

Application Note



Accessing IEC Variables from OPC UA A750000

Version 1.1.0

© 2016 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Tel.: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: <http://www.wago.com>

Technical Support

Tel.: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

We wish to point out that the software and hardware terms, as well as the trademarks of companies used and/or mentioned in the present document are generally protected by trademark or patent.

Table of Contents

1	Notes about this Documentation.....	4
1.1	Copyright.....	4
1.2	Number Notation.....	4
1.3	Font Conventions	4
1.4	Symbols.....	5
2	Legal Bases	7
2.1	Subject to Change	7
2.2	Personnel Qualification	7
2.3	Limitation of Liability.....	8
3	Scope of Functions	9
3.1	Used devices and tools and their versions.....	9
4	Configuring variables for OPC UA access	9
4.1	Defining access to variables in the Symbol Configuration	12
4.2	Configuring access to variables using pragmas	13
5	Test	14
5.1	Writing a variable.....	17
6	Appendix – Publishing structure variables	18

1 Notes about this Documentation

1.1 Copyright

This documentation, including all figures and illustrations contained therein, is subject to copyright protection. Any use of this documentation that infringes upon the copyright provisions stipulated herein is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying), as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will entail the right of claims for damages.

1.2 Number Notation

Table 1: Number Notation

Number code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

1.3 Font Conventions

Table 2: Font Conventions

Font type	Indicates
<i>italic</i>	Names of paths and data files are marked in italic-type. e.g.: <i>C:\Programme\WAGO-I/O-CHECK</i>
Menu	Menu items are marked in bold letters. e.g.: Save
>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: File > New
Input	Designation of input or optional fields are marked in bold letters, e.g.: Start of measurement range
“Value”	Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under Start of measurement range .
[Button]	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: [Input]
[Key]	Keys are marked with bold letters in square brackets. e.g.: [F5]

1.4 Symbols

DANGER

Personal Injury!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

DANGER



Personal Injury Caused by Electric Current!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

WARNING

Personal Injury!

Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

CAUTION

Personal Injury!

Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE

Damage to Property!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

NOTICE



Damage to Property Caused by Electrostatic Discharge (ESD)!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

Note



Important Note!

Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.



Information

Additional Information:

Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

2 Legal Bases

2.1 Subject to Change

WAGO Kontakttechnik GmbH & Co. KG reserves the right to make any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from granting patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

2.2 Personnel Qualification

The use of the product described in this document is exclusively geared to specialists having qualifications in PLC programming, electrical specialists or persons instructed by electrical specialists who are also familiar with the appropriate current standards.

Moreover, the persons cited here must also be familiar with all of the products cited in this document, along with the operating instructions. They must also be capable of correctly predicting any hazards which may not arise until the products are combined.

WAGO Kontakttechnik GmbH & Co. KG assumes no liability resulting from improper action and damage to WAGO products and third-party products due to non-observance of the information contained in this document.

2.3 Limitation of Liability

This documentation describes the use of various hardware and software components in specific example applications. The components may represent products or parts of products from different manufacturers. The respective operating instructions from the manufacturers apply exclusively with regard to intended and safe use of the products. The manufacturers of the respective products are solely responsible for the contents of these instructions.

The sample applications described in this documentation represent concepts, that is, technically feasible application. Whether these concepts can actually be implemented depends on various boundary conditions. For example, different versions of the hardware or software components can require different handling than that described here. Therefore, the descriptions contained in this documentation do not form the basis for assertion of a certain product characteristic.

Responsibility for safe use of a specific software or hardware configuration lies with the party that produces or operates the configuration. This also applies when one of the concepts described in this document was used for implementation of the configuration.

WAGO Kontakttechnik GmbH & Co. KG is not liable for any actual implementation of the concepts.

3 Scope of Functions

This application note explains in some small steps how to publish variables defined in e!Cockpit for external access via OPC UA.

3.1 Used devices and tools and their versions

The following tools and devices were used during the creation of this application note:

	Description	Version
Devices:	PFC200 750-8202	V02.04.22(06)
Tools:	e!Cockpit	1.1.0.8106
	Unified Automation UaExpert	1.3.1 206

