

OPC Router 4.9

User Guide



inray Industriesoftware GmbH
Holstenstraße 40
DE-25560 Schenefeld

Edition of 2018/11/02
Software Release: 4.9
© Copyright 2018 inray Industriesoftware GmbH

Contents

OPC Router: Welcome	4
Security Guideline and Liability Limitations	5
The Manual's Manual	7
Installation/Update	9
System Requirements	9
Update from OPC Router 3.x	10
Installation	12
OPC Router Configuration	15
Run on 64-bit environments	16
Licensing	17
OPC Router Settings	19
Settings notifications (alarm-mail)	21
Settings redundancy	23
Plug-in Configuration	24
OPC DA Server Plug-in	24
OPC UA Server Plug-in	26
Database-Plug-in	28
Plug-in for SAP Systems	37
Plug-in for Rest connection	40
ERP Webservice Plug-in	42
Mqtt Connectivity Plug-in	44
Telegram Bot-Plug-in	46
Plug-in Thingworx®	49
Mail server plug-in	51
Euromap-77-plug-in	52
Euromap 63 Plug-in	54
Charm Plug-in	56
Windows Printer Plug-In	57
Printer Plug-Ins	60
Script Plug-in	62
Variable plug-in	67
Scanner-Plug-in	68
Plug-in RFID Reader	70
Defining Connections (Projecting)	73

Creating and Managing Connections	73
Activate/Deactivate Connections	74
Project Connections	75
Sources and Destinations	78
Trigger Objects	117
Working with templates	148
Export OPC Router 4 project	153
Import OPC Router 4 project	155

OPC Router: Welcome

Dear reader,

inray's OPC Router is a software router for data routing between OPC Servers (and therefore any PLCs) and databases (and therefore almost any MES, ERP or other business administration software. Other plug-ins connect SAP systems and mail servers, print labels, update excel files (or create them), communicate with web services or process individual scripts for customized connections.

OPC Router's license per plug-in is unlimited regarding the number of connections, number and volume of data transfers. Data transfers are triggered by time, events, bits or scripts.

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.

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

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.

Security Guideline and Liability Limitations

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.

The Manual's Manual

This manual uses the following notations:

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

Terms used

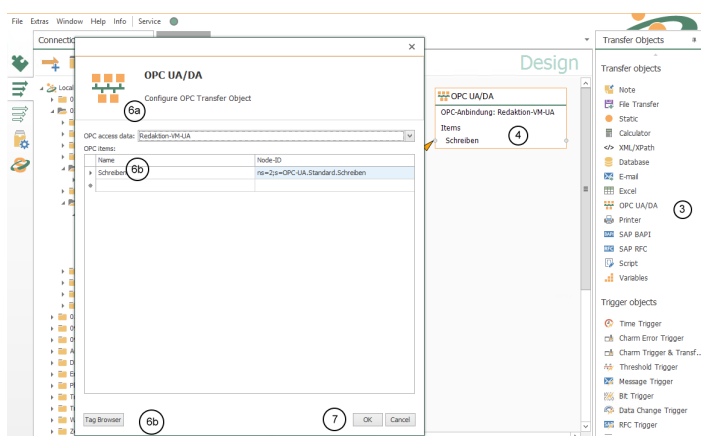
(set of) access data	Technically speaking: instance of a plug-in, e. g. the access data to one specific OPC Server, a specific database server etc. as in OPC access data, database access data etc
element (of a transfer object)	Part of a transfer object, e. g. a table's column within a database transfer 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:



Installation/Update

OPC Router Setup will install OPC Router including configuration GUI, Windows Service, OPC Router Status (runtime supervision) including all plug-ins in trial mode (2 hours runtime). 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

- hardware: We strongly recommend to run server applications on adequate hardware (multi-core processor, 4 GB RAM min., Raid-5-HDD). Should the database server run on the same system, 8 GB RAM is minimum
- Operating system: We recommend Microsoft Windows Server operating systems. Client operating systems (64 bit) can also be used for test purposes. The following Windows versions are supported:
 - Windows Server 2008 x64
 - Windows Server 2008 R2 x64
 - Windows Server 2012 x64
 - Windows Server 2012 R2 x64
 - Windows Server 2016 x64
 - Windows 7 x64
 - Windows 8 x64
 - Windows 8.1 x64
 - Windows 10 x64
 - Windows 10 IoT x64

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

Client operating systems, such as Windows 10, may experience unwanted behavior, unplanned reboots, and incomplete records due to unforeseen forced and 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.

- .NET 4.6 will be possibly installed automatically. We recommend that you download and install Microsoft manually in Microsoft .NET 4.6.1 before the update. Background are .NET updates required by Windows, which were not always offered automatically, and need sometimes longer time and may need several re-starts. This is not within the control of inray. Downtime is kept as low as possible if you uninstall the old versions after a successful installation of .NET 4.6.1. Preinstalled on Windows Server 2016 and Windows 10.
- accessible OPC Server to communicate with PLCs (OPC DA 4.0, UA, XML)
- accessible database server (MS SQL, Oracle, MySQL and all others via ODBC. MS Access and writing/reading XML files supported as well)
- for connection to Oracle databases: installed Oracle instant client (ODAC)
- SAP plug-in supports SAP systems 4.0b or newer, log-on-pad has to be installed.

Update from OPC Router 3.x

OPC Router versions 3 and 4 may run on the same computer and connect the same components without problems. Projects of version 3.x (.ort files) can be opened in Router 3 configuration GUI and thus be converted into the new format (.rcf).

Important: After conversion, the connections are present in both router 3 and router 4. Then both services can not run simultaneously on the productive system, because each transfer would be duplicated, including all SPS handshakes.

Note: The converter has been developed very carefully. However, due to the countless possibilities of OPC Router projects, we can't take any warranty for converted projects being able to run under OPC Router 3. We recommend to treat converted projects like new projects and test them carefully.

Installation, Environment

1. Backup your system.
2. Install OPC Router 4 according to the [manual](#). If at hand, use your deployment or QA system.
3. Configure the OPC Router service according to your old system. This refers especially to the user context, under which the Router service is running. Thus you ensure, that OPC Router service is granted access rights to distributed systems. In your computer's service administration, you find Router 3.x as in-rayOPCRouter and Router 4.x as OPCRouter4Service. A manual might be found in chapter network settings.

Conversion

The OPC router 4 configuration is stored in a provide database. The converter reads the data from the RCF file (OPC router 3 configuration) and and saves it in the database.

1. Copy your Router-3-project (.rcf file) and work with this copy.
2. Open Router 4 configuration. Chose "File" and "Import OPC router 4 project".
3. To open the converter, press "OK".
4. In the converter, select the RCF file. You have the following options:

<i>Ignore existing objects</i>	Objects already created in the OPC router 4 remain unaffected. If objects in the RCF file have the same name but other properties (e.g. credentials), they are not applied
<i>Overwrite existing objects</i>	Overwrites any existing objects that have the same name in the RCF file and router 4
<i>Delete all objects before converting</i>	Deletes all objects present in the Router 4 database
<i>Rebuild the database before converting</i>	Restores the entire router 4 database. An existing router 4 configuration is thus completely deleted

5. Then, select "Start Convert".

When reinstalling the user context remains.

Test and Deploy Project

- Check whether the converter has generated errors or warnings.
- For OPC-UA, DB and SAP access data, use the button "Check connection" within the plug-in configuration.
- Take a look at the single connections to see whether they seen "right".
- In the Router's Settings you can enforce logging for all transfers and adjust the log level to "Warning" and "Error". Add "Trace" if an error occurs (for short time only, as log files will be huge)

- Save and build the project.

Start OPC Router and test the project within your QA environment. The Status Window delivers a quick overview on the transfers.

If you haven't got any deployment or QA environment and thus are forced to convert directly into your productive system, please consider suitable tests. If you follow the steps above, Router 3 is still working. Please note, that data transfers will be executed twice if both Router versions are running. Stop Router 3 before starting Router 4. Monitor Router 4 carefully via Status Window. If errors occur, stop Router 4 and restart Router 3 to keep your productive system working (while you remove the cause for the error).

Installation

Important: Before installing, check whether all system requirements are met. Read the [security guidelines](#) and follow the manual carefully.

OPC Router Setup installs all components needed and configures as follows:

- Windows Service, starting automatically, meaning the runtime service executing data transfers will be started with the startup of the system. Immediately after the installation, the router service does not run because no configuration has yet been carried out. After the configuration (and after each update), start the service once manually via the button in the router configuration.

Installation

Attention: Do not remove your OPC Router 3.x. Please read the [update manual from OPC router 3 to 4](#)

1. Backup your system.
2. OPC Router Setup might update or install Windows components. Therefore, quit all running programs.
3. Check whether system requirements are met.
4. Execute set up .exe on your instal

Attention: If .NET 4.6 installation is performed, Windows will require a reboot. You may, however, install .NET 4.6 prior to installing OPC Router manually. OPC Router installation on a computer with .NET 4.6 already running will *not* require a reboot.

lation source.

5. Follow the setup routine.

After setup is completed you can configure OPC Router.

Update

Updating from a 4.x to a newer 4.x version means removing the old version and install the new version. Your project files remain unaffected.

We recommend to test a new version within your deployment or QA environment. Thus you ensure that your project file is compatible with the new version. As a rule, OPC Router 4 automatically adjusts project files to new versions (not for updates from version 3 to 4).

Attention: Do not remove your OPC Router 3.x. Please read the [update manual from OPC router 3 to 4](#)

1. Backup your system.
2. OPC Router Setup might update or install Windows components. Therefore, quit all running programs.
3. Check under which user the OPC Router service runs, and remember.
4. Remove OPC Router 4.x via control panel. Project files remain unaffected
5. Check whether system requirements are met.
6. Execute set up .exe on your installation source.
7. Follow the setup routine.
8. Check the user context of the service. If necessary, reconfigure according to step 3.
9. Open the OPC Router configuration.
10. Turn the project productively.
11. Restart OPC Router Service.

Update OPC Router to .NET 4.6

The OPC Router versions require the installation of .NET Framework 4.6 on the server. .NET 4.6 is a Windows component and requires some Windows updates, which unfortunately are not always offered automatically by Microsoft. Some of these updates are quite extensive and can take several hours. There are several restarts of the server necessary:

The following updates are needed in this order:

- KB2919442
- clearcompressionflag.exe
- KB2919355

- KB2932046
- KB2959977
- KB2937592
- KB2938439
- KB2934018

If the installation of .NET 4.6 fails, it may have other causes. However, the absence of KB2919355 and the following updates was the only one that was manifested in our tests. We recommend the following procedure:

1. From the control panel, verify that the above updates are installed.
 - If so, install .NET 4.6 manually. If the installation fails, follow the help link suggested in the installer.
 - If not, search support.microsoft.com for the update KB2919355. You will find detailed instructions there. Remember to run KB2919442 and `clearcompressionflag.exe` before.
2. Then install your OPC router version.

OPC Router Configuration

The OPC router 4 configuration contains the configuration, the possibility of productive switching and the status display at runtime.

You will also find there

- OPC router converter, which converts OPC router 3 projects
- OPC router service Status, which executes the data transfers as a run-time component.

Configuration is being started via start menu (Programs – inray – OPC Router).

OPC Router Configuration executes all steps of bringing OPC Router into service, including licensing:

1. *Configuration of plug-ins needed*, e. g. access data of accessible OPC and database servers, definition of variables (Chose plug-in in project configuration tree, left-hand side, then edit on corresponding tab).
2. *Projection (of connections)*: Which data is to be transferred when wherefrom where? (Create and open connections in left-hand side project configuration, chose transfer objects on right-hand side toolbox, edit on corresponding tab)
3. Productive switching of the whole project or individual connections.
4. Start service – if service has been running (default), restart is executed automatically (button).
5. Monitoring and error diagnostics left over the snap-in directory on button “Status”.

The following chapters describe the steps in detail. Please note that only licensed plug-ins are accessible.

The entire project is stored in local databases (MS SQL Local-DB), including the transfer protocol.

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.

OPC Router Service is the runtime component, working in the background as a Windows service and executing the data transfers. The service must be restarted after changes to plug-in configurations, individual connections can also be productive without restarting. Individual configuration regular is only necessary for [use in systems with windows authentication](#).

Run on 64-bit environments

The OPC router setup will automatically set up the service for the 64-bit architecture.

Important: If you provide 64-bit drivers for connection of external systems (e. g. SAP systems, OLE-DB driver) only in a 32-bit version, you must configure the router service manually.

1. Go to “Service” in the top menu bar.
2. Under menu “Service”, select “Registry”.
3. Force the entry to “Register with: 32 bits”.
4. Restart the OPC router.

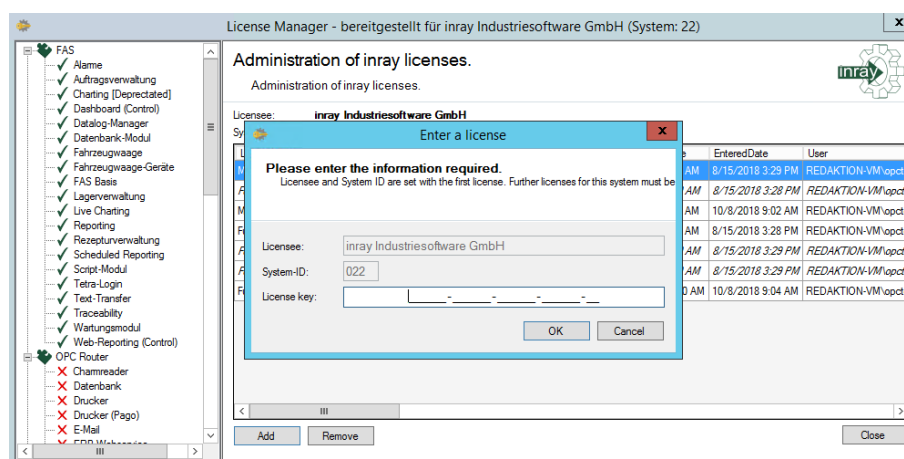
Licensing

Without license each inray software is executable for two hours after installation. You can test all standard modules in trial mode. After entering the license key the licensed modules are usable only.

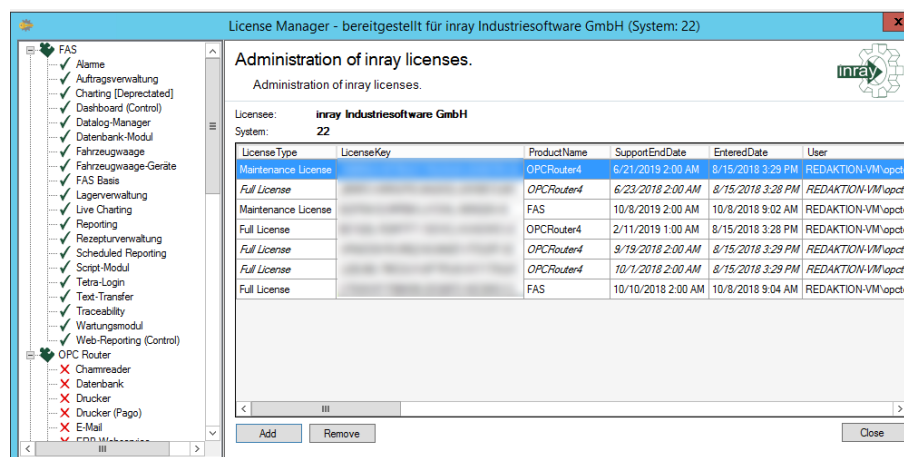
If you need a longer demo license, this can be issued by our sales department, e.g. for a 30-day demo license. Please contact our sales department.

The licensing occurs by a particular program, the inray license manager. You start the license manager either via the Start menu (program - inray) or with corresponding buttons in the programs (FAS configuration, OPC router with “Options” – “Licensing”, FAS-inMOVE assistant). The license manager shows all inray licenses at this system. Have the necessary data ready for licensing.

1. Open the license manager.
2. Choose “Add” to enter a new license.
3. Enter the company name, system ID and license key you got from inray or your reseller. Input fields are case sensitive. If you enter a second or further license, the key has to be issued for exactly the same name and system ID:



4. Save with "OK". The license manager will be shown the successful licensing in the overview:



5. Restart your software.

Future system extensions are licensed for your specific customer name and system ID. Then simply enter the new license key.

We recommend the additional conclusion of a Maintenance License in order to be able to use the support of the company inray Industriesoftware GmbH optimally. We offer you support during normal business hours or, if booked accordingly, a round-the-clock service. If you have signed a Maintenance License for your licensed modules (Full License), then the SupportEndDate shows the date, how long you can use the support. During this time, upgrades to a new major version are also possible. (If a Maintenance License is concluded, a major upgrade from FAS 3.8x to FAS 3.9x, for example, is possible if the upgrade has appeared before the expiry of your contract. The release date applies.) Without the Maintenance License only upgrades within one version are possible (e.g. FAS v3.8.5001.52 to FAS v 3.8.5007.60) or bugfixes can be loaded.

OPC Router Settings

The following options can be set by opening “Settings” in the project configuration:

Runtime

<i>Cycle time</i>	Internal cycle time in milliseconds. This interval is used e. g. to check whether trigger conditions apply. Set a shorter interval if necessary for your project.
<i>Maximum number of threads</i>	Select the maximum number of threads.
<i>Size of ring buffer</i>	Select the needed size of ring buffer.
<i>Transfer break in case of errors</i>	Transfer pause in seconds in case of error.
<i>Connection time-out</i>	Time (in s) that the connection may take before a connection error is returned.

Notification Manager

<i>Storage time for events</i>	Time range in which the data are kept.
<i>Storage time for transferred values</i>	Time range in which the data are kept in case of faults.
<i>Log transferred values</i>	Activate if transferred values are to be recorded
<i>Size of ring buffer</i>	Select the needed size of ring buffer.
<i>Maximum log size for transferred values (GB)</i>	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 a LocalDB database.
<i>Maximum log size for events (GB)</i>	Specify the maximum log size for events in GB.
<i>Maximum log size at escalation level 1 (bytes / min)</i>	Specification of the log size at escalation level 1 in bytes / min. Setting when connections are considered too data intensive. If connections transmit more bytes per minute than set, they will be identified as the "spam connection" and will be disabled first in case of memory problems. For this purpose, the number of transfers is not directly relevant, but the amount of data transferred.

Log output

<i>Log all transfers</i>	Writes all transfers to log file. This option overrides deactivation of logging in single connections.
<i>Zip log files</i>	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.
<i>Error</i>	Logs errors. Recommended during normal operation.
<i>Warning</i>	Logs warnings. Recommend during normal operation.
<i>Trace</i>	Detailed logging to trace errors. Should be deactivated during normal operation.
<i>Debug</i>	Logs debug information for developers. Should be deactivated during normal operation.
<i>Log folder</i>	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.

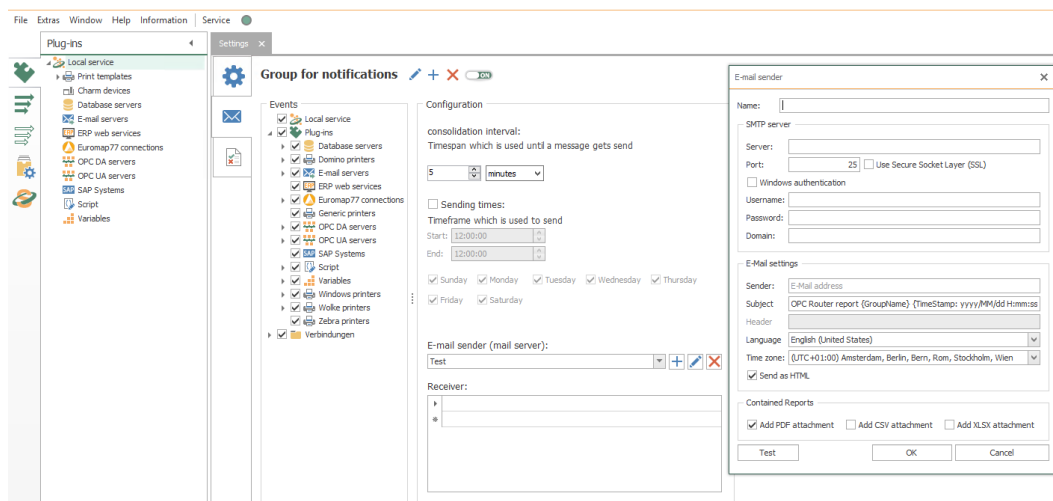
Settings notifications (alarm-mail)

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)
- service 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.



Einstellungen

<i>consolidation interval</i>	The amount of time before a message is sent.
<i>Tiemsplan which is used until a message gets send</i>	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.
<i>E-Mail sender (mail server)</i>	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.
<i>Receiver</i>	At this point, enter the recipients who should receive this notification.

E-mail sender name

<i>Name</i>	Give the e-mail notification a name.
-------------	--------------------------------------

SMTP-Server

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

E-Mail settings

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

Enthaltene Reports

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

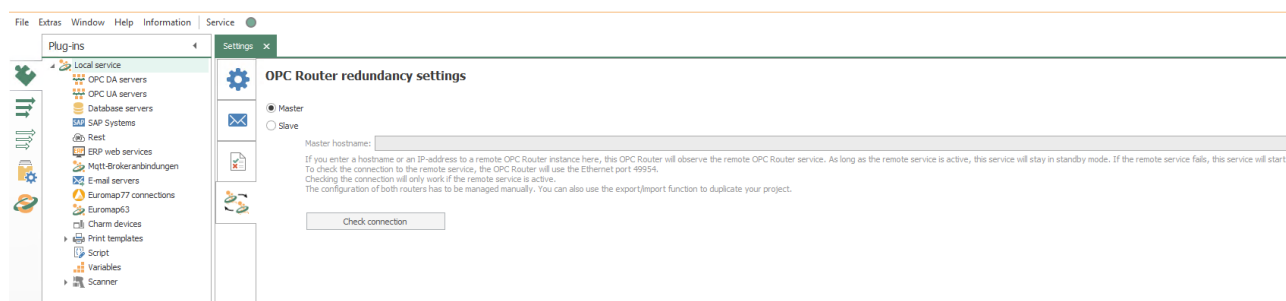
The "Test" button allows you to verify that the server connection is working.

Confirm your entries with "OK".

Settings redundancy

Via the menu item Extras - Settings you will find the redundancy area at button for the basic settings. This way, the OPC router can be protected against failures.

You can configure a master host name in this area and test the connection. There is a “master” and a “slave”. The slave starts in standby mode and monitors the master. Each OPC router is automatically a master. Here only the port in the firewall must be released. For a slave only IP / DNS of the master has to be configured. If the connection leaves from the client to the master, the connections are switched to active and begin to transfer. This is shown in green in the status display. If the client can reconnect to the master, the connections switch back to standby. This is shown in blue in the status display.



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.

Plug-in Configuration

OPC DA Server Plug-in

OPC plug-in connects OPC Router to all accessible OPC Servers.

If an OPC Server runs on a different computer, ...

- ... DCOM configuration has to be adjusted for *OPC DA* connections.

Operation is standard:

1. Open plug-in by double-click "OPC DA servers". The list of created OPC connections is displayed.
2. Chose "Add" or double-click an existing entry to edit.
3. Specify access data. (chapter "[Properties of OPC access data](#)").
4. Confirm with "OK".

The OPC server can now be used to define a connection.

Properties of OPC DA access data

The following properties have to be specified for each set of OPC access data:

OPC DA Access Data

<i>Name</i>	This set's name. The name will be shown in projection. Select a name you will recognize!
<i>Host/IP</i>	IP or host name of OPC Server's computer.
<i>OPC Server</i>	OPC Servers accessible on computer specified under <i>Host/IP</i> . Select by drop-down
<i>Update rate</i>	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.
<i>Dead band</i>	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 reli-

	even the network. This function is optional and does not have to be implemented by all OPC servers.
<i>Access path</i>	Access path on OPC Server

Extended Properties

<i>Transfer OPC null values</i>	Invalid OPC values will be transferred as well, e. g. as NULL into databases.
<i>Define types manually</i>	Define data types of OPC items manually to notify the OPC Server what type is expected.
<i>Activate Redundancy</i>	The redundancy is activated if the primary server is not available and (optional) if the primary server's life can not be read or assumes the value.

Use *Access Path* to simplify projection: If an access path is specified, you only have to enter the OPC Items' names later on.

Just define a set of access data for each PLC to connect. In your OPC Server, define an access path to each PLC. This access path has to be entered here.

If you want to connect different PLCs with identical items, you can use this alternative in combination with templates. Just define the OPC access data as variable within

your template. Specify the set of access data when creating the connection from the template.

OPC UA Server Plug-in

OPC plug-in connects OPC Router to all accessible OPC Servers.

If an OPC Server runs on a different computer, ...

- ... for *OPC UA* the router have to be approve as “Trusted Client”, if an appropriate safety level is set on the OPC server

Operation is standard:

1. Open plug-in by double-click “OPC UA servers”. The list of created OPC 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 OPC 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 server can now be used to define a connection.

Properties of OPC DA access data

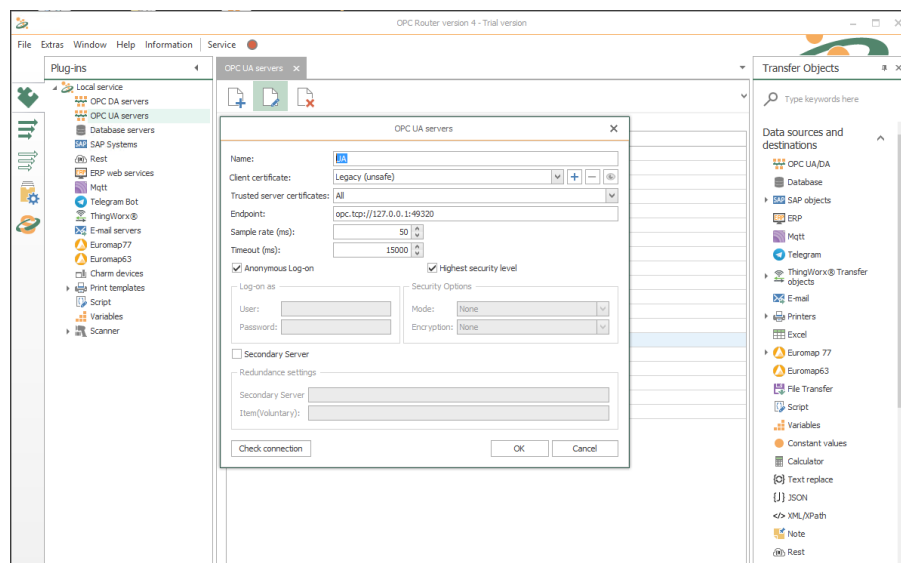
The following properties have to be specified for each set of OPC access data:

OPC UA Access Data

<i>Name</i>	This set's name. The name will be shown in projection. Select a name you will recognize!
<i>Client certificate</i>	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.
<i>Trusted server certificates</i>	Select which server certificates to accept.
<i>Endpoint</i>	OPC-Server's address (protocol: opc.tcp, http or https)

	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. Then the certificate is rejected by the OPC server.
<i>Sample Rate</i>	Sample rate (ms) used by OPC Server to refresh values. Set "0" to use smallest update rate.
<i>Timeout</i>	Time (ms) allowed to establish before a connection error is thrown.
<i>Anonymous log-on</i>	Activate if your OPC Server allows anonymous connections. If not, deactivate and provide user credentials.
<i>User</i>	User name to log on OPC Server (<i>not</i> : Windows Server)
<i>Password</i>	Password to log on OPC Server (<i>not</i> : Windows Server)
<i>Highest security level</i>	Activate to use highest security level provided by your OPC Server. If connection problems occur, deactivate and set security options manually.
<i>Mode</i>	"None", "Sign" or "SignAndEncrypt".
<i>Encryption</i>	Encryption process. Use highest encryption provided by your OPC Server.
<i>Redundance server</i>	Activate to specify a redundancy OPC server. Redundancy is activated if the primary server cannot be reached and (optionally) if the life item of the primary server cannot be read or assumes the value "0". Then enter the redundancy server

and, if required, an object (item) on the selected redundancy server.



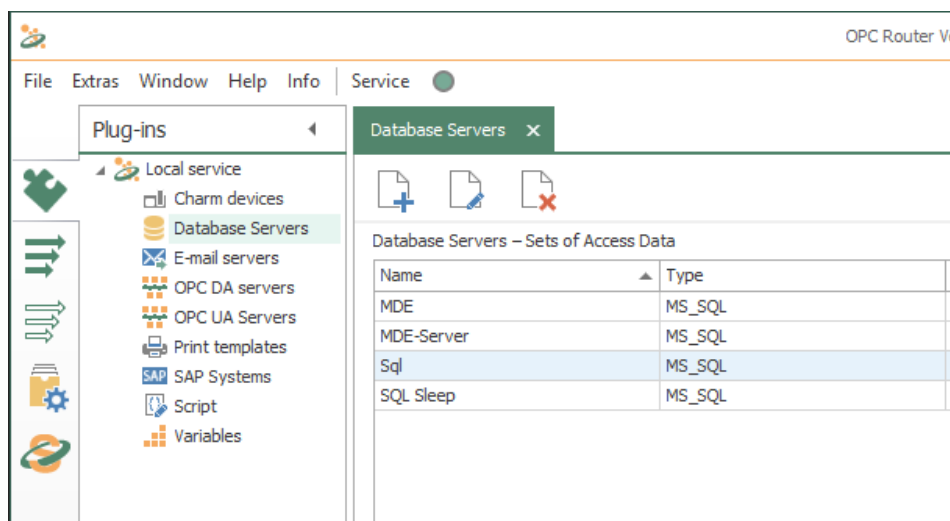
Property *Access Path* is not available with OPC UA. If you want to connect several PLCs with identical data points, project a connection template and define “Access Path” as template variable. This might be a real path if your OPC Server provides strings as node IDs or the namespace’s ID. You’ll have to configure an individual namespace for each PLC. The data point IDs have to be identical.

