (transfer flows), Timetrigger Cycle 0ms (fast as possible)

QoS 0

Maximum throughput [msg/s]: 65
Average throughput [msg/s]: 64.283
Number of received messages: 3857
Per second throughput [msg/s]: 64, 64, 65, 64, 64, 64, 65, 64, 64, 63, 64, 65, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 63, 65, 64, 64, 64, 64, 64, 65, 64, 65, 64, 65, 64, 65, 64, 64, 65, 64, 65, 65, 65, 64, 64, 65, 65, 65, 65, 64, 64, 65, 64, 64, 65, 65, 64, 64

QoS 1

Maximum throughput [msg/s]: 65
Average throughput [msg/s]: 63.098
Number of received messages: 3849
Per second throughput [msg/s]: 64, 64, 65, 65, 64, 64, 65, 64, 65, 65, 64, 64, 64, 65, 64, 64, 64, 65, 63, 64, 64, 64, 64, 64, 64, 64, 65, 64, 64, 65, 64, 63, 64, 64, 64, 64, 65, 64, 65, 64, 63, 65, 63, 65, 64, 64, 64, 64, 65, 64, 64, 63, 64, 64, 64, 64, 64, 63, 64, 64, 1

QoS 2

Maximum throughput [msg/s]: 66
Average throughput [msg/s]: 64.133
Number of received messages: 3848
Per second throughput [msg/s]: 65, 65, 64, 64, 64, 64, 64, 64, 65, 64, 64, 64, 66, 63, 64, 64, 64, 64, 63, 64, 65, 64, 64, 64, 65, 64, 64, 64, 65, 64, 64, 64, 66, 64, 64, 63, 64, 64, 65, 65, 64, 64, 64, 64, 64, 64, 64, 64, 63, 64, 64, 64, 65, 64, 64, 64, 63, 64, 64, 64

10 OPC Router Connections (transfer flows), Timetrigger Cycle 0ms (fast as possible)

QoS 0

Maximum throughput [msg/s]: 650
Average throughput [msg/s]: 640.083
Number of received messages: 38405
Per second throughput [msg/s]: 640, 640, 640, 640, 640, 640, 640, 640, 640, 640, 640, 640, 634, 640, 640, 644, 640, 641, 640, 640, 641, 647, 620, 640, 649, 640, 640, 640, 640, 650, 640, 640, 640, 640, 644, 636, 650, 630, 640, 640, 640, 630, 640, 640, 640, 640, 650, 640, 640, 640, 640, 640, 640, 640, 640, 640, 639, 640, 640, 640, 640

QoS 1

Maximum throughput [msg/s]: 648
Average throughput [msg/s]: 616.426
Number of received messages: 37602
Per second throughput [msg/s]: 630, 615, 645, 616, 613, 616, 648, 639, 623, 640, 618, 622, 626, 634, 626, 629, 642, 639, 601, 637, 589, 626, 630, 638, 627, 635, 632, 620, 635, 626, 624, 618, 634, 628, 639, 623, 608, 624, 624, 617, 629, 622, 627, 633, 630, 634, 625, 627, 632, 635, 620, 614, 638, 634, 621, 636, 635, 587, 631, 626, 10

QoS 2

Maximum throughput [msg/s]: 615
Average throughput [msg/s]: 565.233
Number of received messages: 33914
Per second throughput [msg/s]: 575, 599, 559, 572, 514, 551, 559, 562, 589, 562, 565, 567, 550, 532, 568, 589, 613, 574, 564, 559, 553, 574, 547, 567, 582, 577, 546, 577, 590, 546, 539, 583, 577, 548, 523, 589, 597, 543, 615, 540, 593, 520, 564, 534, 558, 578, 590, 598, 556, 594, 603, 544, 529, 589, 552, 560, 560, 530, 560, 566

50 OPC Router Connections (transfer flows), Timetrigger Cycle 0ms (fast as possible)

QoS 0

Maximum throughput [msg/s]: 3117
Average throughput [msg/s]: 2670.117
Number of received messages: 160207
Per second throughput [msg/s]: 2530, 2718, 3051, 2544, 2782, 2552, 2601, 2736, 2728, 2274, 2412, 2397, 3069, 2968, 2752, 2367, 2902, 2679, 2800, 2751, 2411, 2667, 2414, 2940, 2734, 2679, 2873, 2631, 2177, 2642, 2630, 2366, 2711, 2660, 2557, 2559, 2815, 2722, 2415, 2501, 2611, 2978, 2809, 3022, 3117, 3008, 2764, 2579, 2666, 2397, 2624, 2859, 2302, 2870, 2475, 2612, 2693, 2436, 2789, 2879

QoS 1

Maximum throughput [msg/s]: 2659
Average throughput [msg/s]: 2415.600
Number of received messages: 144936
Per second throughput [msg/s]: 2515, 2416, 2441, 2417, 2450, 2231, 2256, 2434, 2440, 2523, 2304, 2453, 2437, 2408, 2305, 2178, 2659, 2555, 2654, 2650, 2432, 2210, 2372, 2414, 2283, 2547, 2499, 2561, 2506, 2494, 2655, 2256, 2309, 2281, 2367, 2633, 2319, 2493, 2467, 2417, 1990, 2232, 2338, 2227, 2202, 2074, 2524, 2540, 2416, 2615, 2399, 2439, 2440, 2415, 2571, 2570, 2517, 2460, 2359, 2367

QoS 2

Maximum throughput [msg/s]: 2164
Average throughput [msg/s]: 1895.950
Number of received messages: 113757
Per second throughput [msg/s]: 1874, 1898, 2164, 1806, 1896, 2030, 1952, 2005, 1875, 1676, 2073, 1831, 1878, 1881, 1962, 1962, 1862, 1869, 1986, 1958, 2037, 1687, 1795, 1955, 1558, 1962, 1890, 1856, 1873, 1873, 1713, 2135, 2051, 1911, 1780, 1969, 1802, 1863, 1975, 1979, 1931, 2087, 2044, 1662, 1668, 1854, 1888, 1884, 1786, 1731, 2048, 1866, 1908, 1886, 1912, 1953, 2006, 1910, 1701, 1930

100 OPC Router Connections (transfer flows), Timetrigger Cycle 0ms (fast as possible)

QoS 0

Maximum throughput [msg/s]: 5286
Average throughput [msg/s]: 4232.721
Number of received messages: 258196
Per second throughput [msg/s]: 4962, 3818, 3976, 3902, 4773, 4216, 4550, 4165, 4658, 4486, 4897, 3660, 5286, 4286, 4257, 4434, 3804, 4358, 4582, 3965, 4051, 4425, 4627, 3948, 3962, 4726, 4880, 4500, 4037, 4627, 3999, 4087, 4331, 5161, 4622, 4045, 3927, 4371, 4300, 4493, 4128, 4292, 3655, 4065, 4442, 4012, 4111, 3646, 4285, 4577, 3966, 4139, 4342, 4086, 4691, 4642, 4129, 4676, 3852, 4240, 66