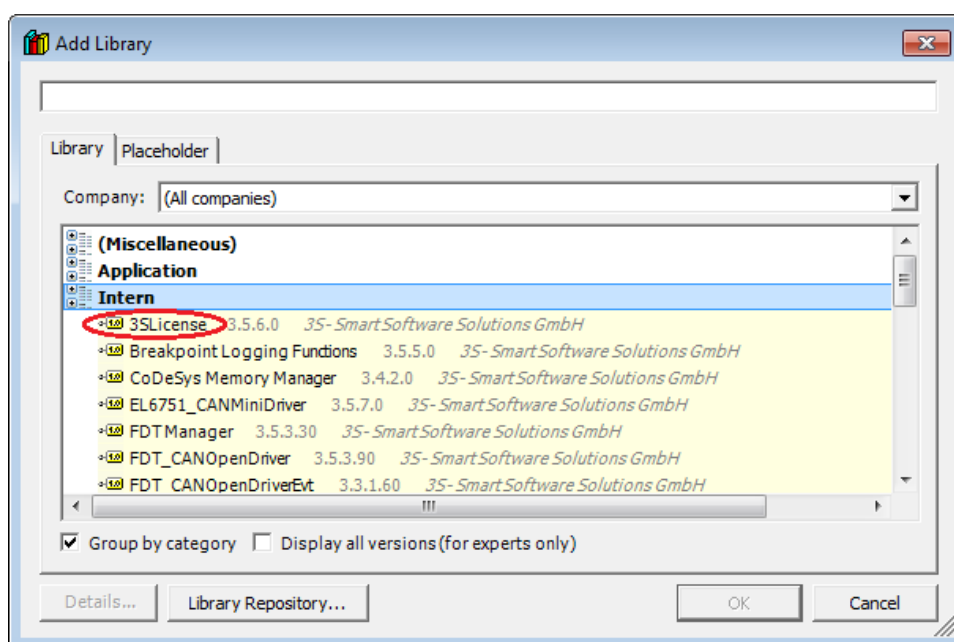
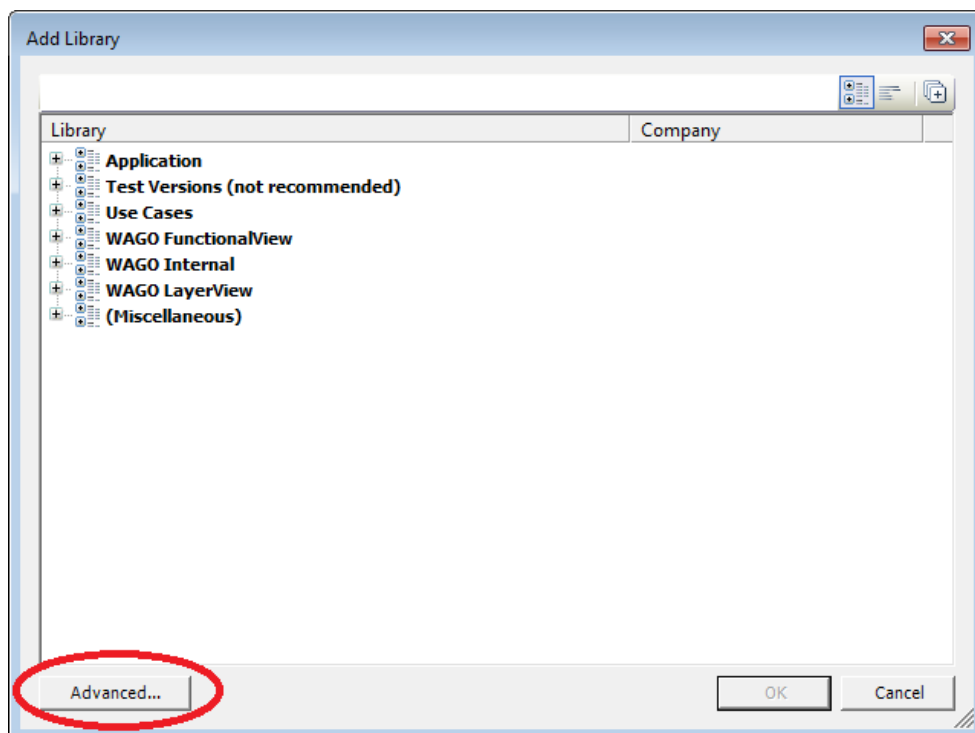
4 Configuring variables for OPC UA access

This application note assumes you have a running IEC application as a starting point.

Alternatively you are able to create a demo application by just creating a new project for your WAGO controller and adding a handful of test variables in PLC_PRG.