Use the “Check connection” button to check in advance if your data connections are working.

Database-Plug-in

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 plug-in by clicking “Database Servers” in the project configuration’s plug-in list.
2. Chose “Add” or double-click an existing entry to edit.
3. Specify access data. (chapter “[Properties of DB access data](#)”).
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.

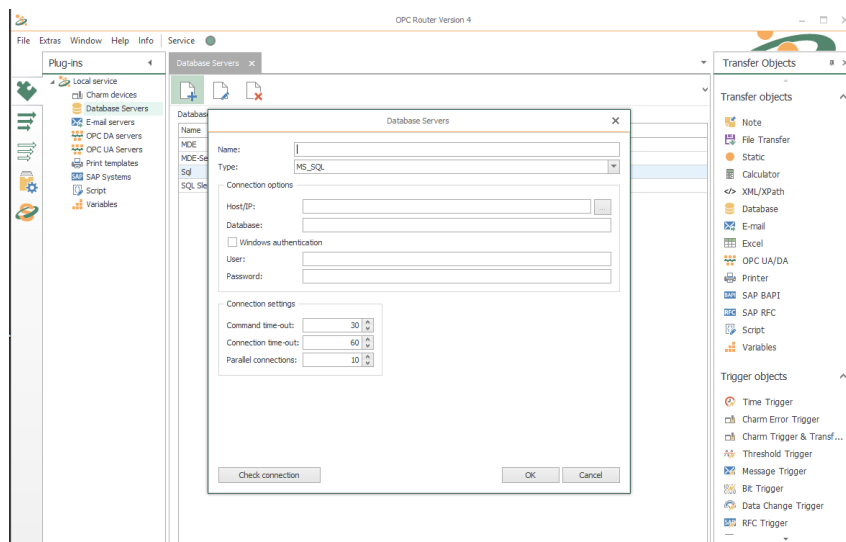
Properties of Database Access Data

The following properties have to be specified for each set of database access data:

<i>Name</i>	This set’s name. The name will be shown in projection. Select a name you will recognize!
<i>Type</i>	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.



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.

Options for MS SQL Servers

<i>Host/IP</i>	Name of MS SQL Server (named pipe or <i><IP>: <Port></i>), specifying a port number enforces communication via TCP/IP.
<i>Port</i>	Selection of the port for the database connection. If necessary, the port can be changed.
<i>Database</i>	Database on server to be used.
<i>Windows authentication</i>	Activate to use Windows authentication instead of <i>User name</i> and <i>Password</i> . Local accounts, DCOM users and Windows domain users (NT authentication) are supported. To use <i>Windows authentication</i> , 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!
<i>User</i>	User name to log on MS SQL Server
<i>Password</i>	Password to log on MS SQL Server
<i>Command time-out</i>	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.
<i>Connection time-out</i>	Time (seconds) that the Router may need to establish connection.
<i>Parallel connections</i>	Number of connections that the Router may establish parallelly. 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.
<i>Polling interval</i>	Distance at which the database is queried.

Options for XML files

The XML file must exist.

<i>File</i>	Select file via "..."
-------------	-----------------------

Please note that the file's structure must be readable, e. g.:

```
<measurement>
  <temperature>
    <ID>1</ID>
    <time>2012-01-31T12:00:00</time>
    <value>5</value>
  </temperature>
  <temperature>
    <ID>2</ID>
    <time>2012-01-31T12:05:00</time>
    <value>4</value>
  </temperature>
  <pressure>
    <ID>1</ID>
    <time>2012-01-31T12:00:00</time>
    <value>1200</value>
  </pressure>
  <pressure>
    <ID>2</ID>
    <time>2012-01-31T12:05:00</time>
    <value>4</value>
  </pressure>
</measurement>
```

Speaking in database terms, this means, we have got a database measurement containing tables temperature and pressure, each containing columns ID, time and value. Each table in this file has got two records. The XML elements must not contain attributes!

From this file, you will be able to read data. Please note that a column ID has to be part of each table.

To write data to an XML file, each table needs a primary key. This can be realized with an XML schema (XSD = XML Schema Definition). This XSD file is referenced in the root element of your XML file. The following example shows the reference to an XSD file stored in the same location as the XML file.

```
<measurement xmlns="measurement.xsd">
...
</measurement>
```

The referenced file measurement.xsd contains the definition:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="measurement" targetNamespace="measurement.xsd"
  xmlns:mstns="measurement.xsd" xmlns="measurement.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"
  xmlns:msprop="urn:schemas-microsoft-com:xml-msprop"
  attributeFormDefault="qualified" elementFormDefault="qualified">

  <xs:element name="temperature"
    msprop:Generator_UserTableName="temperature">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ID" msdata:ReadOnly="true"
          msdata:AutoIncrement="true" type="xs:int" />
        <xs:element name="time" msdata:Caption="Zeit"
          type="xs:dateTime" minOccurs="0" />
        <xs:element name="value" msdata:Caption="Wert" type="xs:int"
          minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="pressure"
    msprop:Generator_UserTableName="pressure">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ID" msdata:ReadOnly="true"
          msdata:AutoIncrement="true" type="xs:int" />
        <xs:element name="time" msdata:Caption="Zeit"
          type="xs:dateTime" minOccurs="0" />
        <xs:element name="value" msdata:Caption="Wert" type="xs:int"
          minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="indexes" msdata:IsDataSet="true">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="temperature" />
        <xs:element ref="pressure" />
      </xs:choice>
    </xs:complexType>
    <xs:key msdata:PrimaryKey="true"
      msdata:ConstraintName="temperatureID" name="temperatureID">
      <xs:selector xpath="./temperature" />
      <xs:field xpath="ID" />
    </xs:key>
  </xs:element>
</xs:schema>
```

```

    </xs:key>
    <xs:key msdata:PrimaryKey="true"
msdata:ConstraintName="pressureID" name="pressureID">
      <xs:selector xpath="//pressure" />
      <xs:field xpath="ID" />
    </xs:key>
  </xs:element>
</xs:schema>

```

The first two elements define the tables' layout, the third element the indexes. Relations could be realized as well. Please note that each table has to have at least one record with a valid ID so that the auto increment function works correctly (IDs of new records are set automatically).

Important: In the Router's plug-in configuration, you specify the XML file, not the schema (XSD). The XSD file is only referenced from the XML file as shown above.

Options for MySQL/MySQL-5 Servers

<i>Host/IP</i>	IP address of MySQL server.
<i>Port</i>	Selection of the port for the database connection. If necessary, the port can be changed.
<i>Database</i>	Database on server to be used.
<i>User</i>	User name to log on MySQL
<i>Password</i>	Password to log on MySQL
<i>Command time-out</i>	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.
<i>Parallel connections</i>	Number of connections that the Router may establish parallelly. 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.
<i>Polling interval</i>	Distance at which the database is queried.

Options for MS Access

<i>File</i>	Choose Access file via button "...". In addition to the .mdb format .accdb files are also supported.
-------------	--

<i>Database</i>	Name of “database” to be used
<i>User</i>	User name (if file is protected)
<i>Password</i>	Password (if file is protected)
<i>Command time-out</i>	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.
<i>Parallel connections</i>	Number of connections that the Router may establish parallelly. 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.
<i>Polling interval</i>	Distance at which the database is queried.

Options for Oracle

To connect to Oracle servers, the Oracle client has to be installed.

<i>TNS name</i>	TNS name of Oracle configuration
<i>Port</i>	Selection of the port for the database connection. If necessary, the port can be changed.
<i>Schema</i>	Oracle schema
<i>Windows authentication</i>	Activate to use Windows authentication instead of <i>User name</i> and <i>Password</i>. Local accounts, DCOM users and Windows domain users (NT authentication) are supported. To use <i>Windows authentication</i>, 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. Observe carefully if transfers are executed correctly!
<i>User</i>	User name to log on Oracle
<i>Password</i>	Password to log on Oracle
<i>Command time-out</i>	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.
<i>Parallel connections</i>	Number of connections that the Router may establish parallelly. 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.
<i>Polling interval</i>	Distance at which the database is queried.

Options for OLE DB Providers

OLE DB connections need a connection string. The OLE DB provider needed has to be installed and must be active.

<i>Connection string</i>	Select a connection string from the drop-down menu.
<i>Command time-out</i>	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.
<i>Connection time-out</i>	Time (seconds) that may take to establish a connection to the database.
<i>Parallel Connections</i>	Number of connections that the Router may establish parallelly. 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.
<i>Polling interval</i>	Distance at which the database is queried.

Options for ODBC drivers

The ODBC driver needed must be installed and active.

<i>ODBC entry</i>	Select configured ODBC driver.
<i>User</i>	User name to log on database
<i>Password</i>	Password to log on database
<i>Command time-out</i>	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.
<i>Parallel connections</i>	Number of connections that the Router may establish parallelly. 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.
<i>Polling interval</i>	Distance at which the database is queried.

Options for dBASE

<i>Directory</i>	Select an appropriate directory from the drop-down menu.
<i>Command time-out</i>	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.
<i>Parallel connections</i>	Number of connections that the Router may establish parallelly. 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.
<i>Polling interval</i>	Distance at which the database is queried.

Options for SQLite

<i>File</i>	Select an appropriate file from the drop-down menu.
<i>Password</i>	Password to log on database
<i>Command time-out</i>	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.
<i>Connection time-out</i>	Time (seconds) that the Router may need to establish connection.
<i>Parallel connections</i>	Number of connections that the Router may establish parallelly. 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.
<i>Polling interval</i>	Distance at which the database is queried.

Plug-in for SAP Systems

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 GUI must be installed. SAP files have not be delivered by inray Industriesoftware GmbH. Therefore you need a current SAP account (status 2018).

Defining access data:

1. Open plug-in by clicking "SAP systems" in the project configuration's plug-in list.
2. Chose "Add" or double-click an existing entry to edit.
3. Specify access data. (chapter [Properties of SAP access data](#)).
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".

Properties of SAP Access Data

The following properties have to be specified for each set of SAP access data:

<i>Name</i>	This set's name. The name will be shown in projection. Select a name you will recognize!
<i>Login method</i>	Select the desired login method.
<i>User</i>	User name to log onto SAP system.
<i>Password</i>	Password to log onto SAP system.
<i>Client</i>	SAP client (company code).

<i>Language</i>	Enter the language you want to use here. For example, "de" for German, "en" for English etc.
-----------------	--

Direct log-in

<i>Host/IP</i>	Server (Name or IP) with SAP system. Explicit log-on is required (no load balancing).
<i>System number</i>	Number of SAP system.

Load balancing

<i>Message server</i>	Name or IP of SAP message server
<i>Log-on group</i>	Log-on group (of SAP system)
<i>System number</i>	Number of SAP system

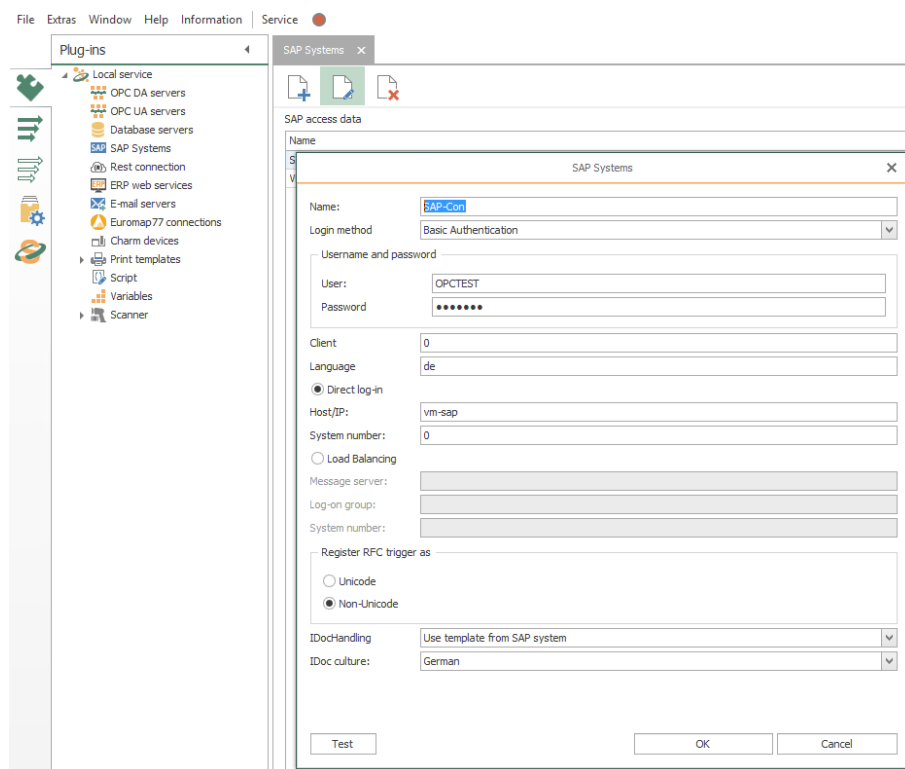
Register RFC Trigger as

<i>Unicode</i>	Register RFC triggers as unicode
<i>Non-Unicode</i>	Register RFC triggers as non-unicode

IDocHandling

<i>IDocHandling</i>	Select when you want to create an empty IDoc or use a template from the SAP system.
<i>IDoc culture</i>	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

You can use the button “Test” to check whether your connection is successful.



If you have selected Secure Network Communications (SNC) at Login method, please make the SNC settings below:

SNC Einstellungen

<i>Name of SNC partner</i>	Entry the SNC partner.
<i>Security level</i>	Selection of the security level from the drop-down menu e.g. Maximum or Standard.
<i>Authentication method</i>	Selection between NTLM Authentification or Kerberos5 Authentication.

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 Wenn das SAP-System einen anderen Port benutzt und das SAP-GUI nicht installiert ist, muss der Port angepasst werden:

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".

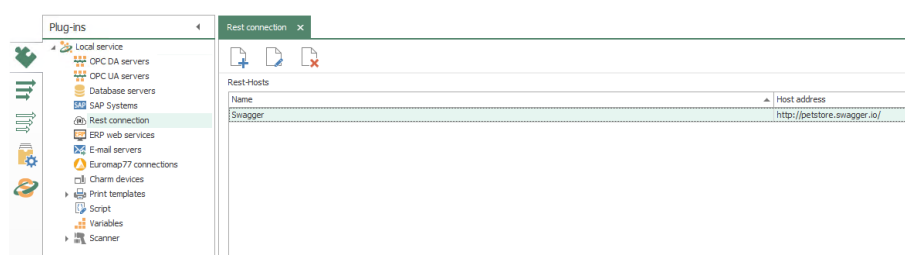
```
sapgw00 <Portnummer>/tcp
```

4. Save the file.

OPC Router will use that specific port.

Plug-in for Rest connection

The Rest plug-in allows you to submit queries from web services and generate mappings from them.



The operation is standard:

1. Open the plug-in by double-click on "Rest connection" at menu "Plug-ins"
2. Choose "Add new set of access data" or double-click an existing entry.
3. Make the necessary settings (see section "[Properties of Rest connection](#)").
4. Confirm with "OK".

The remaining connection is now available for configuring a connection.

With button "Test" you can check the connection.

Properties of Rest connections

You can set the following properties in the configuration of the Rest plug-in

<i>Name</i>	Any name where this connection appears in the connection configuration.
<i>Host - URL</i>	URL the web service to be connected.
<i>OpenAPI - Path</i>	URL, the programming interface API. The API transfer object sends the response data by means of the remaining trigger.
<i>Authentication</i>	If you need to log in to the web service, select the desired type here. You will receive this information from the administrator of the web service.
<i>Http - Header</i>	Fields for transferring the important http parameters.

<i>Timeout (s)</i>	Time (in seconds) that the connection may take before a connection error is returned.
<i>Trusted Certificates</i>	Selection which certificates of the OPC router the user should trust. • Alle: Every certificate is accepted. • Router: Connection is allowed if the certificate is at the router certificate management and trusted. • Windows: Connection is allowed if the certificate is found in the Windows certificate store.
<i>Certificate</i>	Selection of a client certificate from the drop-down list sent by the OPC router. 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.

Rest

Name:

Swagger

Host-URL:

http://petstore.swagger.io/

OpenAPI-File:

Authentication:

None

Http-Header:

	Name	Value
*		

Timeout (s):

30

Trusted Certificates:

All

Certificate:

Without

+

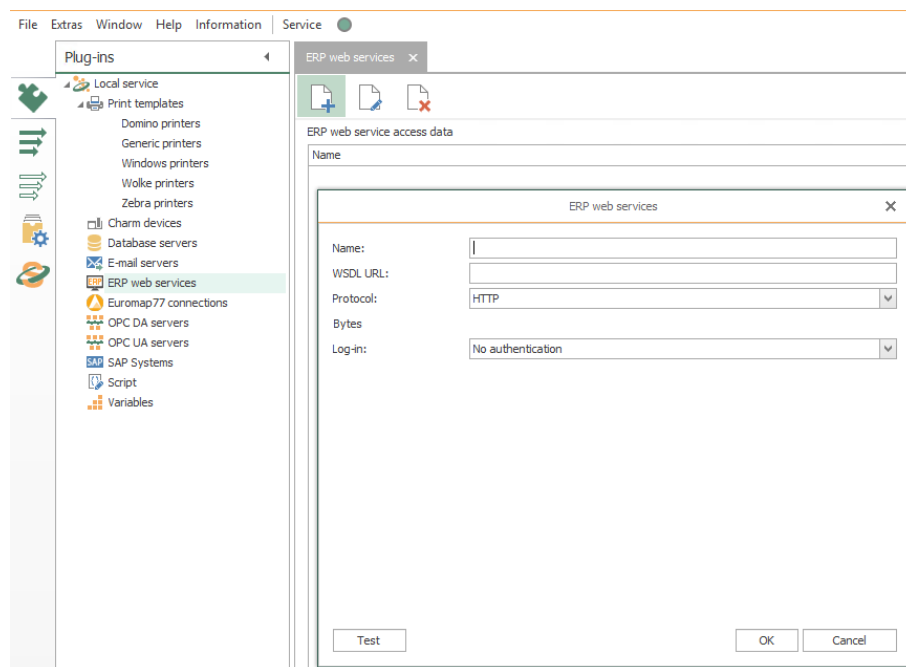
-

OK

Cancel

ERP Webservice Plug-in

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 “[Properties of ERP access data](#)”).
4. Confirm with “OK”.

This set of access data can now be used to define a connection.

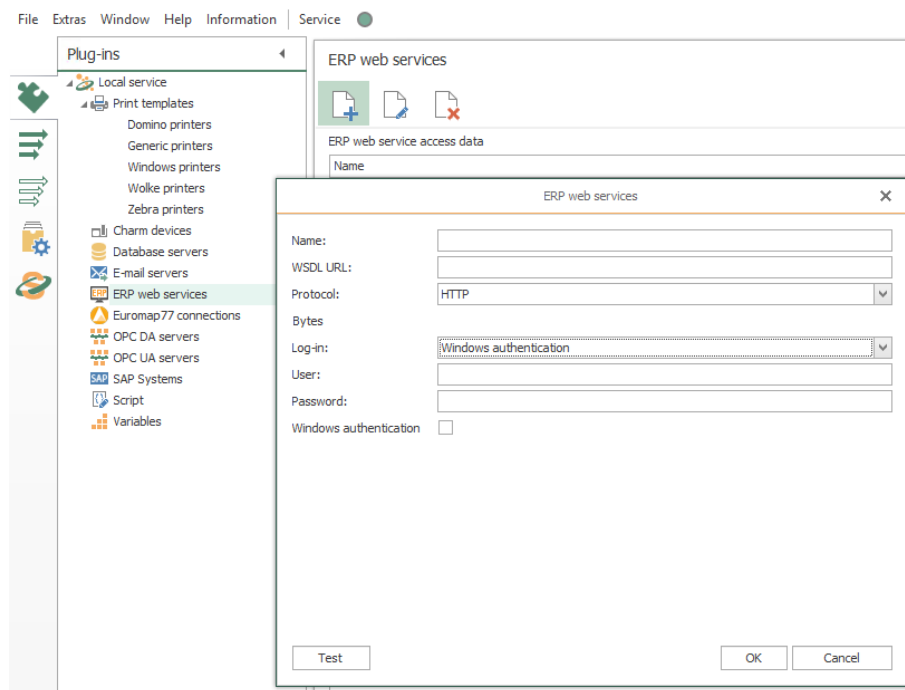
With the button “Test” you can check your connection.

Properties of ERP access data

Set the following properties in the ERP plug-in configuration:

<i>Name</i>	Any name under which this web service appears in the connection configuration. Select a name you will recognize!
<i>WSDL - URL</i>	URL of the WSDL file.
<i>Protocol</i>	Selection of a transmission protocol from the drop-down menu.
<i>Log-in</i>	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 <i>Integrated Security</i>, 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.
<i>User</i>	User name to log on web service (at basic and NTLM authentication)
<i>Password</i>	Password to log on web service (at basic and NTLM authentication)
<i>Windows authentication</i>	When enabled, the user data where the router service is running.



Authentication in Method

If you use the policy “in web service method”, please set log-in parameters via button “Log-in parameters”.

Mqtt Connectivity Plug-in

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 double-clicking on “Mqtt” at the Plug-ins menu.
2. Choose “Add new set of access data ”or double-click on an existing entry.
3. Make the necessary settings (see section “[Properties of Mqtt connections](#)”).
4. Confirm with “OK”.

The Mqtt connection is now available for configuring a connection.

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

Properties of Mqtt Connections

You set the following properties in the configuration of the Mqtt plug-in:

<i>Name</i>	Assign a unique name to the Mqtt connection.
<i>Brokeranbindung</i>	Enter the address of the broker.
<i>Port</i>	Port on the server used for the Mqtt connection
<i>SSL</i>	Uses SSL encryption. The server must support this, the corresponding <i>Port</i> must be set.
<i>Continuous session</i>	Activate if a persistent connection is desired. Missed messages are sent afterwards.
<i>Client ID</i>	Display of the Client ID. The button “Create new Client ID” creates a new ID.
<i>Create new Client ID</i>	Creates a new Client ID that is displayed in the <i>Client ID</i> field.
<i>Username</i>	User name for logging in to the server.
<i>Password</i>	Password for logging in to the server.
<i>Certificate</i>	Selection of a client certificate from the drop-down list, which the OPC Router also sends.
<i>MQTT version</i>	Select the Mqtt protocol version from the drop-down menu.
<i>Quality of Service type</i>	Selection the Standard Quality of Service: • 0 = at most once • 1 = at least once • 2 = exactly one
<i>Wait for transfer</i>	Activate if you want to wait for the transfer by default.

Tab: First will

<i>Activate first will</i>	Activate checkbox to activate the “First will” checkbox. Activity when establishing a connection.
<i>Topic</i>	Specifies which topic the first will to look at.
<i>Payload</i>	Specifies which user data is to be transmitted during communication.
<i>Quality of Service</i>	Selection the Quality of Service: • 0 = at most once • 1 = at least once • 2 = exactly one
<i>Hold</i>	Activate if you want the broker has to hold the data.

Tab: Last will

<i>Activate last will</i>	Activate checkbox to activate the last will. Activity before a connection termination.
<i>Topic</i>	Specifies which topic the load should look at.
<i>Payload</i>	Specifies which user data has to be transmitted during communication.
<i>Quality of Service</i>	Selection the Quality of Service: • 0 = at most once • 1 = at least once • 2 = exactly one
<i>Hold</i>	Activate if you want the broker has to hold the data.

Send last will when clearing connection

Activate if the last will to be sent even when the connection is terminated.

Telegram Bot-Plug-in

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 “[Properties of Telegram Bots](#)”).
4. Confirm with “OK”.

You can now set up telegram bots.

Properties of Telegram Bot Connections

You set the following properties in the configuration of the Telegram Bot plug-in:

<i>Name</i>	Assign an free unique name for your connection.
-------------	---

Abschnitt: Plug in

<i>Bot access token</i>	The Access Token is provided by the “BotFather”: 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 user name. This must always end with <code>_bot</code>. e. g. <code>Temperature_bot</code> 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”.
<i>Maximum file size</i>	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.

Abschnitt: BotFather

<i>Help text</i>	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.

<i>Authentication method</i>	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).
--	--

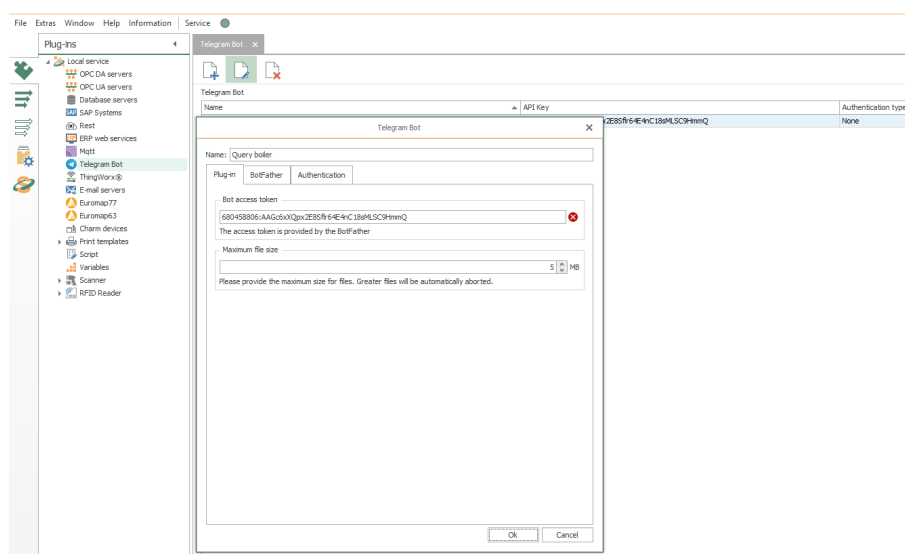
Abschnitt: Bekannte Benutzer für Authentifizierungsmethode Intern

<i>Known users</i>	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.
--------------------	---

Abschnitt Bestätigungsschlüssel für Authentifizierungsmethode Intern

<i>Registration key</i>	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.
-------------------------	--

Confirm your entries with “OK”.



Folgende Edit Bots sind hilfreich:

- /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

Plug-in Thingworx®

Only active "Things" are displayed in the OPC Router Tree. Inactive things, on the other hand, are not browser-enabled and would cause internal server errors.

The data types INFOTABLE and LONG are not supported by the plug-in.

This operation is standard:

1. Open the plug-in by double-clicking on “Thingworx®” in the “Plug-ins” menu.
2. Choose “Add” or double-click an existing entry.
3. Make the necessary settings (see section “[Properties of Thingworx® connections](#)”).
4. Confirm with “OK”.

You can now make connections to your Thingworx® system.

Properties of ThingWorx® connections

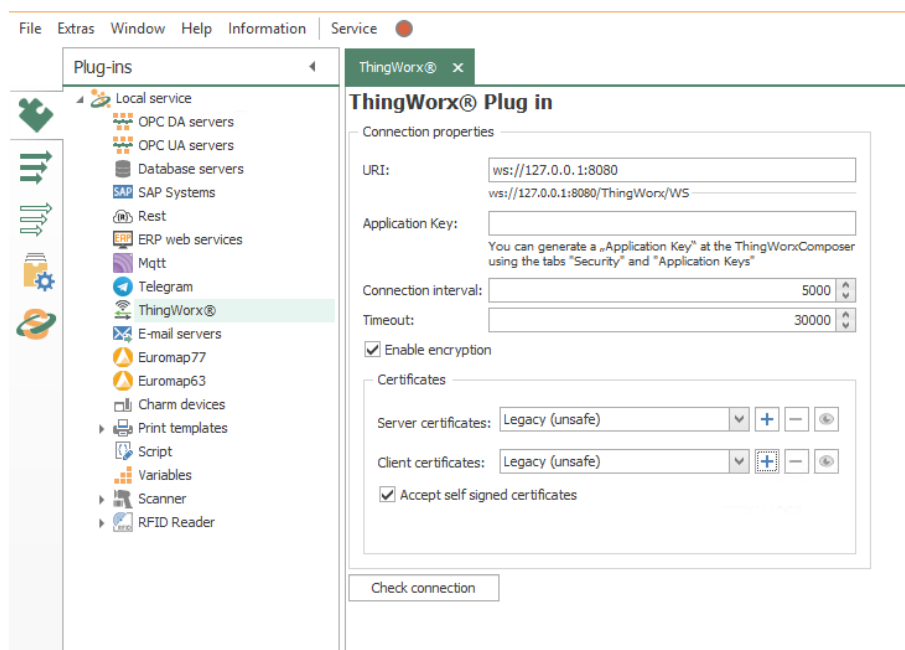
You set the following properties in the configuration of the ThingWorx® plug-in:

<i>URI</i>	Enter the URI for recognizing the ThingWorx® resource, e.g. ws://127.0.0.1:8080/ThingWorx/WS.
<i>Application Key</i>	An “Application Key” can be created via the ThingWorxComposer at the tabs “Security” and “Application Keys”.
<i>Connection interval</i>	Enter an interval in which the connection is to take place.
<i>Timeout</i>	Time that the connection setup may take before a connection error is returned.
<i>Enable encryption</i>	Activate if the connection is to be encrypted.

Certificates

<i>Server certificates</i>	Select a server certificate from the drop-down menu. Only connections to servers with this certificate are established. 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.
<i>Client certificates</i>	Select a client certificate from the drop-down menu, which the OPC Router will send as well. 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.
<i>Accept self signed certificates</i>	Activate the checkbox if self-signed certificates should be allowed.

Click the “Check connection” button to check if your connection is successful.



Mail server plug-in

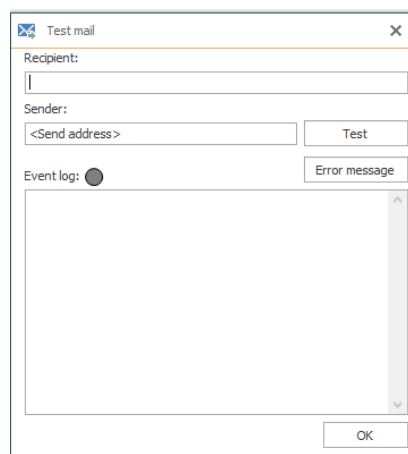
The e-mail plug-in connects the OPC router to a random number of accessible mail server. You just have to state the login information for the mail (outbox) server. Details to mailing are projected later in the connection.

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 authentication, SSL is optional.

The regular handling:

1. Open the plug-in via doubleclicking on “E-mail servers” 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 “[Properties of the Mailserver Link](#)”).
4. With “Sen test mail” you can send a mail, to test the settings. Another window pops up. Fill in an email address and click “Test”. The process log is shown in the dialouge, if sending fails, you can gain further information from the “Error Message”. Confirm with “OK”.



5. Confirm with “OK”.

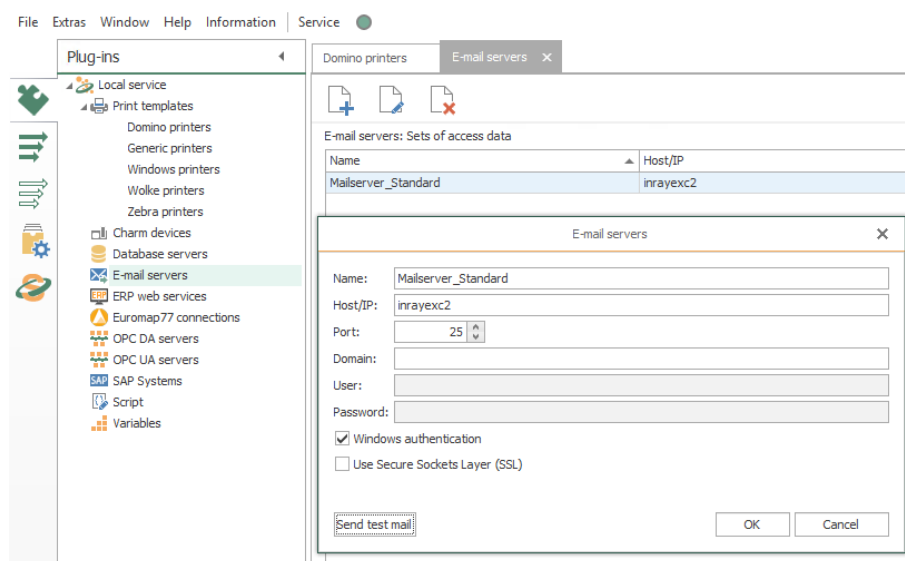
The Mail server is ready to be used in connection projection.

Properties of Mail-Server-Links

The following properties are to be set in the configuration of the E-Mail-Plug-in:

<i>Name</i>	Random name, under which this mail-server is shown in the connection projection.
<i>Host/IP</i>	IP-address or hostname of the mail-server
<i>Port</i>	Port on the server, which is used for the mail traffic (standard: 25, for SSL: 465).

<i>Domain</i>	Domain, on which the <i>User</i> is registered (for logging in on a mailserver with NT-authentication).
<i>User</i>	Username for logging in to the mail server (SMTP- or NT-authentication).
<i>Password</i>	Password for logging in to the mail server (SMTP- or NT-authentication).
<i>Windows authentication</i>	Uses alternatively the user, under which the router service runs (Standard: "System").
<i>Use Secure Sockets Layer (SSL)</i>	Uses SSL encoding. The mail server needs to support this, the needed <i>Port</i> needs to be set up.



With the button "Send test mail" you can check your entries.

Euromap-77-plug-in

The Euromap 77 is a standard for injection molding machines based on the OPC standard. Injection molding machines that support this standard can be connected with the OPC router.

Creating the Euromap connection

1. Open the plug-in "Euromap 77 connections" by double click at the plug-in menu.
2. Please select "Create new profile" or double click on an available entry to change the connection.
3. Please define the settings now (see section "[Properties of Euromap connections](#)"). Select an available command or create a new command set with the buttons. If you change a command set, the changes apply to all existing connections using this set.

4. Confirm with “OK”.

The Euromap connection will be available for projection of connections now.

Properties of Euromap-77 connections

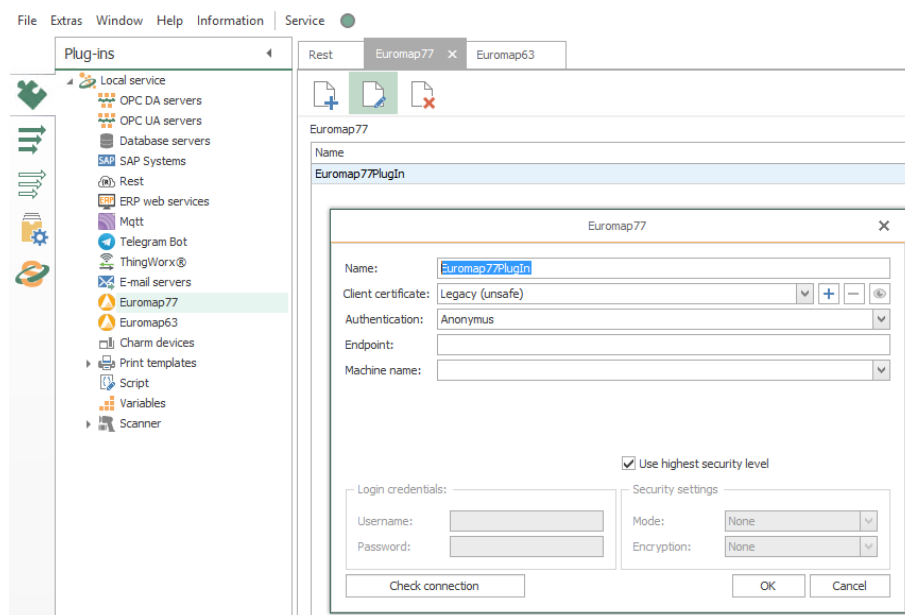
Please define the following properties to configure set of Euromap-77 connections:

<i>Name</i>	Any name where this connection appears in the connection configuration. Choose a name that is clearly recognizable for you!
<i>Client certificate</i>	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.
<i>Authentication</i>	Select authentication from the identity verification drop-down menu.
<i>Use highest security level</i>	Activate the checkbox if you want to use the highest safety level and make the entries accordingly
<i>Login credentials</i>	If you have selected “Username and password” at authentication, entries are required in this area.
<i>Security settings</i>	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

<i>Application name</i>	Assign an application name for the certificate.
<i>Organization</i>	Enter an organization as needed.
<i>Organization unit</i>	If necessary, enter an addition to the organization. For example, a place or hall of the organization.
<i>Application URI</i>	Enter the application URI.
<i>DNS name</i>	Enter the DNS name of your server.
<i>Applicant</i>	Enter the applicant.
<i>Runtime</i>	Enter the duration in minutes.

Confirm your entries with “Create”. You can now select the client certificate in the profile properties.



Euromap 63 Plug-in

Euromap 63 is a standard for injection molding machines. Injection molding machines that support this standard can be connected in the OPC router.

Creating the Euromap connection

1. Open the plug-in by double-clicking on “Euromap 63” at the Plug-ins menu.
2. Choose the “Add new instance” button or double-click on an existing entry to change a connection.
3. Make the necessary settings (see section “[Properties of Euromap63 connections](#)”). Select an existing entry for editing or create a new one with the buttons. If you change an entry, this change applies OPC router-wide.
4. Confirm with “OK”

The Euromap connection is now available for project planning a connection.

Properties of Euromap 63 connections

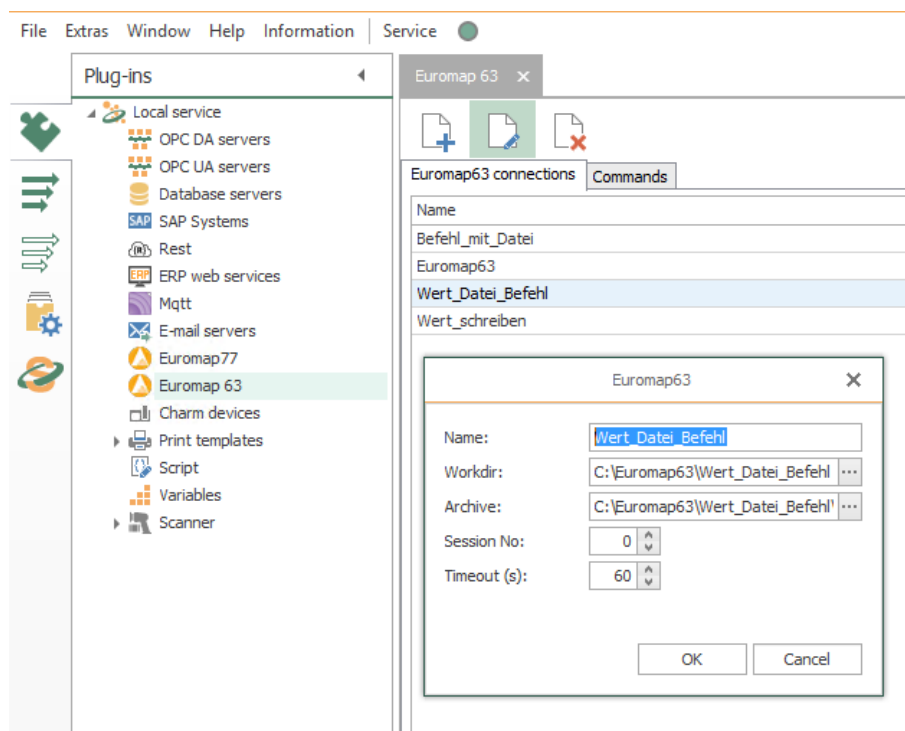
You can set the following profile properties in the configuration of the Euromap 63 connection:

<i>Name</i>	Any name where this connection appears in the connection project planning. Choose a name that you can recognize clearly!
-------------	--

<i>Workdir</i>	Select the path of your Euromap63 connection from the file explorer. The working path is the path in which the OPC router finds or stores the files that are exchanged during a transfer. This is the file path in which the OPC Router and the Euromap machine exchange their data.
<i>Archive</i>	Select the archive area for your Euromapp63 connection from the file explorer. The completed queries are stored in the archive.
<i>Session No</i>	Assign a session number.
<i>Timeout (s)</i>	Time (in s) that the connection setup may take before a connection error is returned.

Confirm your entries with “OK”.

Note: The request file is deleted by the machine software after processing.



Use the tab “Commands” to define the procedure for your Euromap63 connection. Enter a unique name here as well.

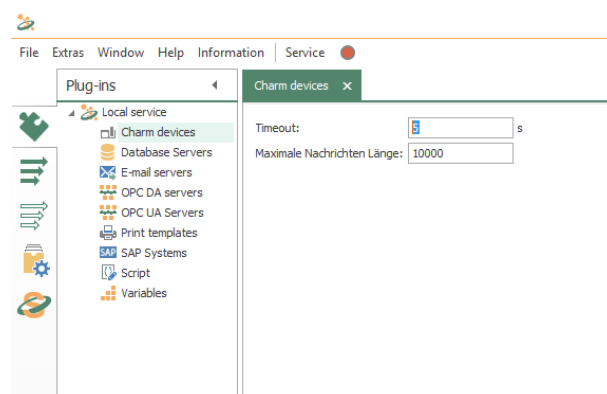
Charm Plug-in

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.



<i>Timeout</i>	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.
<i>Maximal message length</i>	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.

Properties of Charm access data

The following properties are set in the Charm plug-in configuration for each set of Charm access data

<i>Name</i>	This set's name. The name will be shown in projection. Select a name you will recognize!
-------------	--

<i>Port</i>	Port on this computer (hosting OPC Router) to which the specific Charm devices will send data.
-------------	--

Windows Printer Plug-In

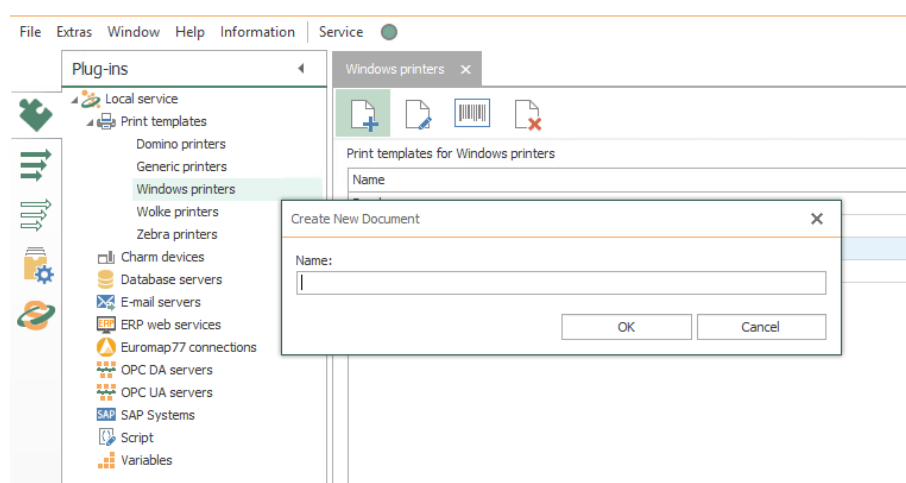
With the printer plug-in you can compile document templates within the OPC router, which are filled with values during run time and which can be printed with a printer installed on the local computer or in network. The following steps are to be done: ,

- 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 clicking 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 “[The designer for document templates](#)”).
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.



Interactive Printers

The OPC router runs as Windows application non stop in the background of the server. An interaction with the user is not planned. That means, that printers, which need user

feed (e. g. PDF output, XPS output with “Save As... -Dialogue” 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.

The document template designer

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”, “Frame” and “Barcode”. Elements from the type “Text” and “Barcode” can be seized with template variables.

Define document size

You can define the document size either independently, or you choose from the printers selection. The second option is especially usefull, 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.
 - Frame: 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.
 - Image: Image data in free position and size, to be chosen from the data system.
1. Choose the button “New Element” and the type in the opening list.
 2. 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.
 3. 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.
 4. Define the other properties.

Tip: With “Fill In 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 “Save”, to save the template.
- With “Print” you’ll start a test print on an installed printer.

- 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 already 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).

Printer Plug-Ins

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 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

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

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".

The screenshot shows the 'Edit print template' dialog box. It has a title bar 'Edit print template' with a close button. Inside, there are two input fields: 'Name' with the value 'DominoPrinterPluginTest' and 'Comment' with the value 'Test plugin'. Below these is a table titled 'Fields'.

Name	Default value
field1	default1
field2	default2
field3	default3
*	

Below the table is a 'Label' section with a text area containing the template: `|Test-Plugin des DominoPrinters. Wert 1: {field1} Wert 2: {field2} Wert 3: {field3}`. At the bottom, there are buttons for 'Insert field', 'Insert control character', 'OK', and 'Cancel'.

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.

Script Plug-in

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.

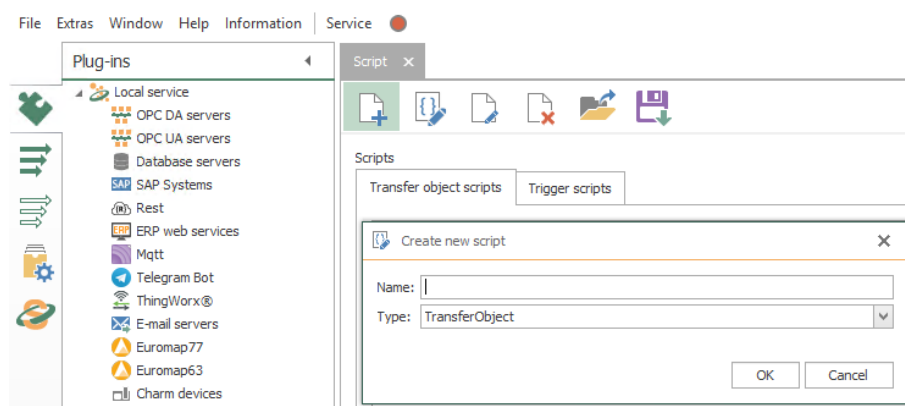
At code-editor

- 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.



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. 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:

Methoden im Transferobjekt

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:

Eigenschaften von Parametern

<i>Name</i>	Name of parameter
<i>Array</i>	Yes No Determines if the parameter is an array. You can use an array as output parameter for B. as a source for an SQL insert.
<i>Type of parameter</i>	Data type
<i>Direction of parameter</i>	• <i>Input</i>: Input parameters. In connection configuration, this parameter can serve as the destination of a data transfer. • <i>Output</i>: Output parameters. In connection configuration, this parameter can serve as the source of a data transfer. • <i>Both</i>: The parameter is both input and output parameters. • <i>None</i>: Variable that is not visible to the connection designer and is used only within the script.

Trigger Script

When you create a new script of type “Trigger”, there are three predefined methods available:

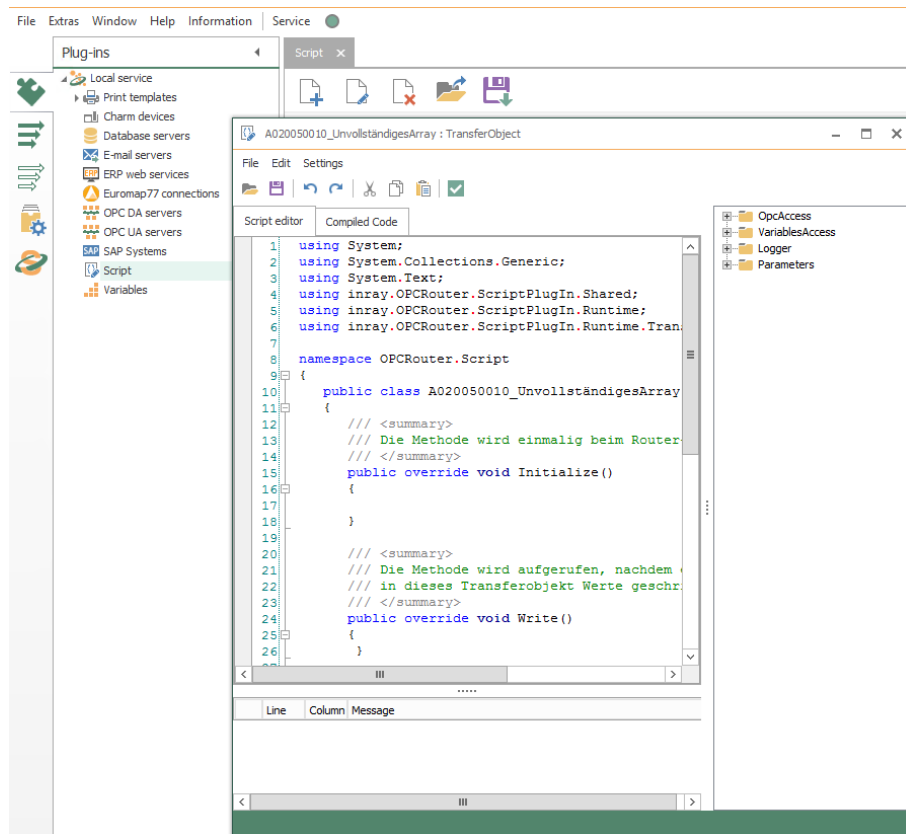
Methoden im Transferobjekt

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 <code>true</code> or <code>false</code> . At <code>true</code> the data transfer is triggered.
ResetTrigger	The method is called after successful data transfer. The parameter is automatically transferred (<code>bool transferError</code>) (<code>true</code> = 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.).

The 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.
- Menu "Settings – Assembly references": Enter here used assemblies.
- Menu "Settings – Edit parameter": Here you create required parameters and variables (s. [Transferobjekte](#) 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>)
```