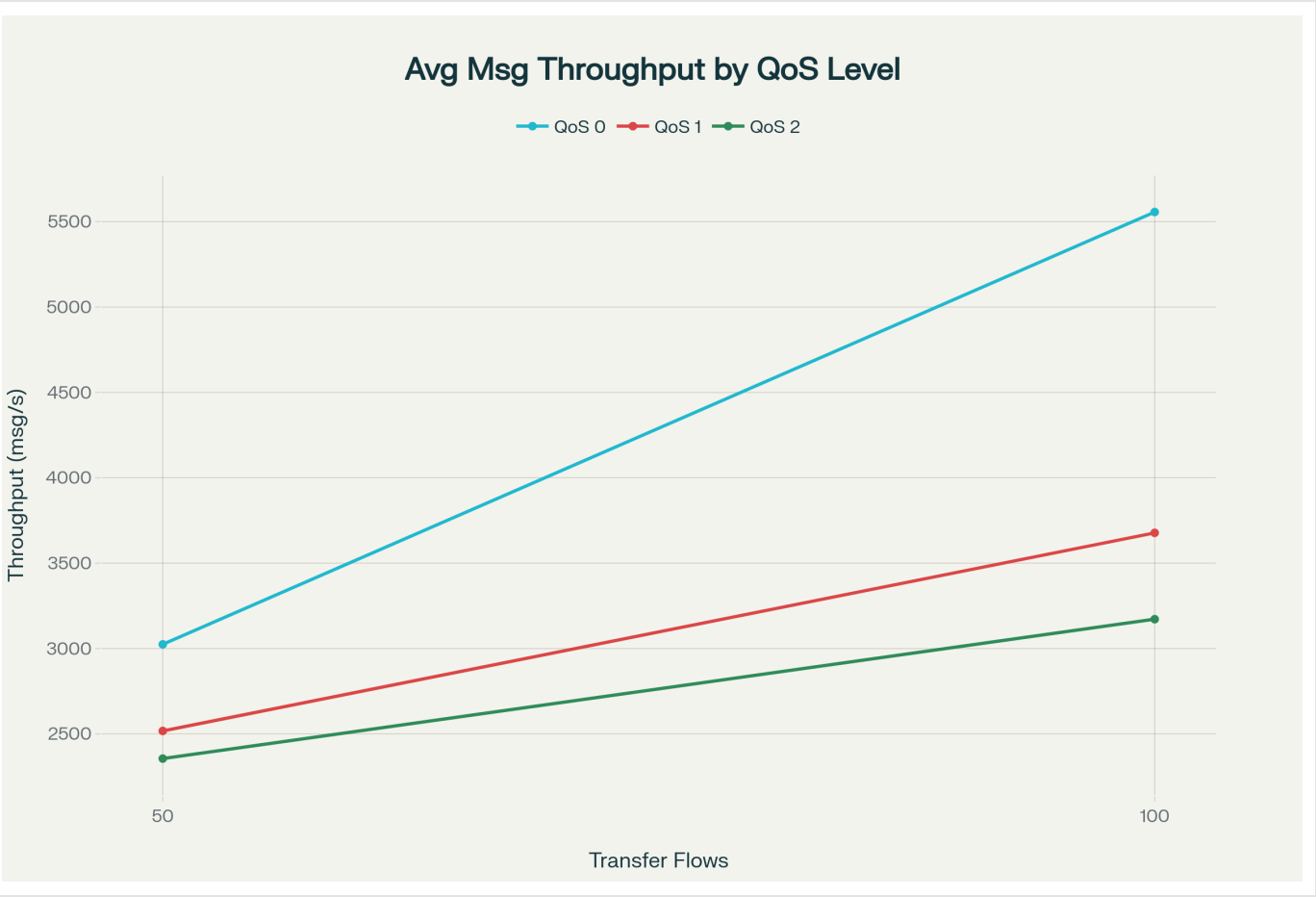
QoS 1

Maximum throughput [msg/s]: 4498
Average throughput [msg/s]: 3783.525
Number of received messages: 230795
Per second throughput [msg/s]: 3909, 3787, 3647, 3879, 4232, 3620, 4025, 4127, 3895, 3814, 4022, 4031, 3607, 3673, 3701, 3965, 4459, 3933, 3422, 4021, 3883, 3802, 3801, 3985, 3322, 4143, 4102, 4040, 4237, 4006, 3836, 3930, 3920, 3669, 3837, 3828, 3863, 3694, 3828, 4498, 3675, 3758, 4175, 4321, 3500, 3680, 2033, 2971, 3649, 3941, 4000, 3864, 3899, 4034, 4009, 3801, 3617, 4319, 3837, 3675, 44

QoS 2

Maximum throughput [msg/s]: 3172
Average throughput [msg/s]: 2664.483
Number of received messages: 159869
Per second throughput [msg/s]: 2482, 2589, 2482, 2309, 2800, 2666, 2762, 2599, 2280, 2836, 2547, 2881, 2786, 2807, 2421, 2845, 2794, 2576, 2922, 2905, 2790, 2531, 2521, 2554, 2757, 2426, 2690, 2745, 2277, 2531, 2474, 2678, 2611, 2410, 2785, 2743, 2364, 2481, 3162, 2812, 2845, 3098, 2526, 2560, 2709, 3172, 2747, 2670, 2677, 2608, 2325, 2616, 2672, 2758, 2704, 2667, 2875, 2560, 2613, 2836

Results with Connection Pool



Connection pool was set to Min: 10 and Max: 100

50 OPC Router Connections (transfer flows), Timetrigger Cycle 0ms (fast as possible)

QoS 0

Maximum throughput [msg/s]: 3230
Average throughput [msg/s]: 3024.600
Number of received messages: 181476
Per second throughput [msg/s]: 2933, 3140, 3057, 2997, 3147, 3104, 3047, 3230, 2845, 2997, 3033, 3083, 3000, 3137, 3042, 2877, 2992, 2916, 3092, 3050, 3026, 2901, 3087, 3079, 3023, 3025, 3160, 2918, 3100, 2957, 2971, 3009, 3049, 2898, 3003, 2980, 3156, 3013, 3000, 2968, 3031, 3027, 3023, 2904, 3102, 2991, 2963, 2886, 3019, 3013, 3097, 3091, 2959, 2998, 3175, 3004, 3074, 2977, 3011, 3089

QoS 1

Maximum throughput [msg/s]: 2890
Average throughput [msg/s]: 2517.607
Number of received messages: 153574
Per second throughput [msg/s]: 2632, 2519, 2852, 2284, 2619, 2526, 2324, 2750, 2440, 2273, 2499, 2595, 2851, 2440, 2537, 2502, 2646, 2611, 2336, 2745, 2890, 2348, 2640, 2292, 2416, 2756, 2363, 2465, 2659, 2366, 2514, 2554, 2483, 2824, 2861, 2467, 2501, 2857, 2518, 2453, 2615, 2453, 2428, 2773, 2420, 2673, 2274, 2845, 2638, 2475, 2510, 2662, 2613, 2630, 2737, 2590, 2467, 2333, 2721, 2455, 54

QoS 2

Maximum throughput [msg/s]: 2774
Average throughput [msg/s]: 2355.721
Number of received messages: 143699
Per second throughput [msg/s]: 2406, 2272, 2376, 2673, 2289, 2190, 2343, 2607, 2411, 2437, 2157, 2414, 2444, 2693, 2193, 2094, 2551, 2341, 2139, 2105, 2244, 2471, 2268, 2162, 2495, 2620, 2425, 2284, 2442, 2606, 2623, 2362, 2323, 2498, 2629, 2480, 2343, 2311, 2638, 2386, 2360, 2303, 2374, 2734, 2386, 2176, 2426, 2361, 2607, 2446, 2245, 2352, 2341, 2774, 2495, 2183, 2293, 2523, 2262, 2263, 50

100 OPC Router Connections (transfer flows), Timetrigger Cycle 0ms (fast as possible)

QoS 0

Maximum throughput [msg/s]: 6118
Average throughput [msg/s]: 5556.230
Number of received messages: 338930
Per second throughput [msg/s]: 5746, 5714, 5629, 5558, 5955, 5545, 5788, 5294, 4577, 5331, 5419, 5866, 5623, 5699, 5864, 5847, 5374, 5583, 5955, 5506, 5211, 5774, 5639, 6001, 5680, 5681, 5249, 5567, 5564, 5918, 5306, 5853, 5517, 5505, 5717, 5925, 5921, 5666, 5388, 5709, 5496, 5403, 5997, 5848, 6118, 5294, 6007, 5574, 5985, 5991, 5464, 6012, 5711, 5737, 5516, 5690, 5640, 5402, 5740, 5635, 6