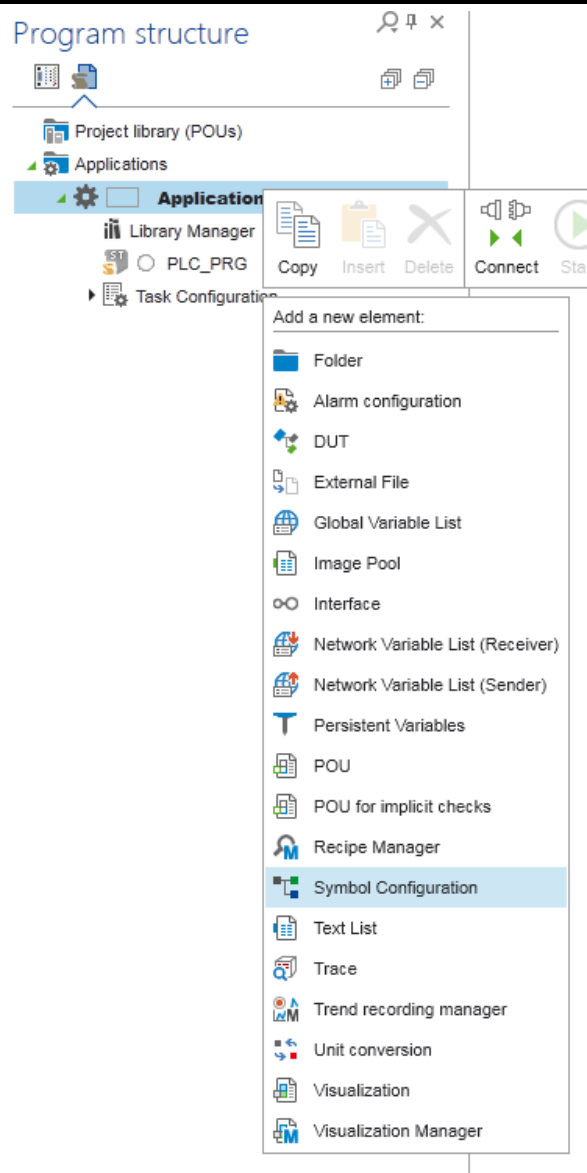
To configure existing IEC variables for OPC UA access you have to add the 'Symbol Configuration' to your application.

1. Add the library '3SLicense' to your application:¹

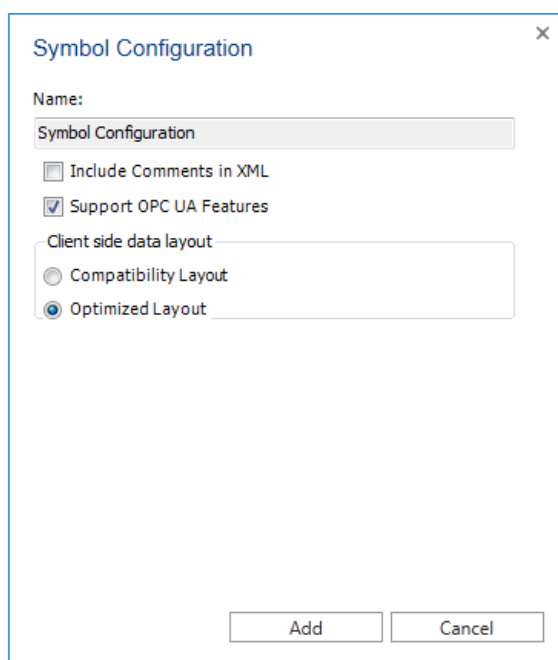


¹ This is a workaround necessary for the Release 6 firmware of the PFC200. Alternatively it is possible to add a WebVisu to activate a valid OPC UA license on the device. Without this the OPC UA server runs in demo mode on the PFC which expires after 2 hours runtime resulting in the shutdown of the server.

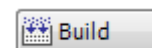
2. Open the context menu of your application and choose 'Symbol Configuration':



3. Choose 'Support OPC UA Features' from the opening dialogue and press 'Add':



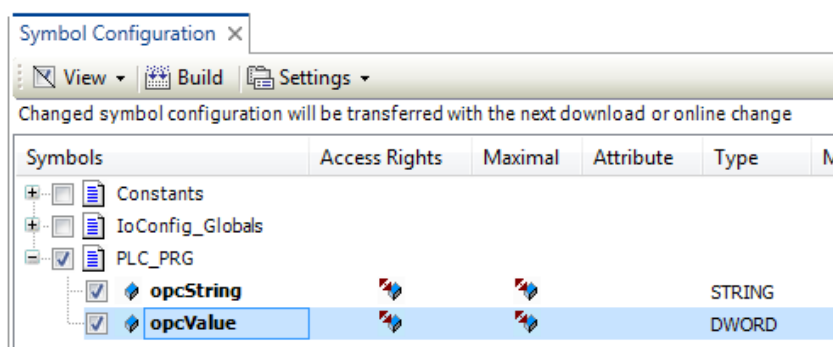
- Press the Button 'build' in the newly opened 'Symbol Configuration' window.