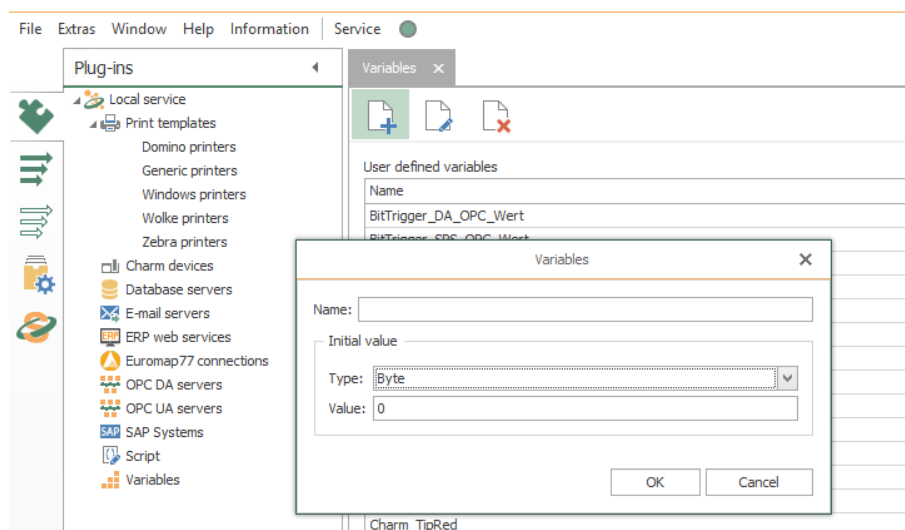
- Return: none
- *strMessage*: message to be written
- *LogLevel*: to be written log level. If the corresponding log level is deactivated, it will not be recorded either. `MessageType.All` logs in any case, regardless of the router configuration.

Variable 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.



The operation is standard:

1. Open the plug-in by double-clicking on “Variables” at the plug-in menu
2. Select “Add new variable” or double-click an existing entry.
3. Make the necessary settings in the dialog:

<i>Name</i>	Name of the variable, appears in the configuration. The variable name must be unique.
<i>Type</i>	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. Typeless is also possible, the variable can then be filled arbitrarily.
<i>Value</i>	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.

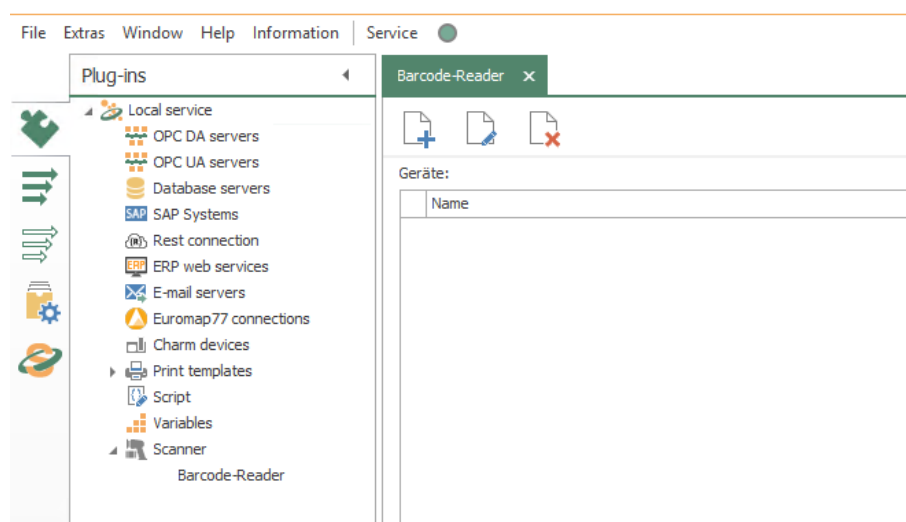
4. Confirm with “OK”.

The variable is now available for configuring a connection.

Scanner-Plug-in

The following scanner devices can currently be connected:

- Barcode-Reader



Properties of scanner connections

You can set the following properties in the configuration of the scanner plug-in for a barcode reader:

<i>Name</i>	Any name where this device appears in the connection configuration.
<i>COM Port</i>	Enter the COM port for communication with the barcode reader.

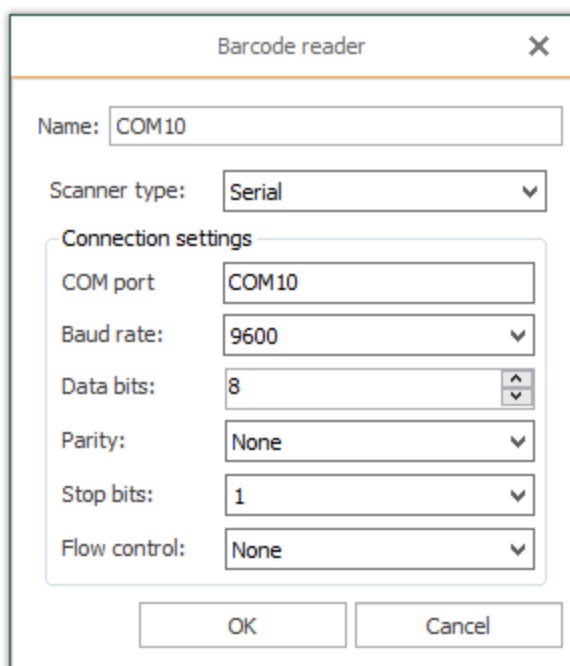
Connection settings serial Barcode reader

<i>Baud rate</i>	Select a baud rate. Specifies how often a signal in a communication channel changes state.
<i>Data bits</i>	Specify how many bits of data to transfer.
<i>Parity</i>	Select a parity from the drop-down menu. For example, Even, Mark, Space.
<i>Stop bits</i>	Specify how many stop bits are needed.
<i>Flow control</i>	Select how the data flow should be controlled. Eg. software, hardware

Connection settings Tcp Barcode reader

<i>IP</i>	Enter the IP of your Tcp barcode scanner.
-----------	---

<i>Port</i>	Enter the port of your Tcp barcode scanner.
-------------	---



The image shows a 'Barcode reader' configuration dialog box. It has a title bar with a close button (X). Inside, there is a 'Name' field with 'COM10' entered. Below it is a 'Scanner type' dropdown menu set to 'Serial'. A 'Connection settings' section is highlighted with a light blue border and contains several fields: 'COM port' (COM10), 'Baud rate' (9600), 'Data bits' (8), 'Parity' (None), 'Stop bits' (1), and 'Flow control' (None). At the bottom are 'OK' and 'Cancel' buttons.

Plug-in RFID Reader

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 "[Properties of RFID Reader connections](#)").
4. Confirm with "OK".

You can now make connections with your RFID reader.

Properties of RFID Reader connections

The following properties are set in the configuration of the RFID Reader plug-in:

<i>Name</i>	Any name where this connection appears in the connection project planning. Choose a name that you can recognize uniquely!
<i>Client certificate</i>	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.

<i>Authentication process</i>	Select an authentication from the Identity Verification drop-down menu. If you select Username, you must enter Username and Password at Registration information.
<i>Endpoint</i>	Enter an endpoint for your connection <code>opc.tcp://</code>
<i>RFID-Reader name</i>	Select your RFID Reader from the drop-down menu.
<i>Scan mode</i>	Selection of scan mode between “Scan manually”, “Scan in regular interval” or “Scan in background”.
<i>Scan interval (ms)</i>	Enter the scan interval in milliseconds.
<i>Trigger if minimum signal strength is reached</i>	Only trigger the scan if the signal strength is sufficient.
<i>Signal strength (RSSI)</i>	Tags with a lower signal strength during scanning are ignored by the OPC router.
<i>Trigger if an RFID tag was not captured for a defined time.</i>	Specifies how long a tag must be outside the reading field before the OPC Router triggers another transfer for this tag.
<i>Hibernation interval (ms)</i>	Period where a RFID tag is not to be recorded.
<i>Highest security level</i>	Activate if the highest security level has to be used.

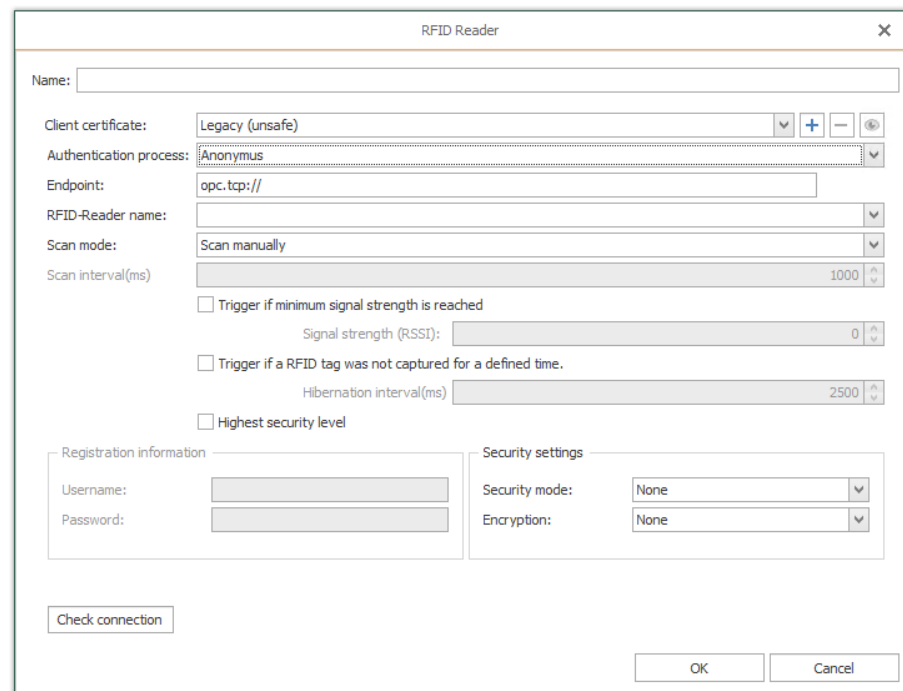
Registration information

<i>Username</i>	Enter a username to authenticate yourself.
<i>Password</i>	Enter a password to authenticate yourself.

