



# IBH OPC UA Editor Manual

Version 7.4.7

---

**IBHsoftec GmbH**  
**Turmstr. 77**  
**64760 Oberzent / Beerfelden**  
**Phone: +49 6068 3001**  
**Fax: +49 6068 3074**  
**info@ibhsoftec.com**  
**www.ibhsoftec.com**

**TTi Ingenieurbüro für**  
**Technologie Transfer**  
**Dipl. Ing. B. Peter Schulz-Heise**  
**Phone: +49 6061 3382**  
**Fax: +49 6061 71162**  
**tti@schulz-heise.com**  
**www.schulz-heise.com**

Windows® is a registered trademark of Microsoft® Corporation.  
Simatic® S5, Step® 5, Simatic® S7, Step® 7, S7-200®, S7-300®, S7-400®, S7-1200®, S7-1500® and GRAPH® 5 are  
registered trademarks of Siemens Corporation, Berlin and Munich.  
Image source: © Siemens AG 2001, All rights reserved.  
Product names are trademarks of their respective owners.

# Content

---

	PLC Sample Projects (PLC Programs) .....	VIII
<b>1</b>	<b>IBH OPC UA Editor .....</b>	<b>1-1</b>
1.1	Using the IBH OPC UA Editor .....	1-2
1.1.1	Calling the IBH OPC UA Editor .....	1-2
1.2	New server connection.....	1-3
1.2.1	Server Connection .....	1-4
	Server connection properties dialog box.....	1-4
	Name of the server connection .....	1-4
	Server address.....	1-4
	Show URL .....	1-5
	Select end point .....	1-5
	Security settings.....	1-5
	Inverse connection .....	1-6
	Login .....	1-6
	Session name .....	1-6
	Variable format.....	1-7
	Classic:.....	1-7
	Compact:.....	1-7
	S7-1500 Compatible: .....	1-8
	Compatible .....	1-8
1.3	Inserting a New control (PLC).....	1-9
	Connection IBH UA OPC UA server – Control .....	1-10
	Control Name (PLC).....	1-11
	Host name / IP address .....	1-11
	Protocol .....	1-11
	S7-200 TCP/IP, S7-1200 TCP/IP, S7-1500 TCP/IP .....	1-12
	Logo8 CPU with TCP / IP Configuration .....	1-12
	Test CPU-Online Connection.....	1-12
	Apply the New control dialog box settings .....	1-12
1.3.1	IBH Link UA - S7 CPU 300 / 400 connection via IBH Link S7++ .....	1-13
1.3.2	Project Window .....	1-15
	Name of the control.....	1-15
	Offline program assignment.....	1-15
	Program type.....	1-15
	Program path .....	1-15
	Station name .....	1-16
	Online connection .....	1-16
	Protocol .....	1-16
	Hostname / Address.....	1-16
	Target module position .....	1-16
	Rack number / slot number .....	1-16
1.4	Assign program .....	1-17
	Program selection .....	1-17
	Offline program assignment.....	1-18
	Special features when selecting TIA projects .....	1-18

<b>1.5</b>	<b>Define OPC tags .....</b>	<b>1-20</b>
	Add new variable (OPC tag) .....	1-21
	Define variable .....	1-22
	Name .....	1-22
	Data Type (OPC UA) / Data Type (S7) .....	1-22
	Address .....	1-23
	Select Variable Area.....	1-23
	DB number .....	1-23
	Byte address / bit address .....	1-23
	Access .....	1-23
	Number of field elements .....	1-23
	Maximum string length .....	1-23
	Comment .....	1-23
	Lower / upper limit .....	1-24
	Release field elements .....	1-24
	Node name .....	1-24
	Accept variable .....	1-24
<b>1.5.1</b>	<b>Change Variable (OPC tag) .....</b>	<b>1-24</b>
<b>1.6</b>	<b>Transfer configuration to the OPC UA server (IBH Link UA). .....</b>	<b>1-25</b>
<b>1.6.1</b>	<b>Configuration Export .....</b>	<b>1-26</b>
<b>1.6.2</b>	<b>Transfer the exported / edited XML file to the IBH Link UA. 1-27</b>	
	Select OPC Editor Project .....	1-27
	Loading the OPC Editor Project file .....	1-27
	Information taken from the OPC Editor .....	1-28
	Multiple CPUs in an OPC Editor project.....	1-28
	OPC editor information.....	1-28
<b>1.7</b>	<b>Configuration Import.....</b>	<b>1-29</b>
<b>1.8</b>	<b>Add external data .....</b>	<b>1-30</b>
<b>1.8.1</b>	<b>External data – context menu .....</b>	<b>1-31</b>
	Read external data from the server.....	1-31
	Transfer external data to the server .....	1-31
	Export / Import.....	1-31
	New external server connection – configuration .....	1-32
<b>1.8.2</b>	<b>Variable transfer .....</b>	<b>1-33</b>
<b>1.8.3</b>	<b>User-defined variables.....</b>	<b>1-34</b>
<b>1.8.4</b>	<b>MQTT configuration .....</b>	<b>1-35</b>
	Standard connection .....	1-36
	Connection to the Microsoft Azure IOT Hub .....	1-36
	Message in case of connection failure (Last Will Message) .....	1-36
<b>1.9</b>	<b>IBH OPC UA Editor Server window.....</b>	<b>1-36</b>
	Name of the server connection .....	1-37
	Server address .....	1-37
	Security settings .....	1-37
	Authentication settings .....	1-37
	Other settings .....	1-37
<b>1.10</b>	<b>Displayed Online OPC UA server information .....</b>	<b>1-37</b>
	Configuration Display .....	1-38
	Variable transfer .....	1-39
	UA Nodes display .....	1-39

<b>1.11</b>	<b>Certificates .....</b>	<b>1-40</b>
	Security setting via Select end point .....	1-40
<b>1.12</b>	<b>Modbus configuration .....</b>	<b>1-44</b>
<b>1.12.1</b>	<b>Add new Modbus device .....</b>	<b>1-44</b>
<b>1.12.2</b>	<b>Modbus TCP interface / Modbus variable .....</b>	<b>1-45</b>
	Modbus Functions – Partially – (Modbus Organization) .....	1-46
	Modbus data formats .....	1-46
	Example - Listing: Access to variables (fictitious Modbus devices manufacturer information) .....	1-47
	Modbus register addressing.....	1-47
<b>1.12.3</b>	<b>Defining Modbus variables in the IBH OPC UA Editors.....</b>	<b>1-48</b>
<b>1.13</b>	<b>Modbus variable – Properties dialog box .....</b>	<b>1-48</b>
	Modbus TCP / RTU interface .....	1-49
	Name (Variable) .....	1-49
	Unity-ID .....	1-49
	Type .....	1-49
<b>1.13.1</b>	<b>Addresses when accessing read / write .....</b>	<b>1-50</b>
	Data type UInt16 and Int16 .....	1-50
	Data types UInt32, Int32 and Float .....	1-50
	Data type Boolean.....	1-50
<b>1.13.2</b>	<b>Access Read Only .....</b>	<b>1-51</b>
	Read Discrete Input (read only – Bit access) .....	1-51
	Read Input Registers (Read only).....	1-51
	Read Input Registers (Read Only).....	1-51
	Read Input Registers (Read only).....	1-52
<b>1.13.3</b>	<b>Access Read and Write .....</b>	<b>1-52</b>
	Read Coils / Write Single Coil .....	1-52
	Read Coils.....	1-53
	Read Coils / Write Multiple Coils.....	1-53
	Read Holding Register / Write Single Register .....	1-53
	Read Holding Registers / Write Multiple Registers .....	1-54
	Read Holding Registers / Write Multiple Registers.....	1-54
	Read Holding Registers .....	1-55
<b>1.13.4</b>	<b>Analog-limits .....</b>	<b>1-55</b>
<b>1.13.5</b>	<b>History.....</b>	<b>1-56</b>
	OPC UA Node detection .....	1-56
<b>1.14</b>	<b>Transfer Modbus configuration to the OPC UA server (IBH Link UA).....</b>	<b>1-56</b>
<b>1.15</b>	<b>IBH OPC UA Editor Server Window.....</b>	<b>1-58</b>
<b>1.16</b>	<b>Mitsubishi configuration .....</b>	<b>1-59</b>
	Insert new CPU.....	1-60
<b>2</b>	<b>IBH OPC UA Editor - Configuration Examples .....</b>	<b>2-1</b>
<b>2.1</b>	<b>Example 1: CPU 416 S7 .....</b>	<b>2-3</b>
	Calling the IBH OPC UA Editor .....	2-3
<b>2.1.1</b>	<b>Server connection.....</b>	<b>2-3</b>
	Check connection to the IBH Link UA.....	2-4
<b>2.1.2</b>	<b>Insert new control .....</b>	<b>2-5</b>
<b>2.1.3</b>	<b>Program assignment .....</b>	<b>2-6</b>
	Define variables as OPC tags.....	2-7

	Select / search OPC tags .....	2-8
<b>2.1.4</b>	<b>Transfer configuration to the OPC UA server (IBH Link UA)</b>	<b>2-9</b>
	IBH Link UA - Siemens Slots project CPU 416 Counter .....	2-10
<b>2.1.5</b>	<b>Online OPC UA Server Information</b> .....	<b>2-10</b>
	Show under UA Nodes.....	2-11
<b>2.2</b>	<b>Example 2: S5 CPU 103U</b> .....	<b>2-12</b>
	Calling the IBH OPC UA Editor .....	2-12
<b>2.2.1</b>	<b>Inserting a New control (PLC)</b> .....	<b>2-13</b>
	New control dialog box .....	2-13
	Right project window .....	2-14
<b>2.2.2</b>	<b>Offline program assignment</b> .....	<b>2-15</b>
<b>2.2.3</b>	<b>Define variables as OPC tags</b> .....	<b>2-16</b>
	Add new variable (OPC tag) .....	2-17
	Change Variable (OPC tag) .....	2-17
<b>2.2.4</b>	<b>Transfer configuration to the OPC UA server (IBH Link UA)</b> .....	<b>2-17</b>
	IBH Link UA - Siemens Slots project S5_CPU_103U .....	2-18
	IBH Link UA – Browser window Diagnostics.....	2-19
<b>2.2.5</b>	<b>Online OPC UA Server Information</b> .....	<b>2-19</b>
	Show under UA Nodes.....	2-20
<b>2.2.6</b>	<b>Unified Automation UaExpert - The OPC Unified Architecture Client</b> .....	<b>2-21</b>
<b>2.3</b>	<b>Example 3: CPU 300 TIA16</b> .....	<b>2-22</b>
	Calling the IBH OPC UA Editor .....	2-22
<b>2.3.1</b>	<b>Inserting a New control (PLC)</b> .....	<b>2-23</b>
<b>2.3.2</b>	<b>IBH Link S7 ++ setting</b> .....	<b>2-25</b>
<b>2.3.3</b>	<b>Offline program assignment</b> .....	<b>2-26</b>
	Listed transferred project.....	2-27
<b>2.3.4</b>	<b>Define variables as OPC tags</b> .....	<b>2-27</b>
<b>2.3.5</b>	<b>Transfer configuration to the OPC UA server (IBH Link UA)</b> .....	<b>2-28</b>
	IBH Link UA - Siemens Slots - Project CPU 300 TIA 16.....	2-29
<b>2.3.6</b>	<b>Online OPC UA Server Information</b> .....	<b>2-30</b>
	View server.....	2-30
	Show under UA Nodes.....	2-30
<b>2.3.7</b>	<b>Unified Automation UaExpert - The OPC Unified Architecture Client</b> .....	<b>2-31</b>
<b>2.4</b>	<b>Example 4: CPU 1200 TIA 16</b> .....	<b>2-32</b>
	Calling the IBH OPC UA Editor .....	2-32
<b>2.4.1</b>	<b>Inserting a New control (PLC)</b> .....	<b>2-33</b>
<b>2.4.2</b>	<b>Offline program assignment</b> .....	<b>2-35</b>
	Listed transferred project.....	2-36
<b>2.4.3</b>	<b>Define variables as OPC tags</b> .....	<b>2-37</b>
<b>2.4.4</b>	<b>Transfer configuration to the OPC UA server (IBH Link UA)</b> .....	<b>2-37</b>
	IBH Link UA - Siemens Slots - Project CPU 1200 TIA 16.....	2-38
<b>2.4.5</b>	<b>Online OPC UA Server Information</b> .....	<b>2-39</b>
	View server.....	2-39
	Show under UA Nodes.....	2-40
<b>2.4.6</b>	<b>Unified Automation UaExpert - The OPC Unified Architecture Client</b> .....	<b>2-40</b>

<b>2.5</b>	<b>Example 5 - project: CPU 416 TIA 16 server - server .....</b>	<b>2-42</b>
2.5.1	Device configuration S7 project CPU 416 TIA 16 server – server .....	2-42
2.5.2	Calling the IBH OPC UA Editor .....	2-43
2.5.3	Inserting a New control (PLC).....	2-44
2.5.4	Offline program assignment .....	2-46
	Listed transferred program.....	2-47
2.5.5	Define variables as OPC tags .....	2-47
2.5.6	Start external OPC UA server .....	2-48
	External OPC UA server (air conditioning and heating systems).....	2-48
2.5.7	Adding a server .....	2-48
2.5.8	Option 1: Adding a server via the IBH Link UA.....	2-49
	Add external data .....	2-49
	OPC server adopted as external data.....	2-50
2.5.9	Option 2: Add server via the IBH OPC UA Editor .....	2-50
	New external server connection – configuration.....	2-51
	Select end point .....	2-51
2.5.10	Variable transfer - define source and target variables.....	2-52
	Variable links .....	2-53
2.5.11	Transferring the configuration to the OPC UA server (IBH Link UA).....	2-53
2.5.12	Displaying the links .....	2-54
2.5.13	OPC UA server information online.....	2-55
	Show variable transfer .....	2-56
	Status – Data block AirConditionerValue [DB 22].....	2-56
2.5.14	Status – AirConditionerValue [DB 22] – CPU 416 TIA 16 Server - Server.....	2-56
2.5.15	UaExpert - Data Access View .....	2-57
<b>2.6</b>	<b>Example 6 - CPU 1500 / CPU 1200 (Server – Server Connection).....</b>	<b>2-58</b>
2.6.1	Device configuration project CPU 1500 - CPU 1200.....	2-58
2.6.2	Exchanging variables .....	2-58
	PLC 1200 data block ReceivedValues [DB 10] .....	2-58
	PLC 1500 data block ReceivedValues [DB 10] .....	2-59
2.6.3	Calling the IBH OPC UA Editor .....	2-59
2.6.4	Inserting a New control (PLC).....	2-61
2.6.5	Program assignment .....	2-63
	Program assignment PLC 1500.....	2-63
	Transferred 1500 PLC program .....	2-64
	Program assignment PLC 1200.....	2-64
	Transferred 1200 PLC program .....	2-65
2.6.6	Define variables as OPC tags .....	2-66
	Define PLC_1500 variables as OPC tags.....	2-66
	Define PLC_1200 variables as OPC tags.....	2-66
2.6.7	Add external data .....	2-67
2.6.8	Variable transfer - define source and target variables.....	2-67
	Variable links: source – PLC_1500 target – PLC_1200 .....	2-67
	Variable links: source – PLC_1200 target – PLC_1500 .....	2-68
2.6.9	Transfer configuration to the OPC UA server (IBH Link UA). .....	2-69
2.6.10	IBH Link UA web browser window .....	2-70

	IBH Link UA - Siemens Slots – PLC_1500 / PLC_1200 .....	2-70
	IBH Link UA - browser window OPC Client.....	2-71
<b>2.6.11</b>	<b>Online OPC UA Server Information .....</b>	<b>2-71</b>
	IBH Link UA – PLC_1500 – PLC_1200.....	2-71
	Variable transfer – PLC_1500 – PLC_1200.....	2-71
<b>2.6.12</b>	<b>UaExpert - Data Access View.....</b>	<b>2-72</b>
<b>2.6.13</b>	<b>PLC-Status .....</b>	<b>2-73</b>
	PLC_1200 data block <i>CounterData [DB5]</i> .....	2-73
	PLC_1200 data block <i>ReceivedValues [DB10]</i> .....	2-73
	PLC_1500 data block <i>CounterData [DB5]</i> .....	2-73
	PLC_1500 data block <i>ReceivedValues [DB10]</i> .....	2-73
<b>2.7</b>	<b>Example 7 – Exchanging data between several</b>	
	<b>S7 / S5 CPUs.....</b>	<b>2-74</b>
<b>2.7.1</b>	<b>CPU 312 connection via IBH Link S7++ .....</b>	<b>2-74</b>
	Configuration of the IBH Link S7 ++.....	2-74
<b>2.7.2</b>	<b>PLC programs.....</b>	<b>2-76</b>
	IP addresses / PLC programs of the devices used.....	2-76
<b>2.7.3</b>	<b>Calling the IBH OPC UA Editor .....</b>	<b>2-76</b>
<b>2.7.4</b>	<b>Inserting New controls (PLCs).....</b>	<b>2-78</b>
	Dialog box New controller - CPU 416 master .....	2-78
	Dialog box New controller - S7 PLC 1 CPU 312.....	2-79
	Dialog box New controller - S7 PLC 2 CPU 312.....	2-79
	Dialog box New controller – S5 PLC 3 CPU 103.....	2-80
	Dialog box New controller – S5 PLC 3 CPU 103.....	2-80
<b>2.7.5</b>	<b>Program assignment.....</b>	<b>2-81</b>
	Select Program dialog box .....	2-81
	Assigned PLC program .....	2-82
<b>2.7.6</b>	<b>Define variables as OPC tags .....</b>	<b>2-82</b>
	CPU 416 master – OPC tags.....	2-82
	S7 PLC 1 CPU 312 – OPC tags.....	2-83
	S7 PLC 2 CPU 312 – OPC tags.....	2-83
	S5 PLC 3 CPU 103 – OPC tags.....	2-84
	S5 PLC 4 CPU 941 – OPC tags.....	2-84
<b>2.7.7</b>	<b>Add external data .....</b>	<b>2-84</b>
<b>2.7.8</b>	<b>Variable transfer – define source and destination</b>	
	<b>OPC tags .....</b>	<b>2-85</b>
	Variable connection.....	2-85
	Established connection .....	2-86
	Defined Connections.....	2-87
<b>2.7.9</b>	<b>Check connections .....</b>	<b>2-90</b>
<b>2.7.10</b>	<b>Transfer configuration to the OPC UA server</b>	
	<b>(IBH Link UA). .....</b>	<b>2-91</b>
<b>2.7.11</b>	<b>IBH Link UA browser windows .....</b>	<b>2-91</b>
<b>2.7.12</b>	<b>Online OPC UA Server Information .....</b>	<b>2-92</b>
	Show variable transfer.....	2-93
<b>2.7.13</b>	<b>Unified Automation UaExpert - The OPC Unified</b>	
	<b>Architecture Client .....</b>	<b>2-95</b>
<b>2.8</b>	<b>Example 8: Connecting two S7 CPU 300 via an</b>	
	<b>IBH Link S7++ .....</b>	<b>2-96</b>
	IBH Link S7++ .....	2-96
	Connecting the IBH Link UA with 2x S7 PLC CPU 312.....	2-96
	Configuration of the IBH Link S7 ++.....	2-96



	IP / MPI addresses / PLC programs of the devices .....	2-98
<b>2.8.1</b>	<b>PLC programs .....</b>	<b>2-98</b>
	Calling the IBH OPC UA Editor .....	2-98
<b>2.8.2</b>	<b>Inserting a New controls (PLC).....</b>	<b>2-100</b>
	New control dialog box – PLC 1 S7-CPU 1 .....	2-100
	New control dialog box – PLC 2 S7-CPU 2 .....	2-101
<b>2.8.3</b>	<b>IBH Link S7 ++ setting .....</b>	<b>2-102</b>
<b>2.8.4</b>	<b>Program assignment .....</b>	<b>2-102</b>
	Select Program dialog box.....	2-103
	Assigned PLC program.....	2-103
<b>2.8.5</b>	<b>Define variables as OPC tags .....</b>	<b>2-103</b>
	OPC tags of the controller –PLC 1 S7-CPU 1 .....	2-103
	OPC tags of the controller –PLC 2 S7-CPU 2 .....	2-104
<b>2.8.6</b>	<b>Add external data .....</b>	<b>2-104</b>
<b>2.8.7</b>	<b>Variable transfer – define source and destination</b>	
	<b>OPC tags .....</b>	<b>2-105</b>
	Variable connection.....	2-105
	Established connection .....	2-106
	Defined Connections.....	2-107
<b>2.8.8</b>	<b>Transfer configuration to the OPC UA server</b>	
	<b>(IBH Link UA) .....</b>	<b>2-108</b>
<b>2.8.9</b>	<b>IBH Link UA browser windows .....</b>	<b>2-109</b>
<b>2.8.10</b>	<b>Online OPC UA Server Information.....</b>	<b>2-110</b>
	Show variable transfer .....	2-110
<b>2.8.11</b>	<b>Unified Automation UaExpert - The OPC Unified Architecture</b>	
	<b>Client .....</b>	<b>2-112</b>
<b>2.9</b>	<b>Modbus connection - examples .....</b>	<b>2-112</b>
	SIMATIC Manager PLC project ModBus_Test .....	2-113
	Organization block OB1 .....	2-113
<b>2.9.1</b>	<b>Calling the IBH OPC UA Editor .....</b>	<b>2-114</b>
<b>2.9.2</b>	<b>Modbus configuration .....</b>	<b>2-115</b>
<b>2.9.3</b>	<b>Add new Modbus device .....</b>	<b>2-115</b>
<b>2.9.4</b>	<b>Define Modbus variable in the IBH OPC UA editor.....</b>	<b>2-116</b>
	Defines OPC variables (OPC tags).....	2-120
<b>2.9.5</b>	<b>Transfer Modbus configuration to the IBH Link UA. ....</b>	<b>2-121</b>
<b>2.9.6</b>	<b>IBH Link UA browser window .....</b>	<b>2-121</b>
	Browser-Fenster-ModBus .....	2-121
	Browser window diagnostics .....	2-121
<b>2.9.7</b>	<b>IBH OPC UA Editor Server Information - Online.....</b>	<b>2-122</b>
<b>2.9.8</b>	<b>UaExpert – The OPC Unified Architecture Client .....</b>	<b>2-122</b>

## PLC Sample Projects (PLC Programs)

<p><b>OPC UA – Editor Examples (SPS Projects)</b></p>	<p><b>Folder with Projects</b>          Example 1 - CPU 416 Counter S7          Example 2 - S5 CPU 103U          Example 3 - CPU 300 TIA 16          Example 4 - CPU 1200 TIA 16          Example 5 - Server - Server connection TIA 16          Example 6 - CPU 1500 CPU 1200 connection          Example 7 - Multi CPUs S7          Example 8 - 2x CPU 312 S7          Example 9 - modbus connection\ ModBus_Test S7</p>
<p><b>OPC UA – Editor Examples and exports</b></p>	<p>Example 1 - CPU 416 S7.opu          Example 1 - CPU 416 S7.opx          Example 2 - S5 CPU 103U.opu          Example 2 - S5 CPU 103U.opx          Example 3 - CPU 300 TIA 16.opu          Example 3 - CPU 300 TIA 16.opx          Example 4 - CPU 1200 TIA 16.opu          Example 4 - CPU 1200 TIA 16.opx          Example 5 - Server - Server connection TIA 16.opu          Example 5 - Server - Server connection TIA 16.opx          Example 6 - CPU 1500 - CPU 1200 connection.opu          Example 6 - CPU 1500 - CPU 1200 connection.opx          Example 7 - Multi CPUs S7.opu          Example 7 - Multi CPUs S7.opx          Example 8 - 2x CPU 312 S7.opu          Example 8 - 2x CPU 312 S7.opx          Example 9 - Modbus connection.opu          Example 9 - Modbus connection.opx</p>
<p><b>All examples</b></p>	<p><b>OPC UA Editor - Examples.zip</b></p>

# 1 IBH OPC UA Editor

---

The **IBH OPC UA Editor** is used to create the configuration of the PLC - IBH Link UA connection without opening the PLC-project. This also applies if neither change can be made in the project or in the PLC.