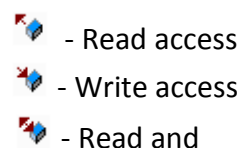
Else you won't be able to select your variables for OPC UA access.

4.1 Defining access to variables in the Symbol Configuration

- Choose the variables you want to publish.



- Choose the access rights for your variables by clicking on the icon in the 'Access Rights' column.



It may happen that not all access rights are available depending on the 'Maximal' access

Write access

rights which depend on the symbol type (e.g. constants are read only).

3. Build the application, download to the device and start the program.

4.2 Configuring access to variables using pragmas

Instead of defining the access rights to variables in the symbol configuration you can use the pragma `{attribute 'symbol' := 'read'}` to specify the access rights in the variable declaration.

```
PROGRAM PLC_PRG
VAR
  {attribute 'symbol' := 'read'}
  opc_var : DWORD;
  {attribute 'symbol' := 'readwrite'}
  opc_string: STRING := 'Hallo Welt';

  opc_struct: opc_struct;
END_VAR
```

By adding a pragma in the form `{attribute 'symbol' := 'accessright'}` you grant access via OPC UA to the following variable.

Possible access rights are:

- none
- read
- write
- readwrite - (default if no value is given)

To grant access to all declared variables of the object the pragma has to be placed in the first line of the declaration part. Learn more about that in the e!Cockpit online help page under *attribute 'symbol'*.

```
{attribute 'symbol' := 'readwrite'}
PROGRAM PLC_PRG
VAR
  {attribute 'symbol' := 'read'}
  opc_var : DWORD;
  opc_string: STRING := 'Hallo Welt';
  {attribute 'symbol' := 'none'}
  opc_struct: opc_struct;
END_VAR
```

In the above example the following rights are granted:

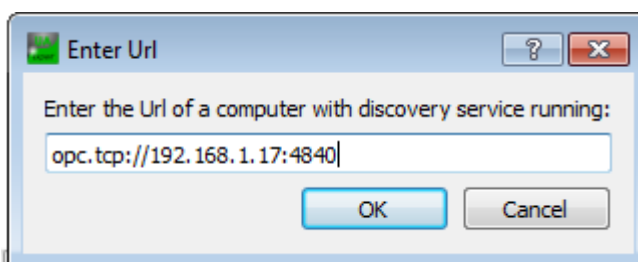
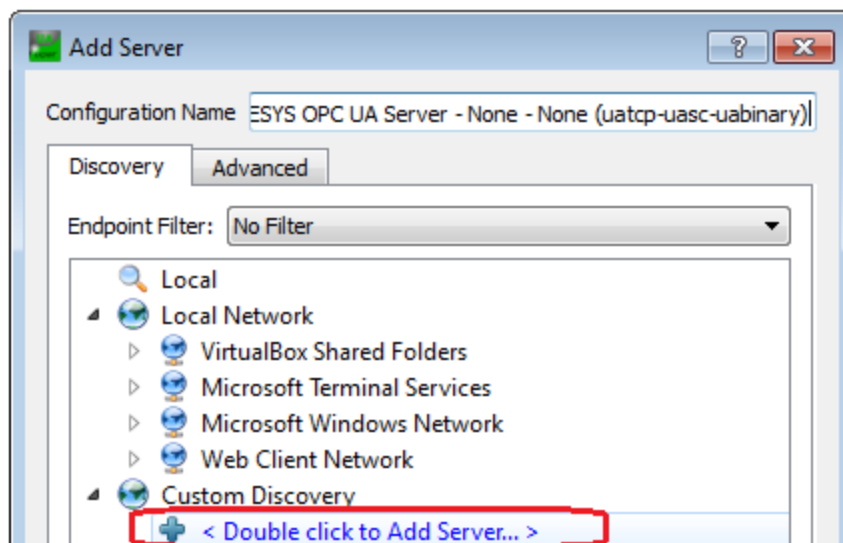
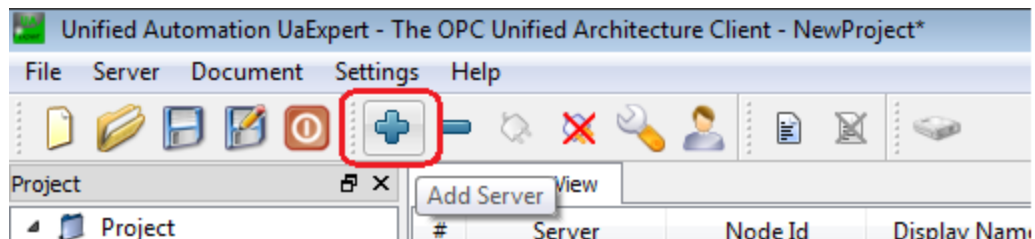
- opc_var – read