QoS 1

Maximum throughput [msg/s]: 4741
Average throughput [msg/s]: 4231.533
Number of received messages: 253892
Per second throughput [msg/s]: 4735, 3911, 3956, 4481, 4631, 4540, 4548, 3769, 3903, 3829, 4349, 4133, 3973, 4535, 4302, 3905, 4043, 4015, 4353, 4211, 4391, 3909, 4379, 4616, 3761, 4537, 4242, 3902, 4018, 4405, 3939, 4400, 3952, 4223, 3664, 4072, 4103, 4582, 4088, 4569, 4270, 4595, 4052, 4207, 4199, 3940, 3946, 4741, 4116, 4000, 4416, 4517, 4094, 4341, 4687, 4577, 4055, 4422, 4376, 4467

QoS 2

Maximum throughput [msg/s]: 5090
Average throughput [msg/s]: 3677.557
Number of received messages: 224331
Per second throughput [msg/s]: 3863, 3489, 4634, 2873, 4884, 3806, 3844, 3513, 3068, 3456, 4227, 3471, 3013, 4340, 3546, 3734, 3845, 3800, 3067, 3080, 4453, 3952, 3606, 3546, 3697, 2809, 5000, 3543, 3831, 3766, 3444, 4796, 3680, 3189, 3921, 4339, 4280, 2840, 3457, 3466, 4582, 2583, 4902, 3256, 2953, 4613, 3884, 3592, 2858, 3000, 5090, 4149, 3870, 4191, 3710, 3773, 3478, 4477, 3320, 2879, 3

Status Indication

The button for the status area is located in the snap-in bar.

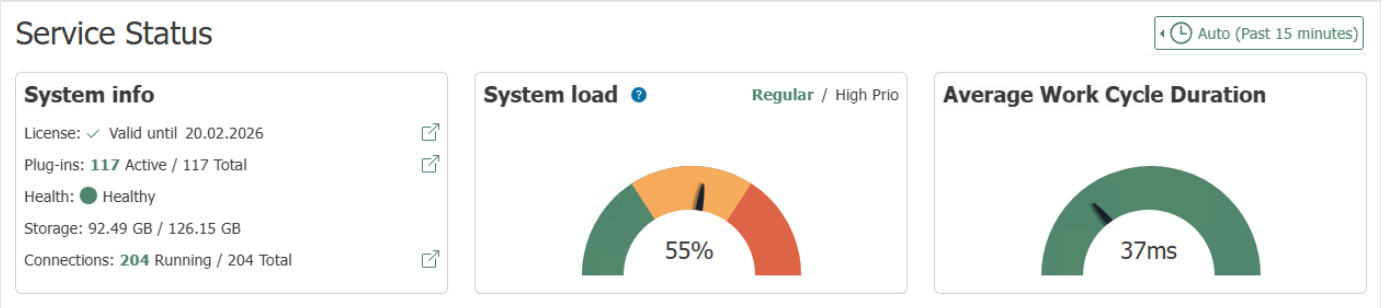
After clicking on the button and establishing a connection, the status window opens:

\

Service status

If you click on ![](</assets/Service status button.png>) in the status view, a dashboard with values for the service status opens.

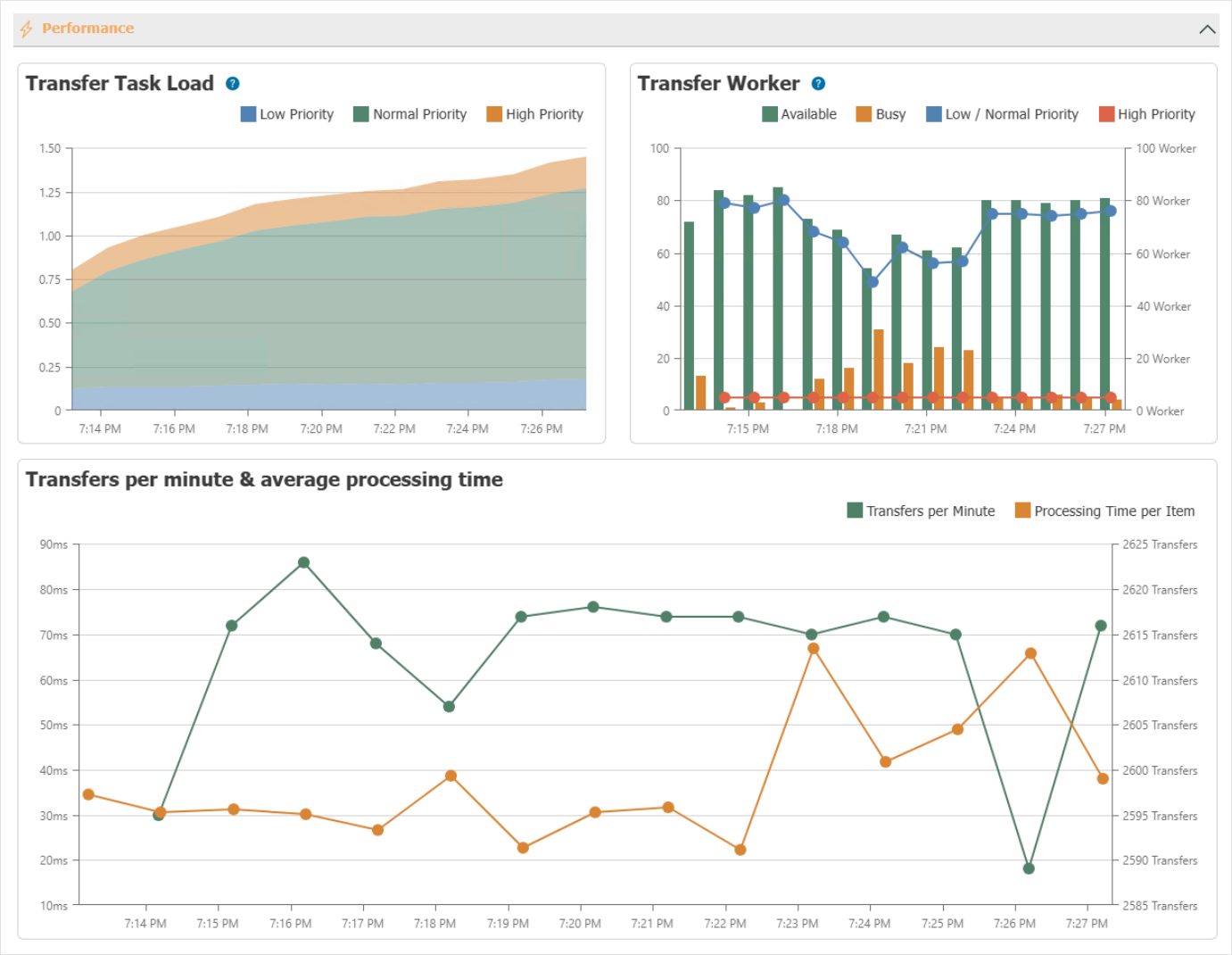
At the top right, you can set the time period for which data should be displayed.