In combination with the IBH Link UA the **IBH OPC UA Editor** is available as a download free of charge.

After installation, the **IBH OPC UA Editor** can be used for the following PLC connections:

- S7 300/400 TCP/IP
- S7 200 TCP/IP
- S7 1200 TCP/IP
- S7 1500 TCP/IP
- S7 300/400 with IBH Link S7++ via the S7 TCP/IP protocol
- SIMATIC S5 with IBH Link S5++ via the S7 TCP/IP protocol
- LOGO 8
- SINUMERIK 840D
- Modbus connection
- Mitsubishi Controls / Robots
- Rockwell Controls

Symbolic variables and data from data blocks can be transferred as OPC variables (OPC tags) to PLC projects (PLC programs):

- TIA 13/14/15/16 Project
- STEP 7 (Simatic Manager)
- S5W Projects (IBHsoftec S5/S7 for Windows® Programming System)
- SIMATIC S5
- NC VAR Selector
- LOGOSoft Comfort Software

OPC variables (OPC tags) can be edited (add, remove, assign access rights) using the **IBH OPC UA Editor**.

Configurations generated in the **IBH OPC UA Editor** can be transferred directly to the **IBH Link UA** or saved as **XML** files.

After transferring the defined OPC variables (OPC tags) to the **IBH Link UA**, the online status of the defined OPC variables (OPC tags) can be observed with the IBH OPC UA editor. The **UA nodes** of the online connected **IBH Link UA** server are also displayed.

**Note:**

To assign OPC variable (OPC Tag), operands must be defined as symbols and data blocks must have a symbolic name.

**Note:**

Projects generated in the **IBH OPC UA Editor** can be directly transmitted into the **IBH Link UA**.

Likewise, the created projects can be exported and imported as an **XML** file (\*.opc).

An exported **XML** file optional edited in the **IBH OPC UA Editor**, can be transferred into the **IBH Link UA** (OPC Slot) using the web-browser.

## 1.1 Using the IBH OPC UA Editor

When installing the **IBH OPC UA Editor**, an icon is created on the desktop to start the program.

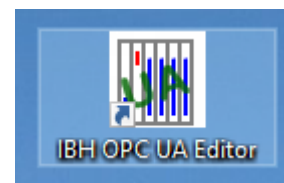
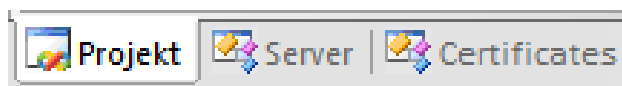
**Note:**

This manual describes the **IBH OPC UA Editor version 7.4.7**. Later versions may contain additional functions.

### 1.1.1 Calling the IBH OPC UA Editor

Double-click the **IBH OPC UA Editor** icon to open the program window.

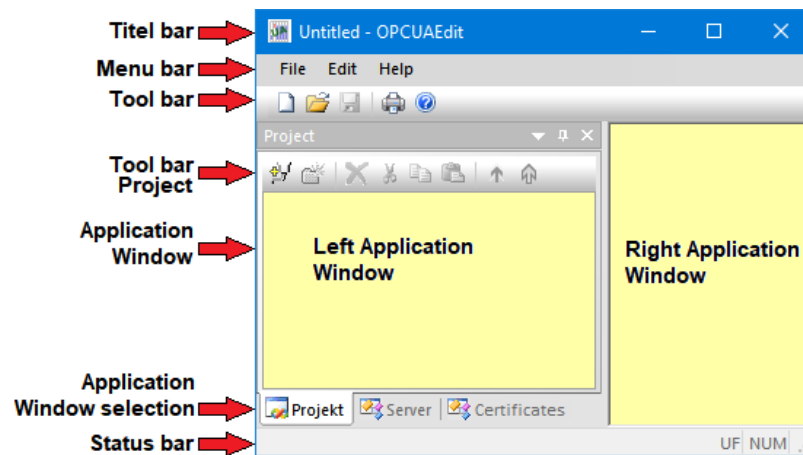
The **IBH OPC UA Editor** has three application windows, which are opened via the tabs **Project**, **Server** or **Certificates** are displayed.



By clicking the tab **Certificates**, the own certificate or the server certificate are displayed. Server certificates can be read from files.

For a better distinction, the background of the windows is displayed with different colors (selectable).

The work area is divided (area windows left / right).



### Title bar

The name of the active project is specified **OPCUAEdit**.

### Menu bar

The menu bar is the same for the **Project**, **Server** and **Certificates** application windows.

### Toolbar

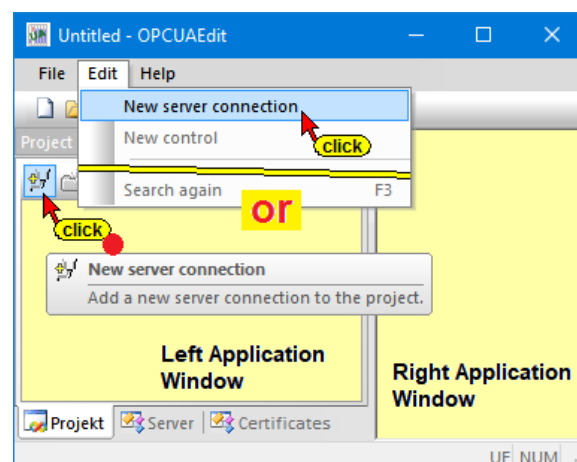
The toolbar is the same for the **Project**, **Server** and **Certificates** application windows.

### Project toolbar

Only the **Project window** has an additional toolbar.

## 1.2 New server connection

The **New Server Connection command** from the **Edit** menu or clicking the icon opens the **New Server Connection** dialog box.



## 1.2.1 Server Connection

To establish a connection to an OPC UA server, the connection data must be specified. The **New server connection** dialog box makes it easier to specify the connection data.

### Note:

The connection data specified by the **New server connection** dialog box will be displayed in the right part of the Project window after completion.

Some of the connection data can be changed at any time in the right part of the project window.

## Server connection properties dialog box

The fields for the general settings for the connection to an *OPC UA* server must be filled out.

### Name of the server connection

The name is freely selectable.

Name of the server connection: IBH Link UA

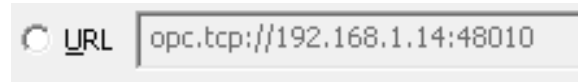
### Server address

If the **IBH Link UA port** to be used for connection is in a network with a **DHCP server**, the actual host name may be entered.

If there is no DNS server, the absolute IP address of the **IBH Link UA** (192.168.1.14) must be entered with the port (48010).

## Show URL

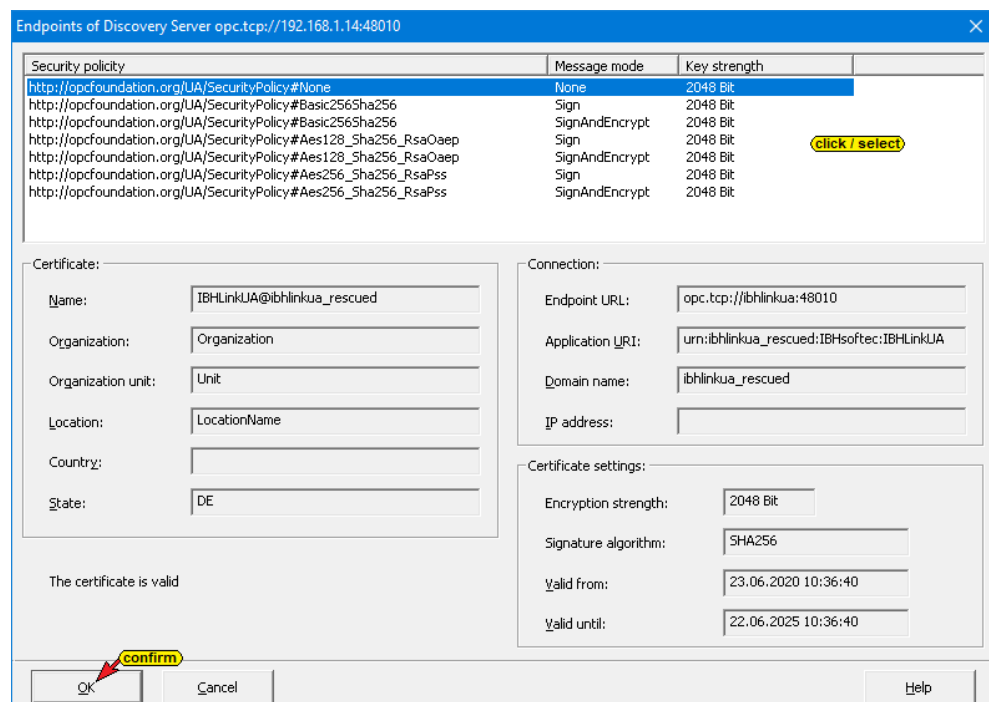
The URL of the selected OPC server is displayed.



## Select end point

By clicking the **Select end point...** button, a connection to the specified OPC UA server is established. If the connection is successful, possible transfer data encryptions are displayed in the opened dialog box.

In addition, the existing certificate in the OPC UA server with its settings and the connection path to the OPC UA server are displayed.



## Security settings

In this field the security procedure and the message mode can be selected.



If a security procedure is selected, certificates must be exchanged between the **IBH OPC Editor** and the **OPC UA Server** (IBH Link UA).

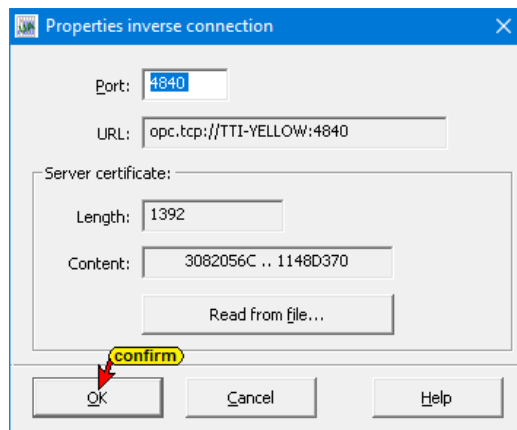
As the message mode signature (Sign) as well as signature and encryption (Sign and Encrypt) are available.

## Inverse connection

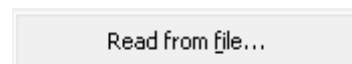
An inverse server connection (reverse connection) can be set up if the server is in a better protected area behind a firewall than the client.



Clicking the **Properties...** button opens a dialog box for entering the endpoint URL of the OPC UA client. This simplifies the configuration of the firewall. Of course, the client must support incoming server connections.

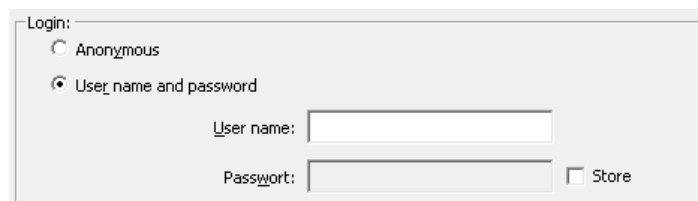


The specified server certificate can be replaced by an existing one.



## Login

In this field, the **User name** and the associated password can be specified. The proposed registration mode is anonymous.



## Session name

The name of the session is freely selectable and can be left blank.





## Variable format

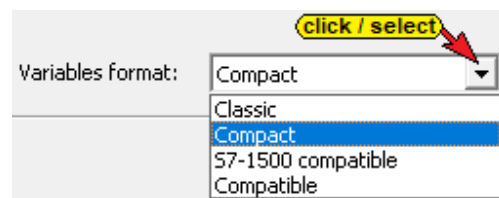
The representation of the variables can be adjusted. The **IBH OPC UA Editor** offers four options for acceptance. With this selection, the limitation of the OPC UA specification of not allowing dots in variable names can be avoided.

### Note:

The programming systems S7 SIMATIC Manager and the TIA Portal allows dots in variable names (e.g. **Switch 7.1**).

**The OPC UA specification does not allow dots in variable names.**

**The selection of the variable format must match the system setting in the IBH Link UA browser window !**



### Classic:

The IBH Link UA software accepts only variable names that comply with the OPC UA specification. Dots in variable names must be removed from the symbol tables (S7 SIMATIC Manager or TIA Portal) before transfer to the IBH Link UA. The identifiers ".GlobalVars", ".Programs" ".Generic" are added to the name of a variable.

The IBH UA Editor accepts variable names with a dot. Variable names containing a dot are put in quotation marks by the IBH Link UA software during transmission.

Identifier	CPU 416.CPU 416-3 PN/DP.Programs.Data block.Var_INT
Identifier	CPU 416.CPU 416-3 PN/DP.GlobalVars.Bit_Var
Identifier	IBH Link UA.CPU414.Generic.OFF_2

### Compact:

Dots in variable names must be removed from the names (S7 SIMATIC Manager) prior transfer to the IBH Link UA. Variable names with dots are accepted in the TIA Portal.

The IBH UA Editor accepts variable names with dots.

Variable names containing dots are put in quotation marks by the IBH Link UA software during transmission.

The identifiers **".GlobalVars"** **".Programs"** **".Generic"** are omitted in the variable names. If such a name occurs in a variable name, it is placed in quotation marks.

If **Compact** is marked, the identifier of a variable is shorter than marking **Classic**.

Identifier	CPU 416.CPU 416-3 PN/DP."Generic"
Identifier	CPU 416.CPU 416-3 PN/DP."Bit_Var"
Identifier	CPU 416.CPU 416-3 PN/DP."Data block"."Var_INT"
Identifier	CPU 416.CPU 416-3 PN/DP."Data block"."Programs"
Identifier	S7-400-Station_1.CPU 416."On_5.3"
Identifier	IBH Link UA.CPU414."OFF.2"

### S7-1500 Compatible:

Dots in variable names must be removed from the names (S7 SIMATIC Manager) before transfer to the IBH Link UA.

Dots in variable names are permitted in the TIA Portal programming system. Variable names not corresponding to the S7-1500 format are put in quotation marks by the IBH Link UA software during the transfer and thus brought to the name format of the S7-1500.

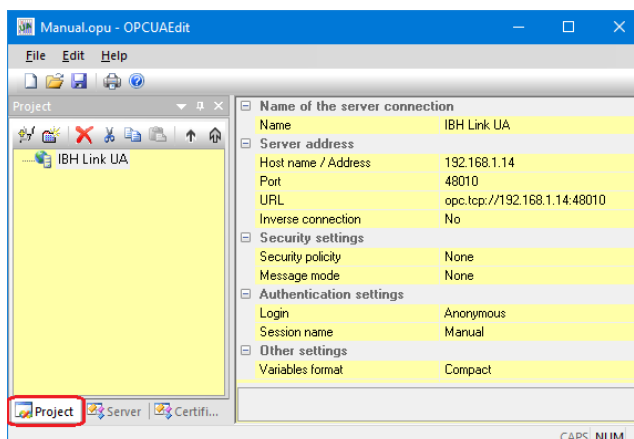
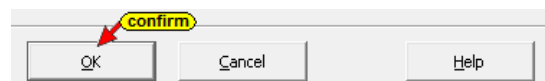
The identifiers **".GlobalVars"** **".Programs"** **".Generic"** are omitted in the variable names.

Identifier	CPU 416.CPU 416-3 PN/DP."GlobalVars"
Identifier	CPU 416.CPU 416-3 PN/DP.Data block.Var_Bool
Identifier	CPU 416.CPU 416-3 PN/DP.Lamp
Identifier	IBH Link UA.CPU414."OFF_47.B"
Identifier	IBH Link UA.CPU414.OFF_2
Identifier	IBH Link UA.CPU414.Bit_Var

### Compatible

Mark **Compatible** if data block variables (OPC tags – defined in the **IBH OPC UA Editor / Variable Transfer**) "GlobalVars" in the target name have. Only required with older IBH OPC UA Editor version (2017...2019).

To apply the settings, click **OK**. The **New server connection** dialog box closes.



The right part of the project window displays the specified settings for the connection to the **OPC UA server**.

**Note:**

In one project several **OPC servers** (IBH Link UA) can be collected.

A right-clicking on the **Server icon** (IBH Link UA) opens the context menu.

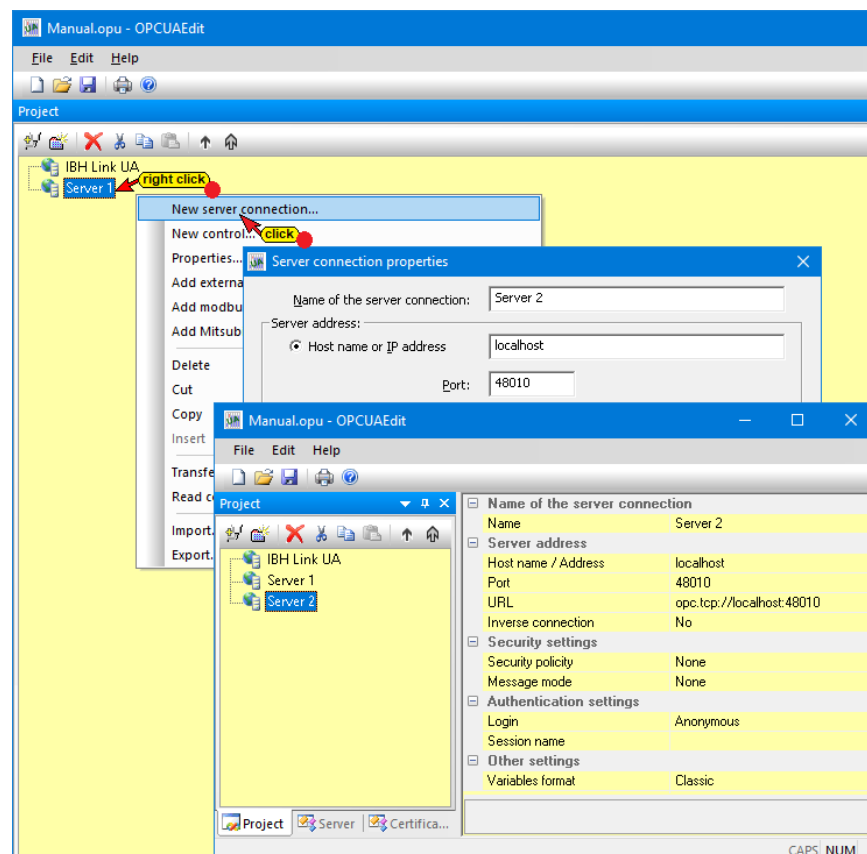


The **New server connection** command, additional OPC servers can be integrated into the project. The **New server connection command** is also available on the Edit menu.

For each new server connection, the **Server connection properties** dialog box opens. The settings for the connection to the OPC UA server must be completed accordingly.

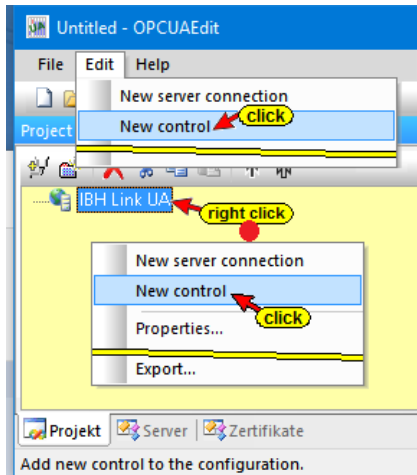
If a server is highlighted in the left part of the project window, the connection settings are displayed in the right part of the window.