Security settings

<i>Security mode</i>	No Security = “None”, Signing = “Sign”, Signing and Encrypting with “SignAndEncrypt”.
<i>Encryption</i>	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.



The image shows a configuration window titled "RFID Reader". It contains several fields and options for setting up the reader's connection and security.

Fields and Options:

- Name:** A text input field.
- Client certificate:** A dropdown menu showing "Legacy (unsafe)" with buttons for adding (+), removing (-), and refreshing (circular arrow).
- Authentication process:** A dropdown menu showing "Anonymus".
- Endpoint:** A text input field containing "opc.tcp://".
- RFID-Reader name:** A dropdown menu.
- Scan mode:** A dropdown menu showing "Scan manually".
- Scan interval(ms):** A text input field with "1000" and up/down arrow buttons.
- ☐ **Trigger if minimum signal strength is reached**
 - Signal strength (RSSI):** A text input field with "0" and up/down arrow buttons.
- ☐ **Trigger if a RFID tag was not captured for a defined time.**
 - Hibernation interval(ms):** A text input field with "2500" and up/down arrow buttons.
- ☐ **Highest security level**

Registration information:

- Username:** A text input field.
- Password:** A text input field.

Security settings:

- Security mode:** A dropdown menu showing "None".
- Encryption:** A dropdown menu showing "None".

Buttons:

- Check connection:** A button located below the registration information fields.
- OK:** A button located at the bottom right.
- Cancel:** A button located at the bottom right, next to the OK button.

Defining Connections (Projecting)

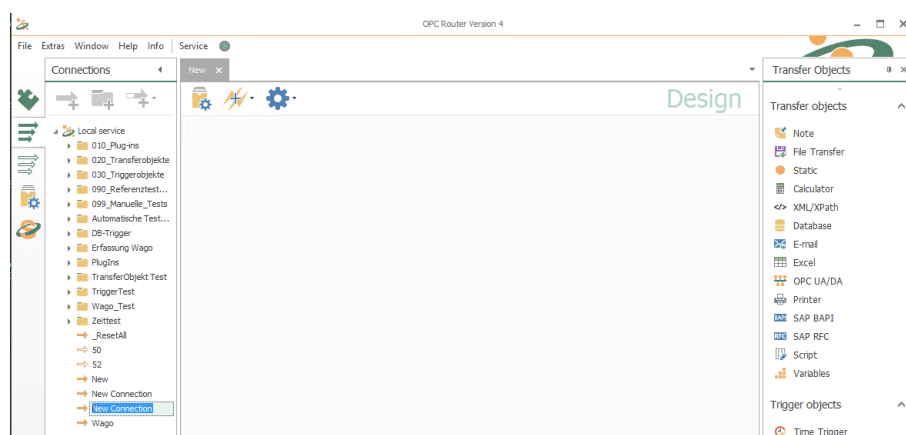
After having configured all plug-ins needed, you can start defining connections. As only one Router project can be loaded as runtime configuration, you want to project all connections within one single project file. You may, however, sort the connections into different folders (=connection group) (similar 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.

Creating and Managing 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”.



The connection has been created and opened in main window. Existing connections can be opened by double-click in the project configuration 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.

Activate/Deactivate Connections

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”.

Project Connections

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.

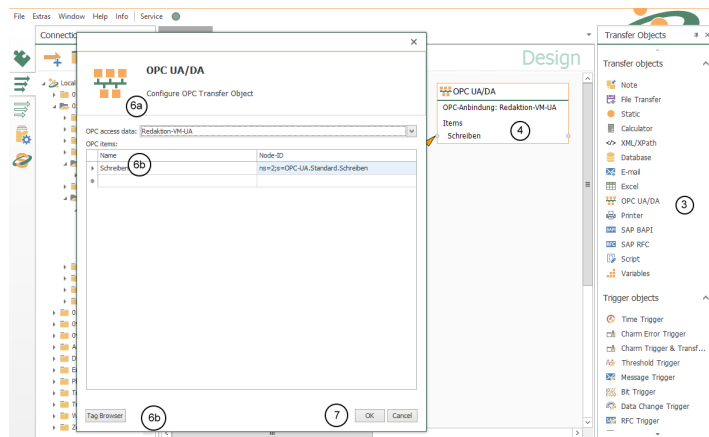
The transfer objects may be found in the right-hand toolbox.

Right mouse-click on a transfer objects with in project allows editing, copying, pasting, cutting, deleting. Right mouse-click on a connection arrow allows defining the step order.

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.
 - a. Selection of the OPC connection.
 - b. Alternatively: Enter data points’ name or open the tag browser.
 - c. Transfer data points to the transfer object by double-clicking.

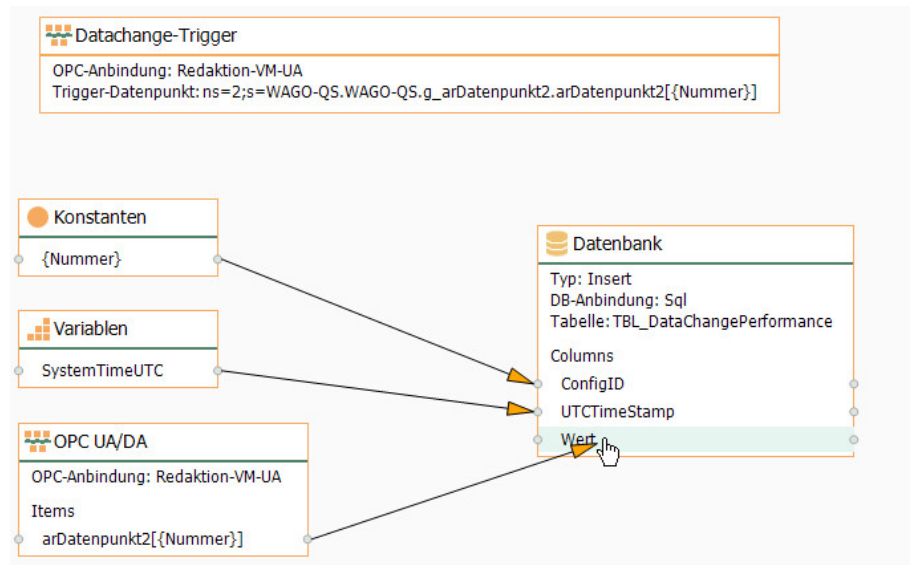
7. Confirm with “OK”.



8. Project a destination, e. g. a database transfer object.

- a. Choose the DB transfer object from toolbox “Transfer Objects”.
- b. Place it within projection area.
- c. Open transfer object with double-click.
- d. Choose a set of DB access data.
- e. Choose a type for this transfer object. If database servers as destination, that would be either INSERT or UPDATE. Choose INSERT in this case.
- f. Choose the table you want to write to.
- g. Choose the columns you want to write to. Sort with “Up” and “Down” to keep the overview in the configuration.
- h. Confirm with “OK”.

9. 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!

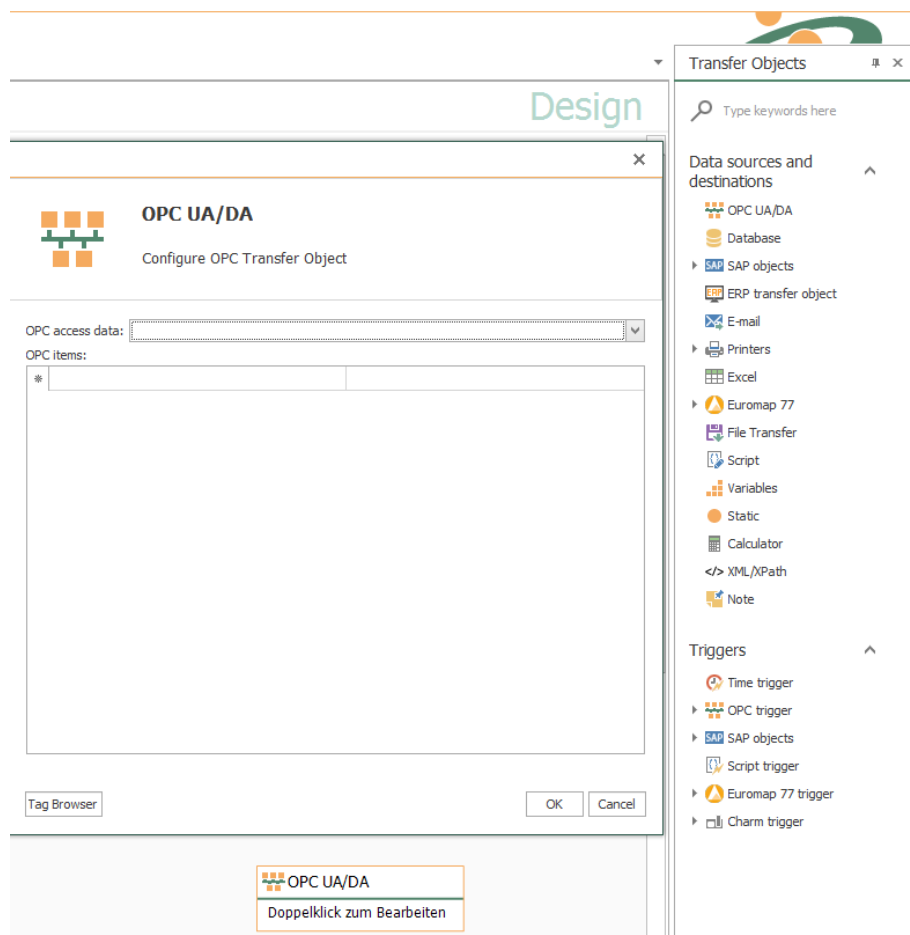


10. Configure a trigger:
 - a. Choose a trigger from the toolbox.
 - b. Place it within projection area.
 - c. Open with double-click.
 - d. Configure the trigger.

Sources and Destinations

Transfer Object “OPC” (DA/UA)

OPC servers and thus PLCs are available both as source and destination of data transfers. The OCP server’s set of access data has to be specified within plug-in configuration. Create a OPC transfer object as [described above](#)..



Double-click on transfer object to assign the properties:

<i>OPC access data</i>	Select a set of OPC access data. Dependent on plug-in configuration, OPC Router will recognize DA/UA connections.
------------------------	---

	CAUTION: If you are using FAS-inMOVE and if you are configuring OPC Router manually as well, do not select an OPC set of access data with prefix [FiMx] when projecting manually. These access data belong to FAS-inMOVE and might be changed or deleted from there. Create a new set of access data within plug-in configuration instead.
<i>OPC items</i>	Enter OPC items needed. Use the Tag Browser to browse items. Note: IF within OPC access data <i>Define types manually</i> is activated, use the drop-down to specify the data type that is expected.
<i>Node ID</i>	Enter Node ID of UA items needed. Use the Tag Browser to browse items.
<i>Quality string</i>	OPC quality as string will be available as transfer element
<i>Quality ID</i>	OPC quality as integer will be available as transfer element.

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 DA: 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

Using the button "Tag Browser", you will open another dialog to browse and select OPC items. The dialog is not modal, you can switch between transfer object and browser.

Enter a search string (OPC DA only) 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.

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
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

Transfer Object “Database”

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](#).

Configure the object's properties by double click:

<i>DB access data</i>	Select one of the configured sets of access data
<i>Type</i>	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. Entries 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 *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)

<i>Table (View, Stored Procedure)</i>	Table, View or Stored Procedure this object refers to. Note: Table has to have a primary key. Tables without primary key are not supported.
<i>Available columns/Columns used</i>	(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)

<i>Filter</i>	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.
<i>Edit filter/filter group</i>	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
<i>Preview</i>	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)

<i>Sorting by column</i>	Column with sorting criteria.
<i>Sorting order</i>	Ascending/Descending
<i>Do not change records</i>	Transferred records will not be changed.
<i>Mark transferred records</i>	(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.
<i>Column to mark</i>	(only with select from table)

	Column to mark transferred rows. Default value set by database should be "0"
<i>Delete transferred records</i>	only with select from table) Transferred rows will be deleted. CAUTION: Rows in database table will be deleted without recover possibility.
<i>Cancel transfer (with empty result)</i>	If the result set is empty, the transfer is canceled.
<i>Log error (with empty result)</i>	If the result set is empty, the transfer is canceled and an error is logged in log file and status monitor.
<i>Limit number of records</i>	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 <i>sorting column</i> should be e. g. a "UtcTimeStamp" with <i>Sorting order</i>"Descending", the <i>number of records</i> limited to "1".
<i>Transfer only with data change</i>	If set, the result set is read with OPC Router start and transferred only after having changed (plus trigger conditions met)

Note: If *Type* is set to "Update", the transfer object will provide the number of rows updated as source.

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 z. B. 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 after transfer flag = 0:

```
CREATE NONCLUSTERED INDEX [IX_TransferTable_Transferflag] ON
[dbo].[TransferTable]
```

```
(
  [Transferflag] ASC
)
WHERE ([Transferflag]=(0))
GO
```

- If the OPC router should filter after 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.

Transfer Object “SAP-RFC”

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:

Tab General

<i>SAP access data</i>	Select SAP set of access data.
<i>Function Group</i>	Select one of the function groups available.
<i>Function Group/Search</i>	Enter a search string (*=place holder) and click “Filter” to ...
<i>Function Group/Selection</i>	... show only those groups matching you search term. Select here!
<i>Function</i>	Select one of the functions available in your <i>Function Group</i> .
<i>Function/Search</i>	Enter a search string (*=place holder) and click “Filter” to ...
<i>Function Group/Selection</i>	... show only those functions matching you search term. Select here!

Registerkarte: Parameter

<i>Parameters available</i>	Select the required parameters by setting the checkmarks accordingly. You can also take whole categories (see below).
-----------------------------	---

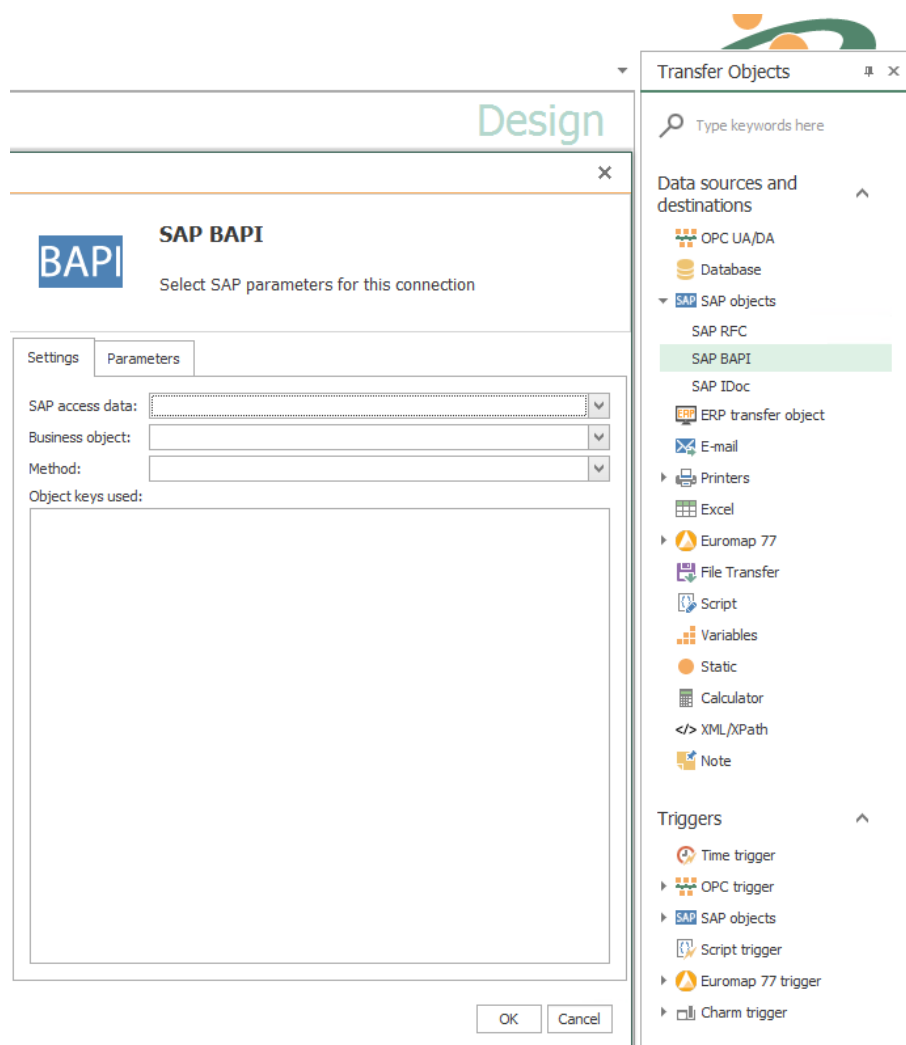
Parameters are categorized:

- 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.

Transfer Object “SAP-BAPI”

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

<i>SAP access data</i>	Select SAP set of access data.
<i>Business Object</i>	Select Business Object.
<i>Method</i>	Select one of Business Object's methods.

<i>Object keys used</i>	Object keys define an object positively and are added automatically. Required if method depends on instance.
-------------------------	--

Registerkarte: Parameters

<i>Please specify a function</i>	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.
----------------------------------	--

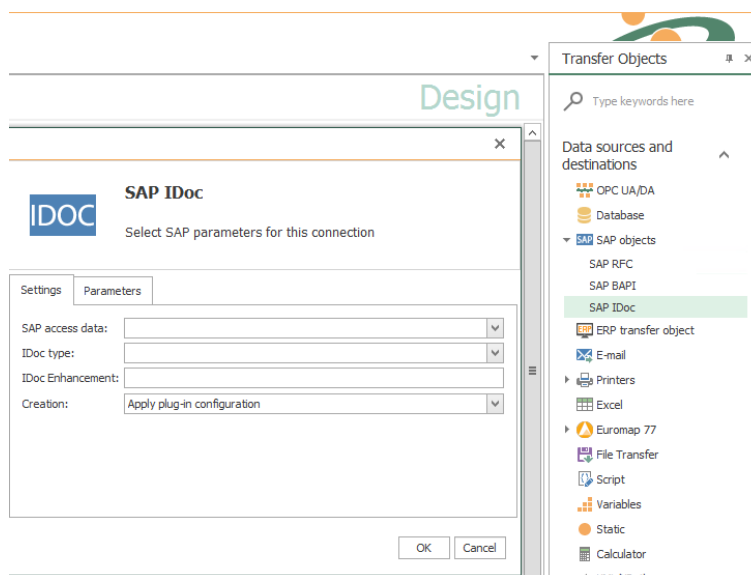
Parameters are categorized:

- 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.

Transfer Object "SAP-IDoc"

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

<i>SAP access data</i>	Select SAP set of access data.
<i>IDoc Type</i>	Select an IDoc type provided by your SAP system
<i>IDoc Enhancement</i>	Class enhancement of IDoc.
<i>Creation</i>	• <i>Apply plug-in configuration</i>: It selects what was selected in the SAP-plug-in by IDoc handling. • <i>Create empty IDoc</i>: When an IDoc is loaded from SAP, the IDoc is sent, but the existing elements are emptied and only their own changes are sent. • <i>Use template from SAP system</i>: The loaded IDoc from SAP is used but edited with its own values.

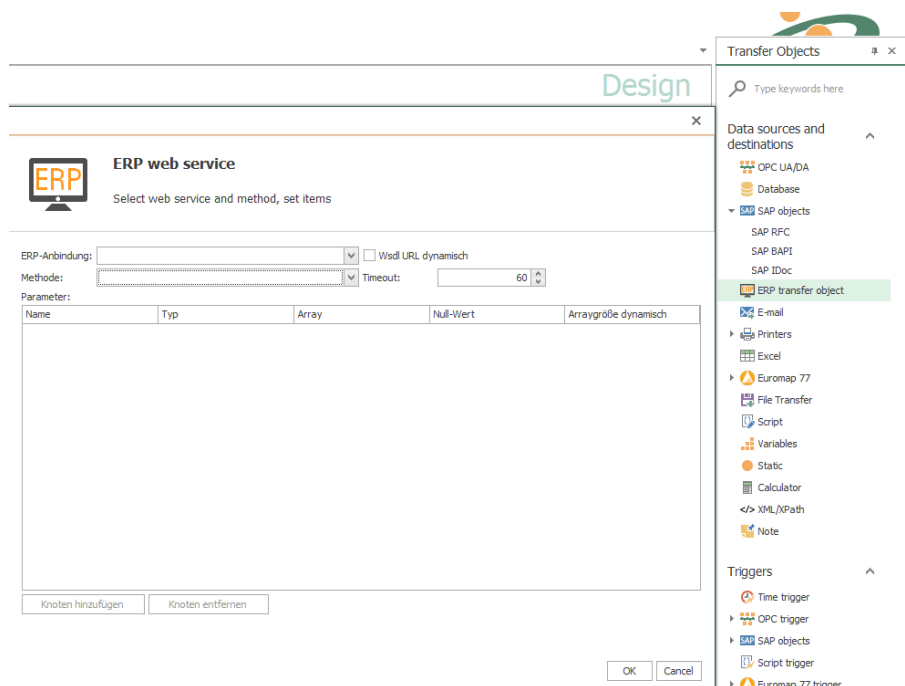
Registerkarte: Parameters

	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.

Transfer Object “ERP Webservice”

Webservices are both available as destination (input parameters) or source (output parameters). Create an ERP object as [described above](#).



Double-click on transfer object to set the following properties:

<i>ERP access data</i>	Select one of the configured web service sets of access data.
<i>Method</i>	Select the method to be called.
<i>WSDL URL dynamical</i>	Enable to pass the URL dynamically at runtime. However, the method and the parameters must be identical.
<i>Timeout</i>	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.
 - a. Leave property *Size of array dynamic* deactivated.
 - b. Select the array.
 - c. Click on “Add node”. These nodes appear within the array.
 - d. Add as many nodes as desired.

Remove a node by selecting it and click on “Remove node”.

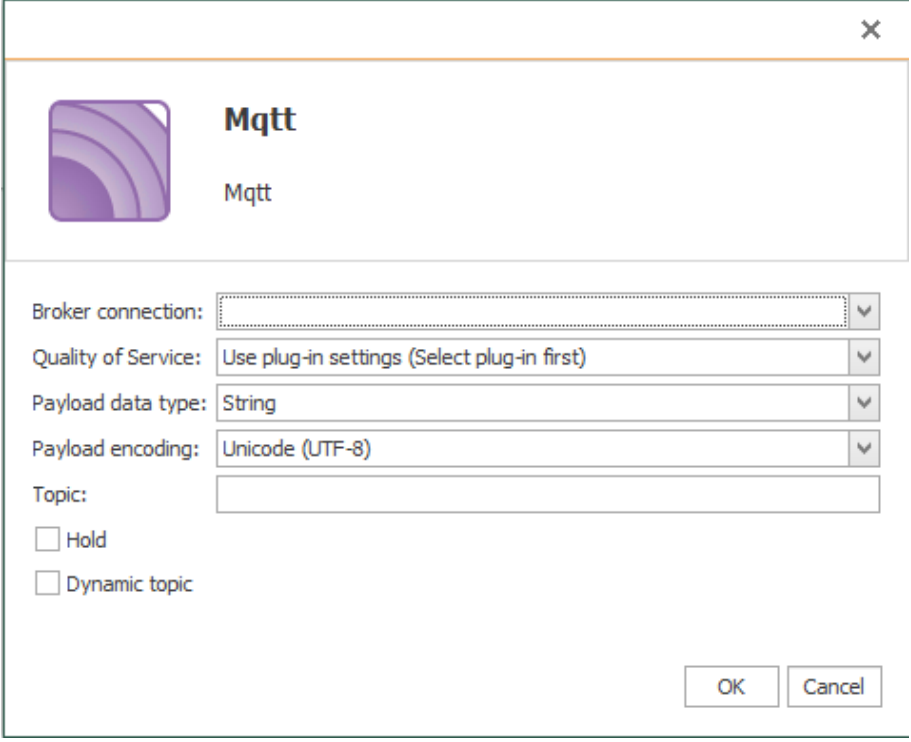
Nested classes with parameters do not support dynamic parameters. Do you still want to use nested classes so must be defined before how many elements the array has since all nodes have to be created by themselves.

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.

Transfer Object “Mqtt”

To enable a connection [proceed as follows](#).



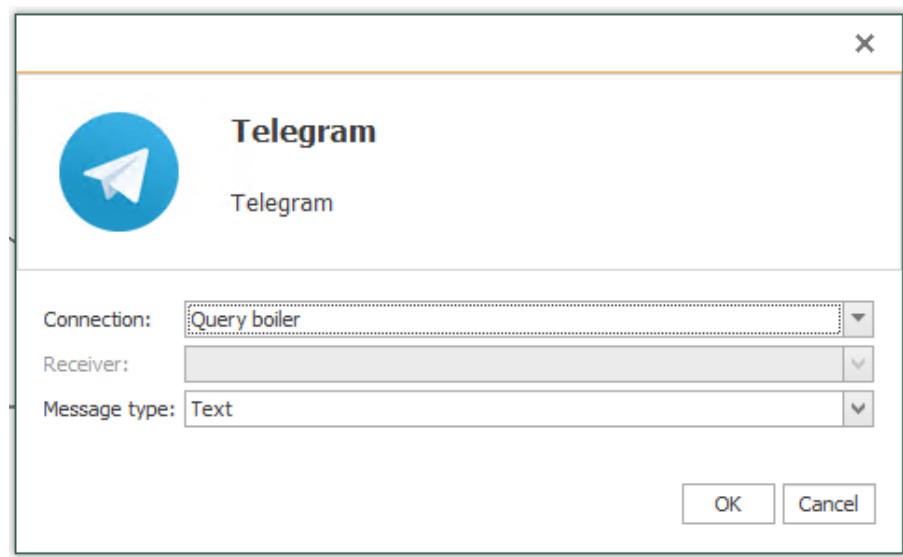
After double-clicking the transfer object, set the following properties:

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

Confirm your entries with “OK”.

Transfer Object “Telegram”

Create a telegram object within the connection [as described](#).



The screenshot shows a dialog box titled "Telegram" with a blue Telegram logo icon. Below the title bar, there are three configuration fields, each with a label and a drop-down menu:

- Connection:** The drop-down menu is open, showing "Query boiler" as the selected option.
- Receiver:** The drop-down menu is currently empty.
- Message type:** The drop-down menu shows "Text" as the selected option.

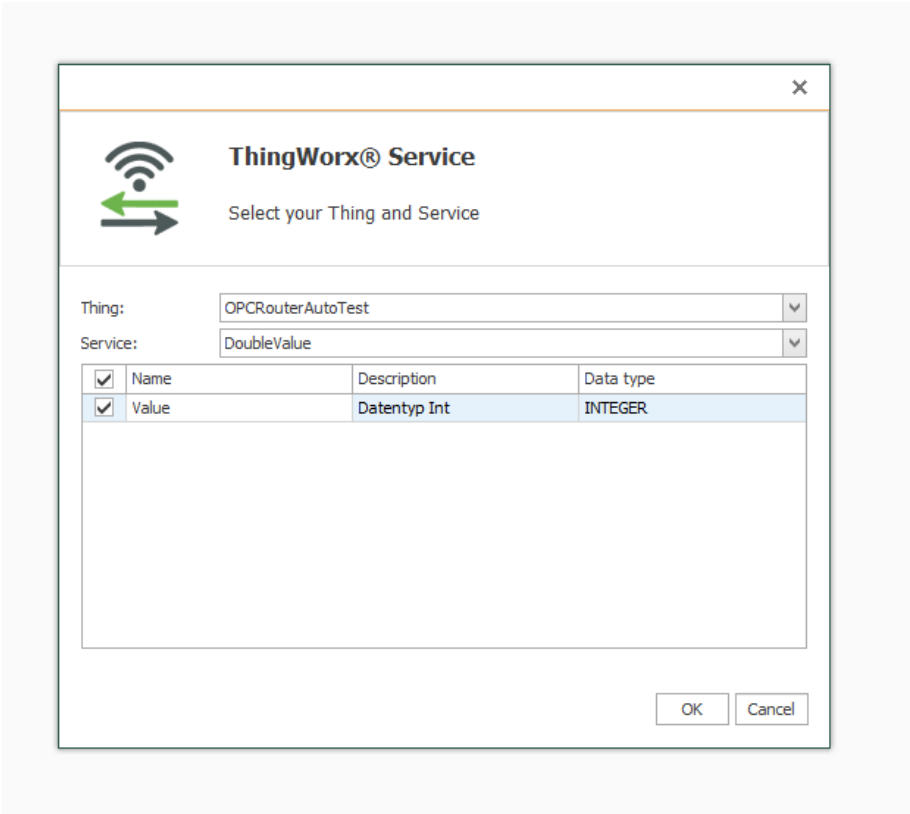
At the bottom right of the dialog box, there are two buttons: "OK" and "Cancel".

Set the following properties after double-clicking the object:

<i>Connection</i>	Select your telegram connection from the drop-down menu.
<i>Receiver</i>	Select the receiver for receiving the telegram from the drop-down menu.
<i>Message type</i>	Select the message type with which the message of the telegram is to be transmitted, e.g. text message or voice message.

Transfer Object “ThingWorx®”

Create a ThingWorx® transfer object within the connection [as described](#). You can create a ThingWorx® Service Transfer Object or a ThingWorx® Property Transfer Object. You can use the Service Transfer Object to have a service run by a Thing.