Performance

This section displays:

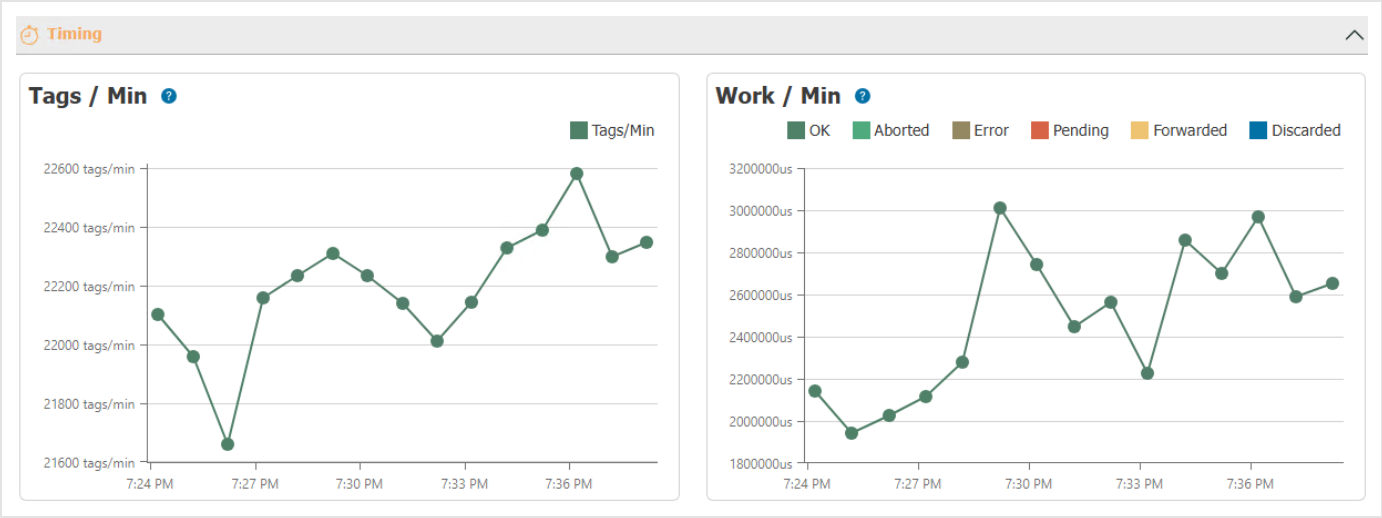
- How much of the OPC Router's capacity is being used, grouped by connection priorities.
- How many transfer workers of each priority are active.
- How many transfer workers are available or busy.
- How many transfers are performed per minute and how long the processing time is for individual items.



Timing

This section displays:

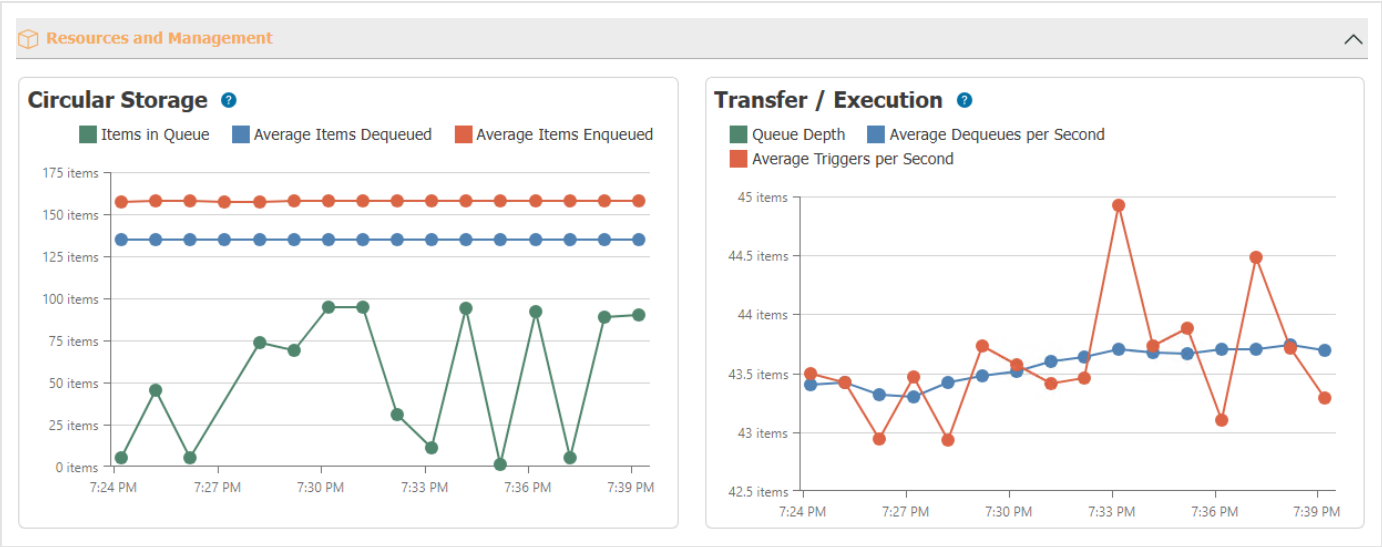
- How many operations the transfer manager performs per minute.
- How long the transfer manager took for the operations, grouped by the final result.



Resources and Management

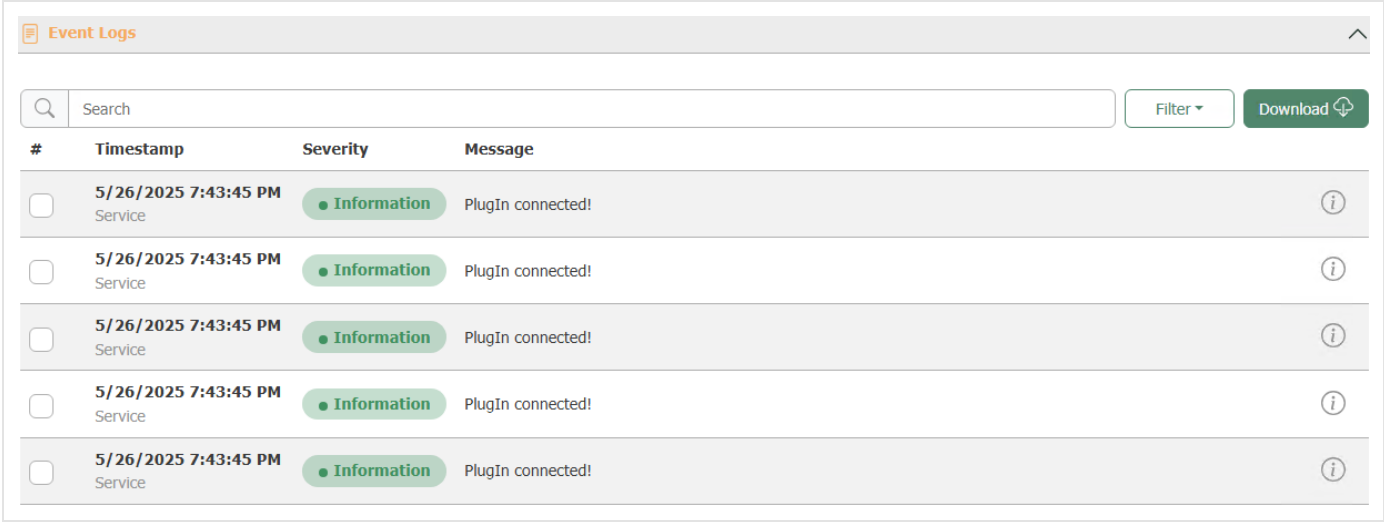
This section shows:

- How many items are in the ring buffer.
- How many elements are added to the ring buffer on average.
- How many elements are removed from the ring buffer on average.



Event Logs

This section lists the log messages of the OPC router.



The messages can be filtered by message type. It is also possible to download the logs.