### Connection to the further OPC UA servers



## 1.3 Inserting a New control (PLC)

The **New Control** command from the context menu opens the **New control** dialog box. The **New control** command is also available in the **Edit** menu.



The **New control** dialog box sets the connection settings to the controller (PLC, CPU, etc.) to be connected to the OPC UA Server.

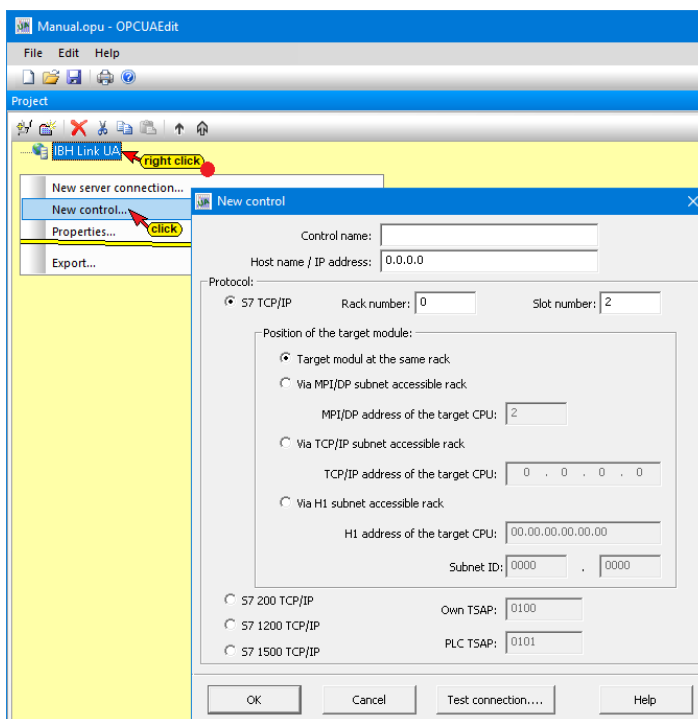
**Note:**  
A project may contain multiple controls in one OPC UA server.

A right-clicking on the **Server icon** (IBH Link UA) opens the context menu.

The command **New control** may assign additional controllers to the OPC UA server. The **New control** dialog box opens for each new control. The connection settings to the controller (PLC, CPU, etc.) to be connected to the OPC UA server are specified here.



### Connection IBH UA OPC UA server – Control



## Control Name (PLC)

The name is freely selectable.  
(*CPU416\_Tank\_Level*).

Control name: CPU 416 Tank Level

## Host name / IP address

The host name or the  
IP address of the device  
(CPU) being online connected with the OPC UA server is defined.

Host name / IP address: 192.168.1.10

## Protocol

Protocol:  
 S7 TCP/IP    Rack number: 0    Slot number: 2

If **S7 TCP/IP** is selected, the Rack number, Slot number and Position of the target module must be specified. This setting is used to connect S7 300 / 400 CPUs having a usable ethernet port (TCP/IP).

## Position of the target module

### Target module at the same rack

If the target module is in the same rack,  
no further information is required.

Target modul at the same rack

### Via MPI/DP subnet accessible rack

If the target module is in a rack accessible via MPI / DP subnetwork, the **MPI / DP address of the target CPU** and the **Subnet ID** must be specified.

Via MPI/DP subnet accessible rack  
 MPI/DP address of the target CPU: 2  
 Subnet ID: 0000 . 0000

### Via TCP/IP subnet accessible rack

If the target module is in a rack accessible via the TCP/IP subnet, the **TCP/IP address of the target CPU** and the **Subnet ID** must be specified.

Via TCP/IP subnet accessible rack  
 TCP/IP address of the target CPU: 192 . 168 . 1 . 55  
 Subnet ID: 0000 . 0000

### Via H1 subnet accessible rack

If the target module is in a rack accessible via an H1 subnet, the **H1 address of the target CPU** and the **Subnet ID** must be specified.

Via H1 subnet accessible rack  
 H1 address of the target CPU: 00.1B.21.25.32.82  
 Subnet ID: 002E . 0005

## S7-200 TCP/IP, S7-1200 TCP/IP, S7-1500 TCP/IP

S7-200 / 1200 / or 1500 CPU having a TCP / IP port, are selected directly. If the connection is made via **ISO on TCP**, the **Own TSAP** and the **PLC TSAP** of the (CPU) must be specified.

S7 200 TCP/IP      Own TSAP: 0000  
 S7 1200 TCP/IP  
 S7 1500 TCP/IP      PLC TSAP: 0000

## Logo8 CPU with TCP / IP Configuration

If a **Logo8** controller is to be connected to the OPC UA server, the following settings must be made in the New Control dialog box:

- S7-200 TCP/ P must be selected as the control type
- **0200** must be entered as the **PLC TSAP** of the CPU
- The default **Own TSAP** is irrelevant.

Control name: Logo 8  
 Host name / IP address: 192.168.1.88  
 Protocol:

S7 200 TCP/IP      Own TSAP: 0000  
 S7 1200 TCP/IP      PLC TSAP: 0200  
 S7 1500 TCP/IP

## Test CPU-Online Connection

If the **New control** dialog box is completed, the online connection to the CPU can be tested. The command **Test connection** is building up the connection to the CPU. Information about the successful connection is displayed.

Test connection....

IBH OPC UA Editor  
 The connection to the PLC CPU 416 has been successfully tested.  
 confirm → OK

## Apply the New control dialog box settings

To apply the settings, click **OK**. The **New control** dialog box closes.

New control  
 S7 1500 TCP/IP  
 confirm → OK    Cancel    Test connection....    Help

### 1.3.1 IBH Link UA - S7 CPU 300 / 400 connection via IBH Link S7++

To establish a connection to a S7 300 / 400 CPU via IBH Link S7 ++, the settings must be according to the screenshots.

**Note:**



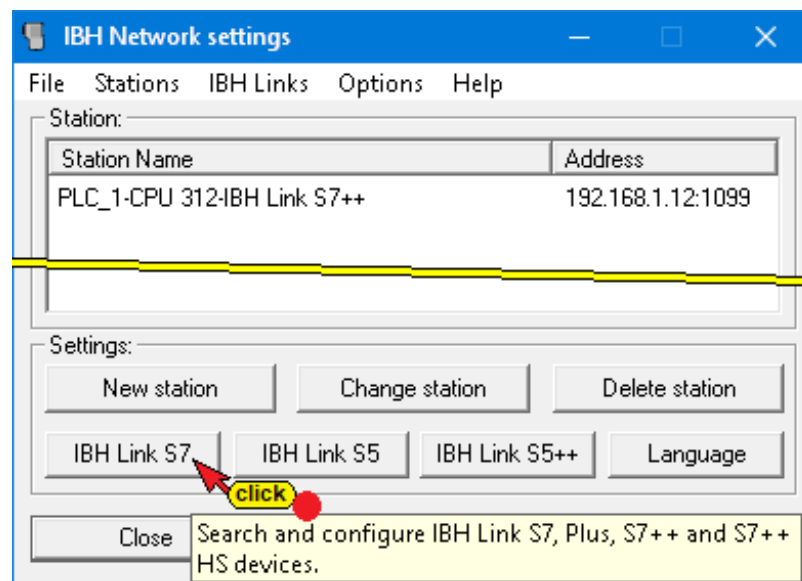
To address an **S7 CPU 300 / 400** via the **IBH Link S7++**, the routing option (dialog box **IBHLink settings / Network tab**) **Configuration with NetPro** must be deactivated (Apply permanently).

**This applies to all S7 300/400 CPUs with IBH Link S7++ connection.**

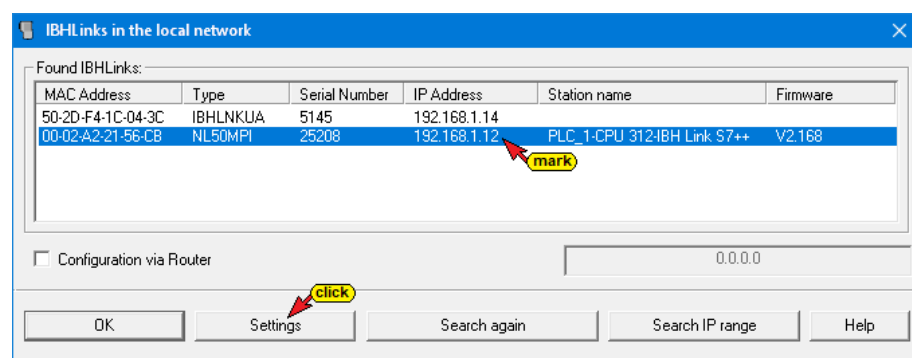
Open the **IBH Link S7 ++** configuration software **IBH Network settings**.



#### Dialog box IBH Network settings

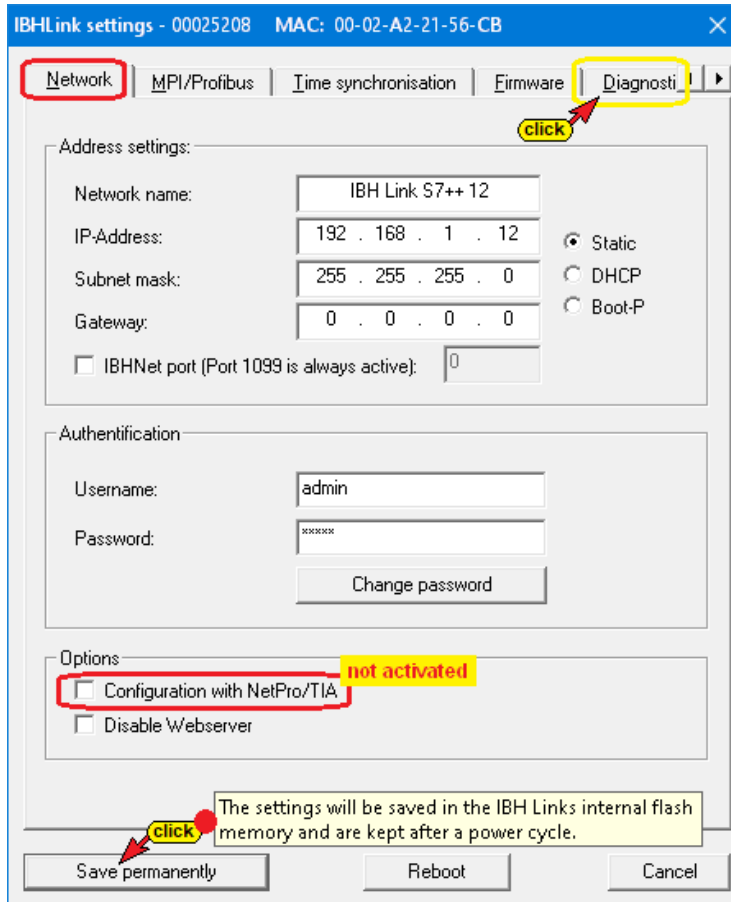


Click **IBH Link S7** to open the **IBH Links in the local network** dialog box.



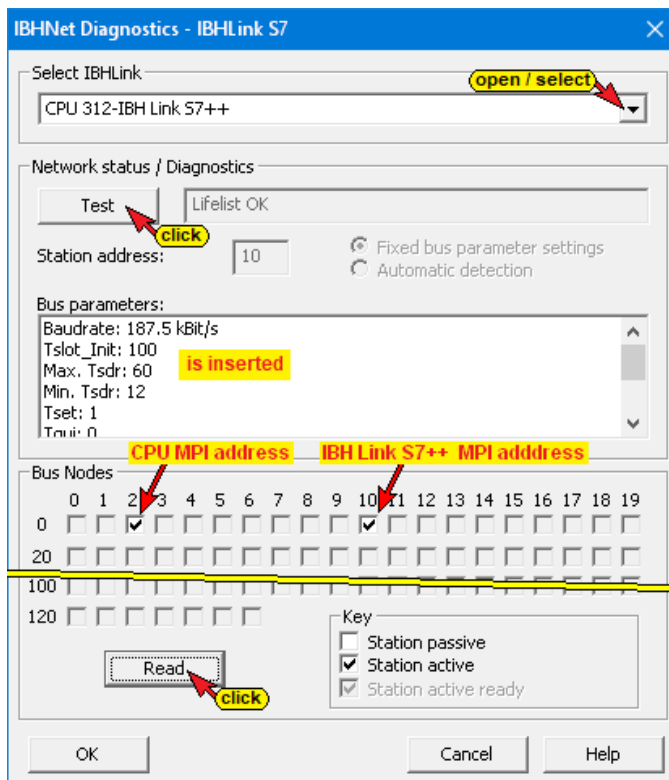
Mark the **IBH Link S7++** connected to the S7 CPU 300 / 400 and click **Settings**.

The **IBHLink settings** dialog box opens.



Deactivating the routing option **Configuration with NetPro**

Open the **Diagnostics tab** click **Test** and **Read**. The MPI addresses of the **IBH Link S7++** and the connected S7 CPU are shown.

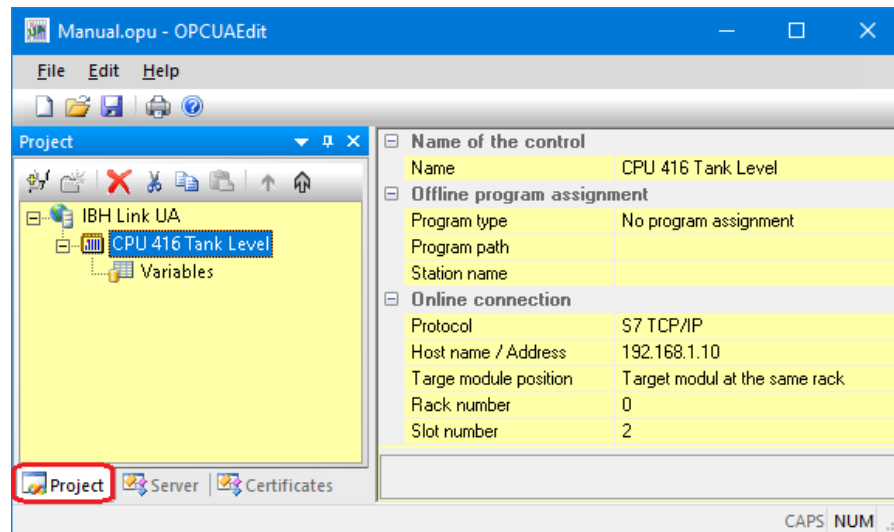


If several **IBH Link S7++** are in the local network, select the correct **IBH Link S7++** connected to the **S7 CPU**.



### 1.3.2 Project Window

The right part of the project window displays the specified **CPU** connection settings.



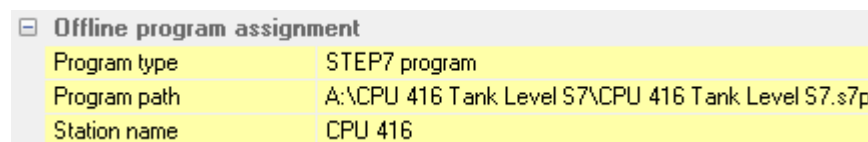
#### Name of the control

The name was specified in the **New control** dialog box.



#### Offline program assignment

If a PLC program is assigned to the selected CPU information are displayed.

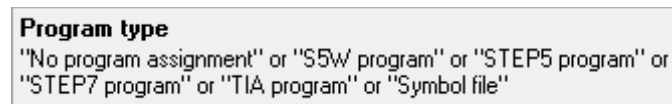


#### Program type

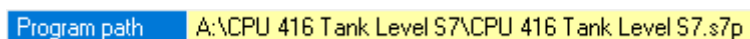


The following program types can be assigned to a CPU:

- No program assignment
- S5W program
- STEP 5 program
- STEP 7 program
- TIA program
- Icon file



#### Program path



Once a program has been assigned, the program path is displayed.

## Station name

Station name CPU 416

The name of the station, the device or the program name of the assigned program is displayed.

## Online connection

The connection to the CPU is defined in the **New control** dialog box.

Online connection	
Protocol	S7 TCP/IP
Host name / Address	192.168.1.10
Target module position	Target modul at the same rack
Rack number	0
Slot number	2

## Protocol

Protocol S7 TCP/IP

The **Protocol** for the online connection to the PLC, S7 TCP/IP, S7-200 TCP/IP, S7-1200 TCP/IP or S7-1500 TCP/IP is displayed.

## Hostname / Address

Host name / Address 192.168.1.10

The host name or the IP address of the online connection CPU / OPC UA Server is displayed.

## Target module position

Target module position Target modul at the same rack

In the New control dialog box, the following target module positions may be defined:

- Target module on the same rack
- Rack accessible via MPI / DP subnet
- Rack accessible via TCP / IP subnet
- Rack accessible via H1 subnet

## Rack number / slot number

These two numbers define the **MPI address** of the target module.

This may be necessary if several CPUs are connected via the MPI bus or the connection is made via an IBH Net S7 ++.

Online connection	
Protocol	S7 TCP/IP
Host name / Address	192.168.1.10
Target module position	Target modul at the same rack
Rack number	0
Slot number	2

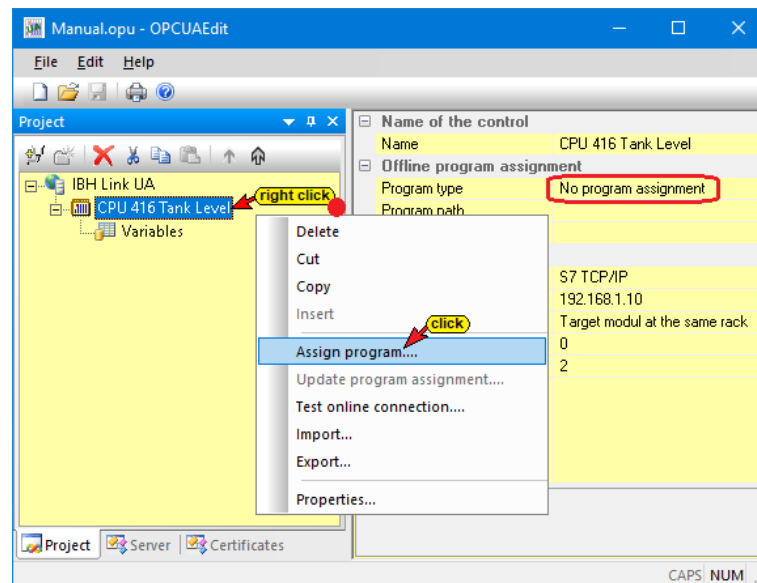
  

MPI Address definition	
Rack No.	Slot No.
3 Bit	5 Bit (0-31)
0 0 0	0 0 0 1 0
0	2

## 1.4 Assign program

An existing PLC program can be assigned to the CPU specified in the New Control dialog box.

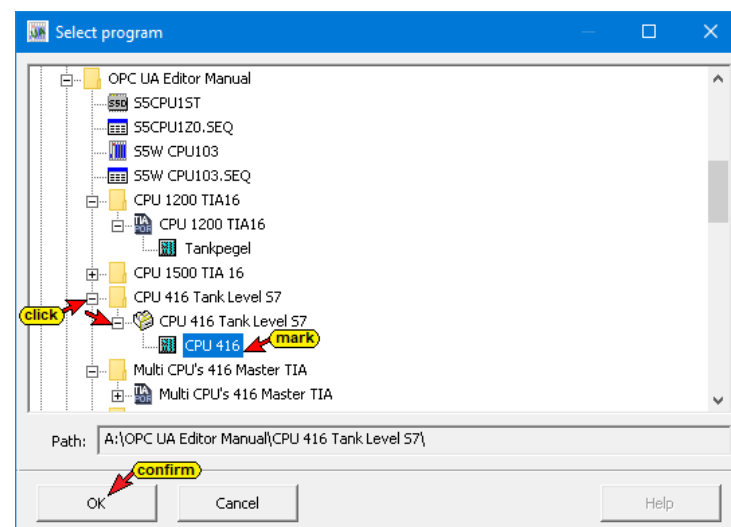
Variables, data, and program information are taken from the PLC program.



The **Assign program** command opens the dialog box **Select program**. The command is also available in the Edit menu.

### Program selection

Select the PLC program to be opened in the **Select Program** dialog box.



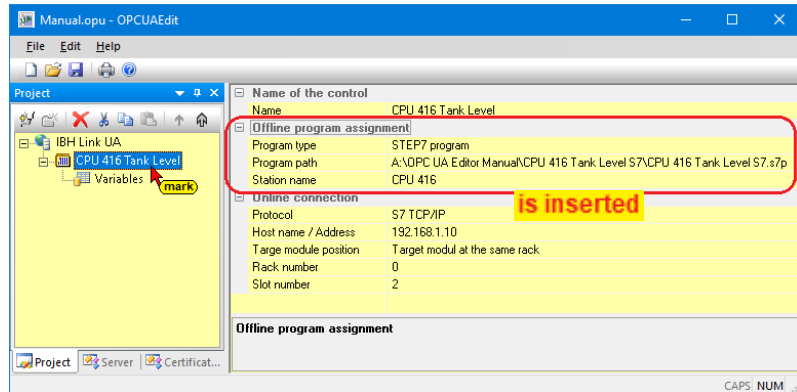
Clicking the symbol **Plus** in front of the symbol of the PLC project (TIA / STEP 7 / S5W / S5 / SEQ) the PLC program reading process is started.