- opc_string – readwrite
- opc_struct – none

5 Test

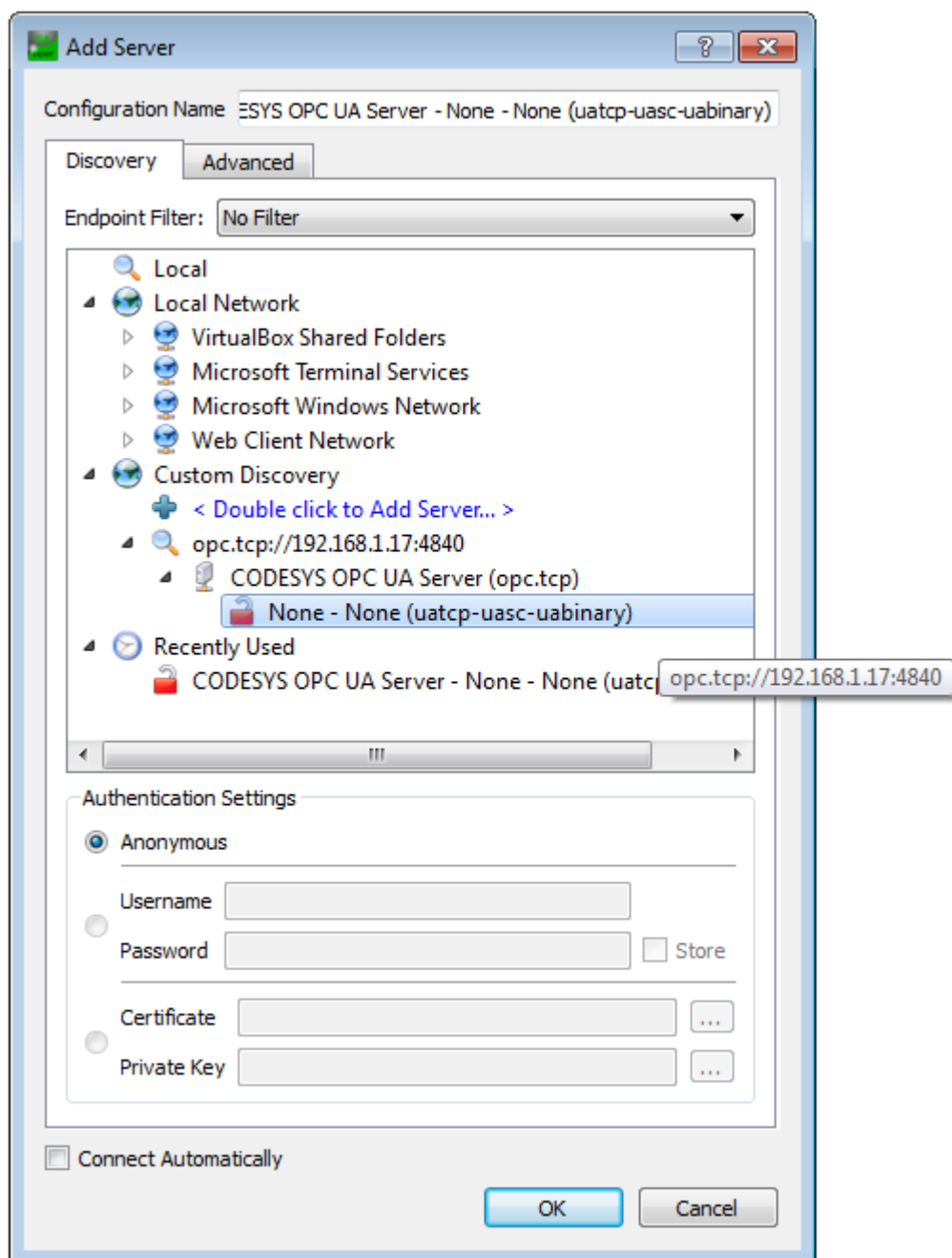
Use any OPC UA client to test the accessibility of the variables. For the creation of this application note we used the tool UaExpert which is available for free from [Unified Automation GmbH](#).