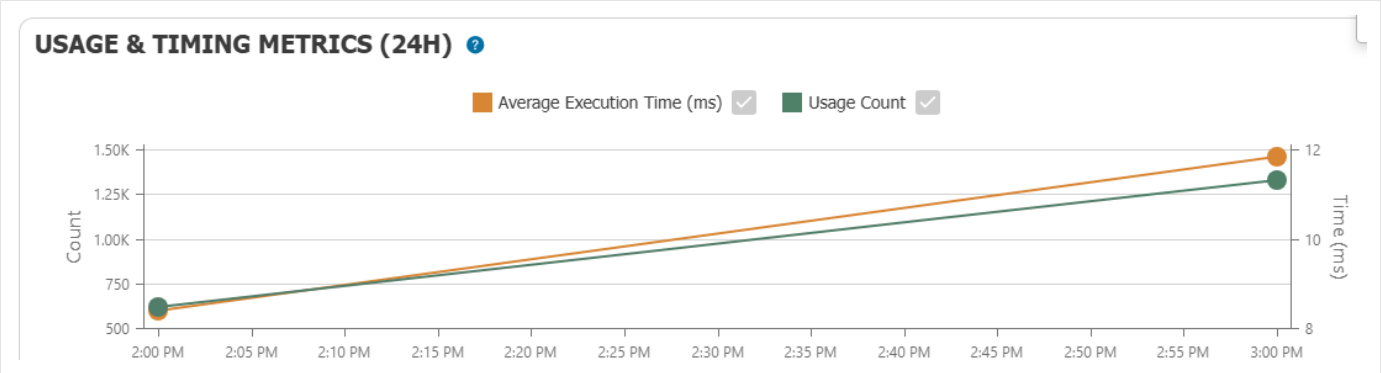
Plug-in status

If you click on a plug-in instance in the status view, a dashboard opens with data on the selected plug-in instance.

Usage & Timing Metrics

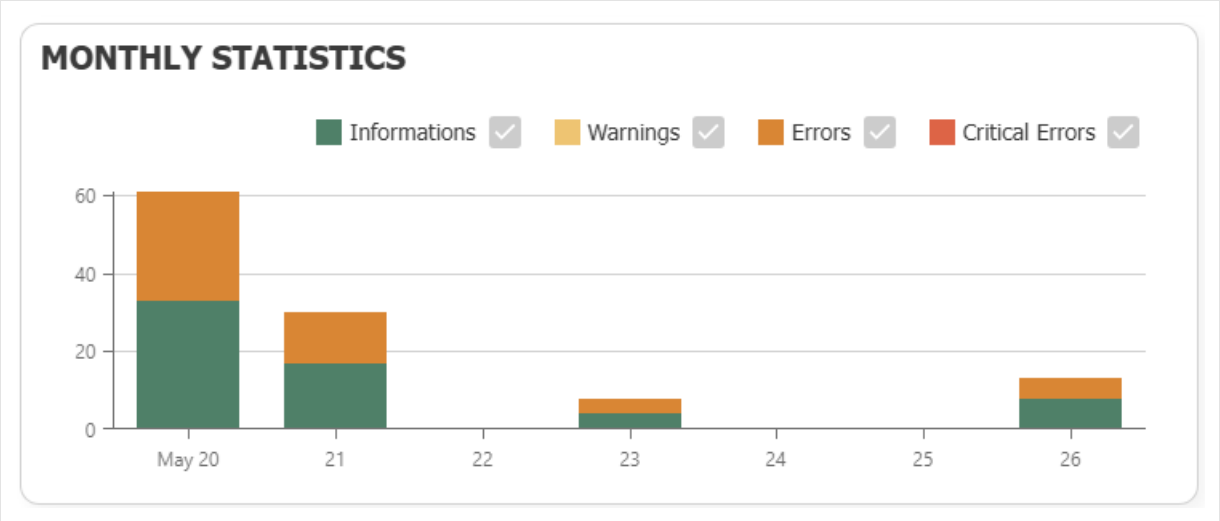
This section displays the following data grouped in full hours:

- Average execution time
- Number of uses



Monthly statistics

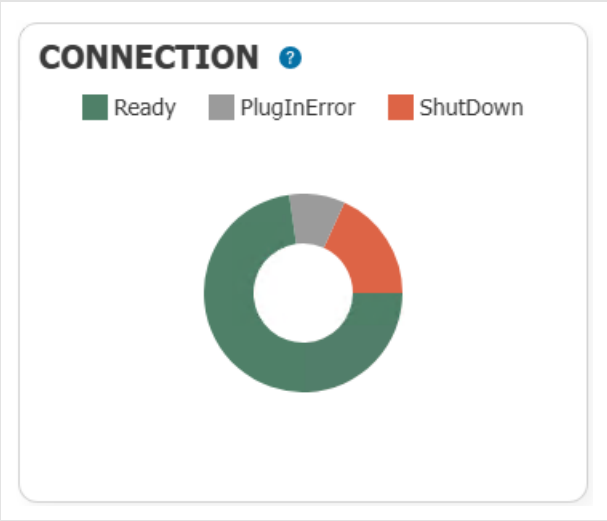
The number and type of log messages is displayed here graphically.



Connections

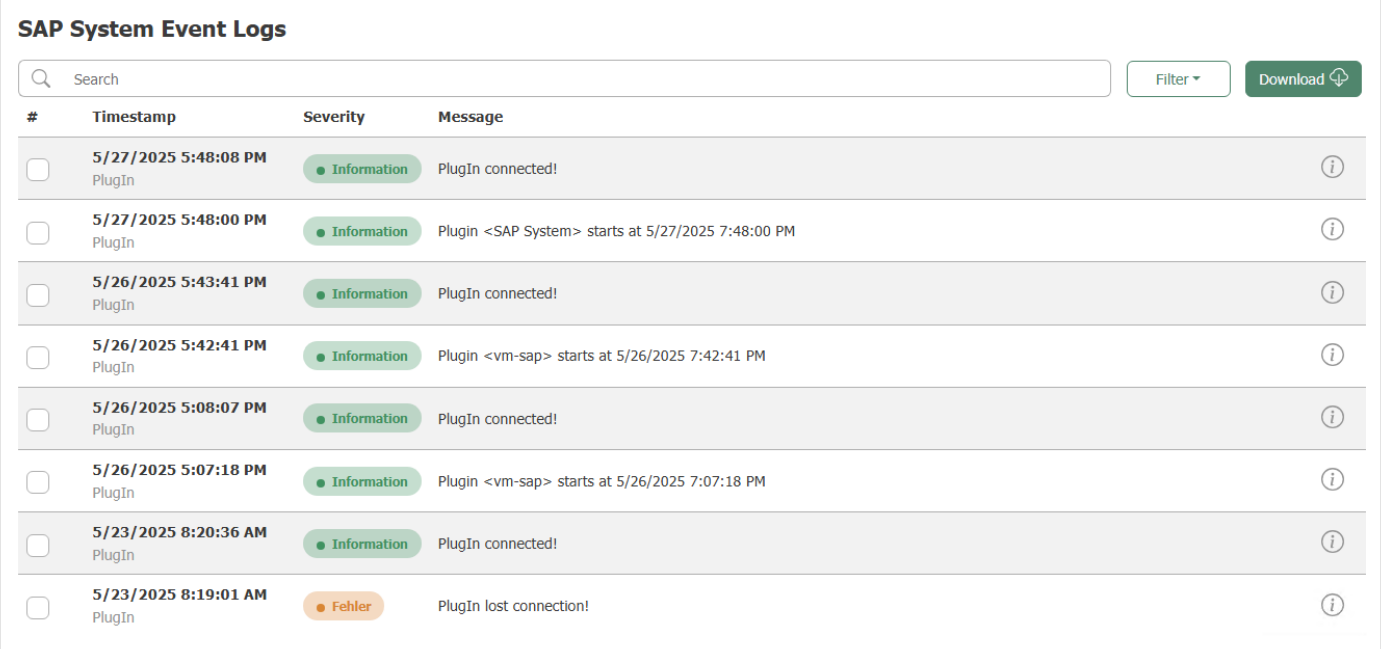
This shows what percentage of connections have which status.