Clicking **OK**, the variables, data and program information are transferred to the **OPC UA Editor**.

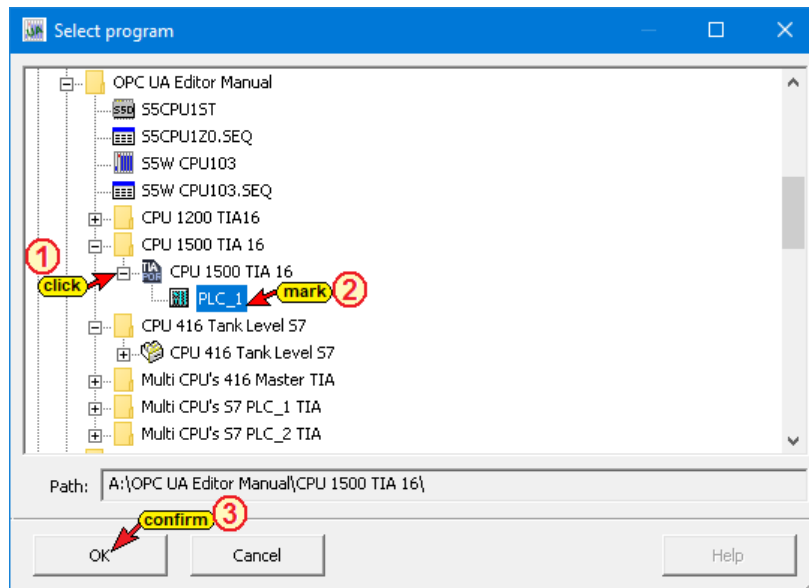
### Offline program assignment

In the right part of the project window, information is displayed under Offline program assignment.

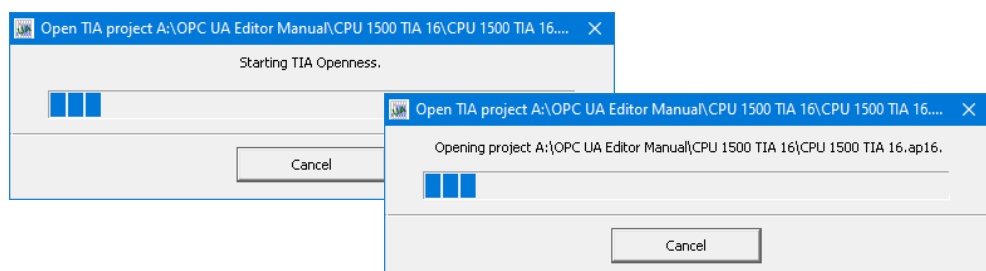


### Special features when selecting TIA projects

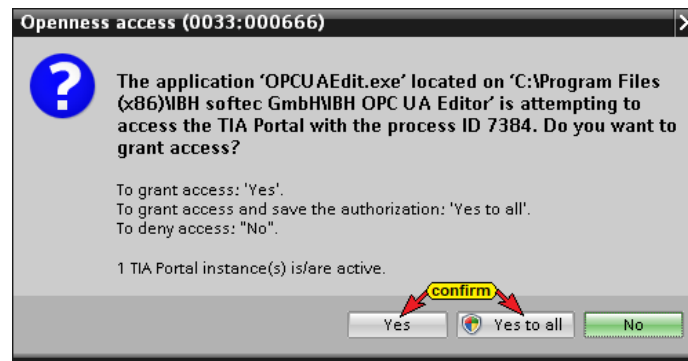
Select the PLC program in the **Select program** dialog box. Clicking the **Plus** symbol in front of the TIA symbol of the PLC project, the PLC program (CPUs) is displayed in the project.



The **SIEMENS support software TIA Openness** is started in the background. Several notices are displayed.



The SIEMENS program **TIA Openness** issues a warning, which must be confirmed, Yes or Yes for all.

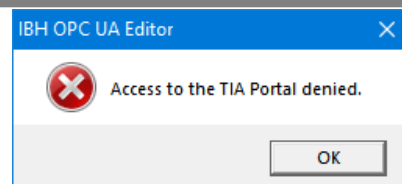


If the support software TIA Openness does not allow access to the selected project, TIA Openness issues the following error message:

**Note:**



The above message may be hidden behind open windows.



**Note:**



To open a TIA project by the SIEMENS support software **TIA Openness** can take some time.

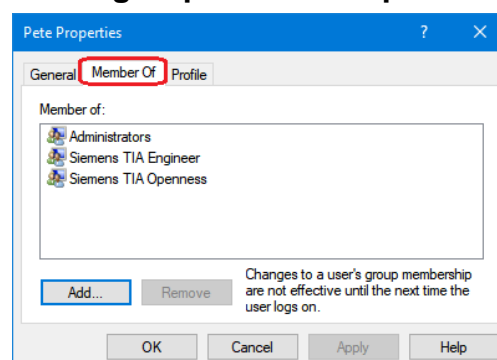
In order to be adopted the PLC program, the TIA Portal software version **TIA 13**, **TIA 14**, **TIA 15** or **TIA 16** must be installed on the PC with the corresponding version of the TIA Openness software.

**It is essential to pay attention to matching software versions.**

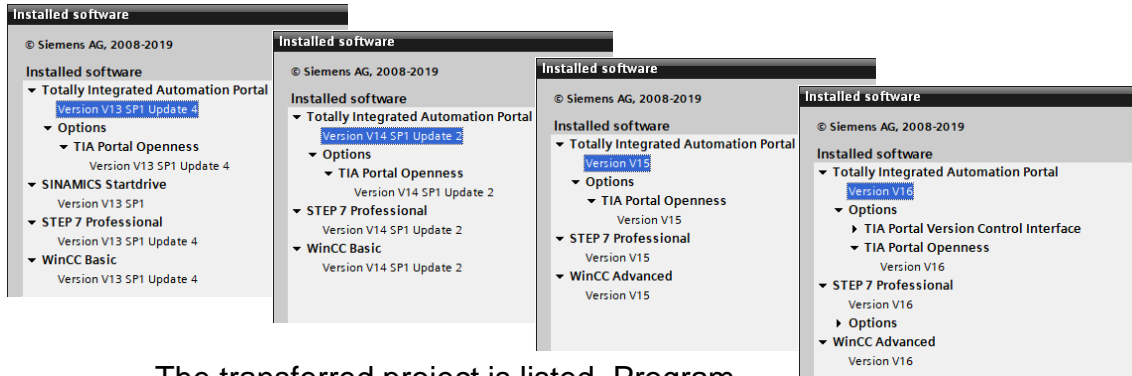
In addition, the user of the PC must be a member of the following groups:

- Administrators
- Siemens TIA Engineer
- Siemens TIA Openness

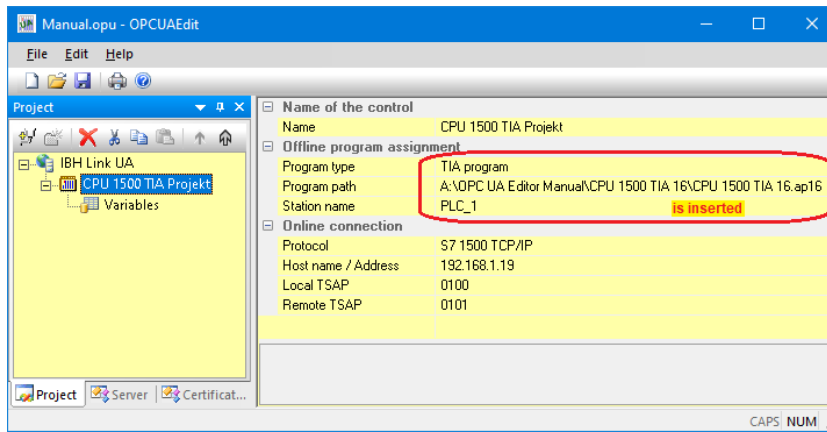
### Registered group memberships in Windows 10



The software versions of the TIA Portal and TIA Openness must be identical.

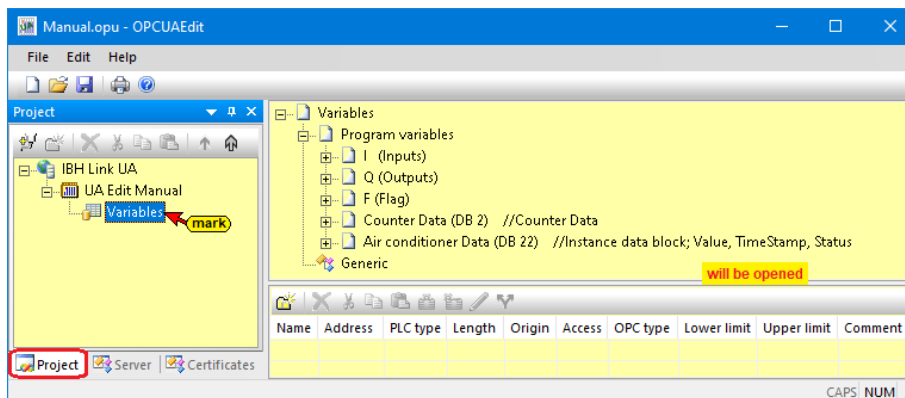


The transferred project is listed. Program assignment information are displayed in the right part of the project window.



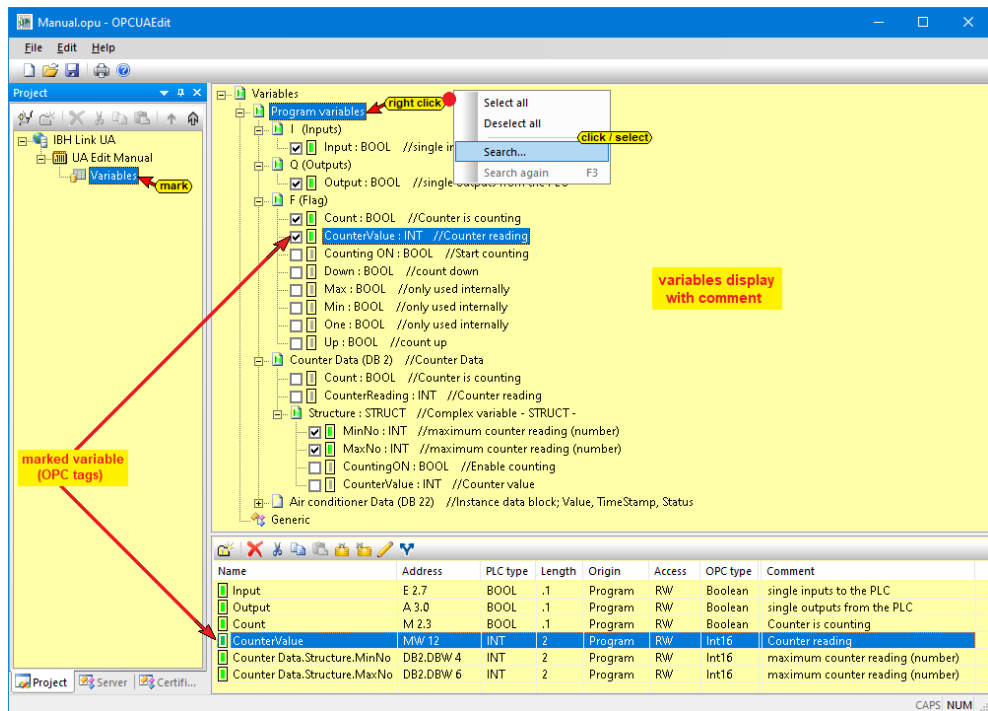
## 1.5 Define OPC tags

A click on the icon **Variables** lists the variables / data in the right part of the project window from the adopted PLC program.



Clicking the symbol **Plus** in front of the icon of the variable area, the existing variables are listed.

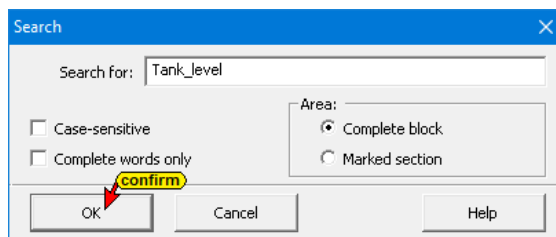
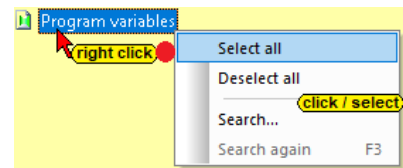




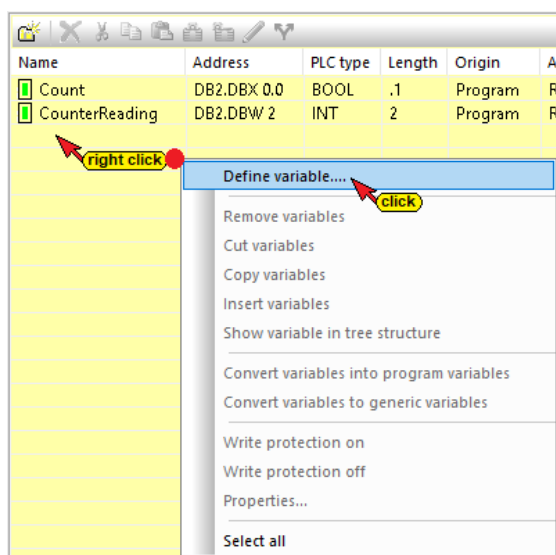
A selected variable is accepted as an OPC tag and displayed in the lower part of the window with additional information.

The context menu offers the commands to select or deselect OPC tags.

A search function is available to handle extensive variable lists.



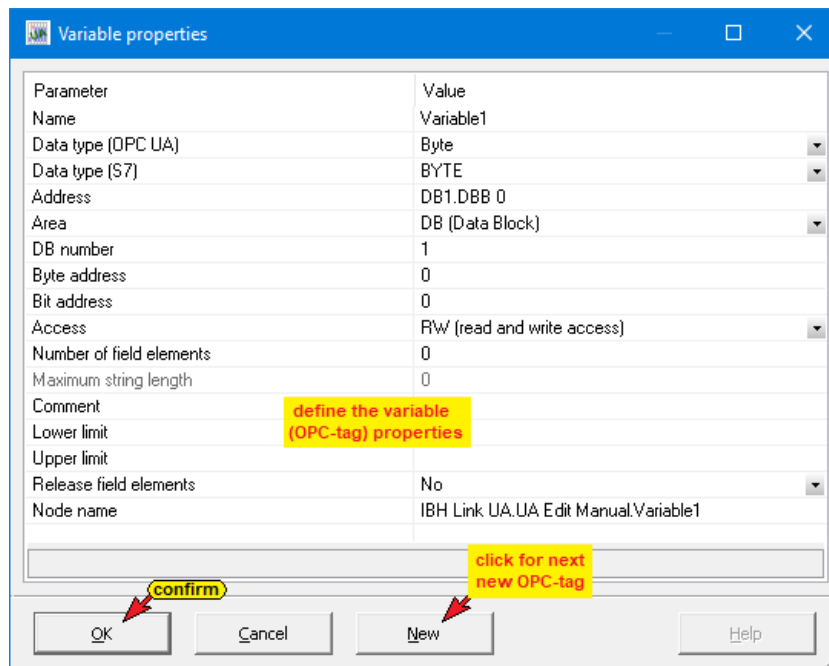
### Add new variable (OPC tag)



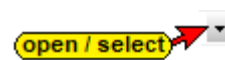
With a right clicking on a line of the variable listing (OPC tag list) the context menu opens. The command **Define variable** (OPC tag) is available.

## Define variable

The context menu command **Define variable** opens the Variable Properties dialog box.



A new variable (OPC tag) can be created. The drop-down list fields make it easier to define a variable.



### Name

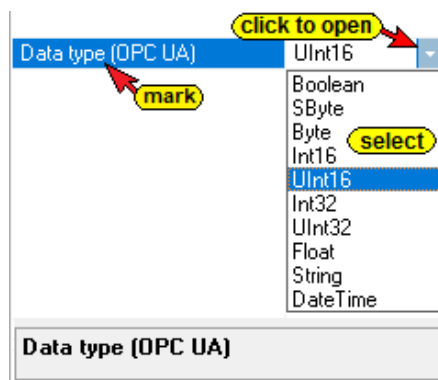


The freely selectable variable name must be unique. A duplicate name is not permitted.

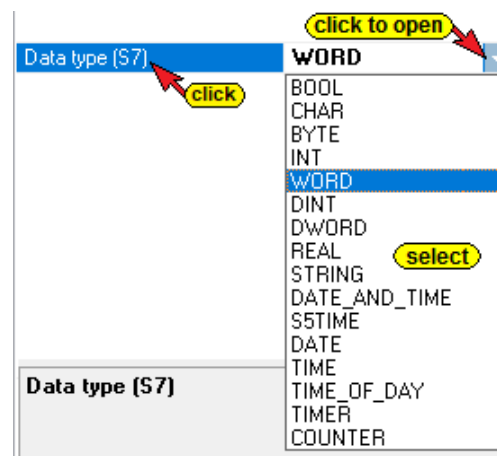
### Data Type (OPC UA) / Data Type (S7)

The **Data Type** can be selected from the drop-down list.

#### Data Type (OPC UA)



#### Data Type (S7)



The data type of the variables needs to be specified only once. Either data type (OPC UA) or data type (OPC UA). The other data type is automatically assigned to the variable.



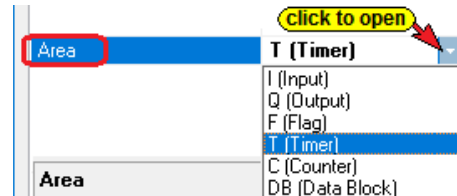
## Address

Address DB1.DBB 0

The address must correspond to the syntax of the control type / programming system. If several field elements are to be specified, the start address must be specified.

## Select Variable Area

The **Area** can be selected from the drop-down list.



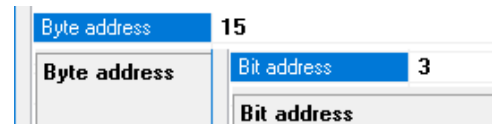
## DB number

The DB number is only relevant if DB (data block) is defined as the area.

DB number 1

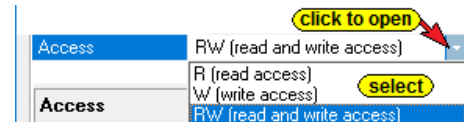
## Byte address / bit address

Depending on the data type, the byte address and, if necessary, the bit address must be entered.



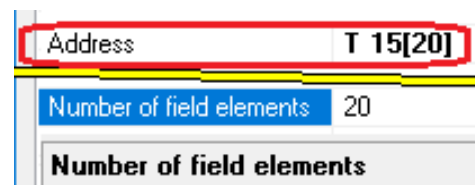
## Access

The access rights of a variable (OPC tag) can be defined in the drop-down list box.



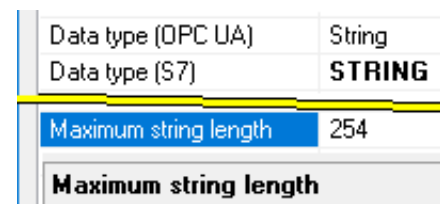
## Number of field elements

A field (array) created with the specified data types, the number of elements must be specified. The number of field elements is automatically added to the start address, in square brackets [ ].



## Maximum string length

If string is selected as the data type, a length of 254 is automatically specified. The length of the string can be adjusted.



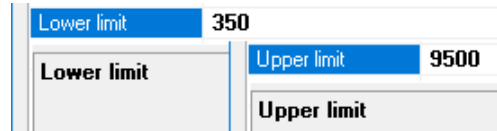
## Comment

A comment can be assigned to a variable (OPC Tag).



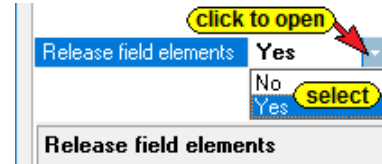
## Lower / upper limit

Limit values can be assigned to a variable (OPC Tag).



## Release field elements

The field elements specified in the Variable Properties dialog box can be released (Yes) or blocked (No) in the drop-down list box.



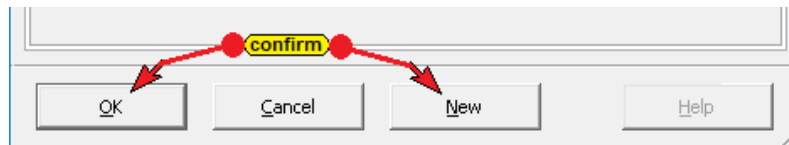
## Node name

The full node name is automatically displayed.



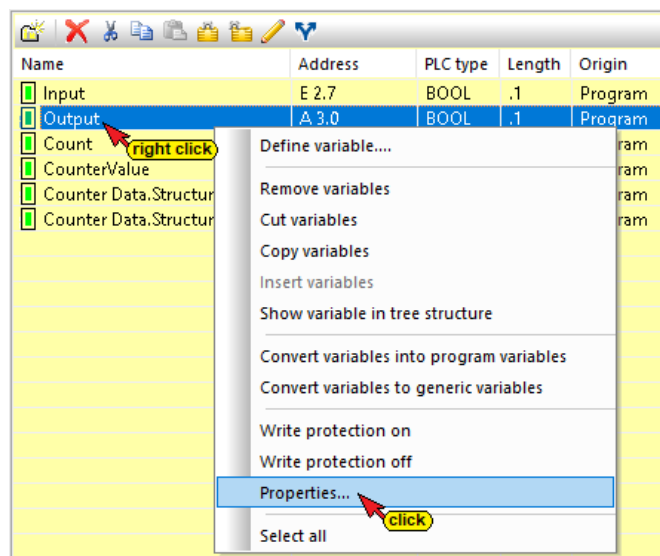
## Accept variable

Click the **New** button accepts the created variable and reopens the dialog box to enter another variable. Clicking the **OK** button accepts the created variable and closes the dialog box.