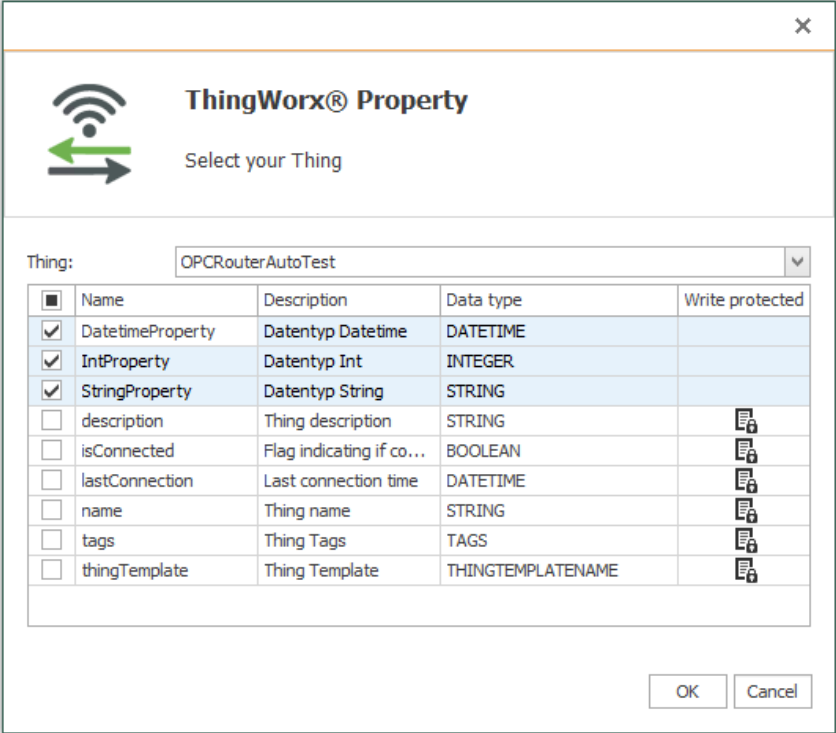


After double-clicking on the transfer object, set the following properties and confirm your entries with “OK”:

Thing	Select your Thing from the drop-down menu.
Service	Select the ThingWorx® service from the drop-down menu.

With the property transfer object properties of a thing can be addressed. For example, values can be written into a Thing. Select your Thing and select the required properties.

Only properties that are not write-protected can be selected. Confirm your selection with “OK”.



The dialog box is titled "ThingWorx® Property" and has a close button (X) in the top right corner. It features a logo on the left consisting of a green Wi-Fi symbol and two green arrows pointing in opposite directions. The text "Select your Thing" is displayed below the logo. A dropdown menu labeled "Thing:" shows "OPCRouterAutoTest". Below this is a table with five columns: "Name", "Description", "Data type", and "Write protected". The table lists several properties, with the first three checked. The "Write protected" column contains lock icons for the last seven properties. At the bottom right are "OK" and "Cancel" buttons.

☐	Name	Description	Data type	Write protected
☒	DatetimeProperty	Datentyp Datetime	DATETIME	
☒	IntProperty	Datentyp Int	INTEGER	
☒	StringProperty	Datentyp String	STRING	
☐	description	Thing description	STRING	
☐	isConnected	Flag indicating if co...	BOOLEAN	
☐	lastConnection	Last connection time	DATETIME	
☐	name	Thing name	STRING	
☐	tags	Thing Tags	TAGS	
☐	thingTemplate	Thing Template	THINGTEMPLATENAME	

Transfer Object “E-Mail”

An e-mail object is a possible transfer destination. The needed e-mail (SMTP) server has to be configured as plug in instance. Create a mail transfer object [as explained](#) within the connection.

Set up the following properties after double clicking the transfer object:

<i>E-mail set of access data</i>	Here the SMTP server are available as configured instances of the OPC plug in.
<i>Number of recipients</i>	Enter the number of receivers here. The transfer object will create that number of elements, which are filled from you during run time with e mail addresses e. g. from variables or a

	data base table. With a dynamic data base filter you can figure the receivers flexible.
<i>Sender</i>	Adresser of the e-mail, has to be in format of an e mail address (name@domain.xx).
<i>Subject</i>	Subject line in the e mail. You can also create placeholders for a dynamic subject line in the subject. Create a new placeholder and then add [%PlaceholderName%] to the subject of the email.
<i>Number of attachments</i>	Provides a corresponding number of file path / name elements in the transfer object, which you can either dynamically fill via a constant object or at runtime. This file(s) will then be sent as attachments
<i>Text as HTML</i>	Gives out the mail in HTML format.
<i>Send mails individually</i>	An e-mail is sent to each recipient individually, so that the individual recipient cannot see to whom the e-mail was still sent. Even if an e-mail address does not work, all other recipients can still receive their message.
<i>Content</i>	Text in the e-mail. Fill in <i>Placeholders</i> via drag & drop, which are filled from you during runtime.
<i>Placeholder</i>	Define any variables, which are filled from you during runtime, e. g. with process values or messages. Create a new placeholder with "New". Mark a placeholder and add it to the text with "Add". You can also add via double click or drag & drop.

Transfer Object "Printer"

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](#).

Domino printer

Set the following properties after double clicking the transfer object for Domino printer:

<i>Label</i>	Document template, created in the Plug in Configuration
<i>Static printer selection</i>	- 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).
<i>IP</i>	Enter the IP address of your static printer.

<i>Port</i>	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.

Generic printer

Set the following properties after double clicking the transfer object for generic printer:

<i>Label</i>	Document template, created in the Plug in Configuration
<i>Static printer selection</i>	- 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).
<i>IP</i>	Enter the IP address of your static printer.
<i>Port</i>	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.

Windows printer

Set the following properties after double clicking the transfer object for generic printer:

<i>Document</i>	Document template, created in the Plug in Configuration
<i>Specify number of copies</i>	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.
<i>Static printer</i>	- 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).
<i>Parallel printing</i>	Activate this option if you want to print on different printers parallel.

<i>Printer</i>	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.

Wolke printer

Set the following properties after double clicking the transfer object for Wolke printer:

<i>Label</i>	Document template, created in the Plug in Configuration
<i>Static printer selection</i>	• 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).
<i>IP</i>	Enter the IP address of your static printer.
<i>Port</i>	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.

Zebra printer

Set the following properties after double clicking the transfer object for Zebra printer:

<i>Label</i>	Document template, created in the Plug in Configuration
<i>Static printer selection</i>	• 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).
<i>IP</i>	Enter the IP address of your static printer.
<i>Port</i>	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.

Transfer Object “Excel”

Excel transfer objects can be used as source and destination. You can write data in defined boxes or read the out. You can also write several data sets from the data base into several lines in the Excel table. A plug in configuration is not necessary, you can project the transfer object [as explained](#) directly in the connection.

Set the following properties after double clicking the transfer object:

Section: *File*

<i>File</i>	Dynamic Static You can define the file to be worked on at this point. Then choose the option “Static” and use the button “...”, to select the file. If the filename is to be found dynamically during runtime (e. g. from a data base, a variable or a script), choose “Dynamic” here. This way, you can connect a data source to the filename within the connection. Note in any case, that the filepath is to be processed in the Windows system. With the button “...” you gain the correct path! (e. g. C:\Documents and Settings\<User>\Own files instead of \Own files)
<i>Action</i>	Edit File Create File “Edit File” refreshes an existing file. Is the file not existing, you can either create it new or abort the transfer. The option “Throw Error” includes breaking up the transfer. “Create new File” creates a new file in any case. This one can be empty or basing on a template (XLT respectively XL - TX). For selecting the template please note the same as mentioned under <i>File</i>.

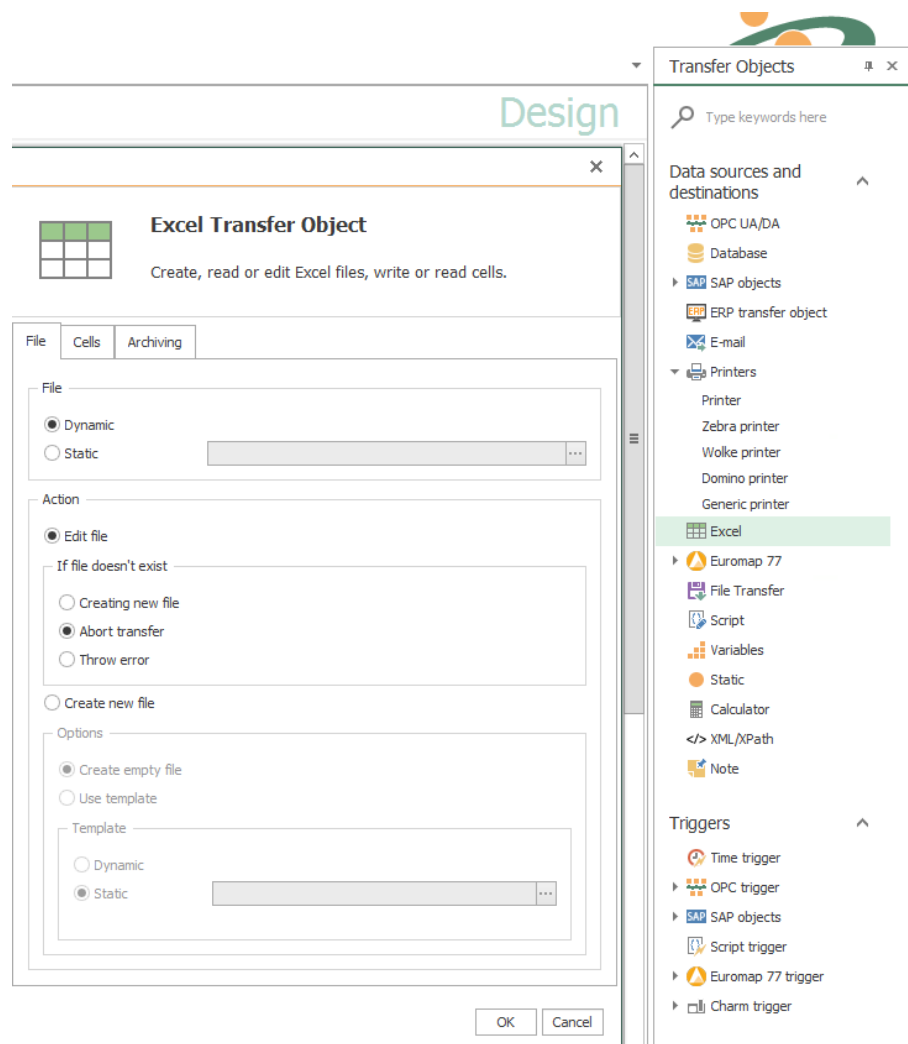
Section: *Cells*

<i>Spread Sheet</i>	Dynamic Static The name of the spread sheet can also be found either the dynamic or the static way. If no name for the table is given, the OPC Router writes the data into Table1. State “Dynamic”, to connect a data source to the tables name within the connection.
<i>Cells</i>	Here state the cells in the usual format “Column/Line”, e. g. A1. The following writing modes are available Update Cell: Overwrites the chosen cell with a new value.

	Insert Cell: Adds a new cell to the chosen position. All other cells are moved down. Append Cell: Finds the first empty cell under the chosen position and fills in the content. This option can be used to, e. g. write the result of an SQL Select (means several data sets) into the table.
--	--

Section: Archiving

Archiving	Activate, to save the Excel file in the state it has before transferring. Note: If transfers are triggered every second, an archive file is created every second, too. This option is obviously smarter used, if for example a daily statistic is needed. The archive files name includes date and time of the archiving (not: the creation of the original file).
Archive Folder	Dynamic Static The notes given under <i>File</i> are to be mentioned here, too – but here you just state one folder, no filename.



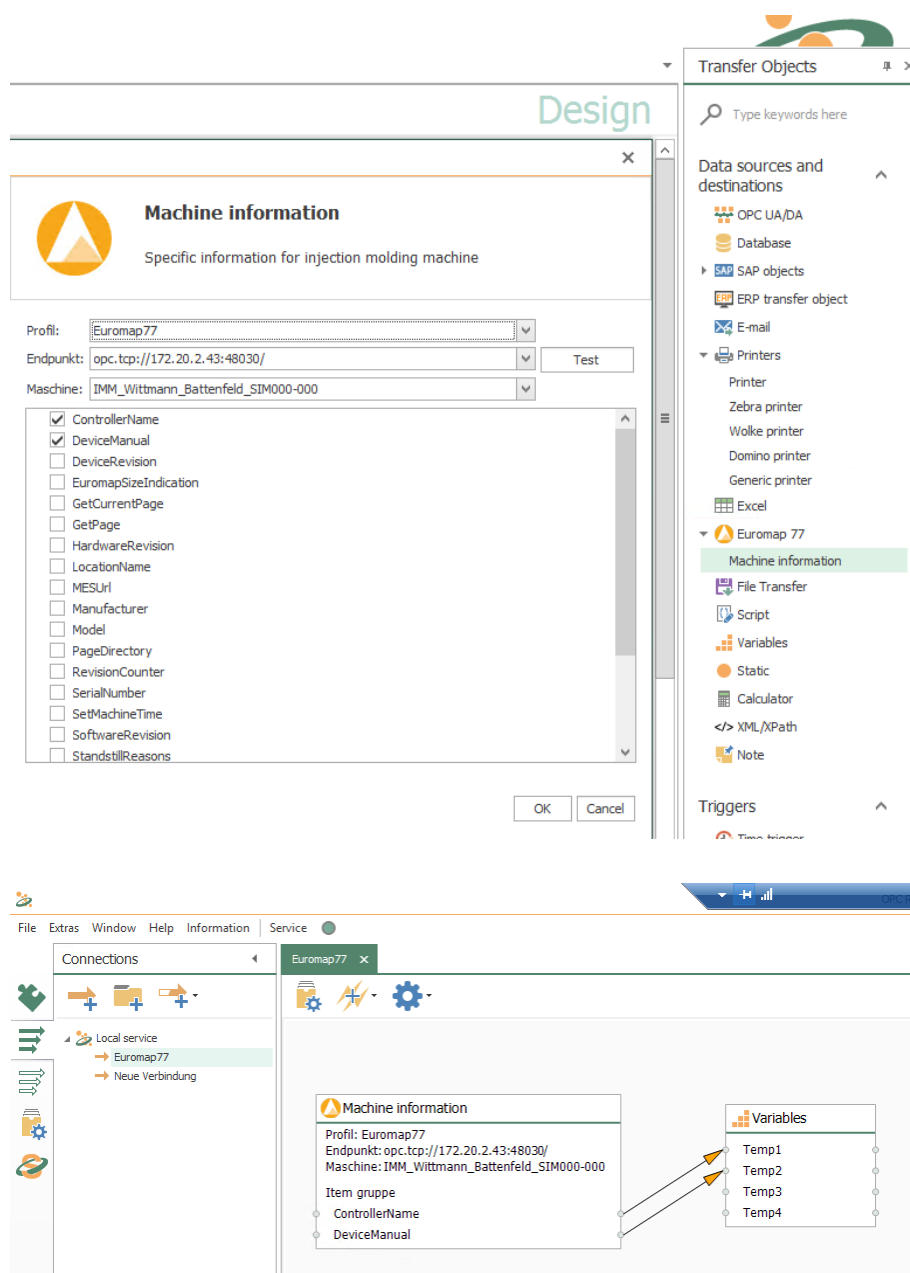
Transfer Object “Euromap 77”

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:

<i>Profil</i>	Here the profiles are available as configured instances of the Euromap77 plug-in. Select a profile from the drop-down list.
<i>Endpoint</i>	Select an endpoint from the drop-down list. With the button “Test” you can check if your connection works
<i>Machine</i>	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.



Transfer Object “Euromap 63”

An Euromap object is available as a transfer source and destination. The required profile must be configured as a plug-in instance. Create a transfer object within the connection [as described](#).

Set the following properties after double-clicking on the transfer object:

<i>Instance</i>	Select the desired Euromap63 instance from the drop-down menu.
-----------------	--

Command	Select the required command for your instance. Use the “Edit” and “Add” buttons to edit the command selection.
---------	--

Confirm your selection with“OK”.



Euromap 63

Create Euromap 63 commands

Instance:

Euromap63

Command:

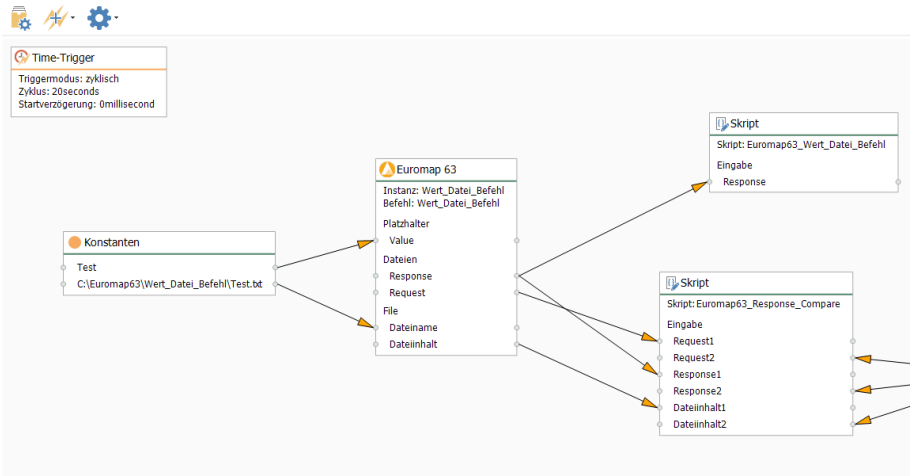
Wert_schreiben

Edit

Add

OK

Cancel



Transfer Object “File”

File 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 transfer object within the connection [as explained](#) projektieren.

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

Tab: Source

Dynamic or Static	Dynamic or static source path selection. For static source path, select the file name to read the file content.
-------------------	---

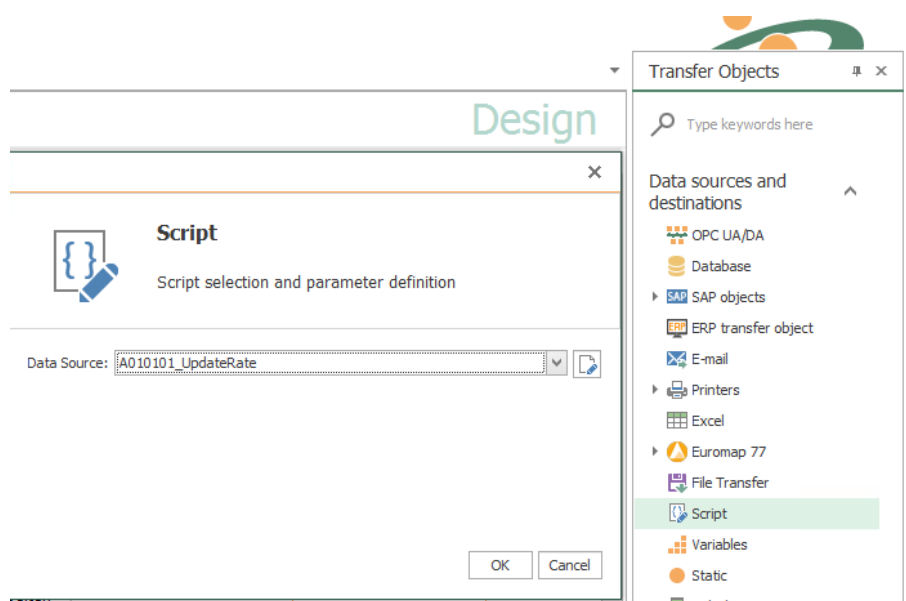
<i>Abort on file not found</i>	Enable to abort the transfer without error message if no file is found.
<i>Encoding</i>	Character encoding can be selected. The default is UTF8.

Tab: Target

<i>Move file after transfer</i>	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_hhmmss_fileName.
<i>Dynamic or Static</i>	Dynamic or static target path. For static target path, select the file name to read the file content.

Transfer Object “Skript”

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](#).



Create a script transfer object within the connection [as explained](#).

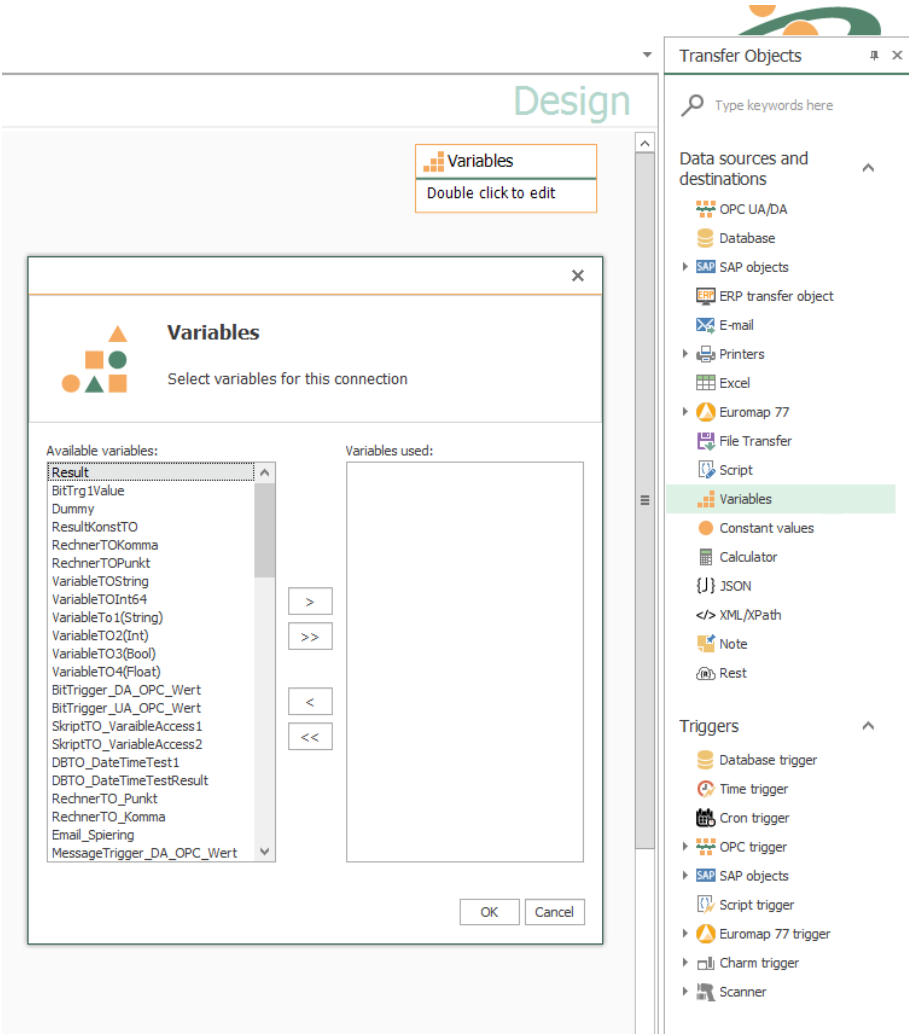
After double-clicking on the transfer object, set the following properties:

<i>Data Source</i>	Available scripts to choose from.
--------------------	-----------------------------------

The input and output parameters of the script are now available to you as elements of the transfer object.

Transfer Object “Variables”

Self-defined variables are available to you both as sources and as targets, system variables as source. The required variables must be configured as plug-in instances. Create a variable transfer object within the connection [as explained](#).



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

Tab: General (all types)

Available variables/Variables used	Available variables: existing, not yet used variables. Columns used: Variables that are displayed as elements (items) of the transfer object and are therefore available for transfers. This distinction primarily serves the overview in the configuration area – for example, if only one variable of 30 existing is needed for the connection.
------------------------------------	--

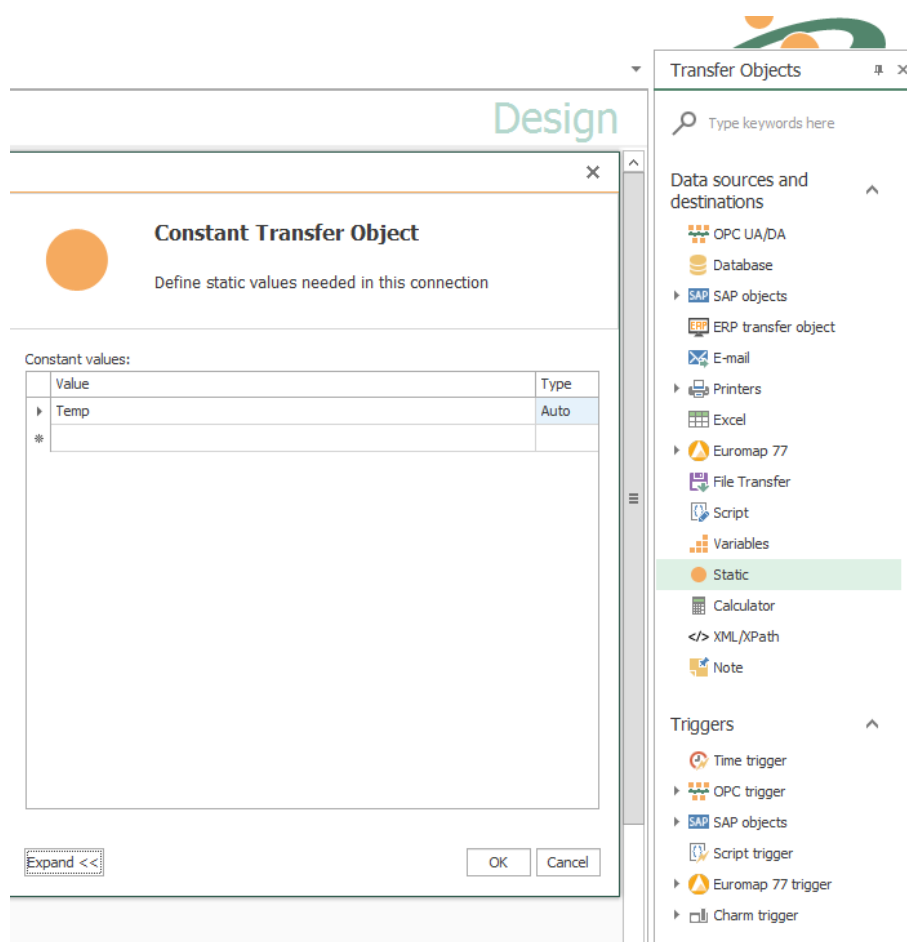
Move the columns by double-clicking on the column name or via the button “<” and “>” (after marking the column). “<<” and “>>” move all columns.

Transfer Object “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](#).

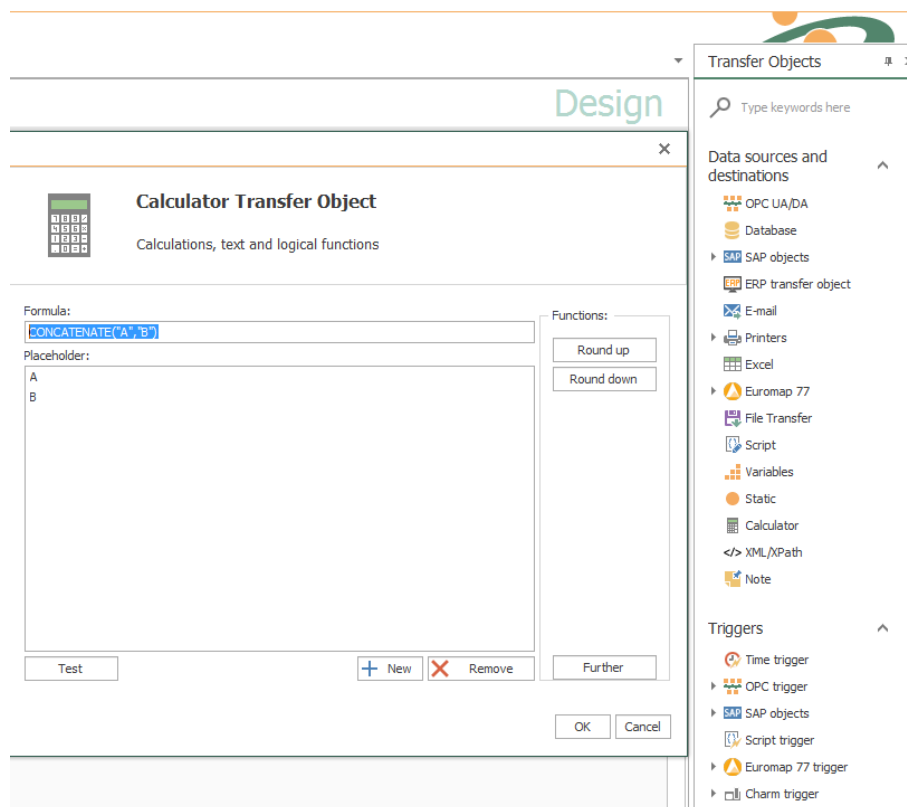


After double-clicking on the transfer object, set the following properties:

<i>Value</i>	Any value
<i>Type</i> (after click on “Expand”)	(ptional) Data type

Transfer Object “Calculator”

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](#).



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.

Transfer Object “Text replace”

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:

<i>Output</i>	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.
<i>Text</i>	Enter your text here. You can enter placeholders by double-clicking on the corresponding entry in the placeholder list to the right.
<i>Placeholder</i>	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

<i>Name</i>	Enter a unique name for the placeholder.
<i>Data type</i>	Specify the data type of the placeholder, for example, String, DateTime.
<i>Default Value</i>	Enter a default value if required.
<i>Preview</i>	The preview shows you the form in which a value will be displayed.
<i>Format</i>	Selection of a format for the data types Int64, Float or DateTime,

<i>Culture</i>	Select the culture to get the corresponding spelling. For example, commas or dots are set differently depending on the selected culture.
----------------	--

Transfer Object “JSON”

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.

A date display should be written as follows:

2012-04-23T18:25:43.511Z

JSON transferobject

Configuration of JSON path values

JSON Items:

Name	JSON Expression
▶ \$.key	\$.key
\$.some	\$.some
\$.thing	\$.thing
*	

.....

JSON Parameters:

Name of parameter	Default value
▶ abc	1
*	

.....