1. Start the UaExpert program.
2. Add a new server.

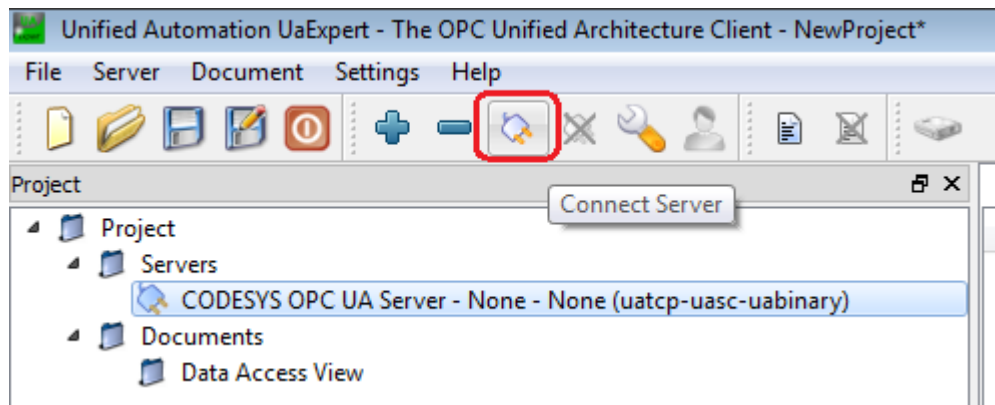


Use the ip address of your device!

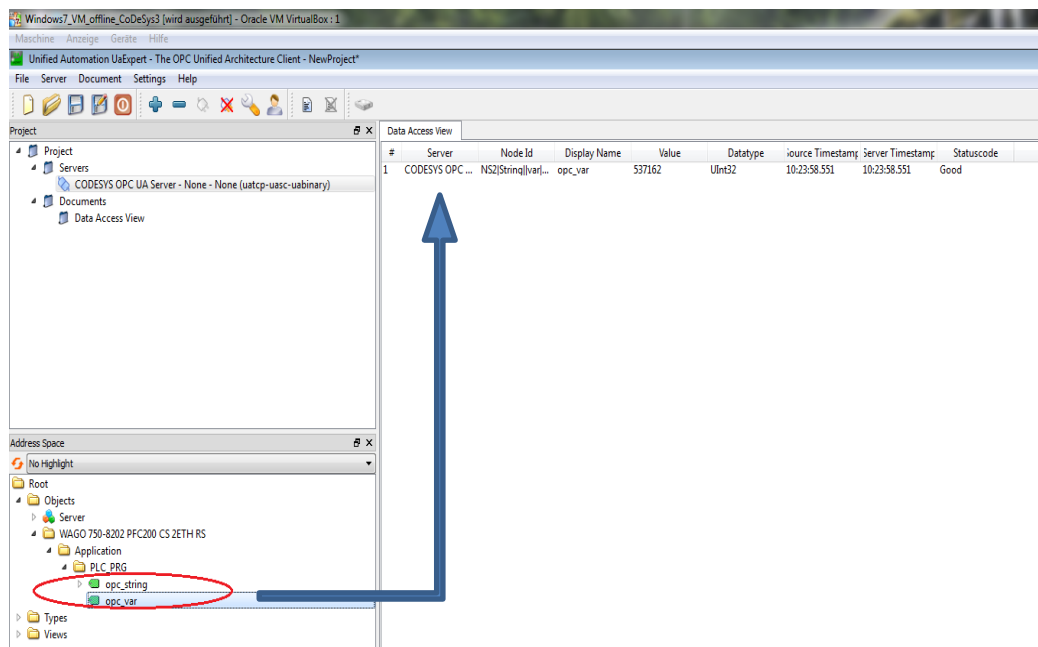
Choose the entry and press <OK>:



3. Connect the server:



4. Find your public variables and read them cyclically. Drag and drop the variable to the center field of the tool:



5.1 Writing a variable

If a published variable has activated write access you can test it by the following steps:

1. Configure the variable for cyclic access in the UaExpert as described above.
2. Double click on the value in the 'Value'-column of the variable.

#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	CODESYS OPC ...	NS2[String]Var...	opc_var	23350	UInt32	11:50:24.577	11:50:24.577	Good