### 1.5.1 Change Variable (OPC tag)

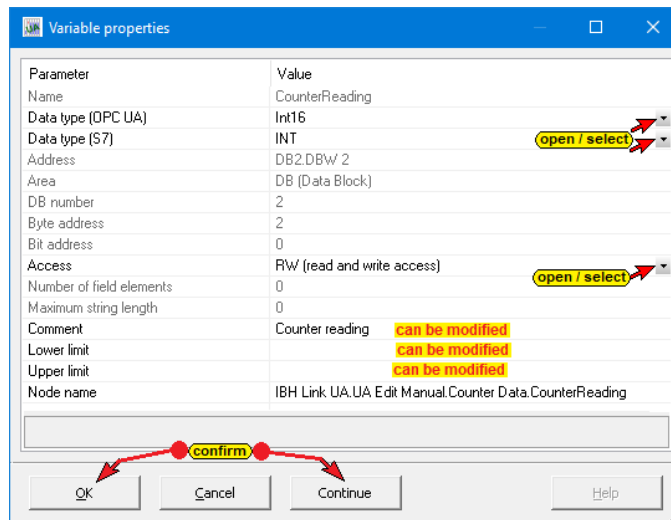
A right-click on a line with a variable (OPC tag), the context menu is opened with commands for editing this variable are available.



The **Properties** command opens the **Variables Properties** dialog box. The marked variable (OPC tag), can be modified.

## Change variable (OPC tag)

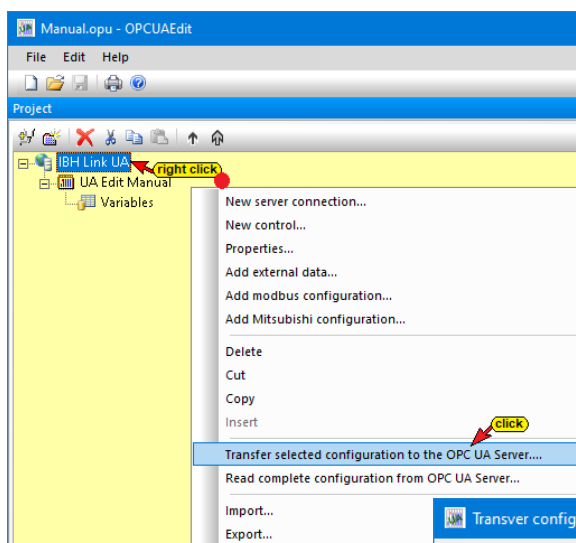
Properties can be changed depending on the data type. The properties shown in light gray cannot be changed.



Click the **Continue** button accepts the modifications. The dialog box reopens to modify the next available variable. Clicking the **OK** button accepts the modifications and closes the dialog box.

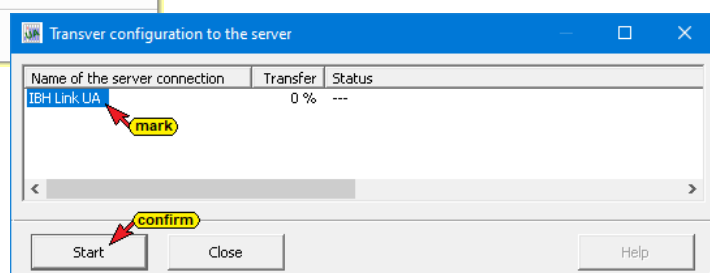
## 1.6 Transfer configuration to the OPC UA server (IBH Link UA).

A right-click on the Server icon (IBH Link UA) opens the context menu.

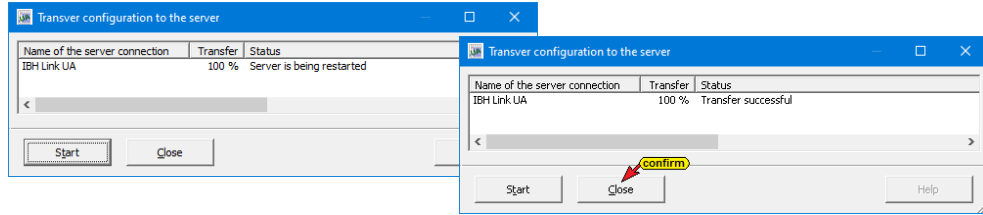


The command **Transfer Selected Configuration to OPC UA Server** opens the **Transfer Configuration to Server** dialog box.

Select the server and then click **Start**.

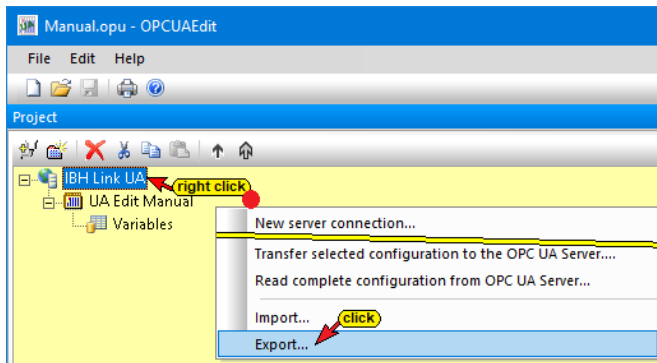


The transfer and the successful transfer are displayed.

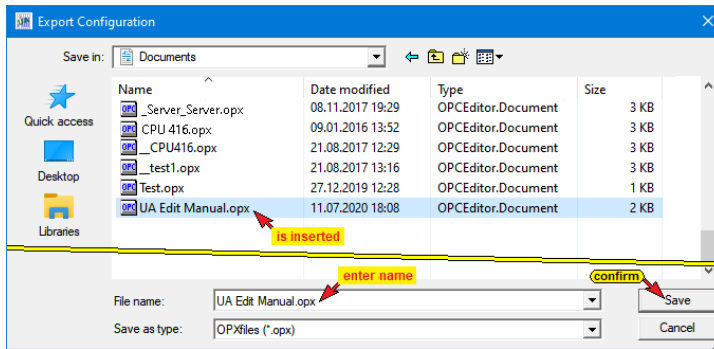


### 1.6.1 Configuration Export

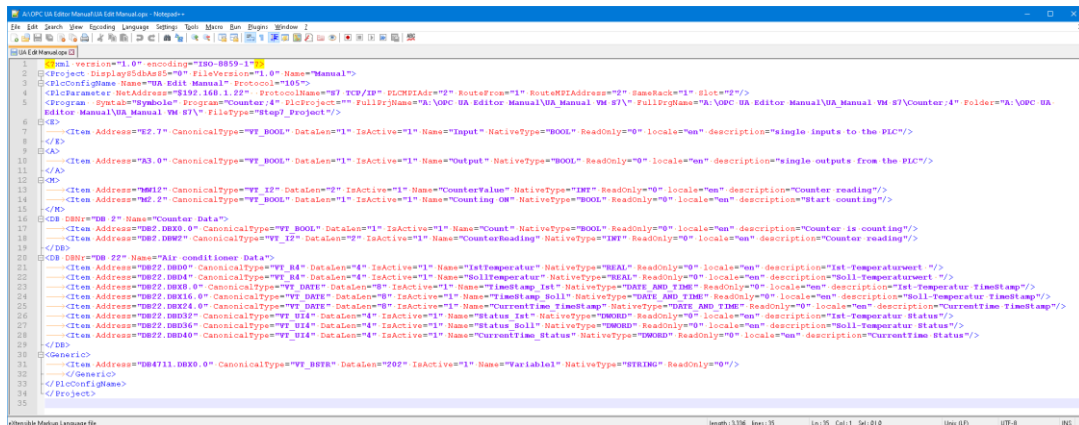
A right-click on the Server icon (IBH Link UA) opens the context menu. The **Export** command is available.



The Export command saves the configuration in **XML** format in a file with the file extension **\*.opx**.



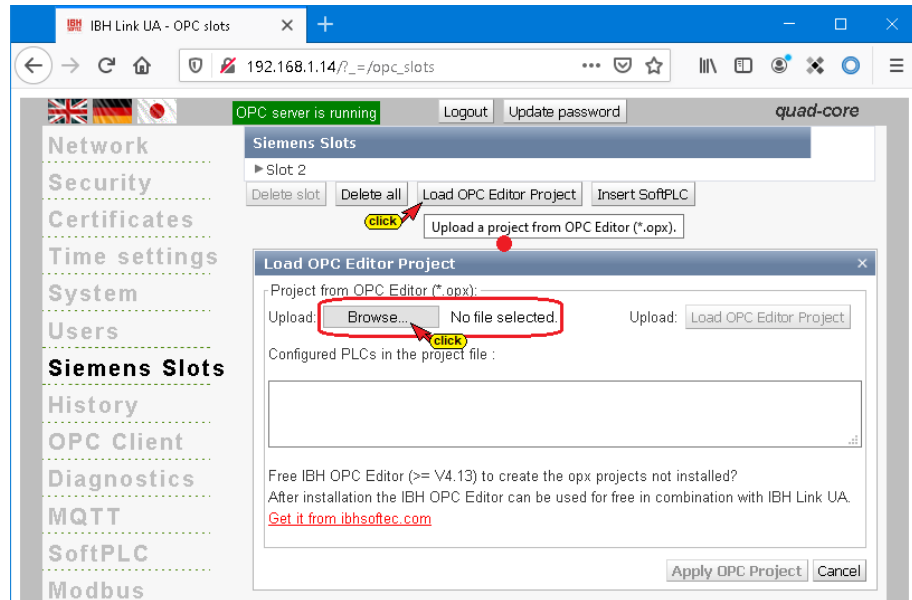
The files exported by the OPC UA Editor in XML format are readable and can be edited directly. A file exported by the OPC UA Editor can be imported into the OPC UA Editor for further processing.



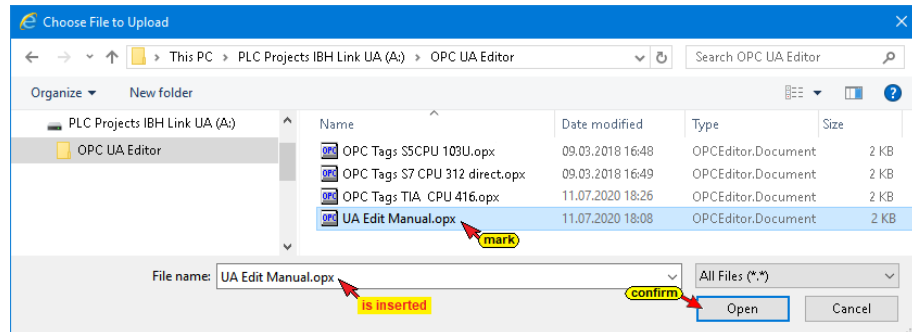
## 1.6.2 Transfer the exported / edited XML file to the IBH Link UA

A file created by the OPC UA Editor can be loaded into the IBH Link UA.

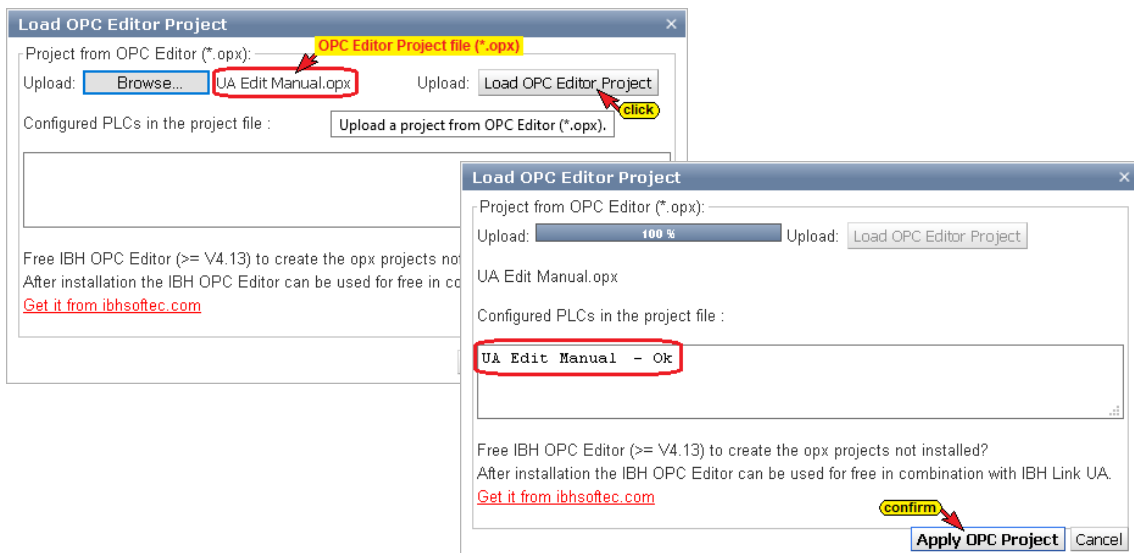
### Select OPC Editor Project



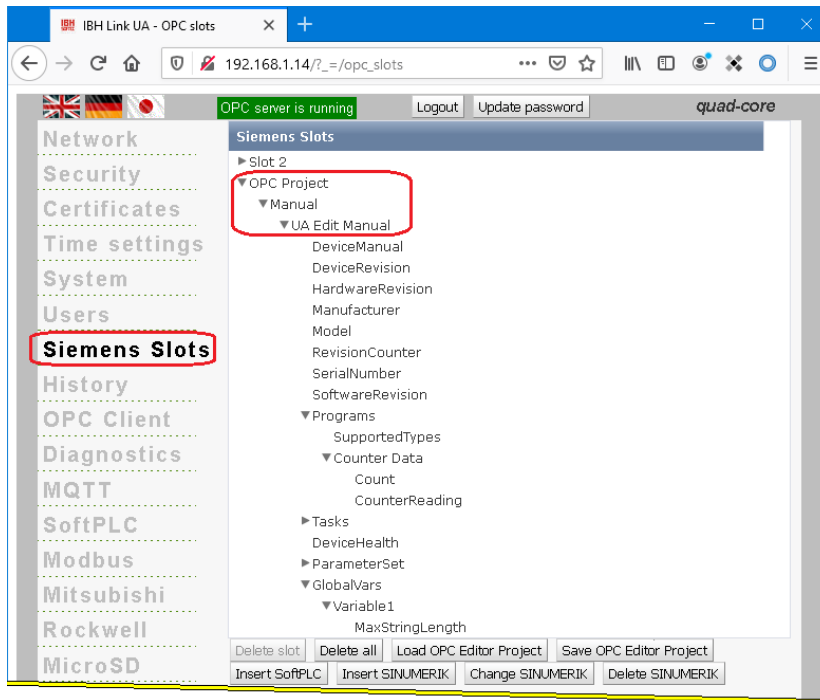
The Windows dialog box to open the OPC editor file is displayed.



### Loading the OPC Editor Project file

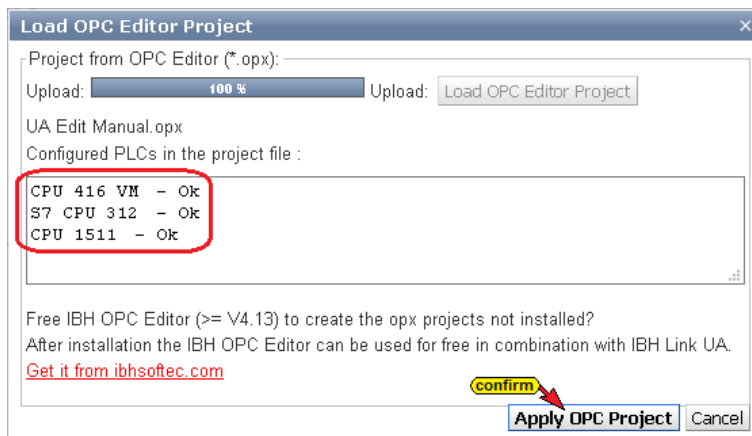


## Information taken from the OPC Editor

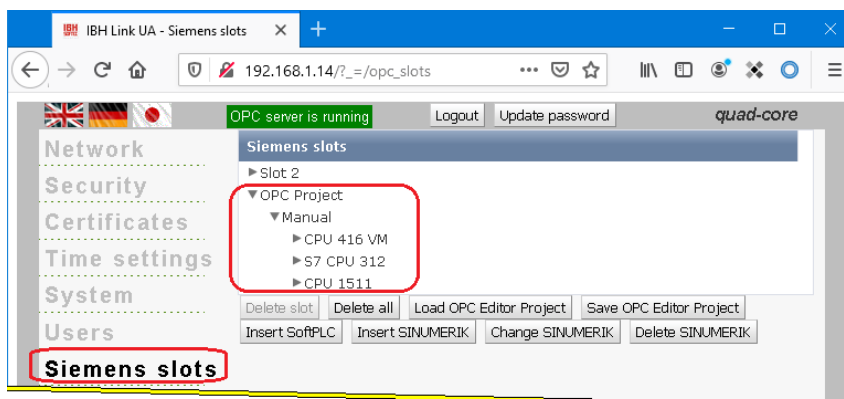


### Multiple CPUs in an OPC Editor project

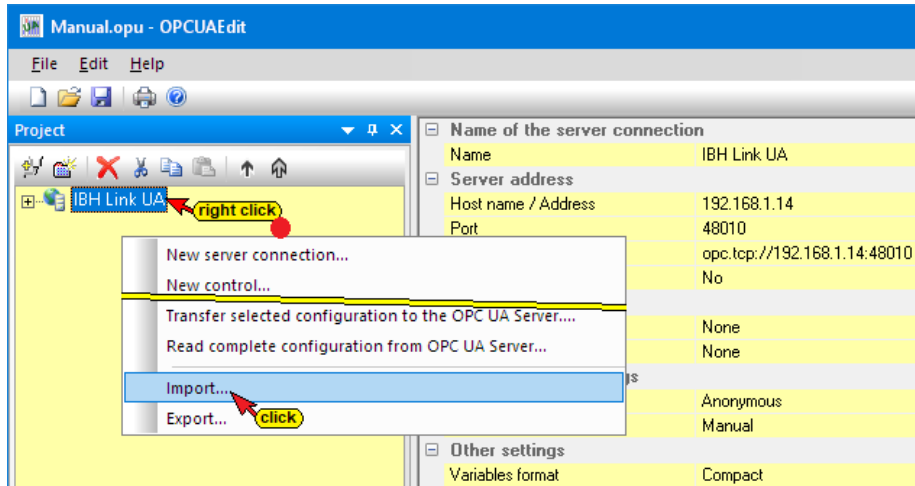
If several CPUs have been merged in one OPC Editor project file, the individual CPUs are listed. All CPUs are taken in one project into the IBH Link UA.



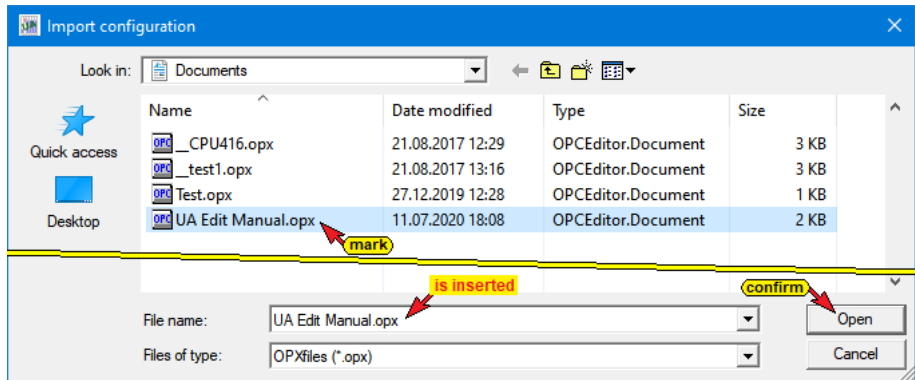
### OPC editor information



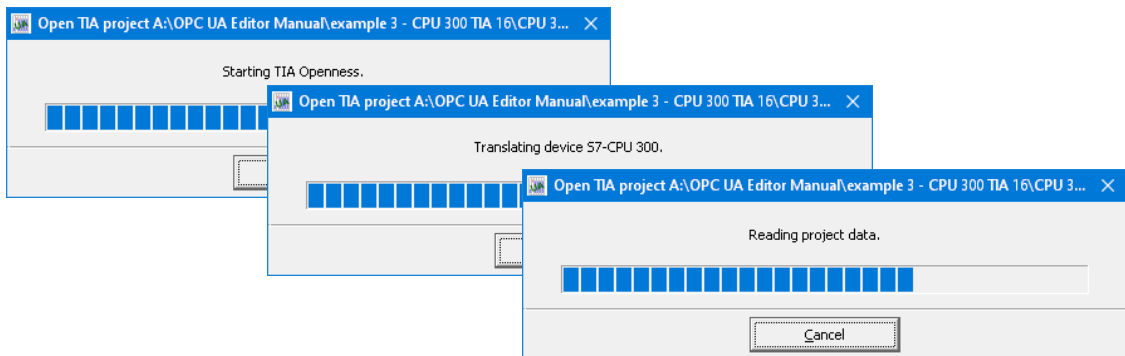
## 1.7 Configuration Import



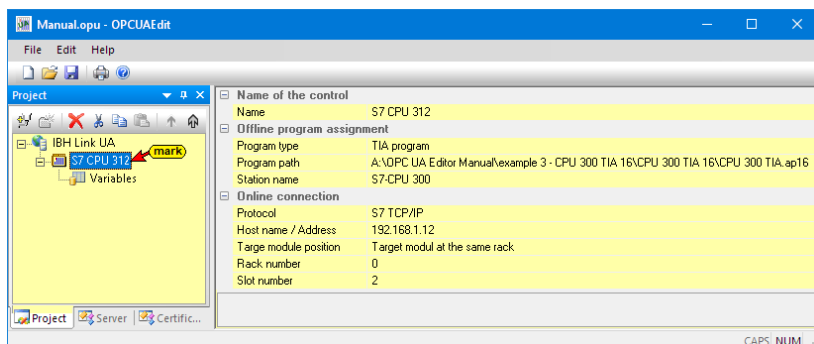
The file (\* opx) to be imported must be in XML format.