JSON Tree

{Object}

key: value

some: [{ a: 1, b: 10 }, { a: 2, b: 20 }, { a: 3 ...

thing: { abc: def }

Edit

OK

Cancel

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

<i>JSON Items</i>	Enter the JSON items directly or select entries from the JSON tree. 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. To delete an entry, place the cursor next to the field to be deleted and press the delete key. If you are still in edit mode, click on the escape key to end editing and then delete.
-------------------	---

<i>JSON Parameters</i>	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.
<i>JSON Tree</i>	The JSON Tree is the standard JSON document during a transfer. This can be read and written to. Click on the "Edit" button and enter your tree structure. Confirm the JSON Editor entries with "OK". Now select an entry and right-click on "Create Expression". The entry now appears at the top of the JSON Items list.

A tree structure is defined in the editor of the JSON tree as follows:

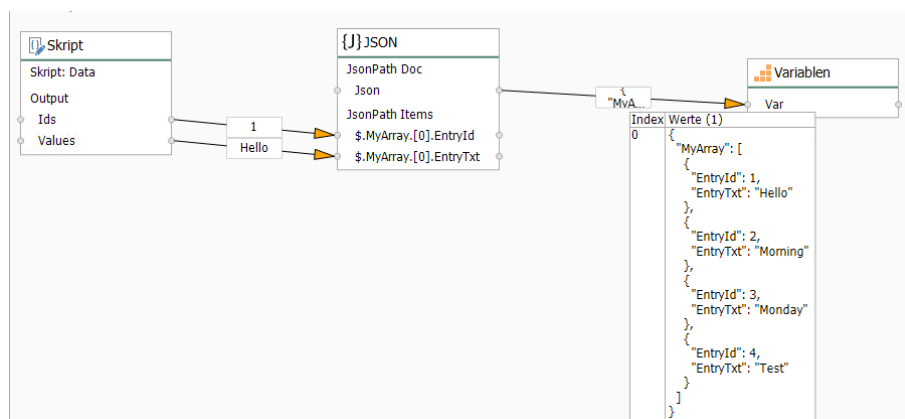
```
{
  "key": "value",
  "some": [
    { "name": "eins", "value": 1 },
    { "name": "zwei", "value": 2 },
    { "name": "drei", "value": 3 }
  ]
}
```

The expression \$.some[?(@.name == 'two')] selects {"name": "two", "value": 2}. Your tree structure would look like this after confirming with "OK":



You have the possibility to describe arrays that contain an object array as content.

Example: Describing the array properties up to n times. The actual array had only 2 entries, but n entries corresponding to the highest count are created within the array.



Arrays can also be further nested. The example below shows how to filter values from different arrays. In the example you want to get the object in the array at “fields” with the key “content” and all other information.

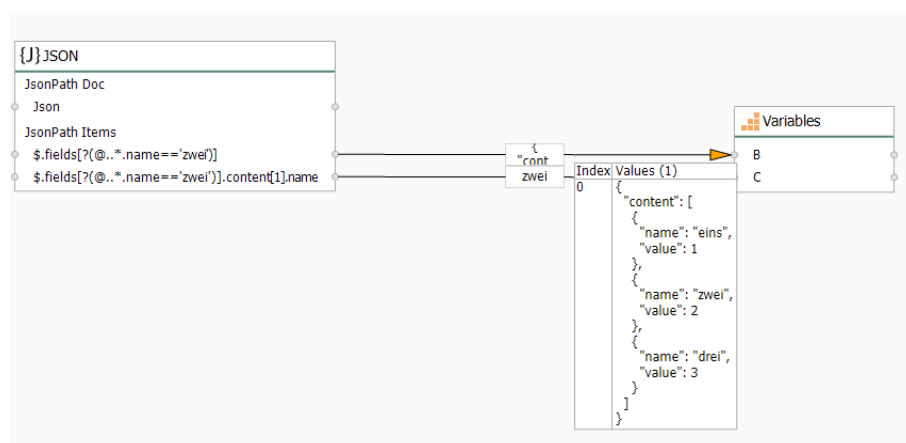
JSON document:

```
{
  "fields": [
    {
      "content": [
        {
          "name": "one", "value": 1
        },
        {
          "name": "two", "value": 2
        },
        {
          "name": "three", "value": 3
        }
      ]
    },
    {
      "content": [
        {
          "name": "eleven", "value": 11
        },
        {
          "name": "twelve", "value": 12
        },
        {
          "name": "thirteen", "value": 13
        }
      ]
    }
  ]
}
```

Output from the item with the expression `$.fields[?(@..*.name=='two')]`:

```
{
  "content": [
    {
      "name": "one", "value": 1
    },
    {
      "name": "two", "value": 2
    },
    {
      "name": "three", "value": 3
    }
  ]
}
```

Reason: The element from the array at “fields” in which the value “two” is for a child at the key “name”.



Transfer Object “XML” (XPath queries)

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

<i>Name</i>	Name where the query appears in the transfer object.
<i>X-Path Expression</i>	Valid X-Path query, in the simplest case the path to the node (see below).
<i>XML Mode</i>	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.

Section: Namespaces and Prefixes

<i>Prefixes</i>	Unique prefix
<i>Namespace</i>	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 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>

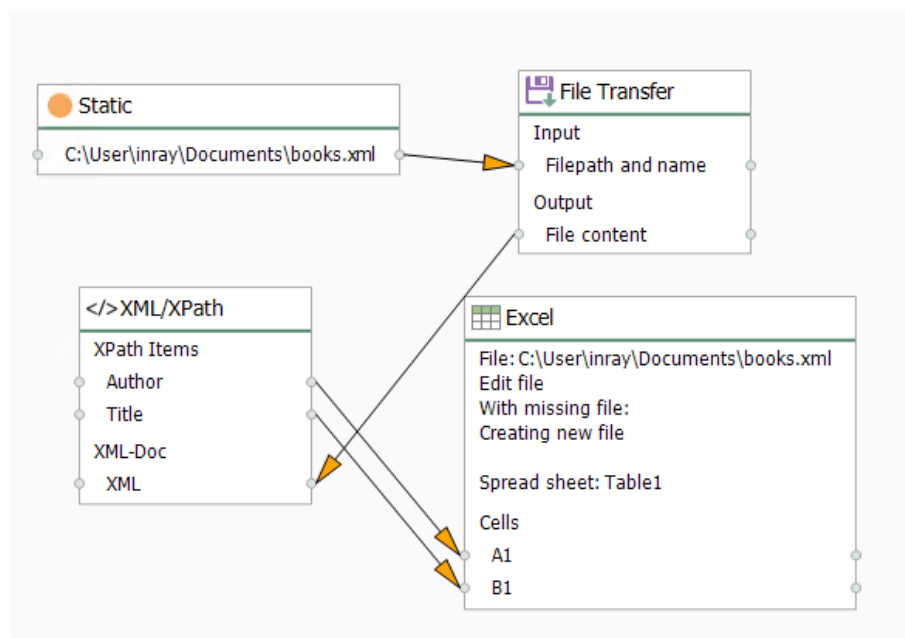
```

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 (e.g. by http://www.w3schools.com/xml/xpath_syntax.asp ; Restriction: Any nodes must be initiated with "//").

For this example, the configuration looks like this:



1. 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.
2. 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.
3. 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 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.

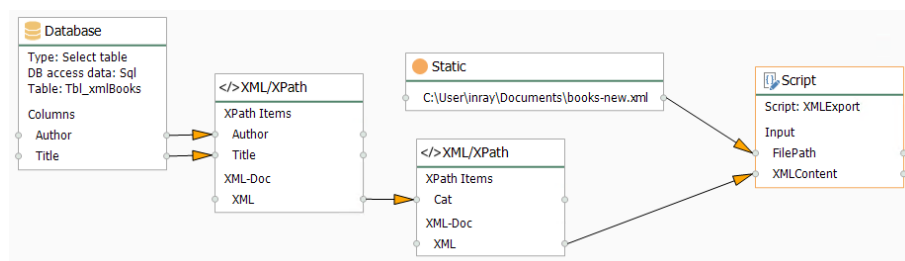
For example, it comes from a table:

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.
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

```
<book>
  <author></author>
  <title></title>
</book>
```

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:

```
<?xml version="1.0"?>
```

```
<catalog>
</catalog>
```

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());
}
```

Transfer Object “Note”

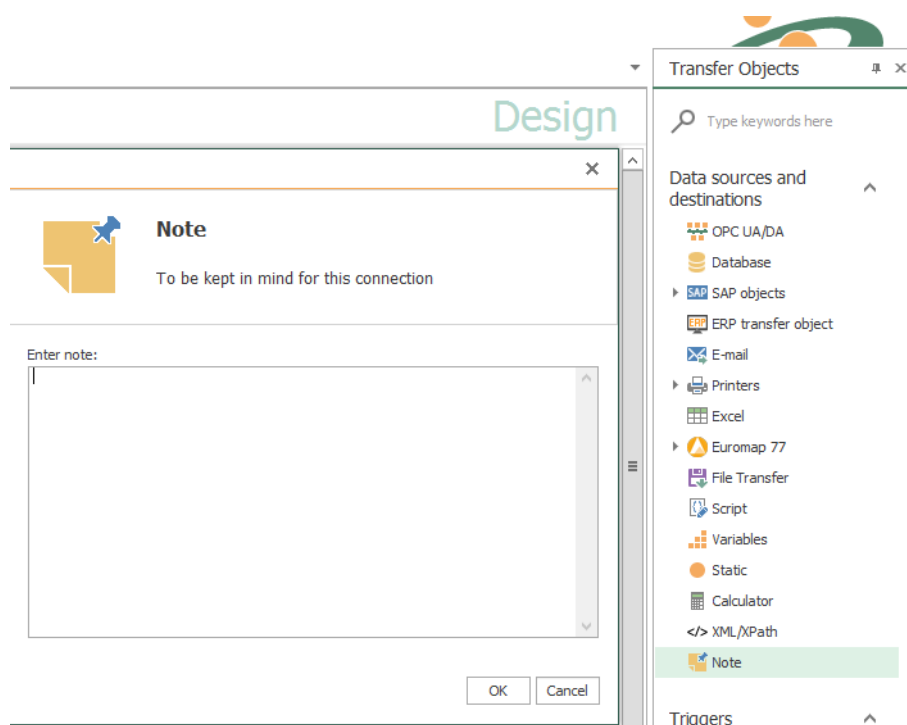
In extensive projects, especially when several colleagues work together, the comment on the connection is quickly no longer sufficient to keep track.

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](#).



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

Enter note	any text
------------	----------

Transfer Object “Rest”

Web service components can be combined with the remainder transfer object. Create a remainder transfer object within the connection [as described](#).

Rest
Compiling web service components

Connection:

☐ Define dynamic host URL

Method

URL-Endpoint:

Method:

Request

Format:

Parameter:

	Name	Value	Parameters type
*			

.....

Response

Parameter:

	Name	Value
*		

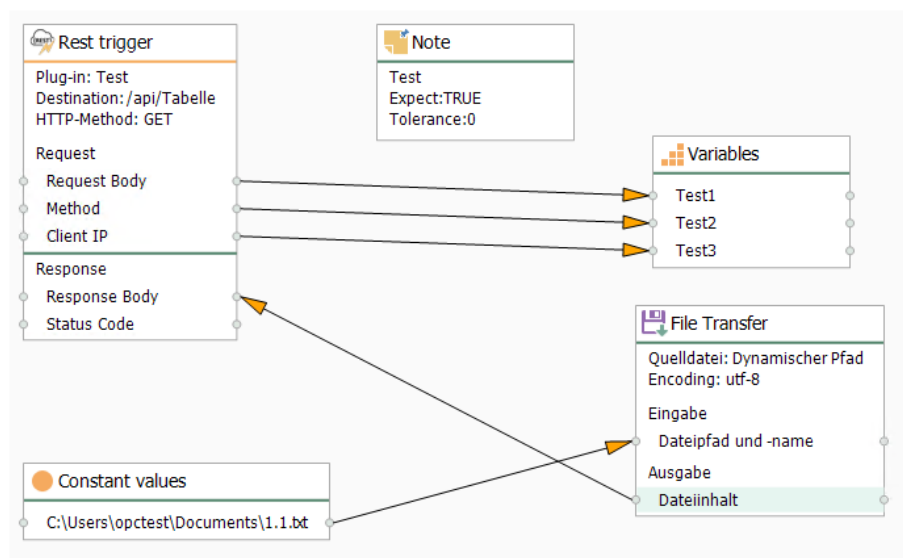
Test

OK Cancel

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

<i>Connection</i>	Hostname of the web service.
<i>Define dynamic host URL</i>	Enable if hostname changes. You can also use the Test button to check the dynamic host.
<i>URL -Endpoint</i>	Enter the address for the endpoint and select an entry from the drop-down menu.
<i>Method</i>	Select a method for the connection from the drop-down list.
<i>Format</i>	Select the format for the connection from the drop-down list. e.g. Json or Xml
<i>Request parameters</i>	Determine the parameters for the request of the connection.
<i>Response parameters</i>	Determine the parameters for the response of the connection.
<i>Test</i>	About button “Test” determine if the connection is made. If you have checked the “Dynamic host” checkbox, this button opens a text field in which you can further define the URL endpoint in the “Host address” field. Check your entry with “Send”.

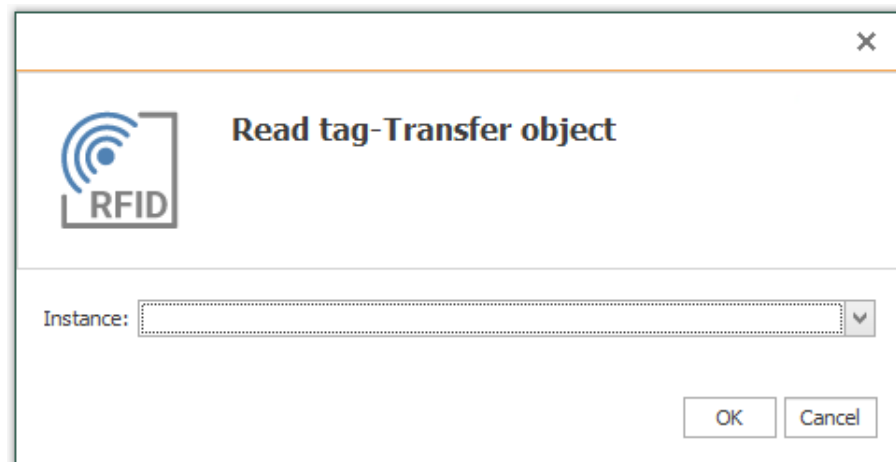
Attention: Please take care when entering the URL on uppercase and lowercase letters. The URLs are Case sensitive!



Transfer Object “RFID Reader”

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](#).

After double-clicking the transfer object, select an instance from the drop-down menu.



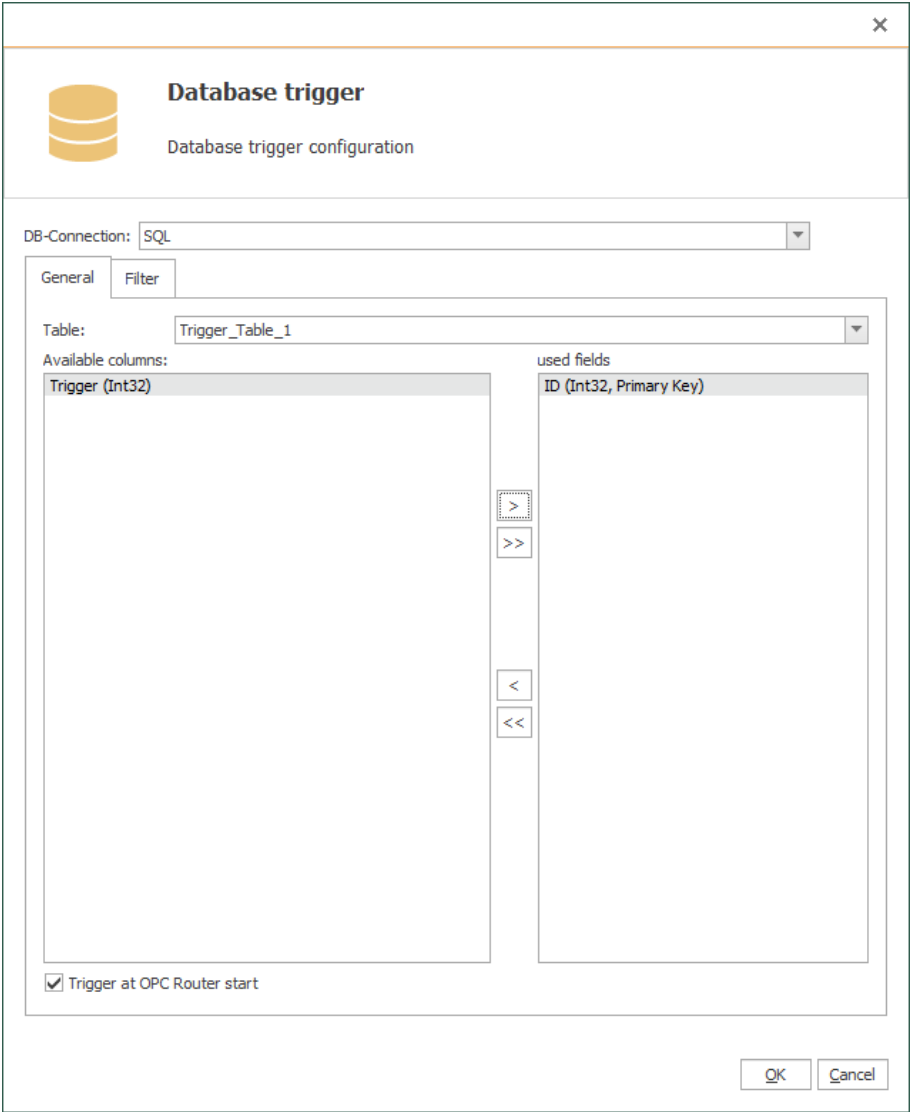
Trigger Objects

So-called trigger objects, triggers for short, execute the data transfer if their conditions are met. Each connection needs at least one trigger. Each trigger contains one condition that has to be met to execute the transfer of that specific connection. Complex conditions can be programmed with a script trigger (C#).

Trigger objects are projected [as described above](#), but not connected with arrows.

Trigger Object “Database”

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.



The individual properties are:

<i>DB-Connection</i>	The configured instances of the DB plug-in are available here.
----------------------	--

Registerkarte: Allgemein (alle Typen)

<i>Table</i>	Table being monitored.
--------------	------------------------

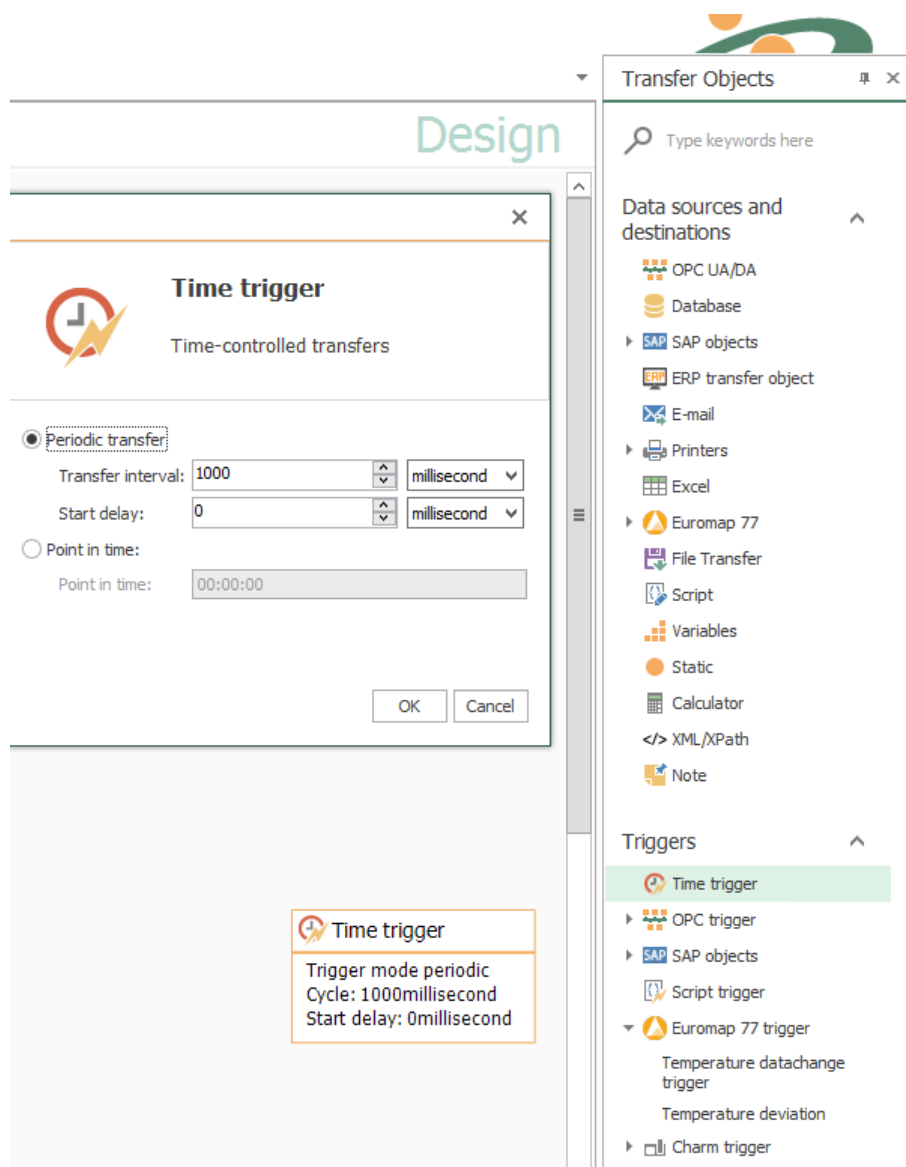
	Note: The specified table must have a primary key. Tables without a primary key are not supported.
<i>Available columns</i>	All available columns that have not yet been selected for the panel “Used fields”. Select the columns to monitor.
<i>Used fields</i>	All already selected fields. Use the arrows between “Available columns” and “Used fields” to add or remove fields.
<i>Triggern at OPC Router start</i>	At router start it will be trigger.

Registerkarte: Filter (Abfragen, Aktualisieren, Löschen)

<i>Filter</i>	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.
<i>Edit filter</i>	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.
<i>Preview</i>	Preview the SQL statement.

Trigger Object “Time”

The time trigger triggers data transfers at variable intervals in a time-controlled interval. If a transfer is to be carried out once a day, you can also specify a fixed time.



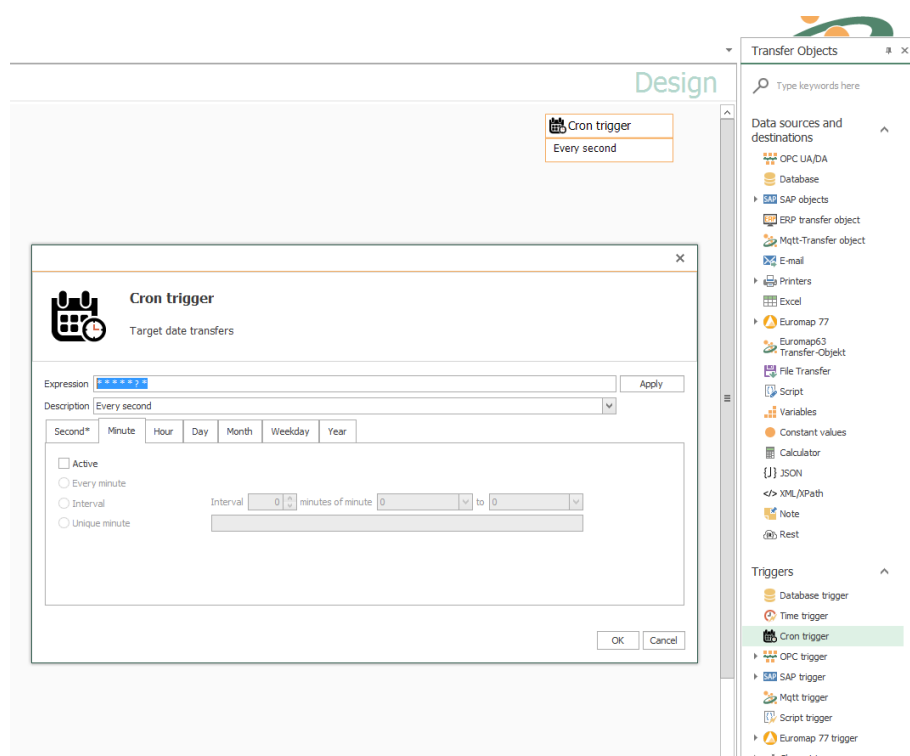
The individual properties are:

<i>Option: Periodic transfer</i>	The transfer will be triggered at the specified interval.
<i>Transfer interval</i>	The interval at which the transfer is triggered. If <i>start delay</i> = 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 or hours 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.
<i>Start delay</i>	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 or hours are to be selected.
<i>Option: Point in time</i>	The transfer will be triggered once a day at the set time.
<i>Point in time</i>	Time at which the transfer is triggered.

Trigger Object “Cron”

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.



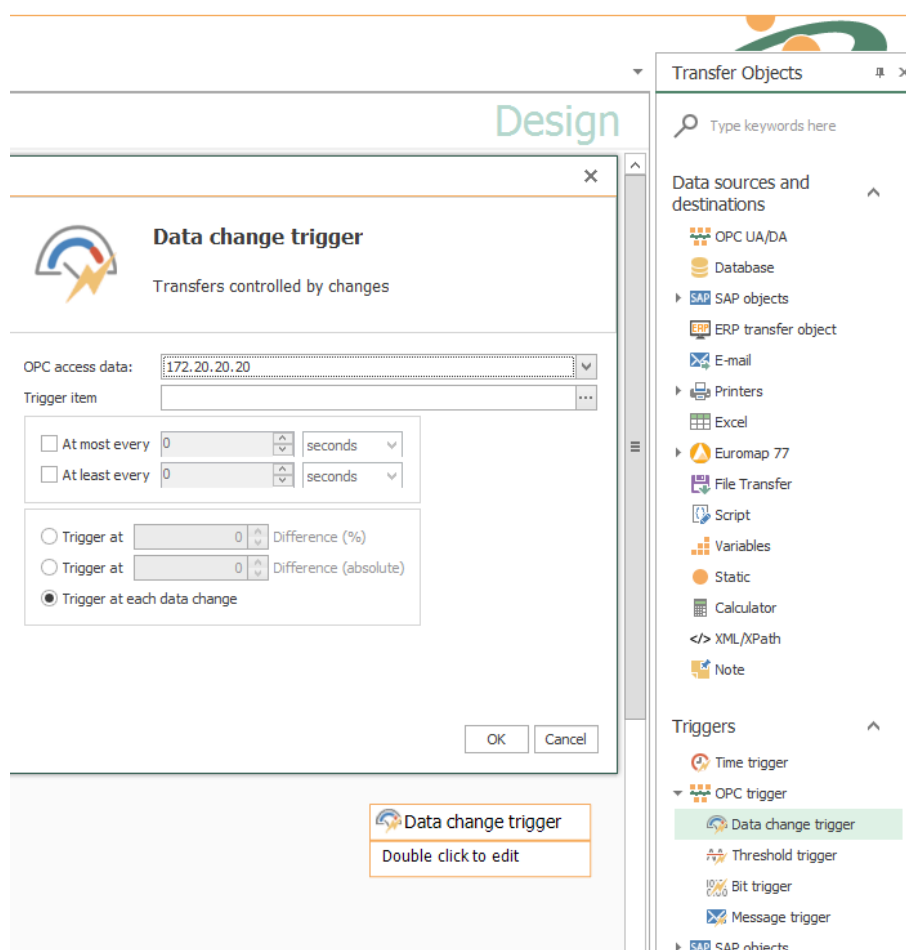
The individual properties are:

<i>Expression</i>	Display of timed transfers after selection of properties. Here is also the possibility of manual input. After entering, confirm with the “Apply” button.
<i>Description</i>	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.
<i>Active</i>	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.

Trigger Object “Data Change”

Data change trigger executes transfers with a changing OPC 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.



<i>OPC access data</i>	Select a set of OPC access data. Dependent on plug-in configuration, OPC Router will recognize DA/UA connections.
------------------------	---

<i>Trigger item</i>	Item whose change will execute the transfer. Browse items with tag browser ("..." button). OPC UA node IDs are registered in background.
---------------------	--

Use the extended button to define further properties

<i>At most every</i>	Minimum interval in which transfer will be executed even if no data change has occurred. Integers only, unit from milliseconds to hours.
<i>At least every</i>	If data change frequently the number of transfers can be limited. An entry means, e. g. Transfer every second maximum even if data changes more frequently. Integers only. Units from milliseconds to hours.
<i>Trigger at ... difference (%)</i>	Transfer will only execute if data changes by this difference.
<i>Trigger at ... difference (absolute)</i>	Transfer will only execute if data changes by this difference.
<i>Trigger at each data change</i>	Default. Overrides tolerance settings.

Tip: To set conditions for relative and absolute tolerance, project two data change triggers linked with AND or OR.