Please note that this is the status of all connections.



Event log

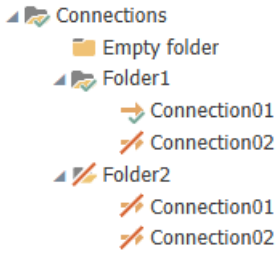
This section lists the log messages of the plug-in.



The messages can be filtered according to the type of message. It is also possible to download the logs.

Connection status

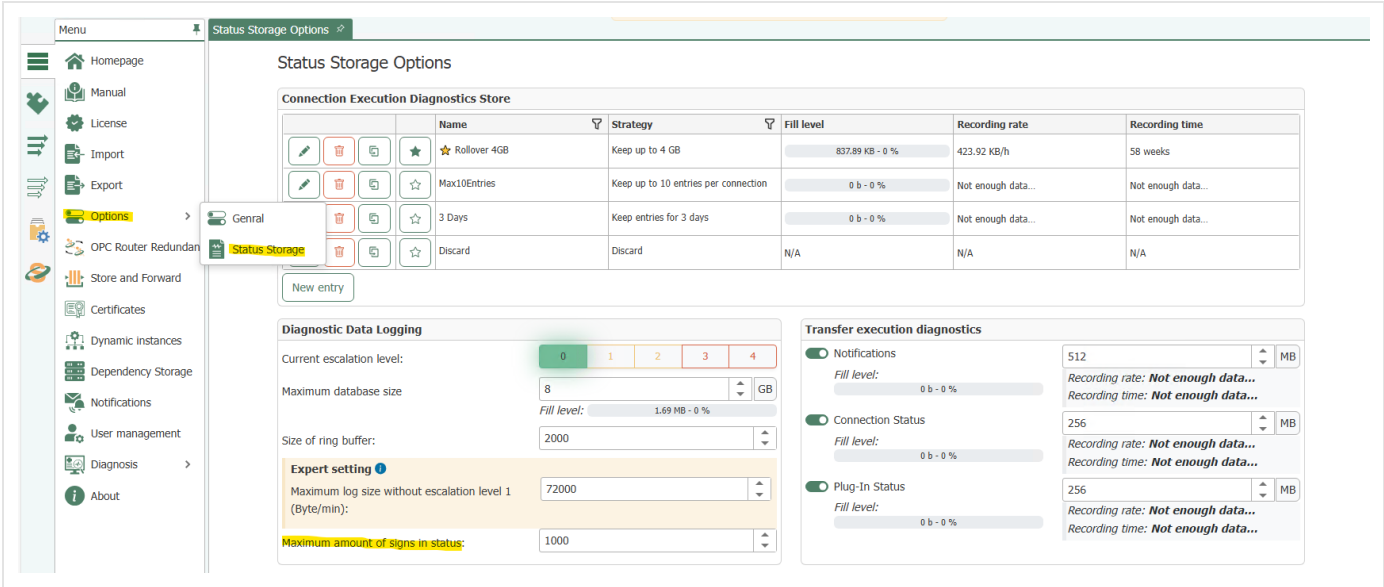
Note: Folders that are contain only deactivated connections are shown crossed out.



The Status chart

The status chart provides detailed information about the individual data transfers. The status chart can be started via the corresponding snap-in in the left margin.

- Green transfer points represent the individual transfers that were in order and the transfer values are stored in the short storage. The lead time is defined in the settings. If you click on a transfer point where the lead time has expired, the message “Transfer values can not be displayed because the lead time of the values has expired”. The red transfer points, however, indicate erroneous transfers and are kept permanently in the Long Storage.
- If you click on a transfer point, it will be shown in light blue and the data will be displayed under the status chart.
- If you move the mouse to a transfer point, the execution time is displayed.
- Above the status chart, you will find the buttons “Previous Transfer” and “Next Transfer” to display the transfer next to the previously selected transfer. With the button “Switch to live view”, you can observe the progress of the individual transfers.
- The zoom can be changed with the mouse wheel.
- If strings or arrays are too long, the preview of the transferred value is shortened at the end – the value is nevertheless transferred completely. The length can be adjusted in the settings under Options/ Status Storage /Maximum amount of signs in status. The default value is 1000 characters.“



- Errors are indicated by the red marking in the status chart and if you move the mouse over the transfer object field under the status chart, you will get detailed information about the error:

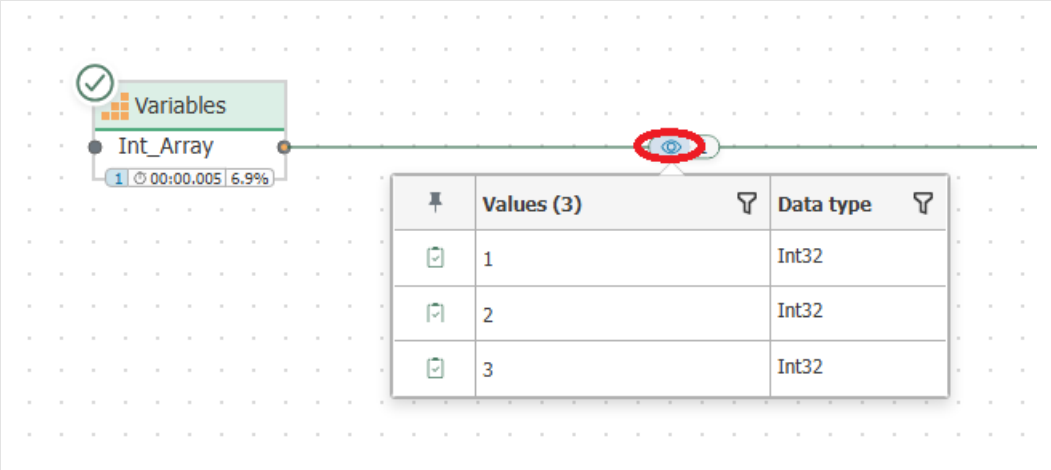
- The status bar changes color when an error occurs, a warning appears, or the connection is broken. The status bar turns

orange (checking the connection) when the connection is checked and shown in red if the connection is not present (error in the plug-in). If everything is OK, the status bar will turn green (Ready).

When the OPC Router shuts down, all connections get the status “Shutdown” In the history, the status then appears in red.

In the status tree, you can start the transfer of a connection manually by right-clicking. It is not necessary for a trigger to be present in the connection. The manually started transfers are displayed in blue in the chart. In addition, it is recorded who started this transfer.

If you click on the "eye" of a transmission, the complete array or the complete stream is displayed.



Escalation levels for status recording

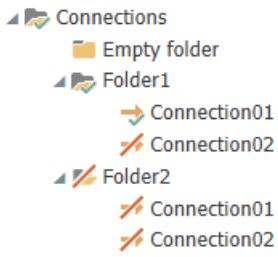
If status recording has been deactivated, there is a warning in the status display. If recording has been completely deactivated at escalation level 4, the affected period is shaded gray. \ A more detailed explanation of the escalation levels can be found on the site escalation-levels.md.

Status Indication backup

In the snap-in bar is the button for the status area.