Several notes are displayed during the configuration import.



The imported configuration is displayed.

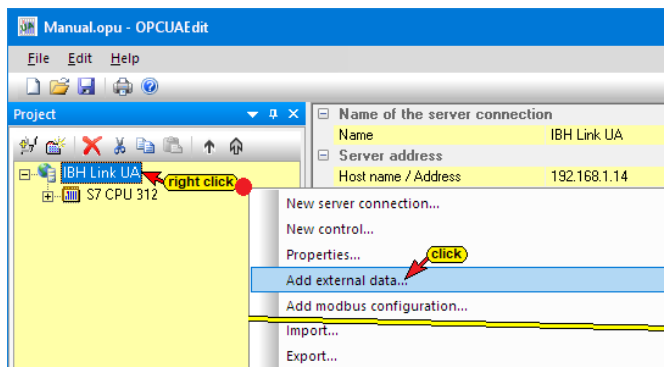


## 1.8 Add external data

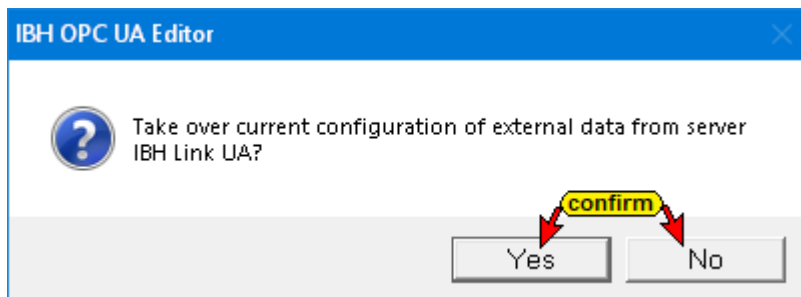
If variables are to be exchanged between two OPC UA servers, the second server and its variables are defined under **Add external data**.

The MQTT configuration also takes place under **External data**.

Right-click on **IBH Link UA** and execute the **Add External Data ...** command.

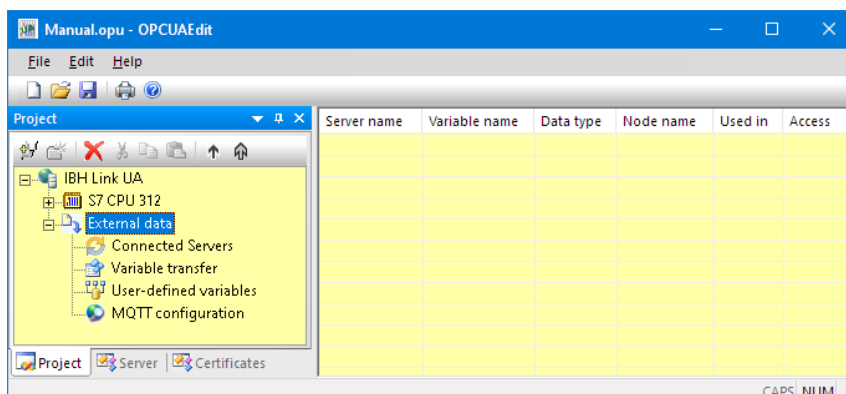


Clicking on **Add External Data...** opens the IBH OPC UA Editor dialog box.



If the desired second OPC UA server has already been inserted in the IBH Link UA, the **Yes** button can be clicked to import the server configuration from the IBH Link UA.

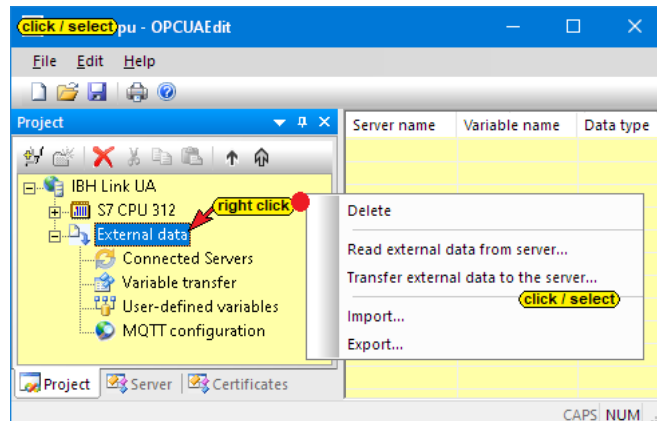
If there is no OPC UA server to be imported in the IBH Link UA, click **No**. The tree structure **External data** is displayed.





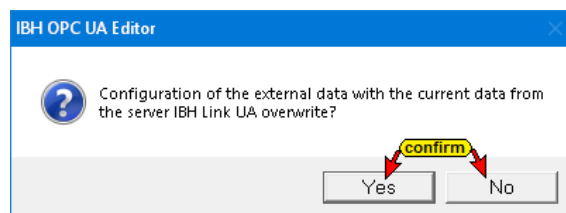
## 1.8.1 External data – context menu

A Right-clicking on **External data** opens the context menu.



### Read external data from the server

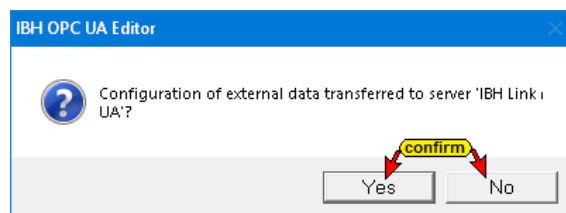
The data of the already configured **OPC UA server** (here - IBH Link UA) are adopted for the external server by clicking **Yes**.



Clicking **No** closes, the information without any action.

### Transfer external data to the server

Clicking **Yes**, the data of the external server are transferred to the already configured OPC UA server (here - IBH Link UA).



Clicking **No** closes, the information without any action.

### Export / Import

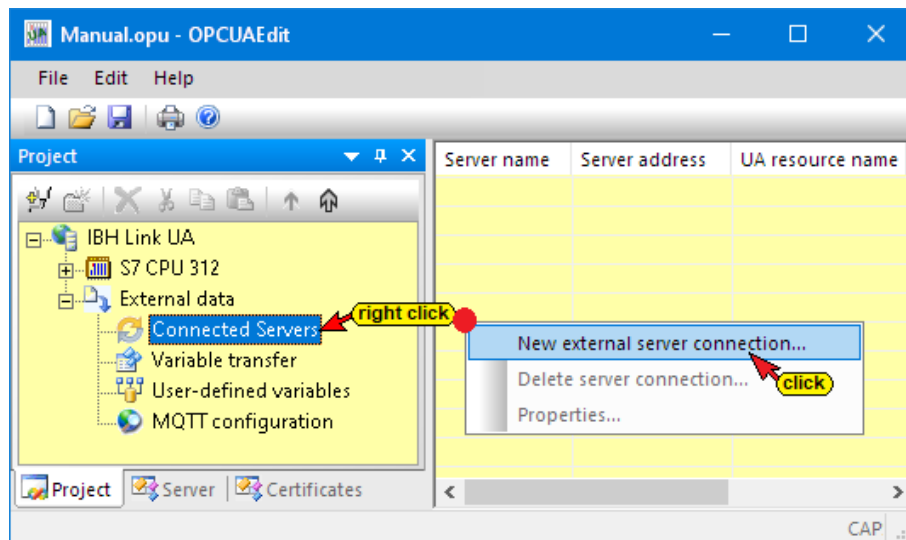
The **Export** command saves the configuration of the external server as an **XML file** (file extension \* .xml).

The file exported by the OPC UA Editor in XML format is readable and can be edited directly.

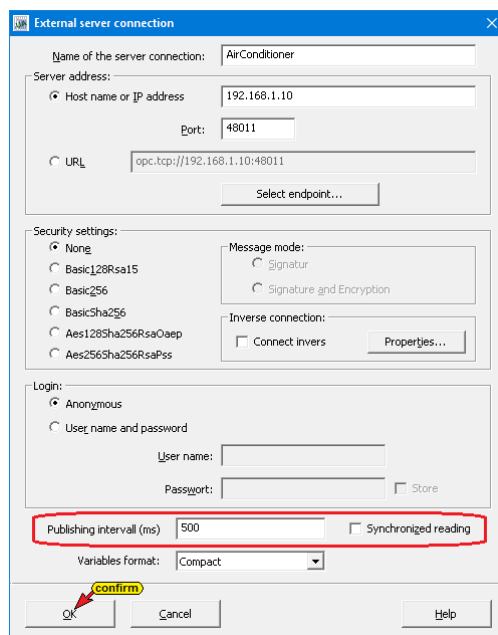
A configuration file of the external server exported by the OPC UA Editor can be imported into the OPC UA Editor for further processing.

## New external server connection – configuration

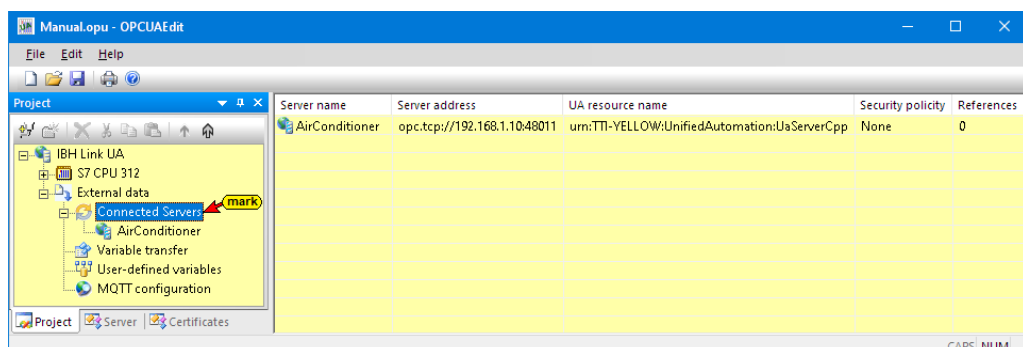
The **New external server connection** command opens the dialog box to configure an OPC UA server to reading variables.



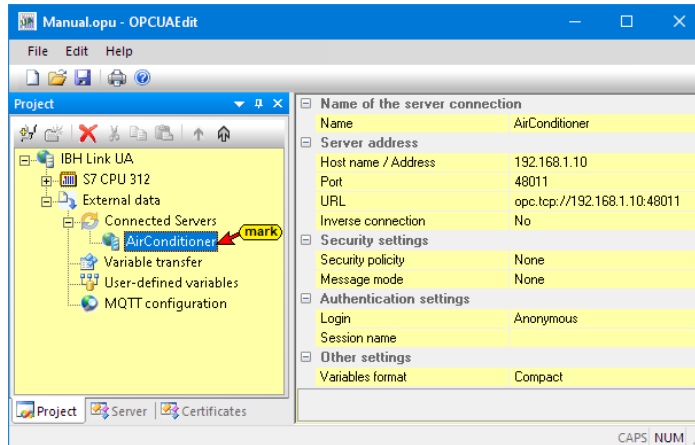
The structure of the dialog box is largely identical to the dialog box Properties of the server connection (see page 4).



In addition, the publishing interval time and the synchronized reading can be initialized.

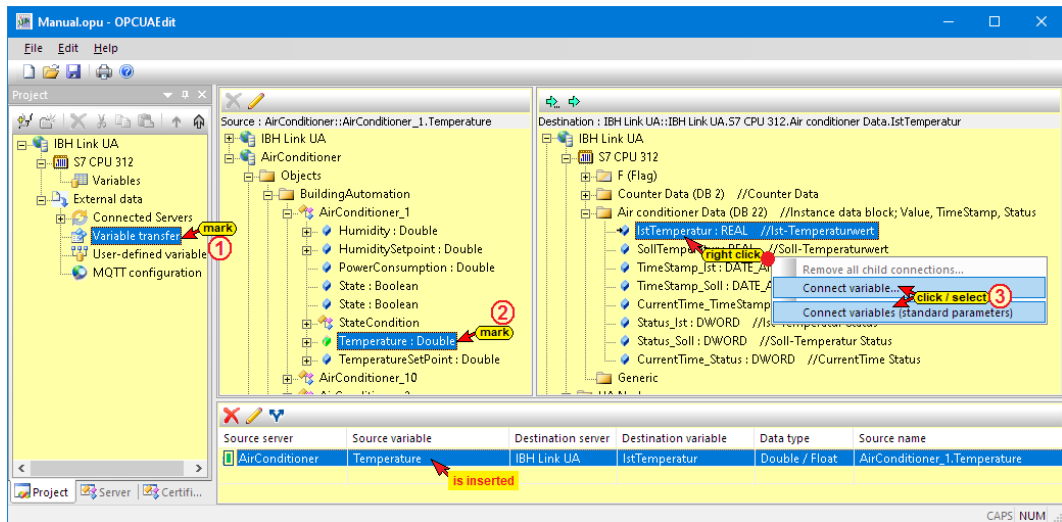


## Connected servers display

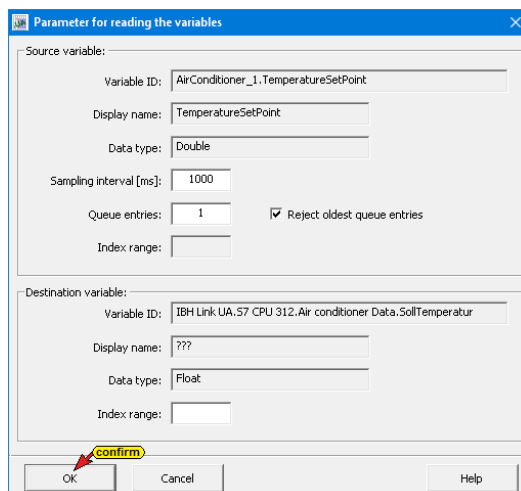


### 1.8.2 Variable transfer

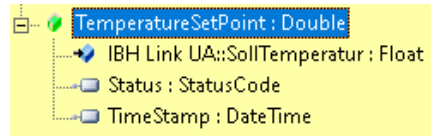
The OPC UA variable connection is accepted by marking the source and target variables and then clicking the **Connect variable** or **Connect variable (standard parameters)** command. The connection is displayed.



The **Connect variable ...** command opens the following dialog box.

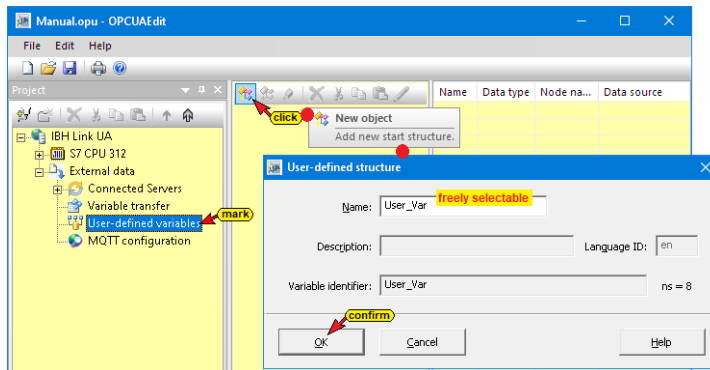


Has the connection of a variable (Value) been carried out the status and the time stamp belonging to the source variable are shown.

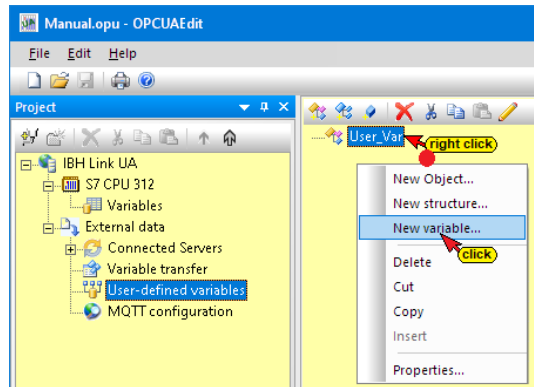


### 1.8.3 User-defined variables

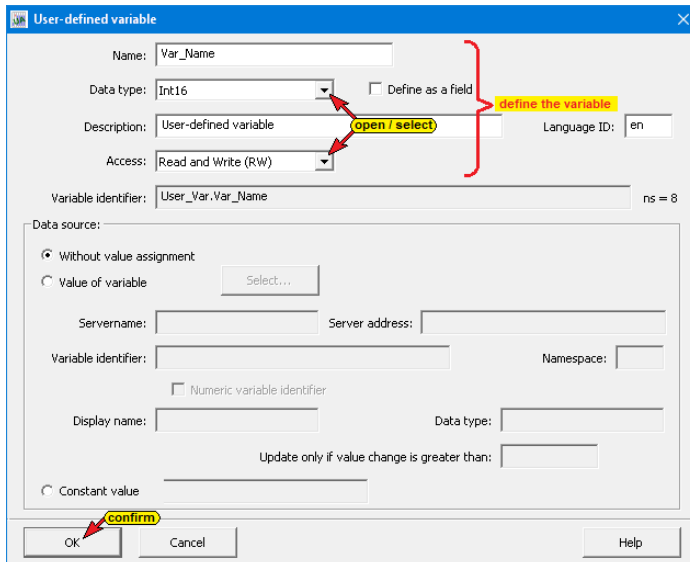
Click on **User-defined variables**, the right project window changes. Click the **New object** icon to open the dialog box to enter the name of the variable structure.



Click **OK** to close the dialog box and display the variable area name. A right-clicking on the **User\_Var** icon opens a context menu.



Dialog boxes can be opened to insert a new object / structure or variable.



Fields are available to define the variable.



## Standard connection

In the dialog box, an MQTT connection to the MQTT broker (RabbitMQ) on `mqtt.ibhsoftec.com` is shown as an example.

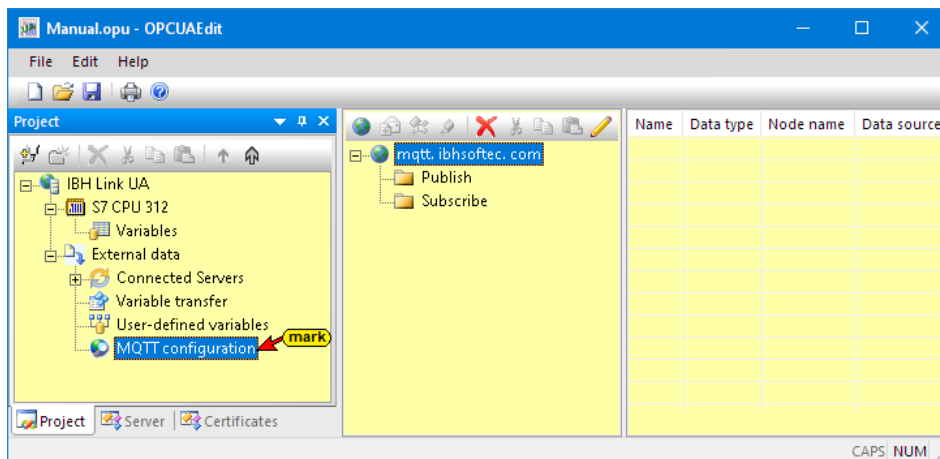
## Connection to the Microsoft Azure IOT Hub

All you need to connect to the Azure IOT Hub is the login text (***AzureIoTConnectionString***). The rest is anchored in the software. Only one topic is possible per Azure IOT Hub.

## Message in case of connection failure (Last Will Message)

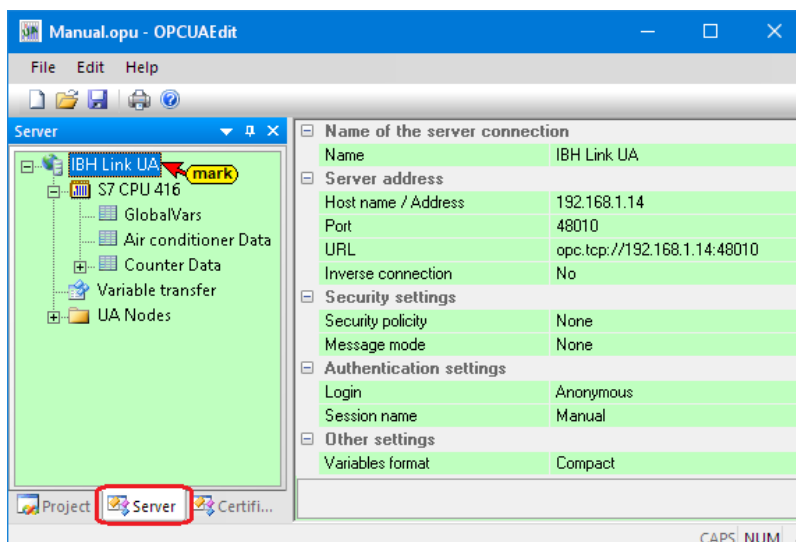
The broker can be advised what should happen if the connection drops unexpectedly.

The configured connections are listed.



## 1.9 IBH OPC UA Editor Server window

The result of a successful project configuration transferred to the **OPC UA server** can be displayed online in the server window.



**Note:**

The configuration available in the *IBH Link UA* window **Siemens slots / OPC Project** is displayed in the Server window.

In the right part of the server window, general settings of the connected **OPC UA server** are displayed. This data is based on the settings from the **New server connection** dialog box (Project window).