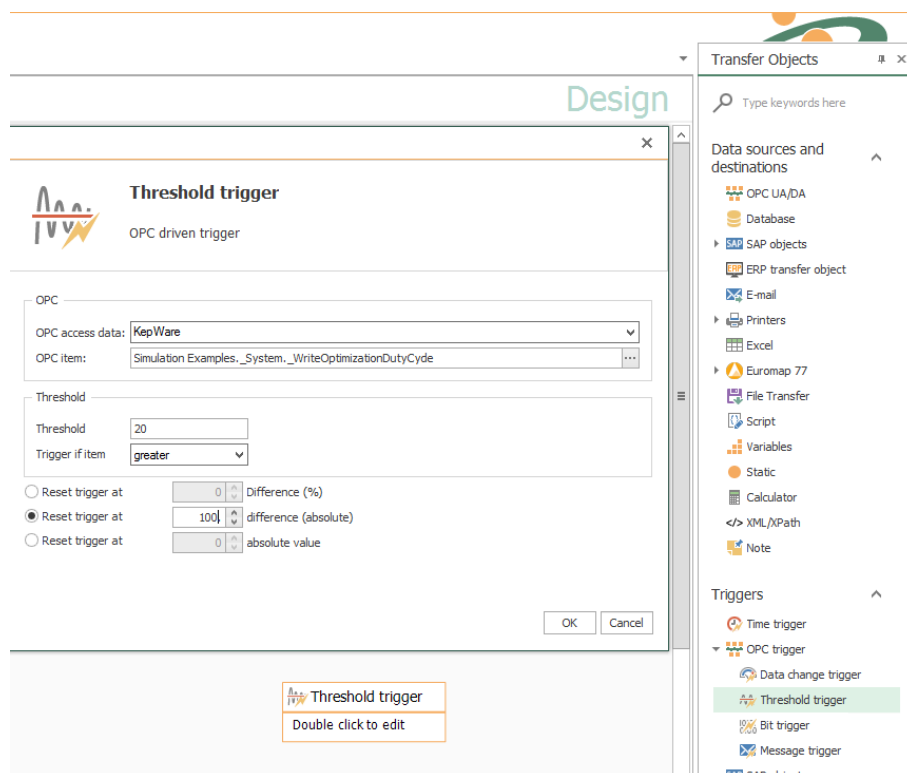
Data length and limit value triggers for analog values

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.

Trigger Object “Threshold Trigger”

Threshold trigger executes a data transfer with off-limit condition (of OPC value)



The properties are:

<i>OPC access data</i>	Select a set of OPC access data. Dependent on plug-in configuration, OPC Router will recognize DA/UA connections.
<i>OPC item</i>	Item whose off-limit condition will execute the transfer. Browse items with tag browser (“...” button). OPC UA node IDs are registered in background.
<i>Threshold</i>	Absolute value.
<i>Trigger if item</i>	Set if threshold has to be under- or overrun.
<i>Hysteresis</i>	If the trigger has executed the transfer once, it might be required that the trigger is re-released only if the OPC item's value is clearly different from threshold value. If this is the case, please specify the difference between OPC 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.

Data length and limit value triggers for analog values

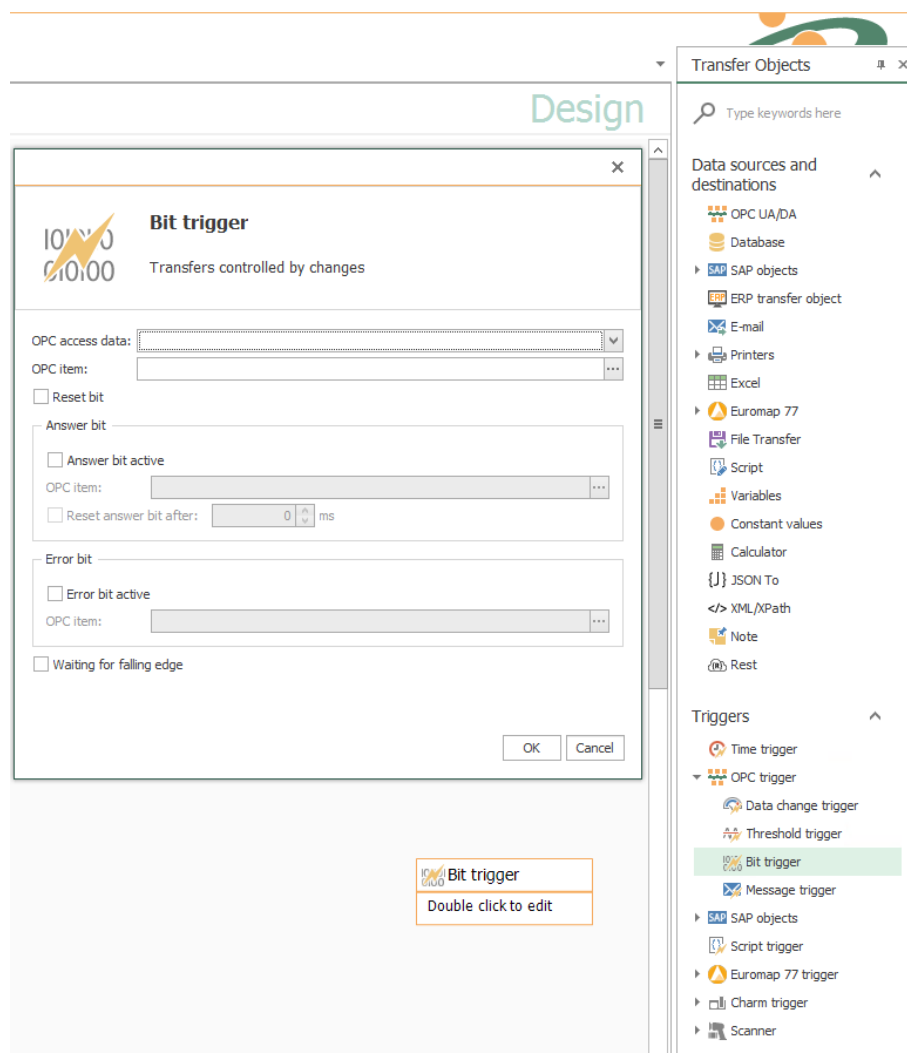
You can set the data length and limit value triggers if you want to transmit a single analog value. Trigger item and transfer item are identical. The transfer starts immediately after the trigger condition has occurred.

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.

Trigger Object "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.



<i>OPC access data</i>	OPC connection which the relevant data point is reached.
<i>OPC item</i>	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) for the interactive selection of an item. For OPC UA connections, the node ID is managed in the background.
<i>Reset bit</i>	Resets the OPC item to zero after the transfer. Should usually be disabled because a bit should only be written from one side. Instead, use the <i>answer bit</i> to tell the controller that the transfer has completed.
<i>Answer bit active</i>	Sends one bit to another data point in response that the transfer has completed.

<i>OPC item</i>	OPC 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.
<i>Reset answer bit after</i>	After the set time, the response bit is reset.
<i>Error bit active</i>	Sends a bit to another data point if the data transfer was faulty.
<i>OPC item</i>	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.
<i>Waiting for falling edge</i>	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.

Bit and message triggers for OPC telegrams

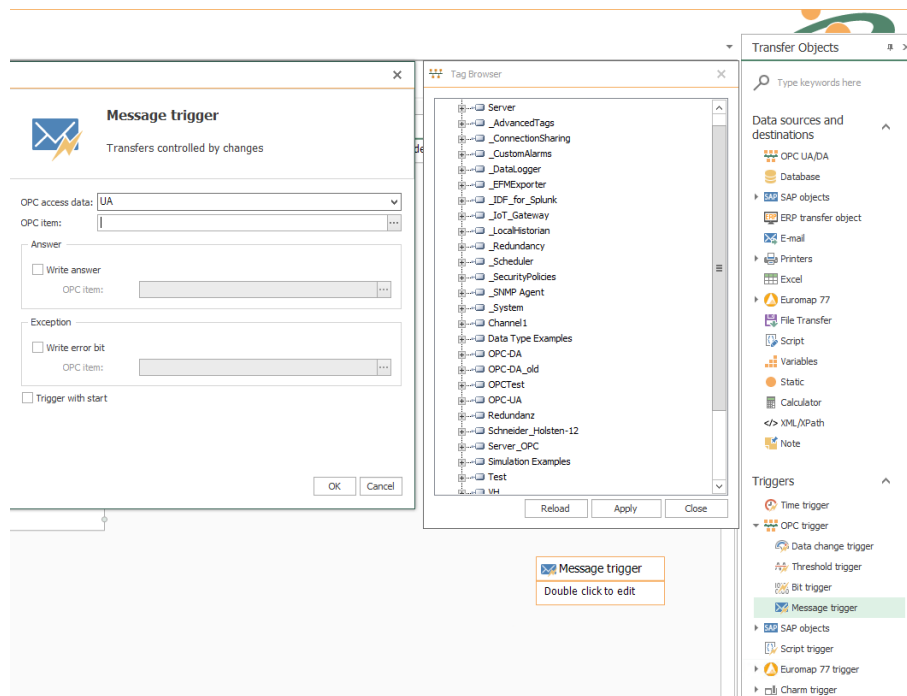
Bit and message triggers are used for OPC telegrams if you want to read several related items. The OPC router waits for another read cycle before the items are read out in the OPC transfer object to ensure that the PLC could write all values. On the SPS side the trigger should be written to the end.

Bit 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 be 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.

Trigger Object “Message”

The message trigger triggers the data transfer if the assigned numeric data point is 1 higher. You can also send an answer and, if the transfer is incorrect, an error message to two other data points.



<i>OPC access data</i>	Select a set of OPC access data. Dependent on plug-in configuration, OPC Router will recognize DA/UA connections.
<i>OPC item</i>	Item whose increase will execute the transfer. Browse items with tag browser (“...” button). OPC UA node IDs are registered in background.
<i>Write answer</i>	Will transfer OPC item’s value (or set a bit) to another item as confirmation that the transfer executes.
<i>OPC item</i>	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.
<i>Write error bit</i>	Activate to set a bit with transfer errors.
<i>OPC item</i>	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.
<i>Trigger with start</i>	Release trigger once with OPC Router start.

Bit and message triggers for OPC telegrams

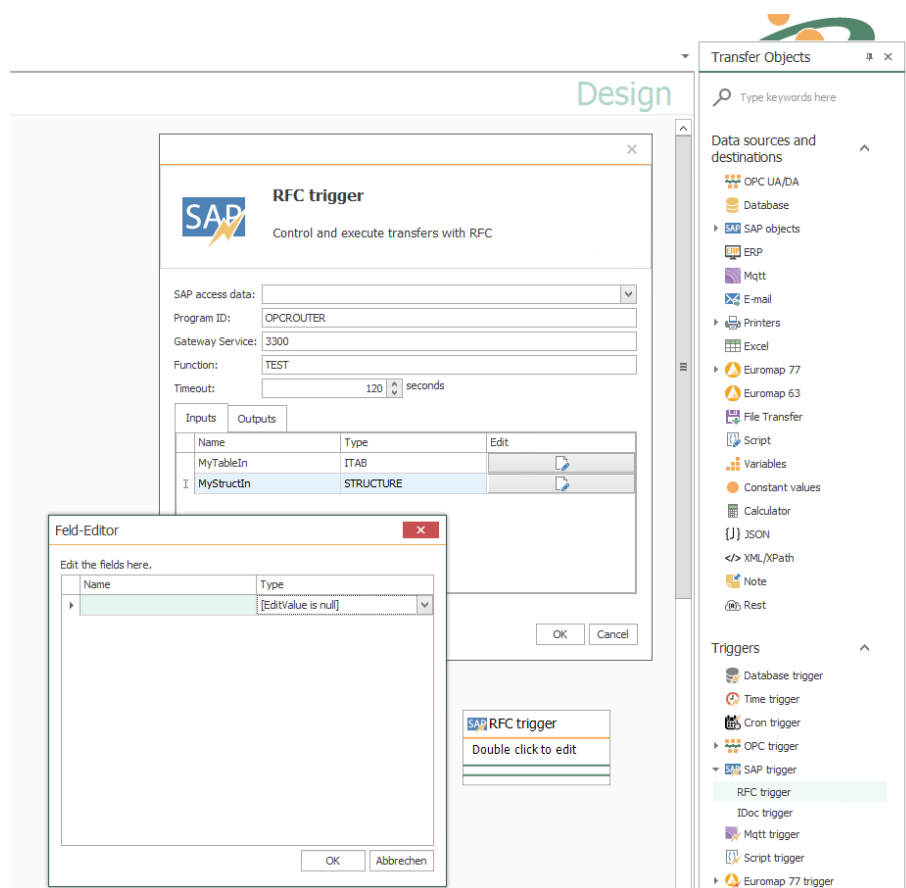
Bit and message triggers are used for OPC telegrams if you want to read several related items. The OPC router waits for another read cycle before the items are read out in the OPC transfer object to ensure that the PLC could write all values. On the SPS side the trigger should be written to the end.

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.

Trigger Object “RFC”

With RFC trigger, SAP systems can trigger data transfers. Within SAP system, in transaction SM59 the corresponding entry has to be made.



Properties are:

<i>SAP access data</i>	Set of SAP access data.
<i>Program ID</i>	Program ID used by SAP system to call destination, identical with program ID of transaction SM59.
<i>Gateway Service</i>	Gateway Service of SAP system

<i>Function</i>	Function defined within SAP system that triggers this connection.
<i>Time-out</i>	Time span after which OPC Router will send time-out to SAP system if task has not been finished

return parameters

<i>TRANSFER_OK</i>	• 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.
--------------------	---

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:

<i>SAP access data</i>	<i><SAP access data></i>
<i>Program ID</i>	INRAYTRIGGER
<i>Gateway Service</i>	Gateway Service of SAP system, e. g. 3302
<i>Function</i>	TRIGGERFUNCTION

Corresponding SAP properties, SM59:

<i>Destination</i>	OPCROUTER
<i>Connection Type</i>	T TCP/IP
<i>Activation Type</i>	Registered Server Program
<i>Program ID</i>	INRAYTRIGGER

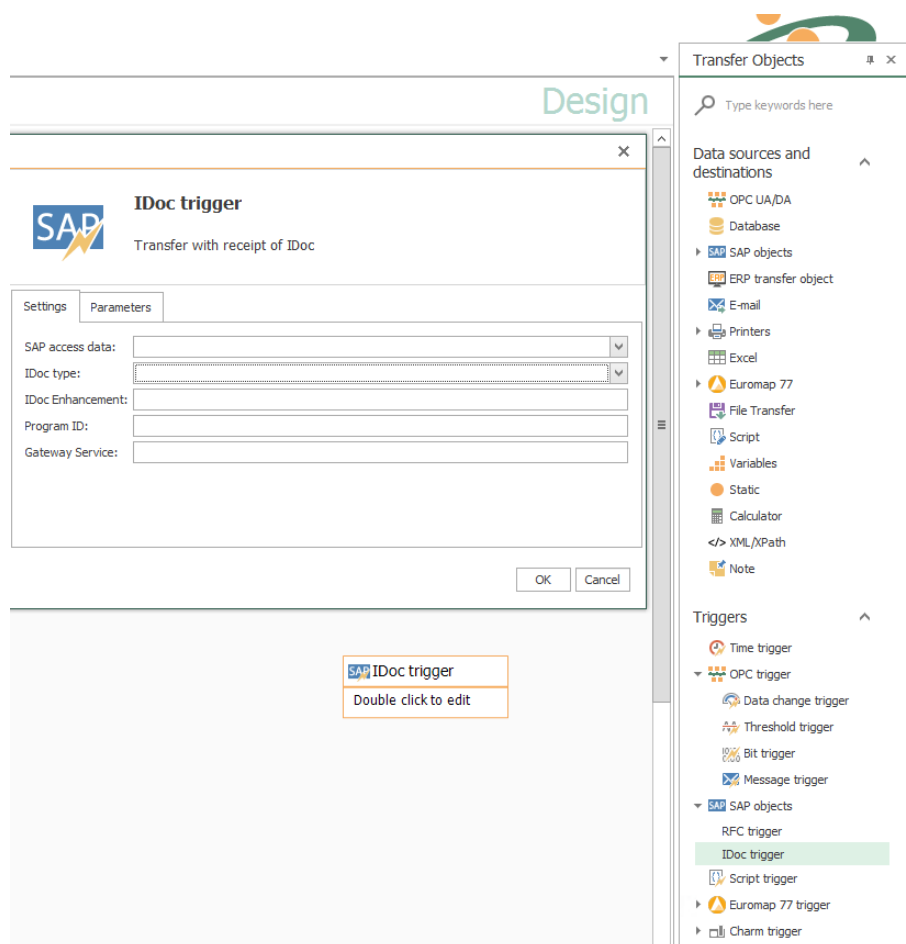
<i>Gateway-Host</i>	(no entry!)
<i>Gateway-Service</i>	(no entry!)

By calling this function from within SAP system, SAP system will trigger OPC Router:

Call function 'TRIGGERFUNCTION' destination 'OPCROUTER'.

Trigger Object "IDoc"

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:

<i>SAP access data</i>	Set of SAP access data.
<i>IDoc type</i>	Type of IDoc the Trigger listens. Must be the same as in corresponding transfer object.
<i>IDoc Enhancement</i>	Class extension of the IDocs.

<i>Program ID</i>	Program ID used by SAP system to call destination, identical with program ID of transaction SM59.
<i>Gateway Service</i>	Gateway Service of SAP system

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.

Example*IDoc Trigger Settings:*

<i>SAP access data</i>	<SAP access data>
<i>IDoc Type</i>	PROCESSORDER
<i>Program ID</i>	INRAYTRIGGER
<i>Gateway Service</i>	Gateway Service of SAP system, e. g. 3302

Logical System within SAP transaction BD54:

<i>LogSystem</i>	OPCRIDOC01
<i>Description</i>	OPC Router with IDoc listener

Corresponding SAP Settings, SM59:

<i>Destination</i>	OPCROUTER
<i>Connection Type</i>	T TCP/IP
<i>Activation Type</i>	Registered Server Program
<i>Program ID</i>	INRAYTRIGGER
<i>Gateway host</i>	(no entry!)
<i>Gateway service</i>	(no entry!)

Communication Port in transaction WE21:

<i>(Type)</i>	Transactional RFC
<i>Port</i>	e.g. IDOCPORT
<i>Version</i>	(select appropriated IDoc version)
<i>RFC destination</i>	OPCROUTER

Partner profile in transaction WE20

<i>(Partner type)</i>	LS (Logical System)
<i>Partner no.</i>	OPCRIDOC01

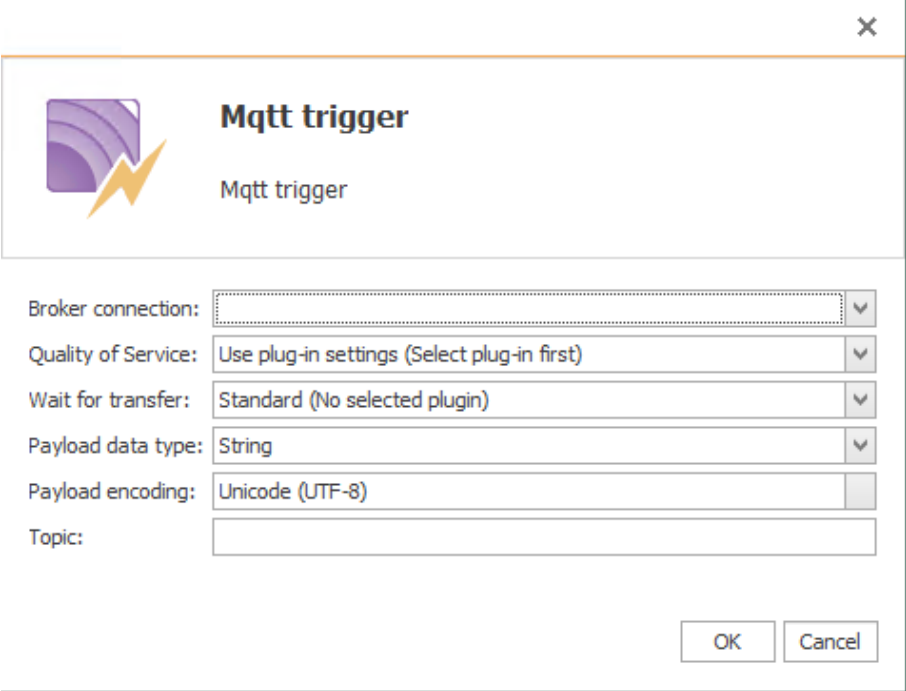
Outbound parameters in WE20:

<i>Receiver Port</i>	IDOCPORT
<i>Outpout mode</i>	Transfer IDoc immed.
<i>IDoc Type/Basic type</i>	PROCESSORDER
<i>Cancel processing af- ter syntax error</i>	(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).

Trigger Object “Mqtt-Trigger”

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



The individual properties are:

<i>Broker connection</i>	Select a configured broker connection from the drop-down menu.
<i>Quality of Service</i>	Selection how often the query should be executed.
<i>Wait for transfer</i>	Select whether and in what form to wait for the transfer.

<i>Payload data type</i>	Selection of the data type Byte Array or String.
<i>Payload encoding</i>	Selection of the encoding for the Mqtt query.
<i>Topic</i>	Enter the topic.

Trigger Object “Telegram”

With the telegram trigger it is possible to query messages via a command.

Set the following properties after double-clicking the trigger object:

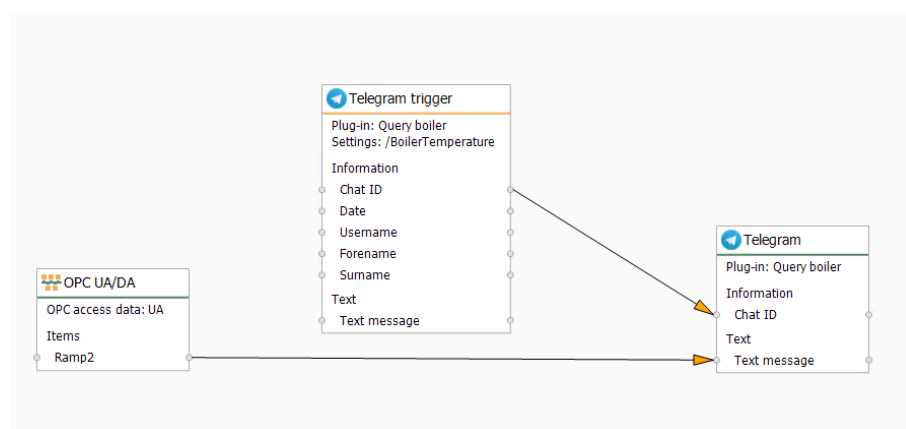
<i>Connection</i>	Select a connection from the drop-down menu.
<i>Command</i>	Type a command <i><Command></i>
<i>Help text</i>	Here you can enter a help text for the command, which will be displayed later in the notification.
<i>Message type</i>	Select a message type from the drop-down menu, e.g. Text message, Voice message...The message type “Any file” can be sent any file format (text, images, videos, etc.).
<i>Authentication required</i>	Activate the checkbox if you want to allow only authorized users to receive the messages.
<i>Authorized users</i>	Select allowed users from the drop-down menu.

<i>User filter</i>	Here you have the possibility to enter a filter to select specific users.
--------------------	---

You can receive any form of a file with the Telegram trigger. To send a file from the OPC Router to Telegram, you need the Telegram Transfer Object. To send a file to the OPC Router, e.g. an image, you have to enter the trigger command as description. However, the file name of the image is not sent and you have to give the file a new name if you want to save the file.

Example Telegram Connection

In our small example the boiler temperature is queried. The data is collected by the OPC router and can be queried via telegram.

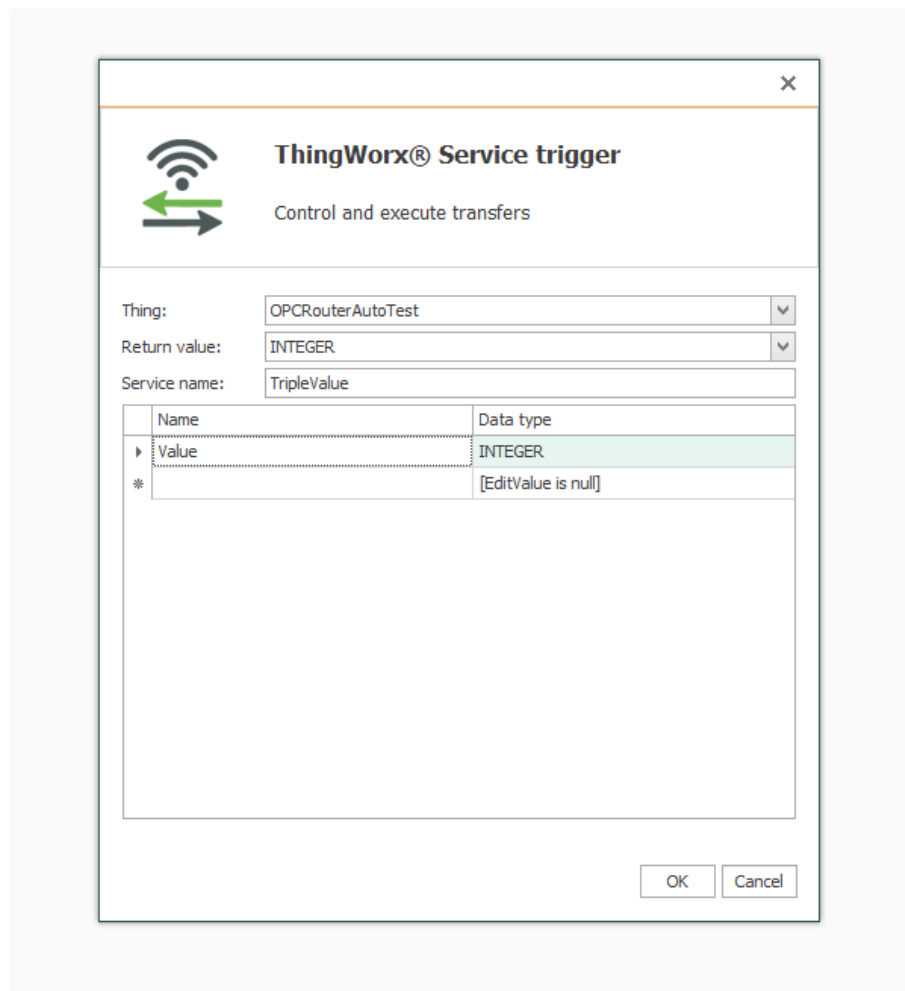


To display the boiler temperature on a remote web-enabled device, proceed as follows:

1. Go to the link of your bot that you received together with the AccessToken from BotFather.
2. Enter the command `/start`.
3. If you click on `/help`, the BotFather gives you all possible commands.
4. Select the command `/BoilerTemperature`.
5. The boiler temperature and the help text will then be displayed on your device.

Trigger Object “ThingWorx®”

With the ThingWorx®-Service-Trigger it is possible to create own services in the OPC Router. If this service is accessed via a service transfer object, for example, the connection is triggered and returns a result to the service transfer object.



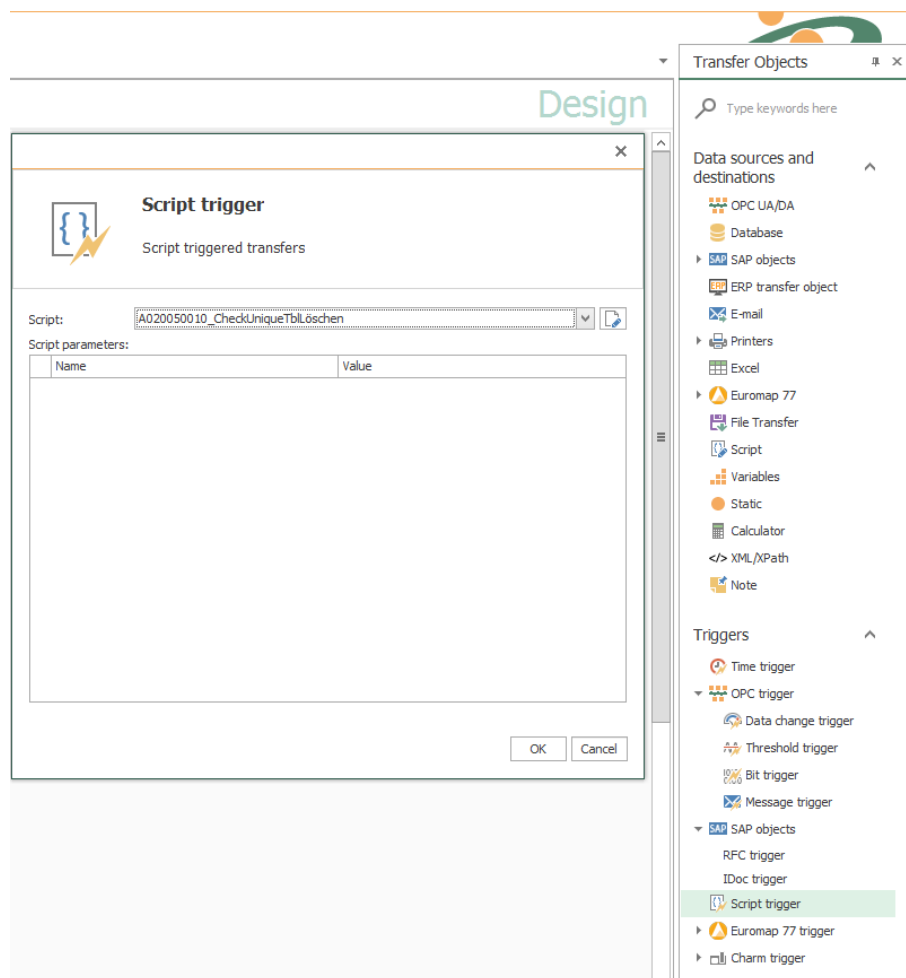
Set the following properties after double-clicking the trigger object:

<i>Thing</i>	Select your Thing from the drop-down menu.
<i>Return value</i>	Select the data type for the return value from the drop-down menu.
<i>Service name</i>	Select the ThingWorx® service from the drop-down menu.

Trigger Object “Script”

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. Meintest du Benötigt der Script-Trigger dynamische Werte, so

sind diese im Trigger selbst zu ermitteln? 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.