3. Enter a new value and press <Enter>. The new value is written to the variable.

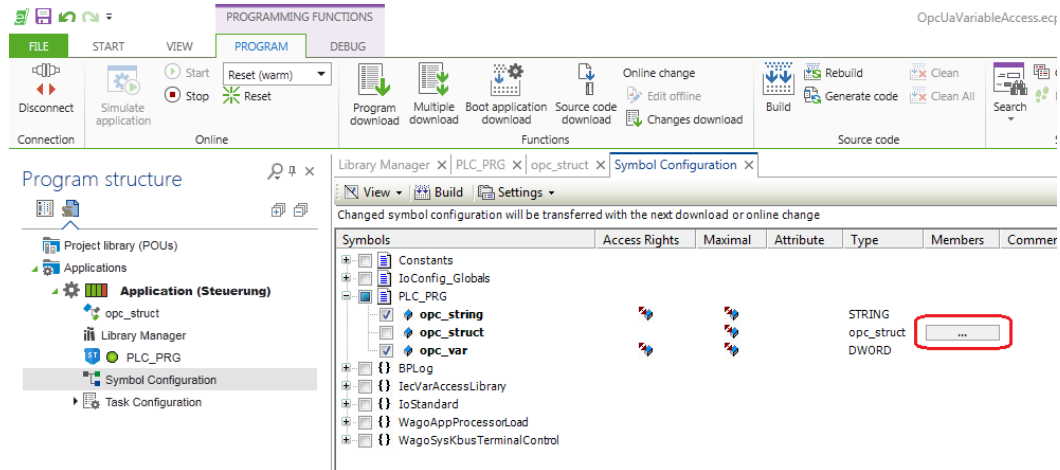
If the variable has no activated write access an error message is displayed in the log of the UaExpert.

Log			
Timestamp	Source	Server	Message
16.03.2016 10:47...	DA Plugin	CODESYS OPC UA Server - None - None (uatcp-uasc-uabinary)	Write to node 'NS2[String]Var[WAGO 750-8202 PFC200 CS 2ETH RS.Application.PLC_PRG.opc_var] failed [ret = BadNotWritable]
16.03.2016 10:46...	DA Plugin	CODESYS OPC UA Server - None - None (uatcp-uasc-uabinary)	Write to node 'NS2[String]Var[WAGO 750-8202 PFC200 CS 2ETH RS.Application.PLC_PRG.opc_var] failed [ret = BadNotWritable]
16.03.2016 10:46...	DA Plugin	CODESYS OPC UA Server - None - None (uatcp-uasc-uabinary)	Write to node 'NS2[String]Var[WAGO 750-8202 PFC200 CS 2ETH RS.Application.PLC_PRG.opc_var] failed [ret = BadNotWritable]

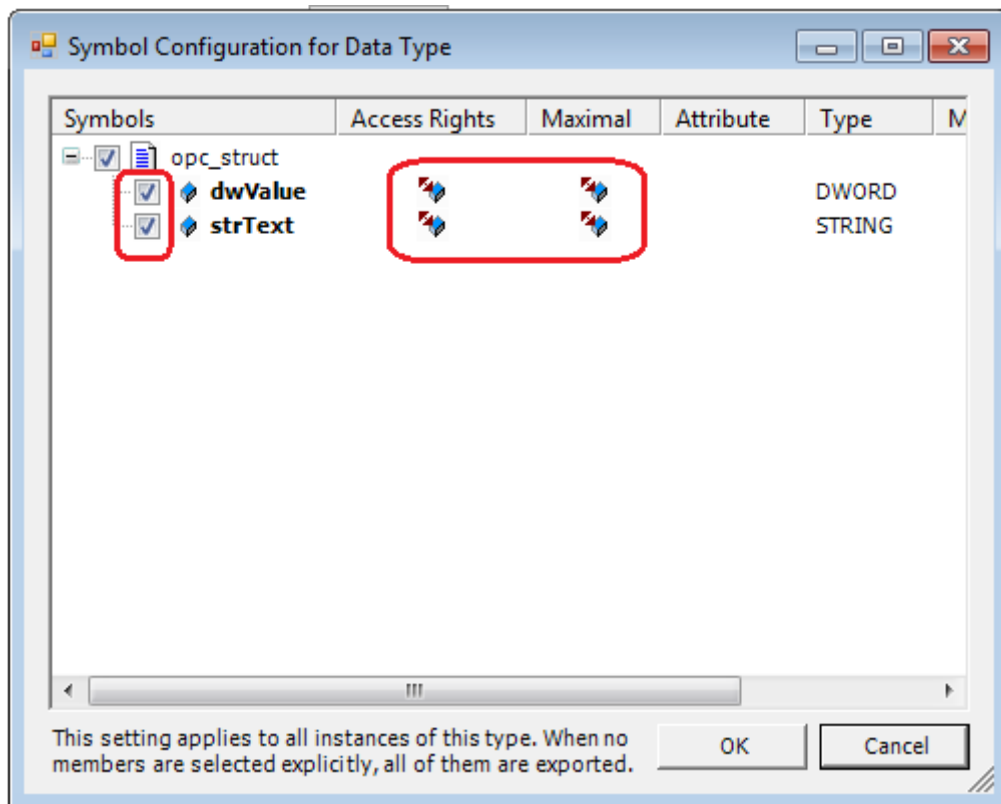
6 Appendix – Publishing structure variables

To publish variables which are members of a data structure additional steps are necessary in the symbol configuration.