**Name of the server connection**

The name is taken from the project window. Name changes can only be done in the project window.

Name of the server connection	
Name	IBH Link UA

**Server address**

The Server address is taken from the project window.

Server address changes can only be done in the project window.

Server address	
Host name / Address	192.168.1.14
Port	48010
URL	opc.tcp://192.168.1.14:48010

**Security settings**

The Security settings are taken from the project window.

Security setting changes can only be done in the project window.

Security settings	
Security policy	BasicSha256
Message mode	Sign and Encrypt

**Authentication settings**

The Authentication settings are taken from the project window.

Authentication setting changes can only be done in the project window.

Authentication settings	
Login	Anonymous
Session name	Workshop

**Other settings**

The variable format representation setting done in the project window is displayed.

Other settings	
Variables format	Compact

**1.10 Displayed Online OPC UA server information**

Information from the OPC UA server and the CPUs both online connected are displayed. It is irrelevant whether the **OPC tags** have reached the **OPC UA server** directly or via the **OPC UA editor**. The configuration in the IBH Link UA Siemens Slots / OPC Project window is displayed.

## Configuration Display

The devices are listed in the left server window. The groups of variables (GlobalVars, data blocks) and the selected controller are listed below.

By clicking on a group, the individual variables (OPC tags) are displayed in the right server window with their status. The status of the OPC tags is continuously updated.

### GlobalVars

Name	Data type	Status	Access	Value	Node name
Count	Boolean	&OK	RW	true	IBH Link UA.S7 CPU 416.Count
CounterValue	Int16	&OK	RW	6039	IBH Link UA.S7 CPU 416.CounterValue
Counting ON	Boolean	&OK	RW	true	IBH Link UA.S7 CPU 416.Counting ON
Down	Boolean	&OK	RW	false	IBH Link UA.S7 CPU 416.Down
Up	Boolean	&OK	RW	true	IBH Link UA.S7 CPU 416.Up

### Data Block – Air conditioner Data

Name	Data type	Status	Access	Value	Node name
CurrentTime_Status	UInt32	&OK	RW	0	IBH Link UA.S7 CPU 416.Air conditioner Data.CurrentTime_Status
CurrentTime_TimeStamp	DateTime	&OK	RW	2020-07-20T15:55:43.452Z	IBH Link UA.S7 CPU 416.Air conditioner Data.CurrentTime_TimeStamp
IstTemperatur	Float	&OK	RW	72	IBH Link UA.S7 CPU 416.Air conditioner Data.IstTemperatur
SoilTemperatur	Float	&OK	RW	72	IBH Link UA.S7 CPU 416.Air conditioner Data.SoilTemperatur
Status_IST	UInt32	&OK	RW	0	IBH Link UA.S7 CPU 416.Air conditioner Data.Status_IST
Status_Soil	UInt32	&OK	RW	0	IBH Link UA.S7 CPU 416.Air conditioner Data.Status_Soil
TimeStamp_IST	DateTime	&OK	RW	2020-07-20T15:55:43.444Z	IBH Link UA.S7 CPU 416.Air conditioner Data.TimeStamp_IST
TimeStamp_Soil	DateTime	&OK	RW	0001-01-01T00:00:00Z	IBH Link UA.S7 CPU 416.Air conditioner Data.TimeStamp_Soil

### Data Block – Counter Data / Structure

Name	Data type	Status	Access	Value	Node name
Count	Boolean	&OK	RW	true	IBH Link UA.S7 CPU 416.Counter Data.Count
CounterReading	Int16	&OK	RW	2204	IBH Link UA.S7 CPU 416.Counter Data.CounterReading

Name	Data type	Status	Access	Value	Node name
CounterValue	Int16	&OK	RW	7075	IBH Link UA.S7 CPU 416.Counter Data.Structure.CounterValue
CountingON	Boolean	&OK	RW	true	IBH Link UA.S7 CPU 416.Counter Data.Structure.CountingON
MaxNo	Int16	&OK	RW	8000	IBH Link UA.S7 CPU 416.Counter Data.Structure.MaxNo
MinNo	Int16	&OK	RW	100	IBH Link UA.S7 CPU 416.Counter Data.Structure.MinNo



## Variable transfer

The online status of the interconnected **OPC tags** (defined in the Project window / External data / Variable transfer) are displayed.

Source server	Source variable	Destination server	Destination variable	Data type	Value	Source name
IBH Link UA	AirConditioner.Temperature	IBH Link UA	IstTemperatur	Double / Float	71.9948	AirConditioner_1.Temperature
IBH Link UA	AirConditioner.Temperature.TimeStamp	IBH Link UA	TimeStamp_Ist	DateTime	2020-07-20T16:14:58.677Z	AirConditioner_1.Temperature.TimeStamp
IBH Link UA	AirConditioner.Temperature.Status	IBH Link UA	Status_Ist	StatusCode / UInt32	0	AirConditioner_1.Temperature.Status
IBH Link UA	AirConditioner.Temperature.SetPoint	IBH Link UA	SoilTemperatur	Double / Float	72	AirConditioner_1.Temperature.SetPoint
IBH Link UA	AirConditioner.Temperature.SetPoint.TimeStamp	IBH Link UA	CurrentTime_TimeStamp	DateTime	2020-07-20T16:14:58.576Z	AirConditioner_1.Temperature.SetPoint.TimeStamp
IBH Link UA	AirConditioner.Temperature.SetPoint.Status	IBH Link UA	Status_Soil	StatusCode / UInt32	0	AirConditioner_1.Temperature.SetPoint.Status
IBH Link UA	AirConditioner.CurrentTime	IBH Link UA	CurrentTime_TimeStamp	DateTime	2020-07-20T16:14:57.566Z	2258
IBH Link UA	AirConditioner.CurrentTime.Status	IBH Link UA	CurrentTime_Status	StatusCode / UInt32	0	2258.Status

## UA Nodes display

The names of the OPC UA Nodes are listed in the left-hand server window (attributes, OPC tags, etc.). The corresponding values are displayed in the right-hand server window.

The screenshot shows the OPCUAEdit interface with the following details:

- Left Pane (Server View):** A tree view showing the hierarchy of UA Nodes. The 'CounterValue' node is selected and highlighted with a red box. A yellow arrow points to it with the word 'mark'.
- Right Pane (Properties View):** Displays the properties of the selected 'CounterValue' node. The 'Value' property is set to 3037, which is circled in red and labeled 'CPU value'.

### Note:

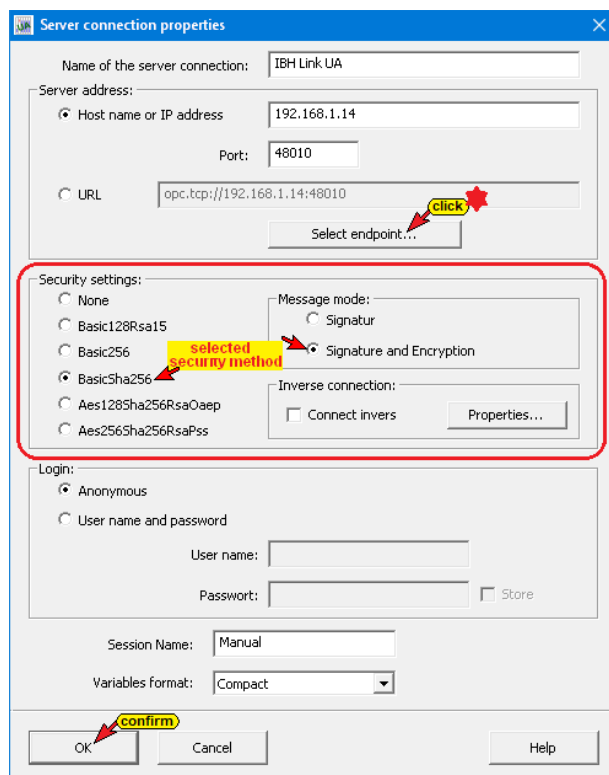
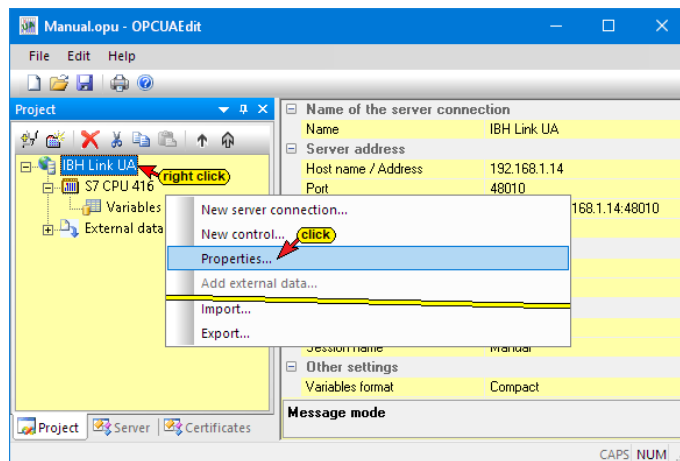
The values are current values. They are taken only once when the OPC UA Node name is clicked.

## 1.11 Certificates

The data transfer from the **IBH OPC UA Editor** to the **IBH Link UA** and vice versa can be encrypted.

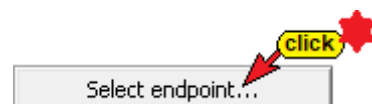
The encrypted data transfer takes place according to the **OPC UA** specifications.

The security method desired may be selected in the Server Connection Properties dialog box.

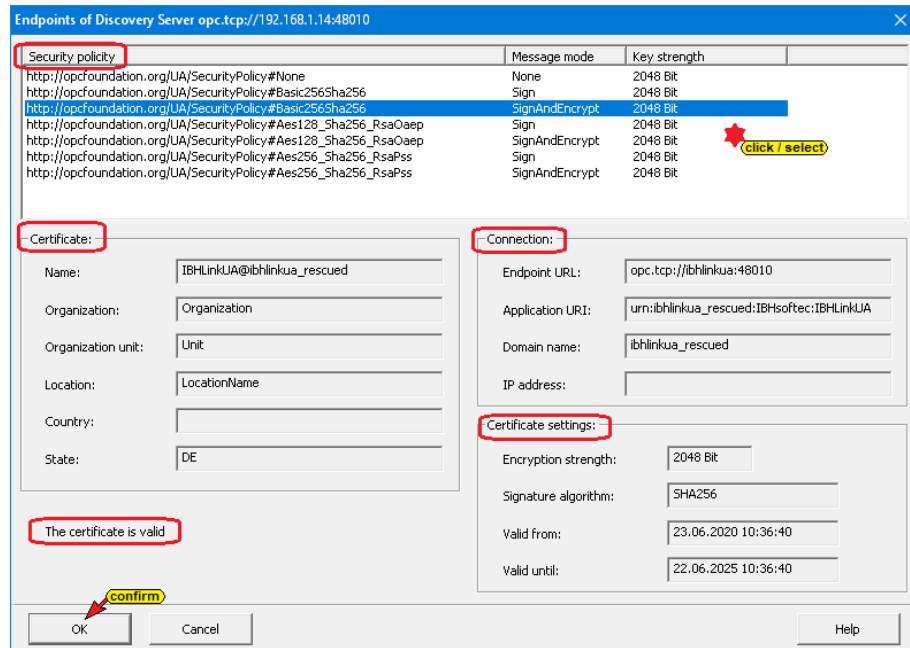


### Security setting via Select end point

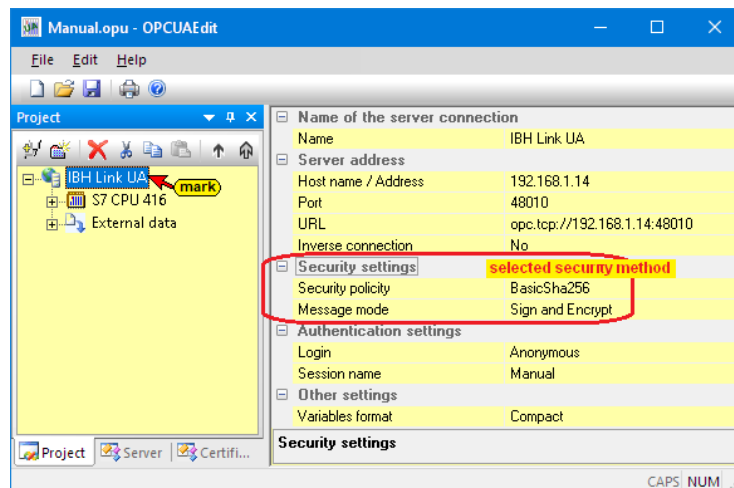
If there is an online connection to the **OPC UA server**, the selection can be made from the encryptions provided by the OPC UA server.



The certificate existing in the OPC UA server with its settings and the connection path to the OPC UA server are displayed.

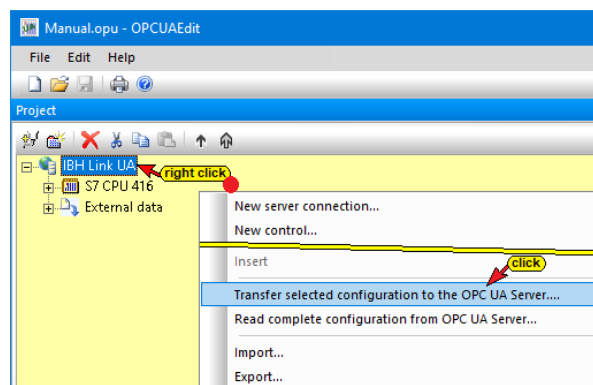


The security procedure selected in the project is displayed.



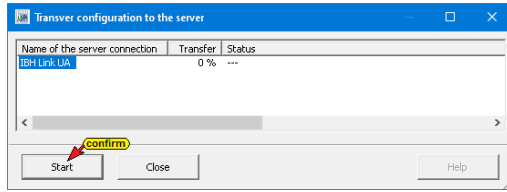
The exchanged certificates must be confirmed as trusted in the IBH OPC UA Editor and in the IBH Link UA.

The exchange of certificates takes place when the configuration is transferred to the OPC UA server (IBH Link UA).

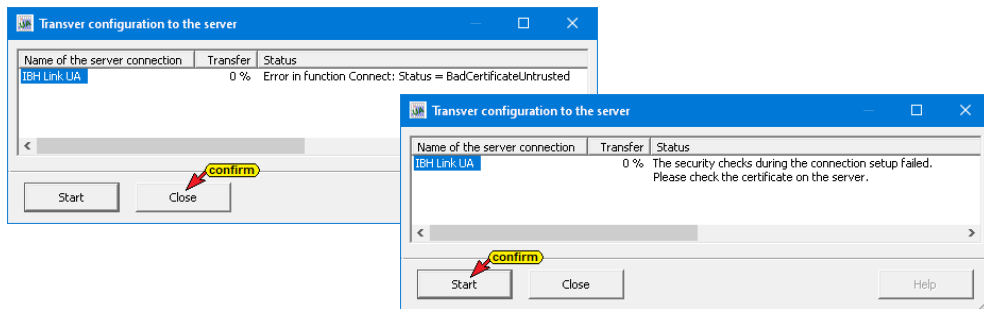


The command **Transfer selected configuration to OPC UA Server** opens the **Transfer configuration to the server** dialog box.

By marking the server and then clicking **Start**, the transfer takes place.

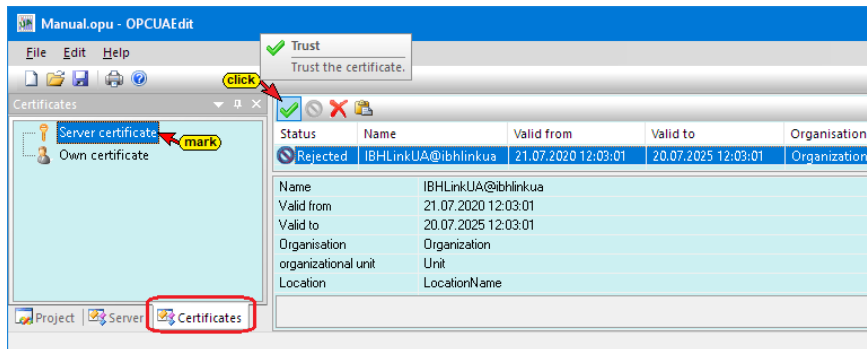


If the exchanged certificates have not yet been confirmed as trusted by the **IBH OPC UA Editor** and the **IBH Link UA**, an error message appears.

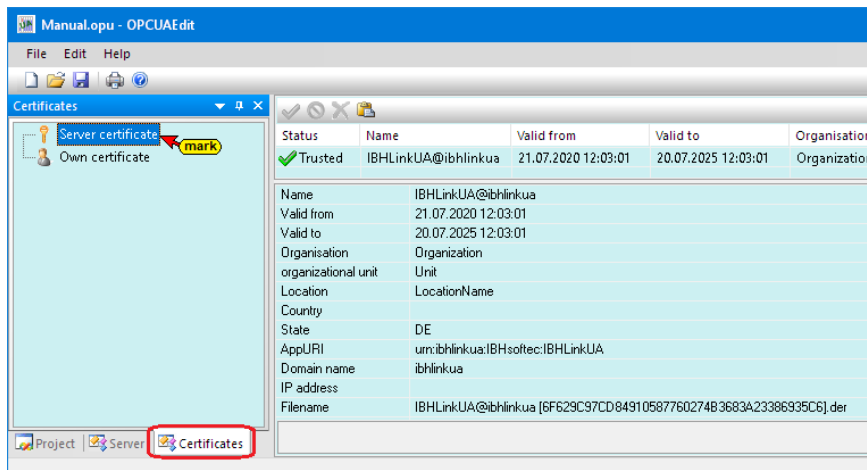


The error message must be closed, and the certificates must be confirmed as trusted.

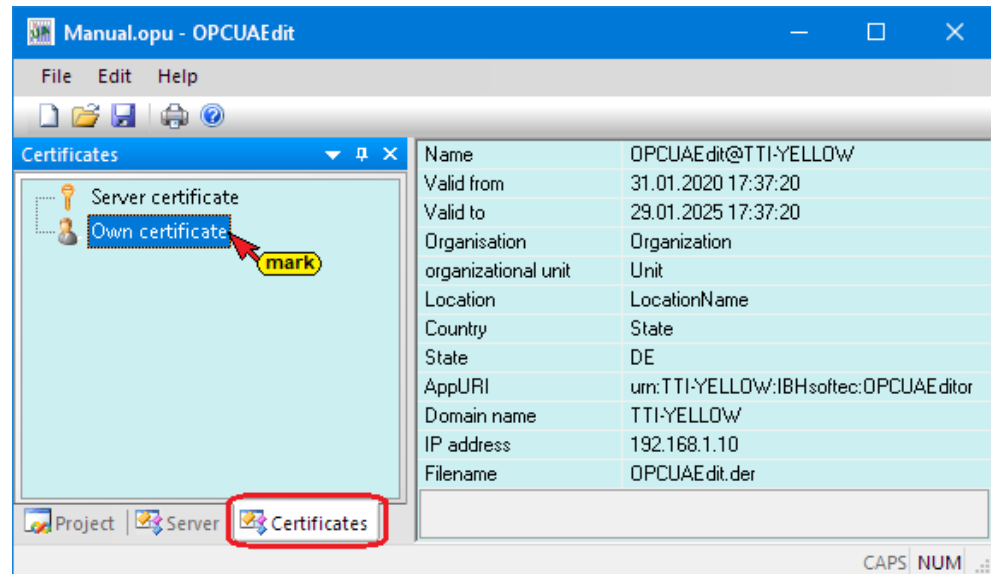
### Rejected IBH Link UA server certificate in IBH OPC Editor



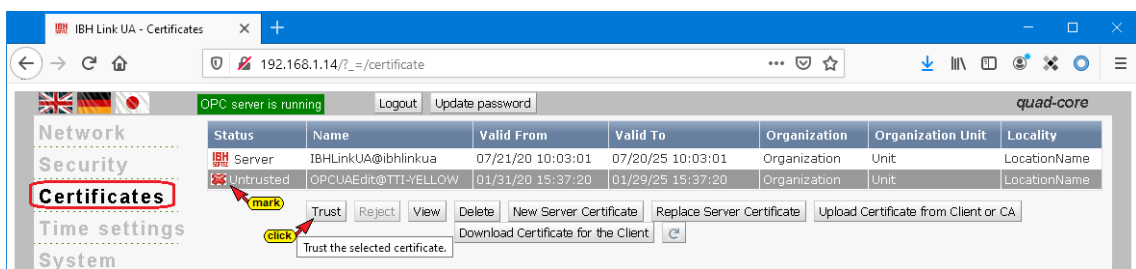
### Trusted IBH Link UA server certificate in IBH OPC Editor



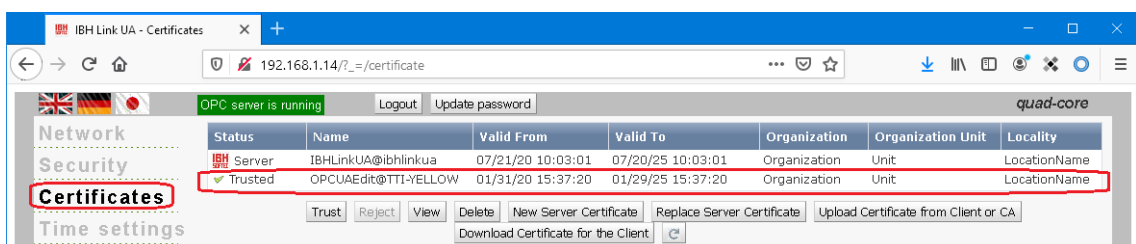
The **IBH OPC UA Editor** has its own certificate.