The individual properties are:

<i>Script</i>	Trigger script being executed.
<i>Script parameters</i>	Input parameters defined in the script. All parameters must be provided with a value or a template variable.

Trigger Object “Euromap77”

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:

<i>Profil</i>	Here the profiles are available as configured instances of the Euromap77 plug-in. Select a profile from the drop-down list.
<i>Destination</i>	Select an endpoint from the drop-down list. The “Check” button allows you to verify that your connection is working.
<i>Machine</i>	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

<i>At most every</i>	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.
<i>At least every</i>	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
<i>Trigger at ... difference (%)</i>	The transfer is only triggered if the value of the data point changes by this entry (relative).
<i>Trigger at ... difference (absolute)</i>	The transfer is only triggered if the value of the data point changes by this entry (absolute).
<i>Trigger at each data change</i>	Default setting, <i>sets trigger at ... difference</i> override.

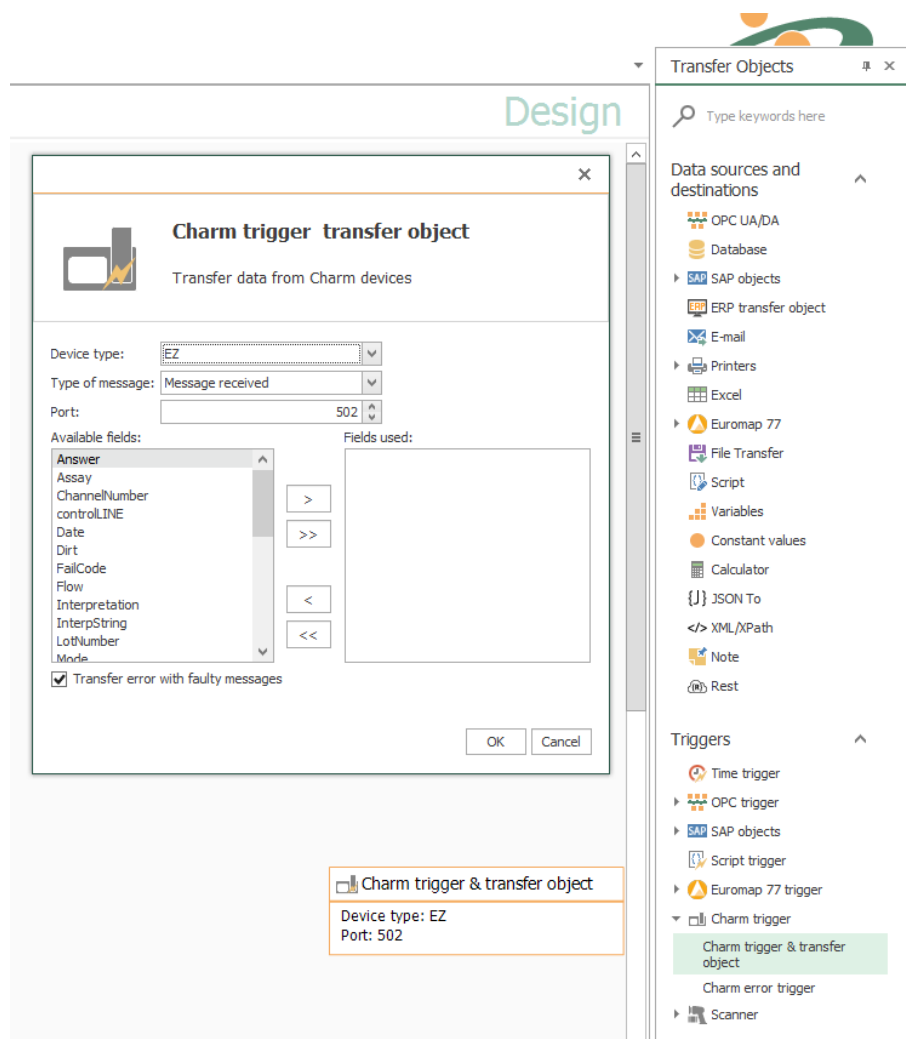
Features temperature deviation

<i>Trigger if Low-Deviation-Temperature 1 exceeds</i>	When activated, the temperature is only triggered when falling below the low deviation.
<i>Trigger if High-Deviation-Temperature 1 exceeds</i>	When activated, the temperature is triggered only when the highest deviation is exceeded.

Trigger Object "Charm"

The Charm trigger releases a transfer when new data are presented by the Charm device observed. In contrast to the OPC Router 3, which provided the transfer object

and trigger separately, both functions are now combined in the Charm trigger object - the trigger is also the data source.



Properties are:

<i>Device type</i>	Type of Charm-Reader. The devices “EZ”, “EZ-M” and “EZ-M2” are predefined. However, you have the option of defining additional device types via an XML-file
<i>Type of message</i>	By default, “Message received” is set. When switching to “Heartbeat received”, an internal device in the background receives a message.
<i>Port</i>	Port on this machine (with the OPC router) to which the Charm device sends the data. Basically, a separate port must be created for each Charm device type. Recommended Action: All Charm devices of one type send to the same port. A single connection is created in the OPC router for this purpose.

	In addition to the results, the OPC router also transmits the information about the sending Charm device to the database. The logic of how the results are processed resides in the database (or other transfer destination). However, if a separate connection is to be created for each Charm device, a separate port must also be available for each device.
<i>Available fields/ Fields used</i>	From the available fields, select the fields you want to use for your connection. Use the arrows to transfer or remove your selection.
<i>Transfer error with faulty messages</i>	Activate to trigger a transfer error in case of erroneous messages. Deactivate to ignore erroneous messages (e.g. to intercept faulty messages with the charm-error-trigger in another connection, see there).

Confirm your entries with “OK”.

In addition to the message fields selected from the available fields, additional information about the message received is automatically output (message information). At “message information”, the complete message, the IP address of the Charm device, the time stamp and a numbering of several consecutive messages are output.

News information

<i>Complete message</i>	The Complete Message of the Charm device.
<i>Client address</i>	IP address of the Charm device.
<i>Message index</i>	For multiple concurrent messages, the OPC Router will number each message.
<i>Timestamp (UTC)</i>	Timestamp of message (UTC).

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

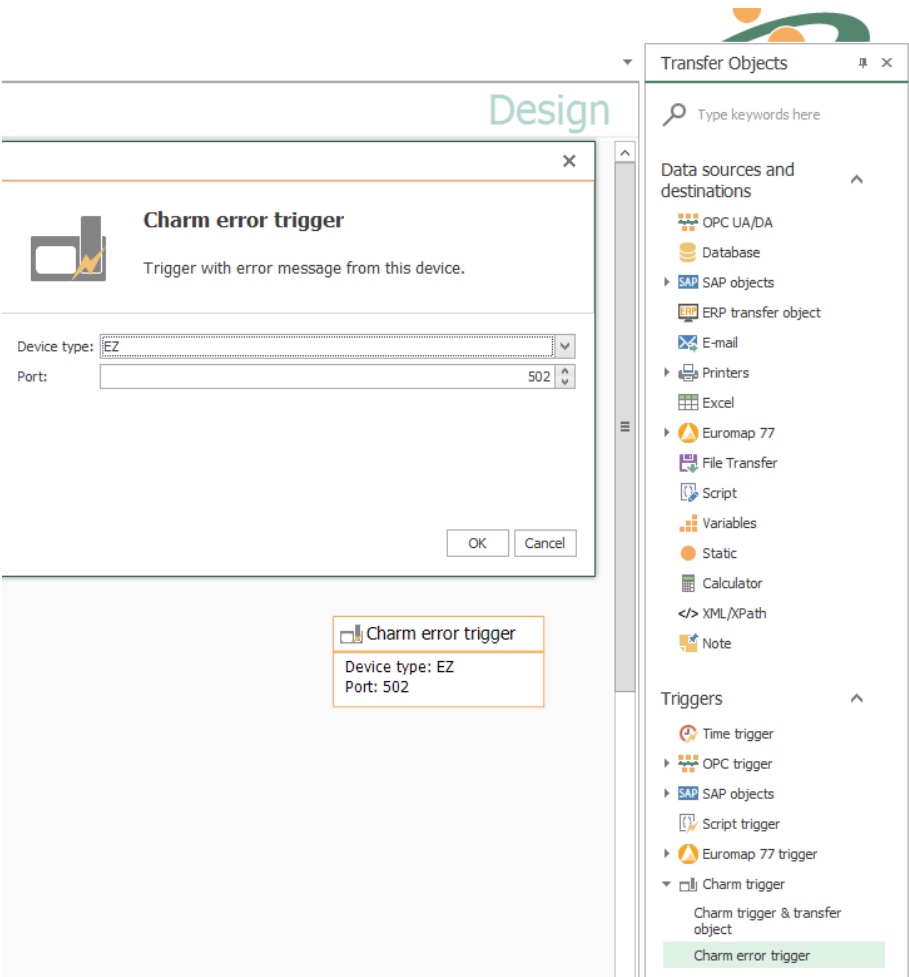
Name	EZ	EZ-M	EZ-M2	Description
<i>Answer</i>	X	X		Numerical test result including unit where necessary
<i>Assay</i>	X	X		sample type

<i>ChannelNumber</i>	X	X	X	EZ channel number
<i>CommodityName</i>		X	X	Raw material / article
<i>controlLINE</i>	X	X	X	Intensity of the Control Line
<i>CurrentCommoditySelectIndex</i>		X		
<i>CurrentDilutionSelectIndex</i>		X		
<i>CurrentMatrixSelectIndex</i>		X		
<i>CurrentTestSelectIndex</i>		X		
<i>Date</i>	X	X	X	Date of the test
<i>DE_Name</i>		X	X	
<i>Dirt</i>	X	X	X	Degree of contamination of the lens
<i>FailCode</i>	X	X	X	Error code on failed test
<i>Flow</i>	X	X	X	Flow value (Correct application of the milk on the test strip)
<i>Interpretation</i>	X	X	X	Interpretation of test result (positive, negative, etc.)
<i>InterpString</i>	X			(Interpretation in newer firmware)
<i>LotNumber</i>	X	X		LOT number (as entered on the unit)
<i>MatrixNumber</i>			X	Matrix number of the test result
<i>Mode</i>	X	X	X	Mode of the Charm device
<i>OperatorID</i>	X	X		Operator ID (as entered on the device)
<i>Ratio</i>		X	X	Ratio of Extraction Solvent to Extraction
<i>Remmissions</i>	X	X		
<i>Result</i>	X	X	X	Test result (positive, negative, invalid)
<i>RFID</i>			X	
<i>SampleID</i>	X	X	X	Test ID (as entered on the device)
<i>StructVersion</i>	X	X	X	
<i>testLINE</i>	X	X	X	Intensity of the test line

<i>TestOutput-Name</i>	X	X	X	
<i>TestTemperature</i>	X	X	X	Incubator temperature at test time
<i>TestType</i>			X	
<i>Time</i>	X	X	X	Time of the test
<i>TipBlue</i>	X	X		Displays value “Blue” when the test color is identified
<i>TipGreen</i>	X	X		Displays value “Green” when the test color is identified
<i>TipRed</i>	X	X		Displays value “Red” when the test color is identified
<i>UnitSN</i>	X	X	X	
<i>XLine</i>	X	X	X	Intensity of the X Line
<i>YLine</i>	X	X	X	Intensity of the Y Line
<i>ZLine</i>	X	X	X	Intensity of the Z Line

Trigger Object “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:

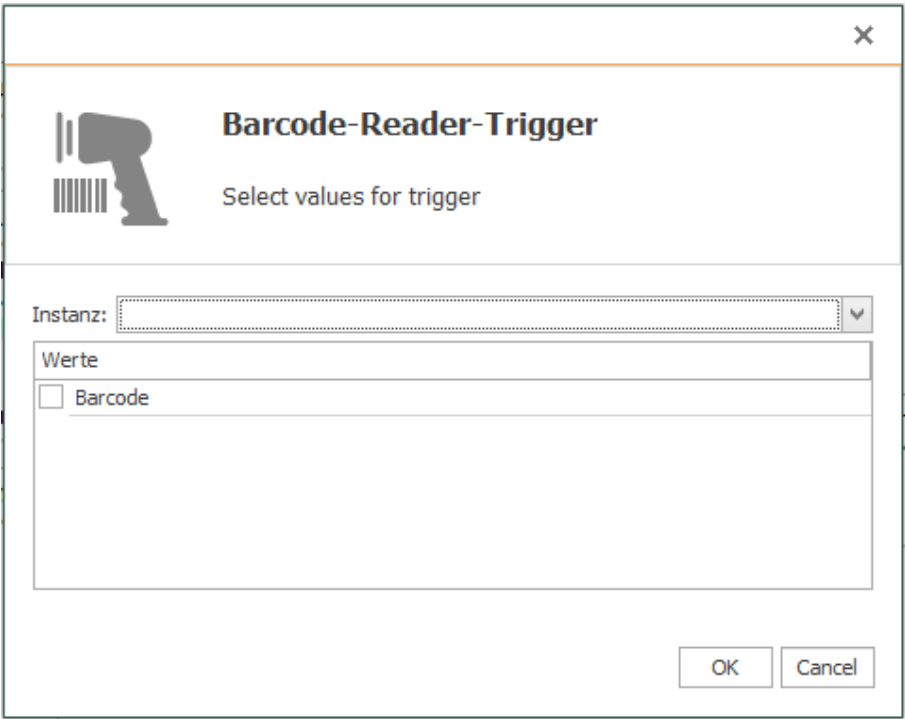
<i>Device type</i>	Type of charm reader.
<i>Port</i>	Port on this machine (with the OPC router) to which the Charm device sends its data.

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.

Trigger Object “Scanner”

Currently a trigger for the barcode reader is available.

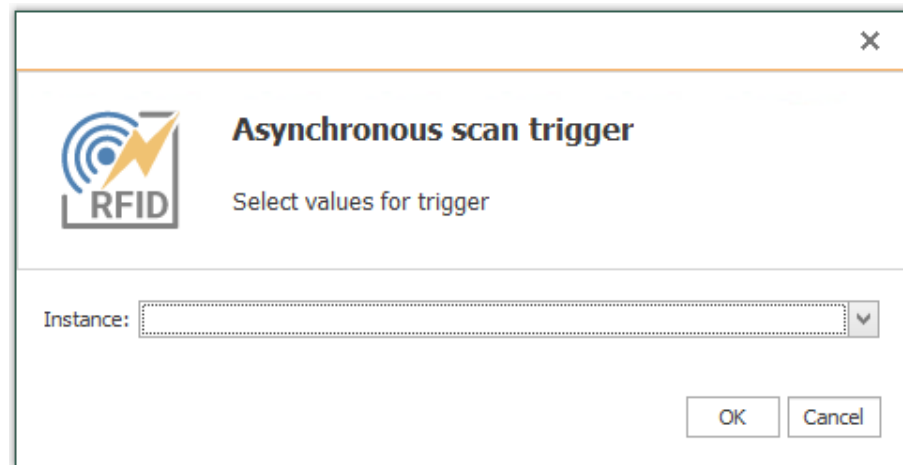


The individual properties are:

<i>Instance</i>	Select an instance from the drop-down menu for the barcode reader whose scan product you want to use.
<i>Values</i>	Select the value to be read, in this case “Barcode”.

Trigger Object “RFID Reader”

Scan asynchronous trigger of the RFID reader to trigger asynchronous scans. Select an instance from the drop-down menu after double-clicking.



Trigger Object “Rest-Trigger”

The Rest trigger provides residual API endpoints. You can define the incoming and outgoing parameters in the properties mask:

Rest trigger
Provides rest API endpoints

Rest API plug-in:

Endpoint:

Http Method:

Request parameters:

	Name	Value	Parameters type
*			

Response format:

Response parameters:

	Name	Value
*		

OK Cancel

The individual properties are:

<i>Rest API Plug-in</i>	Selection of a Rest connection from the drop-down menu.
<i>Endpoint</i>	Address of the OPC server (protocol: opc.tcp, http or https)
<i>Http Method</i>	Selecting the Http method from the drop-down menu.
<i>Request parameters</i>	Determine the parameters for the connection request.
<i>Response parameters</i>	Determine the parameters for the connection response.

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, which 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

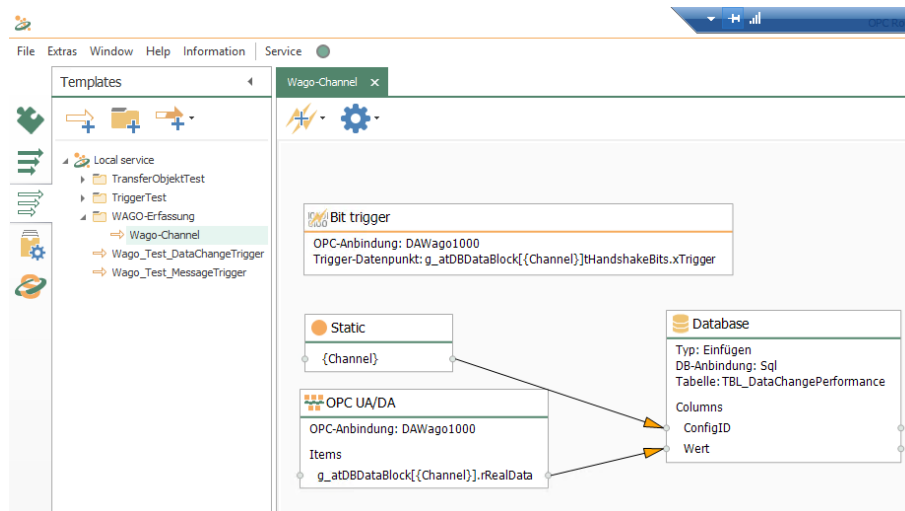
Note: To avoid errors in the configuration, the template variables are type-safe. A combination of multiple template variables or template variables with characters (e.g.. `g_atDBDataBlock[{Variable}]`) is only in type “string” possible.

in the following parts illustrate your possibilities.

2. 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.
3. 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.



1. In the snap-in "Templates", select the "New template" button or right-click on an existing entry and select there "New template".
2. Name it, e. g. like here "Wago Channel".
3. Define the connection properties for the template. The Properties will be applied for all instances.
4. 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.
5. Create a template variable Channel. To do that, click "Add" in the template variable dialouge and state *Name* ("Channel") and *Type* ("Integer").
6. Fill in the properties

<i>OPC Access Data</i>	<i><Name of the Access Data from the Plug-In Configuration></i>
<i>OPC-Items</i>	<i>g_atDBDataBlock[{Channel}].rRealData</i>

First insert the template variable via drag & drop. It is integrated into the transfer object with the correct syntax. Then add the static string.

7. Close the OPC transfer object.
8. 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.

<i>DB-Access Data</i>	<Name Of the Access Data From the Plug-In Configuration>
<i>Type</i>	Insert
<i>Table</i>	TBL_DataChangePerformance
<i>Used Columns</i>	ConfigID, Value

9. 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).
10. Connect the constant value to the ConfigID and the item with the value.
11. Project a Bit Trigger, to start the transfer. Activate either reset bit or insert a response bit, depending on the PLC program.

<i>OPC-Item</i>	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

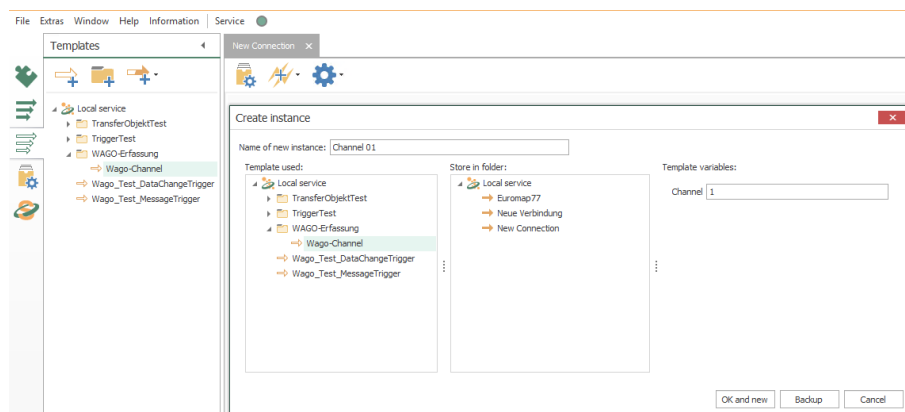
You can generate a template from a connection:

1. In the snap-in pane, select a connection.
2. Right-click to select “Create template from connection”.

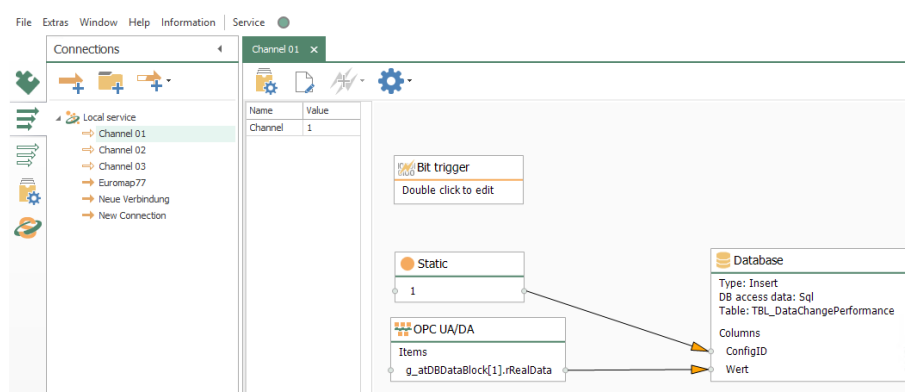
3. Give the new template a unique name and confirm with “OK”.
4. If you activate the “Open template directly” checkbox at the bottom of the window, the template will be opened after it has been created.

Creating Instances

To use the template for the data transfer, you have to create instances, thus actual connections.



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.



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 CSV 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 separated 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.

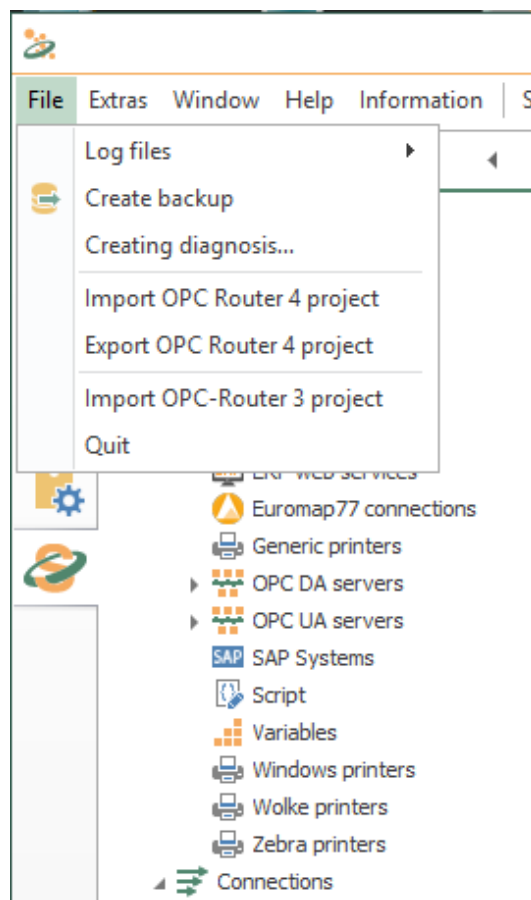
1. In the “Templates” or “Connections” snap-in, select the “Create Instance” drop-down button and then “Create Multiple Instances”.
2. Select the template to use
3. Select the destination folder.
4. Enter the values or open your CSV file.
5. Determine whether existing connections will be retained or over-written.

6. Select the “Create instance” button.
7. Make the new instances productive.

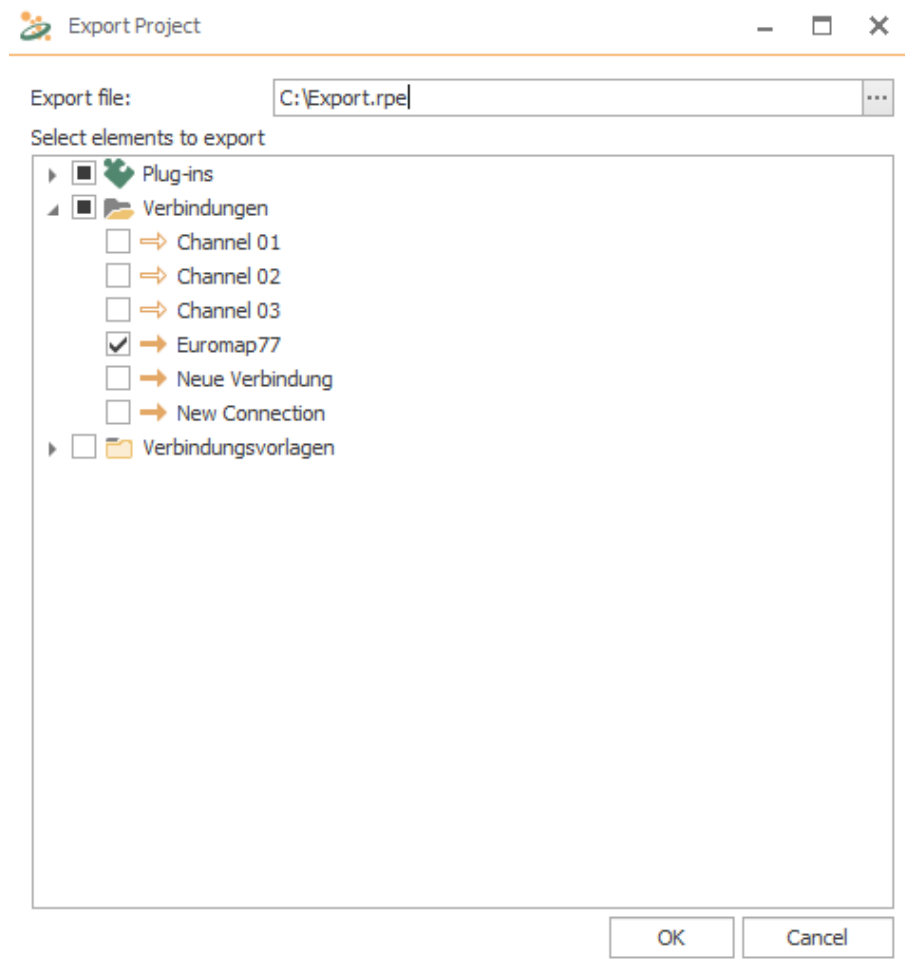
Note: If you change the template later, all instances will be adjusted.

Export OPC Router 4 project

You open the “Export OPC Router 4 project” function in the menu bar under “File”.



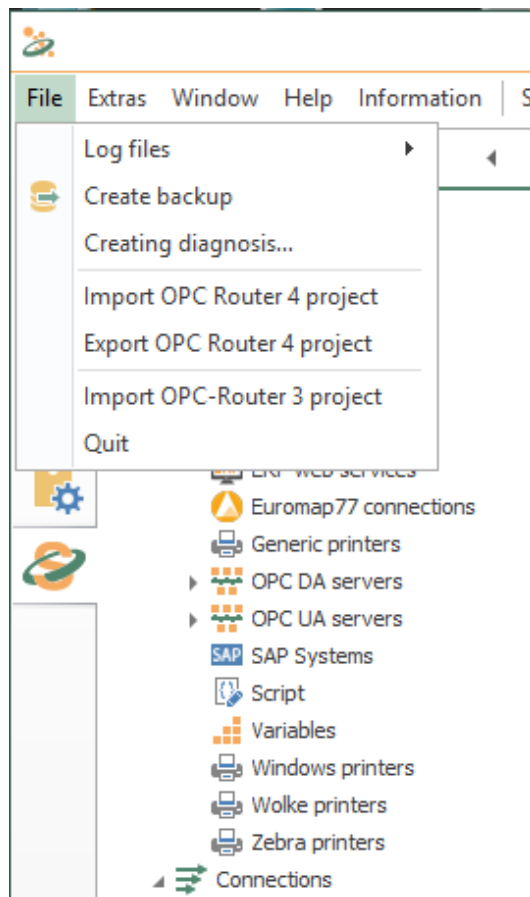
Select a file to export to and select all the connections and connections you want to export. If you select templates and connections, the corresponding structures are automatically selected as well.



Confirm your selection with "OK".

Import OPC Router 4 project

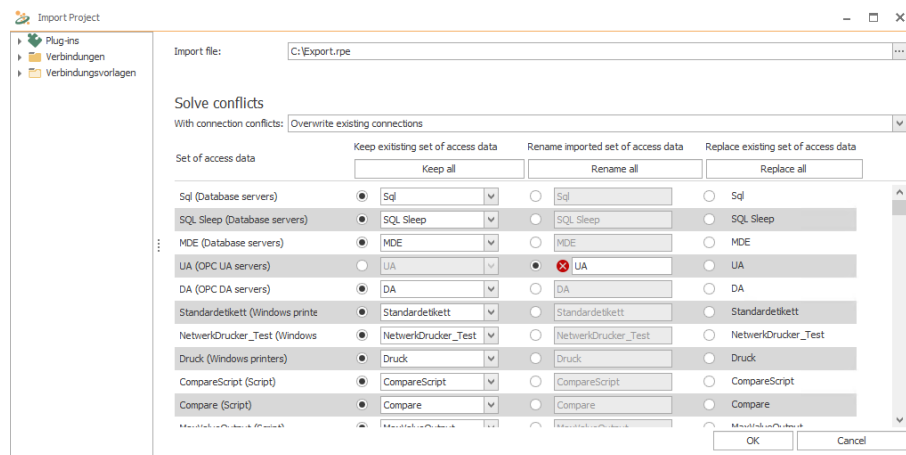
You open the “Import OPC Router 4 project” function in the menu bar under “File”.



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.



Confirm your selection with “OK”.