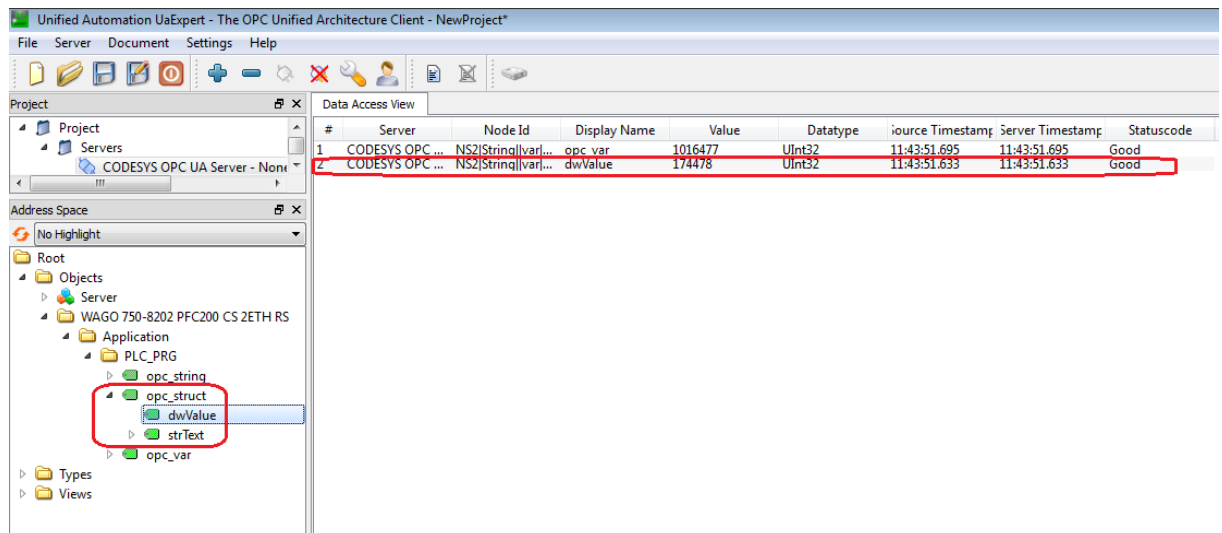
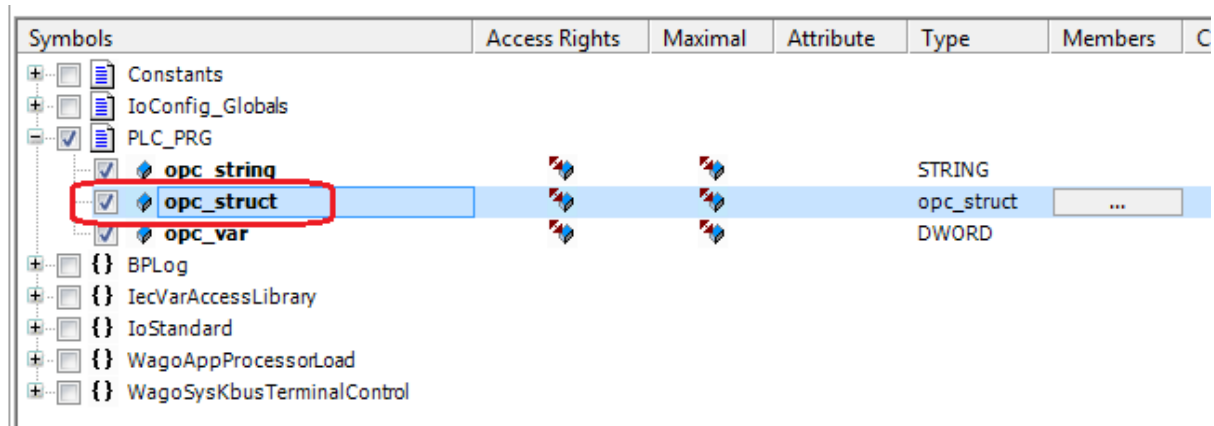
1. Define a struct type and declare a variable in your code.
2. Open the symbol configuration. The new struct variable is shown with an additional button in the 'Members'-column.



3. A dialog opens showing the member variables of the structure. Choose the variables to be published and their access rights.



4. Close the dialog and activate the structure variable. Now the chosen members of the struct are available via OPC UA.



WE! INNOVATE!

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Telefon: +49 (0) 571/8 87 – 0
Telefax: +49 (0) 571/8 87 – 1 69
E-Mail: info@wago.com
Internet: <http://www.wago.com>