After clicking on the button and a connection the status window opens:

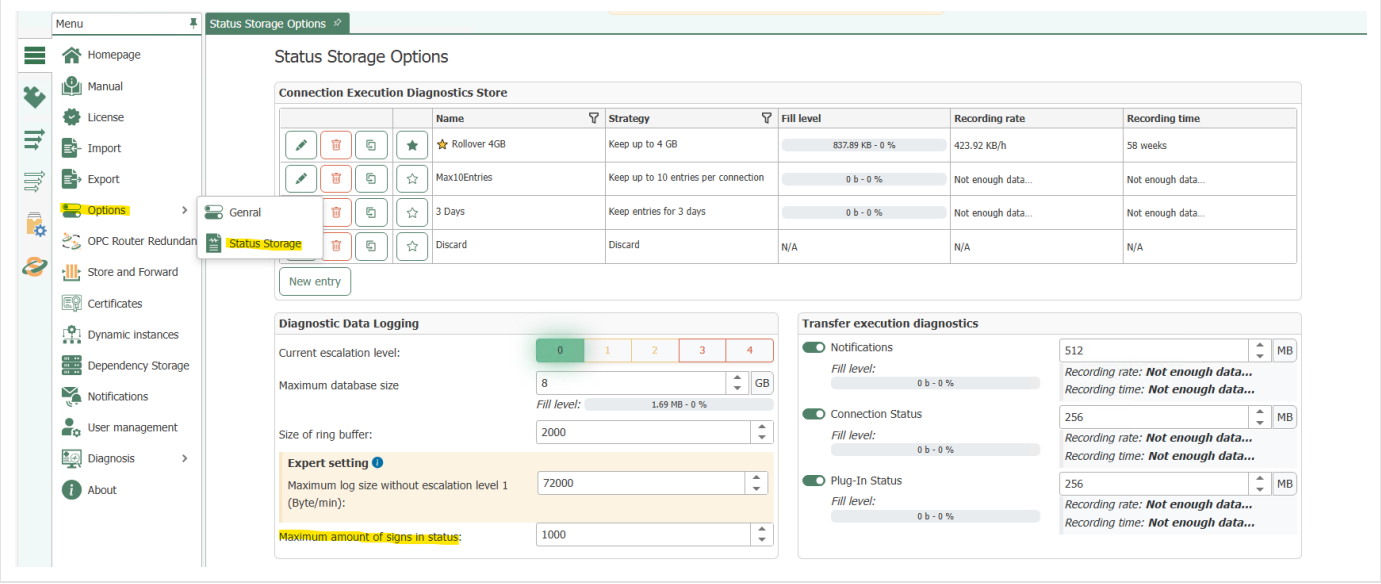
Note: Folders that are contain only deactivated connections are shown crossed out.



The Status Window

The status chart provides detailed information about the individual data transfers. The status chart can be started via the corresponding snap-in in the left margin.

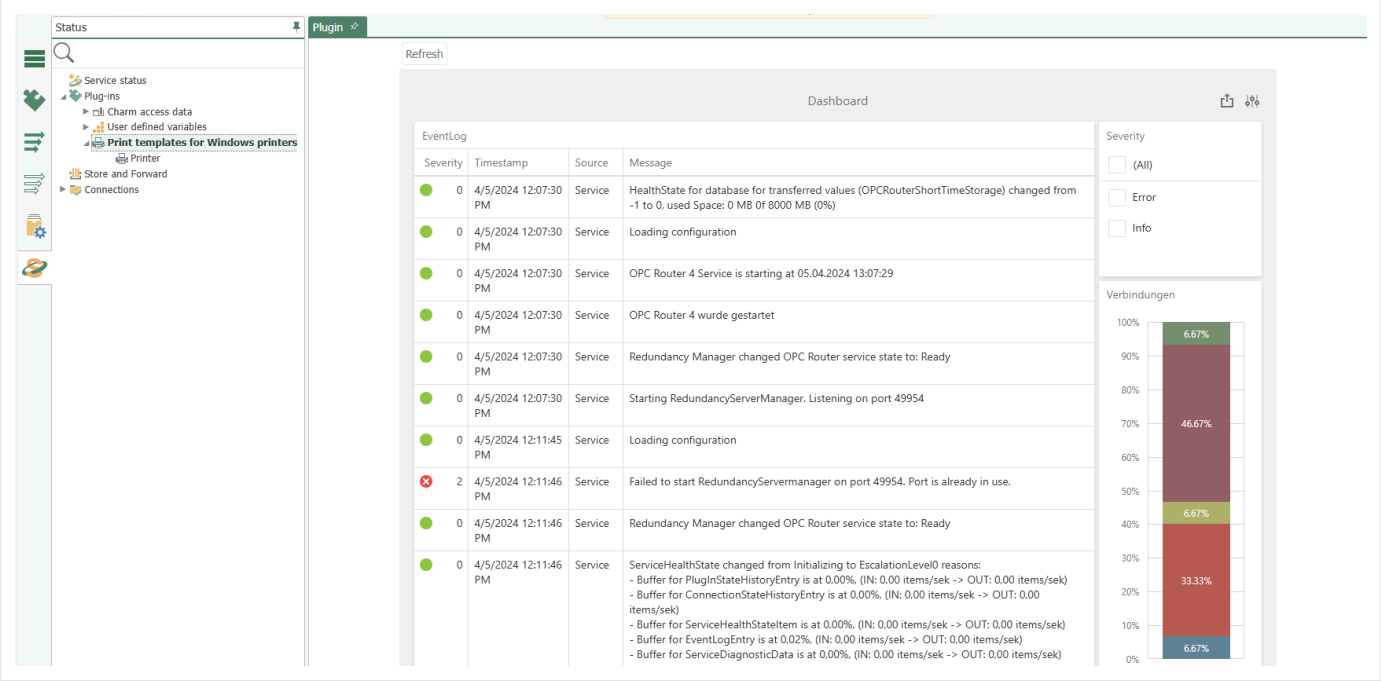
- Green transfer points represent the individual transfers that were in order and the transfer values are stored in the short storage. The lead time is defined in the settings. If you click on a transfer point where the lead time has expired, the message “Transfer values can not be displayed because the lead time of the values has expired”. The red transfer points, however, indicate erroneous transfers and are kept permanently in the Long Storage.
- If you click on a transfer point, it will be shown in light blue and the data will be displayed under the status chart.
- If you move the mouse to a transfer point, the execution time is displayed.
- Above the status chart, you will find the buttons “Previous Transfer” and “Next Transfer” to display the transfer next to the previously selected transfer. With the button “Switch to live view”, you can observe the progress of the individual transfers.
- The zoom can be changed with the mouse wheel.
- If strings or arrays are too long, the preview of the transferred value is shortened at the end – the value is nevertheless transferred completely. The length can be adjusted in the settings under Options/ Status Storage /Maximum amount of signs in status. The default value is 1000 characters.“



- Errors are indicated by the red marking in the status chart and if you move the mouse over the transfer object field under the

status chart, you will get detailed information about the error:

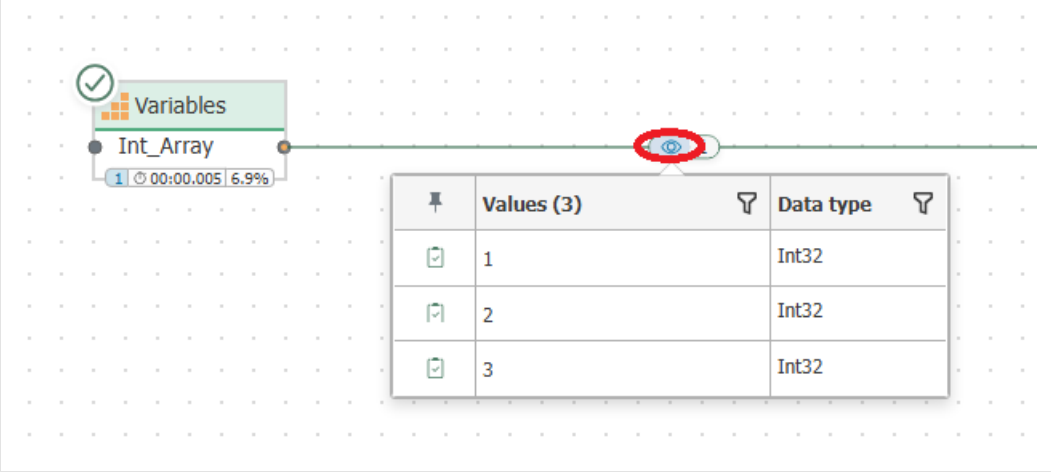
- The status bar changes color when an error occurs, a warning appears, or the connection is broken. The status bar turns orange (checking the connection) when the connection is checked and shown in red if the connection is not present (error in the plug-in). If everything is OK, the status bar will turn green (Ready).
- You can also view the messages of the plug-ins at Status snap-in.
- When you switch from status to plug-in snap-in, you automatically enter the edit mode of the open connection.
- In the status snap-in in the Plug-in tree, select a Plug-in and display the messages for it:



When the OPC Router shuts down, all connections get the status “Shutdown” In the history, the status then appears in red.

In the status tree, you can start the transfer of a connection manually by right-clicking. It is not necessary for a trigger to be present in the connection. The manually started transfers are displayed in blue in the chart. In addition, it is recorded who started this transfer.

If you click on the "eye" of a transmission, the complete array or the complete stream is displayed.



Escalation levels for status recording

The OPC Router now detects when too many status data are recorded or the database memory is almost full. In order to prevent full RAM memory, the OPC Router automatically initiates ever-increasing countermeasures. The transmission of user data is NOT

adjusted by these measures! The measures only affect the status records of the OPC Router. The recording of the user data should be ensured. In the basic settings of the OPC Router, you set how far the OPC Router is allowed to fill the status databases.

The following escalation levels are available:

- Escalation level 0:
  - Normal behavior
- Escalation level 1:
  - Ring buffer exceeded batch size
  - Status database filled to 60%
  - Action: No more transfer values of successful transfers log (only for spam connections)
- Escalation level 2:
  - Ring memory batch size exceeded by three times
  - Status database filled to 80%
  - Action: No more transfer values from successful transfers
- Escalation level 3:
  - Ring buffer size exceeded by six times
  - Status database filled to 90%
  - Action: No longer log transfer values
- Escalation level 4:
  - Ring memory batch size exceeded by ten times
  - Status database filled to 100%
  - Action: Disable status recording of transfers completely

If the status recording has been disabled, there is a warning in the progress bar. If the recording has been completely deactivated at escalation level 4, the affected period will be shaded in gray.

Network configuration

It is common, that the systems connected by the OPC Router are not installed on the same computer. This isn´t a problem for the router. However vulnerable to errors is the correct Windows configuration of the computers involved. Start a distributed router project in any case together with your network administrator. Projecting and starting your system is by the way also a service of inray.

Basically we recommend to notice following:

- Deactivate the Windows firewalls of the computers involved. Especially, if OPC server and OPC Router run on different computers using the firewall is not possible just like that, because of the dynamic port placing with DCOM. If deactivating the firewalls isn´t an option at all, please contact inray.
- Ask your network administrator to allow the communication between the systems involved. Often there are special approvals needed, if the linked server are not within the same network as the OPC Router. (e. g. sending email over an external mail server).
- Create a user,
    - who is logged in to your Windows domain
    - who has administration rights on the computer with the OPC Router
    - under whom the OPC Router service is started
    - who gains the DCOM approvals on the computer with the OPC server
    - who is also logged in over the Windows authentification as user on the MS SQL or Oracle.

User Account for Service

Changing the user account settings for a windows service.

If inray software should access other computers (e. g. with OPC servers or databases), the executing component, a Windows service as a rule, must log on this/these computer/s. It is recommended to create a Windows user within the Windows domain shared by all computers involved – especially if OPC is converned (DCOM, see below). This user should have administrative rights on the inray computer to use configuration environments as well.

1. Open services administration via “Control Panel – Administrative Tools – Services”.
2. Open service to be configured by double-click (OPCRouterService, FAS-Service, inMOVE-Service)
3. On tab “Log on”, choose option “This account”.
4. Via “Browse”, select a user (object name). Important: With “Locations”, select your Windows domain, not the local computer.
5. When having selected the user, click “OK”. You’re back in configuration dialog. Set the password.
6. Confirm with “Apply”. Windows tries to log on your domain.
7. If logging on fails, change the user from *<User>*@*<domain>* to *<domain>**<user>* and retry. No error means “succeeded”
8. Confirm with “OK”.
9. Restart service.

![Dienste anmelden](</assets/dienstanmelden (7).png>)

Distributed OPC Servers/DCOM

Requirements for a working DCOM c.

If inray software should access OPC servers that are not installed loacally (on the “inray” computer), a correct DCOM configuration on the computer with OPC servers (“OPC” computer) is necessary. Requirements:

- a user account (Windows user) within the domain shared by the two computers (NT authentication). inray services will run under this user account. Should, however, no domain user be available, each of the computers needs a local user with exactly the same account name and password.
- the Windows firewall on both computers has to be deactivated (dynamic ports under DCOM!). Should this be impossible, please contact inray.

DCOM Configuration

Short manual to configure DCOM.

Please note that this configuration has to be processed for objects both “OPCEnum” *and each* OPC server to connect. You need administrative rights on the OPC server’s computer.

1. Open component service via “Start – Run – `dcomcnfg`”.
2. Navigate to “Component Services – Computers – My Computer – DCOM Config”.
3. Open component “OPCEnum” (Right-click – Properties).
4. Set the options as follows:

Tab: General

Authentication Level	Connect

Tab: Security

Launch and Activation Permissions	Customize – “Edit”: Add in dialog “Launch and Activation Permission” your Domain-User (which should be added at this point, s. above). All four permissions should be allowed.
Access Permissions	Customize – “Edit”: Add in dialog “Access Permission” your Domain-User (which should be added at this point, s. above). Both permissions should be allowed.
Configuration Permissions	Customize – “Edit”: Add in dialog “Configuration Permission” your Domain-User (which should be added at this point, s. above). Both permissions should be allowed.

1. Confirm with “OK”.
2. Set options accordingly for *each* OPC server. You’ll find KEPServerEx filed under “Kepware Communications Server”.
3. Check whether DCOM is allowed on your computer by default: “Component Services – Computer – My Computer”, “Right-click – properties”, Tab “Default Properties”
4. Close Component Services.

Distributed databases

Accessing non-local data bases with the OPC Router is not much of a challenge. Indeed administrating the rights on the data base servers is the part which could trouble you. Therefore call to your data base systems documentation.

Basically we recommend to notice following:

- Does the data base allow remote access? Does this have to be assigned to every single user?
- Does the log in procedure have to be defined (Windows-/NT-authentification or user/password)?
- Does the access to every single data base or table have to be assigned to the user?
- Do the needed functions have to be assigned explicit to the user?
- Is the needed port in the firewall open(we recommend to deactivate local firewalls)?
- Depending on the data base system the property Server/Data of the data base access data includes a different declaration (s. "Storage").

Distributed mail server

The mail server should normally not be installed locally. Basically accessing is not different as from any other e-mail program. This counts as well for mail server installed within the company network as for external ones e. g. installed server from the internet provider. The access data can be asked from your network administration.

Basically we recommend to notice following:

- The mail server must accept SMTP- or NT authentication as log in procedure.
- Does your network administrator have to explicit allow accessing an external mail server?