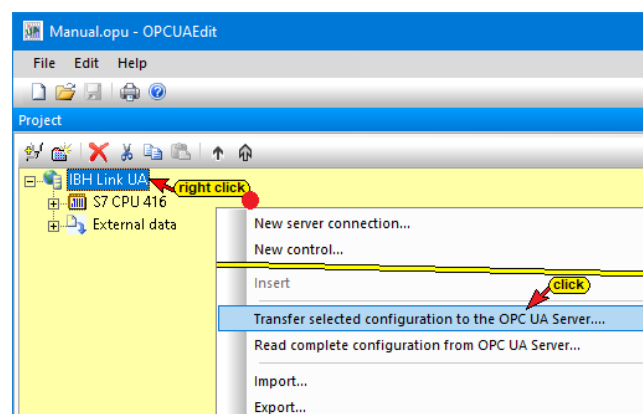
The own certificate of the **IBH OPC UA Editor** must be confirmed **trusted** in the **IBH Link UA**.



The **IBH OPC UA Editor Certificate** must be set to **trusted** in **IBH Link UA**.

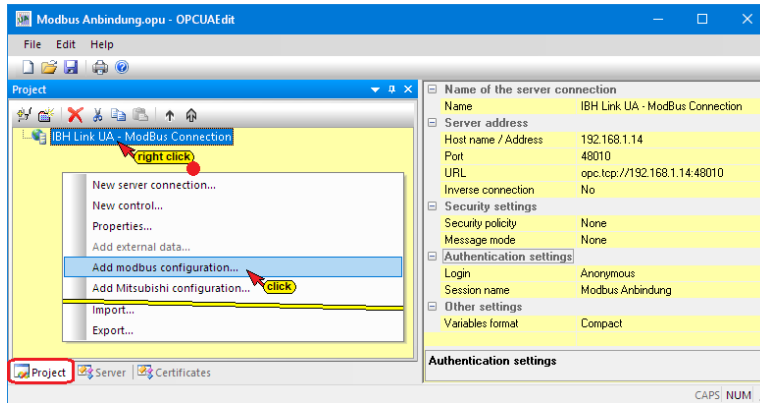


The transfer of the selected configuration to the **OPC UA Server** can be restarted.

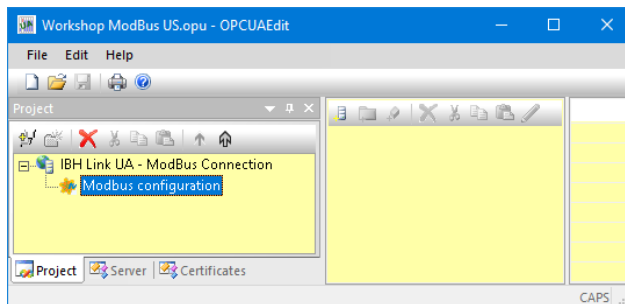


## 1.12 Modbus configuration

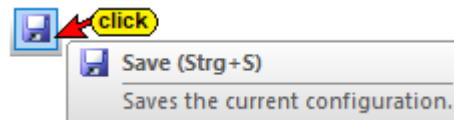
If the connection to the **IBH OPC UA Editor** has been created, a Modbus configuration can be added. The **Add Modbus configuration** command starts the configuration process.



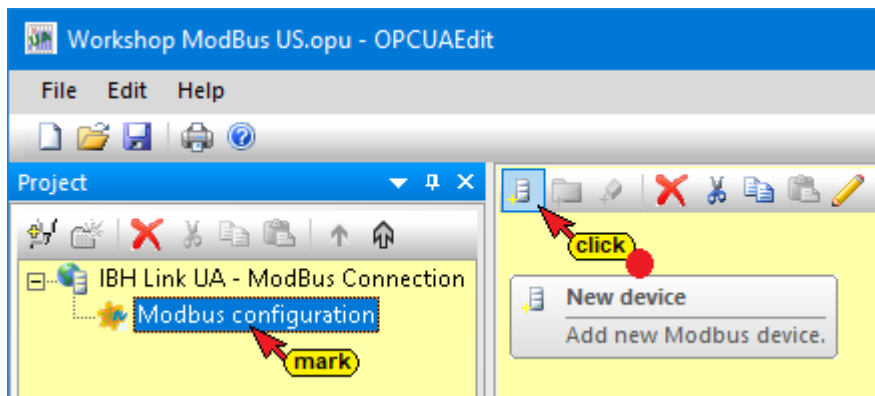
In the right part of the project window, the **Modbus configuration** window opens.



The existing configuration can be saved at any time.



### 1.12.1 Add new Modbus device



If Modbus configuration is marked in the left part of the project window, the Modbus device properties dialog box can be opened by clicking on the New device icon.



The **IBH Link UA** devices **SingleCore** or **QuadCore** have a **USB port**. With a converter (USB / serial), this port can be used as an interface for the Modbus device.

If the device name and the interface are specified, the dialog box can be confirmed with **OK**.

### 1.12.2 Modbus TCP interface / Modbus variable

The Modbus organization specifies the form in which variables can be present.

Device manufacturers follow these guidelines and provide device-specific tables that list the address of the variable and how it is addressed.

To define the Modbus variable as OPC tag in the **Modbus Variable Properties** dialog box of the IBH OPC UA Editor, knowledge of the fieldbus node with its data, data coding, addressing and transactions is required.

Following are excerpts from specifications of the Modbus organization, which are necessary for the definition of the OPC tags. The tables / descriptions of the fieldbus node provide this information in a similar form.

## Modbus Functions – Partially – (Modbus Organization)

Reading and writing variables are determined by functions.

<b>Physical Discrete Inputs</b>	Read Discrete Inputs
<b>Internal Bits or Physical coils</b>	Read Coils
	Write Single Coil
	Write Multiple Coils
<b>Physical Input Registers</b>	Read Input Register
<b>Internal Registers or Physical Output Registers</b>	Read Holding Registers
	Write Single Register
	Write Multiple Registers
	Read/Write Multiple Registers
	Mask Write Register
	Read FIFO queue
<b>File record access</b>	Read File record
	Write File record
<b>Diagnostics</b>	Read Exception status
	Diagnostic
	Get Com event counter
	Get Com Event Log
	Report Server ID
	Read device Identification
<b>Other</b>	Encapsulated Interface Transport
	CANopen General Reference

Most Modbus devices do not provide all the functions. However, writing or reading functions are usually supported by all. Therefore, the IBH OPC UA Editor only supports the functions for reading and writing values.

## Modbus data formats

<b>Name data access (Storage area)</b>	<b>Quantity Data type</b>	<b>Access Type memory access</b>	<b>Comment</b>
<b>Discrete Input</b> physical inputs process image	<b>1 Bit</b> (Single bit)	<b>Read-Only</b>	These data types can be provided by an I / O system.
<b>Coils</b> Discrete outputs physical outputs process image	<b>1 Bit</b> (Single bit)	<b>Read-Write</b>	These data types can be changed by an application program.
<b>Input Registers</b>	<b>16-Bit</b> Wort	<b>Read-Only</b>	These data types can be provided by an I / O system.
<b>Holding Registers</b>	<b>16-Bit</b> Wort	<b>Read-Write</b>	These data types can be changed by an application program.



Addresses of the memory areas and their access provided by the Modbus device manufacturers, in

## Example - Listing: Access to variables (fictitious devices manufacturer information)

### Modbus register addressing

	Addressing 1 Bit Register	Start address	End address	Access	Description
Process Data	X <sup>(1)</sup>	0x0000	0x00CF	Read/Write	Process data Physical input
	X <sup>(2)</sup>	0x00D0	0x00FF	Read/Write	
	X <sup>(3)</sup>	0x0100	0x01CF	Read-only	Process data Physical output
	X <sup>(4)</sup>	0x01D0	0x01FF	Read-only	
	X <sup>(5)</sup>	0x0200	0x02FF	Read/Write	
	X <sup>(6)</sup>	0x0300	0x03FF	Read/Write	
Diagnosis	X <sup>(7)</sup>	0x0400	0x040F	Read-only	Status register
	X <sup>(8)</sup>	0x0410		Read-only	Process in analog output
	X	0x0411		Read-only	Process in analog input
	X	0x0412		Read-only	Process in digital output
	X	0x0413		Read-only	Process in digital input
	Special Register	X <sup>(9)</sup>	0x0420	0x042F	Read/Write
X <sup>(10)</sup>		0x0430	0x043F	Read/Write	Error Register
X		0x0440	0x044F	Read/Write	Command
X		0x0450	0x045F	Read/Write	Internal Register

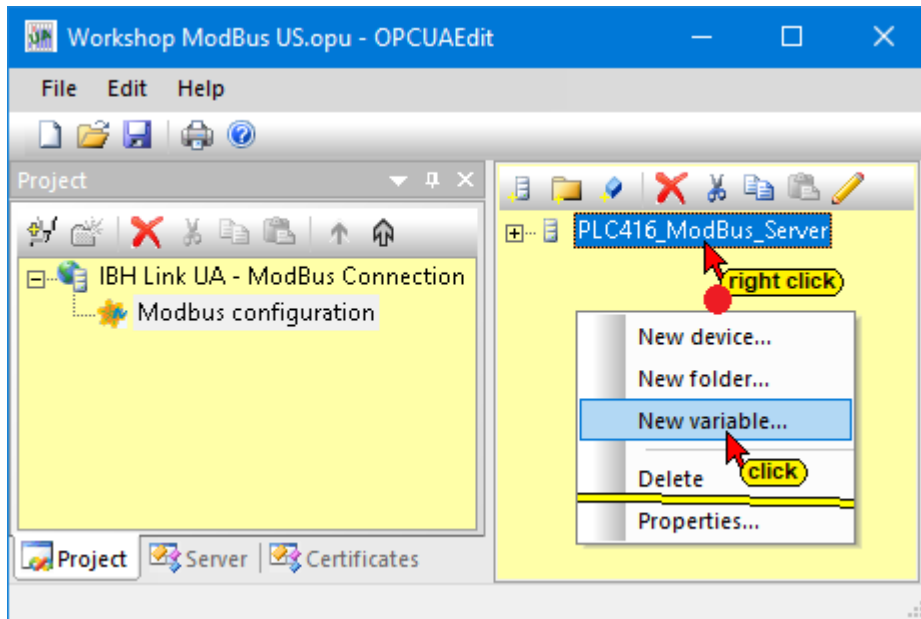
(n) As an example, for the definition of a variable exists.

#### Note:

The address information in the Modbus devices Manufacturer specifications are often in hexadecimal form. These addresses are converted into a decimal address for input in the IBH OI

### 1.12.3 Defining Modbus variables in the IBH OPC UA Editors

Right-clicking on the Modbus device name (PLC 416 ModBus Server) opens the context menu.

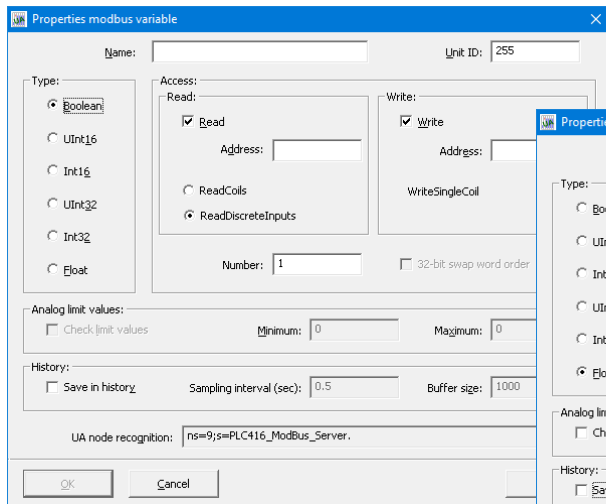


The **New Variable** command opens the **Modbus Variable Properties** dialog box.

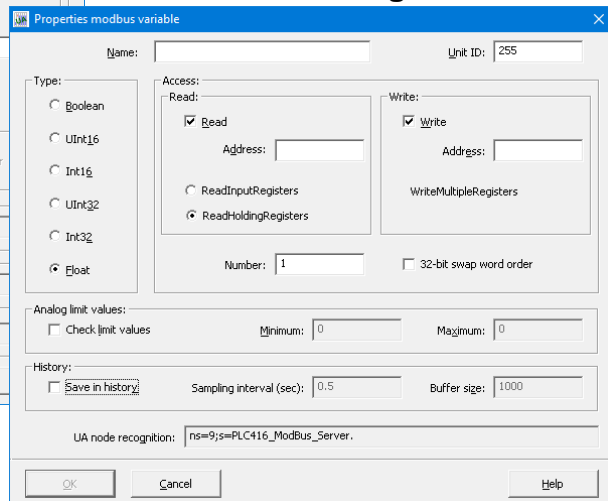
This dialog defines the variables that the OPC server should connect to. This can be a write variable, a read variable, or a read / write variable.

## 1.13 Modbus variable – Properties dialog box

### Read- (Coils / Discrete Inputs) – Write-Coils



### Read- / Write-Registers




## Modbus TCP / RTU interface

Modbus device manufacturers describe in tables the Modbus fieldbus node functions. Based on these descriptions, the definitions of the accesses to variables are defined.

### Name (Variable)

The name can be selected freely but must correspond to the OPC UA specification (letters **A-Z**, **a-z**, numbers **0-9**, no special characters, no symbols, no dots, no colon.) As a special character only **\_** under line.

### Unity-ID

For Modbus TCP, the Unity-ID = 255. This is set  at default. For Modbus RTU, the address must be set according to the slave address.

### Type

#### Boolean

One (1) bit information (**Boolean**), which can have the states TRUE (1) and FALSE (0). A variable of type **Boolean** occupies 1 bit in a register address (16 bits). An array of 16 variables of type Boolean occupy an entire register address (16 bits).

#### UInt16

Positive 16-bit **unidirectional integer** (number) between 0 and 65,535 ( $2^0$  to  $2^{+16}$ ). A **UInt16** variable occupies a register address (16 bits).

#### Int16

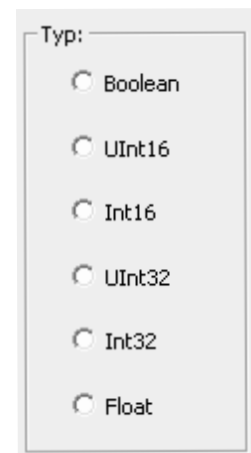
Positive or negative 16-bit integer (number) between -32,767 and +32,767 ( $-2^{+15}$  to  $2^{+15} - 1$ ). An **Int16** variable occupies a register address (16 bits).

#### UInt32

Positive 32-bit unidirectional integer (number) between 0 and 4,294,967,295 ( $2^0$  to  $2^{+32}$ ). A **UInt32** variable occupies two (16-bit) register addresses.

#### Int32

Positive or negative 32-bit integer (number) between -2,147,483,654 and +2,147,483,654 ( $-2^{+31}$  to  $2^{+31} - 1$ ). An **Int32** variable occupies two (16-bit) register addresses.



Typ:

- Boolean
- UInt16
- Int16
- UInt32
- Int32
- Float

## Float

A variable of data type **Float** represents a fractional number that exists as a 32-bit floating-point number (REAL). A **Float** variable occupies two (16-bit) register addresses.

### 1.13.1 Addresses when accessing read / write

#### Note:

The address information in the Modbus devices Manufacturer specifications are often in hexadecimal form. These addresses are to be converted into a decimal address for input in the IBH OPC UA Editors.

### Data type UInt16 and Int16

A word address (16-bit) is used for accessing variables of a data type. Address 0 addresses the variable that occupies the first 16 bits of a data area. With address 1, the variable occupying the second 16 bits of a data area is addressed. Address 3 addresses the third variable (16-bit) of a data area, and so on.

### Data types UInt32, Int32 and Float

Two words (2 x16-bit = 32 bits) are always required for accessing variables of these data types. Address 0 addresses the first 32-bit variable of these data types. Address 1 must not be addressed because this address would address the low word of the 32-bit variable. Address 2 addresses the variable occupying two words 2 and 3 in the data area. The address 4 addresses the third 32-bit variable. etc.

The order of the two 16-bit words, which consist  32-bit swap word order of the data types UInt32, Int32 and Float, can be set.

- **High-Word** – Low-Word or Low-Word – **High-Word**.

### Data type Boolean

A bit address is used to access bits in the data area.

- Address 0 addresses bit 0 in the word address 0 area.
- Address 1 addresses bit 1 in the word address 0 area.
- Address 16 addresses bit 0 in the word address 1 area.
- Address 66 addresses bit 2 in the word address 3 area.

### 1.13.2 Access Read Only

#### Read Discrete Input (read only – Bit access)

Data type Boolean

##### Example: - Fictitious Modbus table (3)

Start Word Address 0111<sub>hex</sub> = Bit  
Address 1110<sub>hex</sub> = 4368<sub>dec</sub> - Read-only  
- Bit Access - Physical inputs, Process  
image. There are 7 input bits to be  
defined as OPC tags.

This setting addresses variables from  
the registers of the **Discrete Inputs**  
whose contiguous status comes from  
digital inputs.

The address of the first variable and  
the number of variables must be specified.

Read\_Discrete\_Inputs\_Boolean

Access:

Read:

Read  
Address: 4368

ReadCoils

ReadDiscreteInputs

Number: 7

**Screenshot:** 7 individual bits are read from bit address 4368 of the  
memory area of the physical inputs.

#### Read Input Registers (Read only)

All data types except Boolean (example: Int16 - 16-bit)

##### Example: - Fictitious Modbus table (4)

Start register address 01E0<sub>hex</sub> =  
480<sub>dez</sub> - Read-only - Word access -  
area **Read Input Registers**. Four  
registers are to be defined as OPC  
tags. The contiguous content of digital  
inputs (analog inputs) is addressed.  
The address of the first variable and  
the number of variables must be  
specified.

Read\_Input\_Registers\_Int16

Access:

Read:

Read  
Address: 480

ReadInputRegisters

ReadHoldingRegisters

Number: 4

**Screenshot:** 4 fixed point numbers from word 480 of the area of the  
**Read Input Register** are read.

#### Read Input Registers (Read Only)

All data types except Boolean (example: UInt16 - 16-bit)

##### Example: - Fictitious Modbus table (7)

Initial register address 400<sub>hex</sub> = 1024<sub>dez</sub> - Read-only - Word access -  
The contents of 10 status registers is addressed as OPC tags.

This setting addresses variables from the diagnostics area (***Input Registers***). The start registers address and the number of registers must be specified.

**Screenshot:** 10 numbers (unsigned) are read with 16 bits each from word 1024 of the **status registers**.

Read\_Input\_Registers\_UInt16

Access:

Read:

Read

Address: 1024

ReadInputRegisters

ReadHoldingRegisters

Number: 10

## Read Input Registers (Read only)

All data types except Boolean (Int16 - 16-bit)

**Example:** - Fictitious Modbus table (8)

Start register address 410hex = 1040dez  
 - Read-only - Word access - The contents of the register of the process image length in bits, of the analog outputs, are to be addressed as OPC tags.

With this setting, variables from the diagnostic area (***input registers***) are addressed.

**Screenshot:** The 16-bit word 1040 of the **register process image** length is read.

Read\_Input\_Registers\_Int16

Access:

Read:

Read

Address: 1040

ReadInputRegisters

ReadHoldingRegisters

Number: 1

## 1.13.3 Access Read and Write

### Read Coils / Write Single Coil

Data type Boolean

**Example:** - Fictitious Modbus table (1)

Register Address 0010hex = Bit Address 100hex = 256dec - Read-Write  
 - Bit Access - Physical Inputs, Process image.

Define 1 input bit as OPC tag. With this setting, variables from the registers of the coils are addressed. The address of the variable and the number of variables (1) must be specified.

Read\_Coils\_Write\_Single\_Coils\_Boolean Unit ID: 255

Access:

Read:

Read

Address: 256

ReadCoils

ReadDiscreteInputs

Write:

Write

Address: 256

WriteSingleCoil

Number: 1  32-bit swap word order

**Screenshot:** 1 single bit is read with bit address 256 of the memory area of the physical inputs.

## Read Coils

### Data type Boolean

#### Example: - Fictitious Modbus table (1)

If such a variable is only defined as a read variable, it is defined as an OPC tag with the status **Read**.

Register Address  $0014_{\text{hex}}$  = Bit Address  $140_{\text{hex}}$  =  $320_{\text{dec}}$  - Read - Bit Access - Physical Inputs, Process image. 8 input bits should be defined as OPC tag.

**Screenshot:** 8 single bits are read from the bit address 320 of the memory area of the physical inputs.

## Read Coils / Write Multiple Coils

### Data type Boolean

#### Example: - Fictitious Modbus table (6)

Register Address  $0310_{\text{hex}}$  = Bit Address  $3100_{\text{hex}}$  =  $12544_{\text{dec}}$  - Read-Write - Bit Access - Physical Inputs, Process image.

12 input bits should be defined as OPC tag.

With this setting, variables are addressed from the registers of the coils whose contents reflect individual bits. These can be individual outputs as well as individual inputs. The address of the first specified variable and the number of variables is specified.

**Screenshot:** 12 individual bits are defined from the bit address 12544 of the memory area of the physical outputs.

## Read Holding Register / Write Single Register

### (All data types except Boolean)

#### Example: - Fictitious Modbus table (2) - Data type INT16 (integer).

Register address  $00D0_{\text{hex}}$  =  $208_{\text{dec}}$  - Read-Write - word access - Physical Inputs, Process image.

One (1) integer number should be defined as OPC tags.

With this setting variables from the ***Holding Registers*** are addressed whose contents reflect individual registers. This can be, for example, an analogue input. The address of the first variable and the number (1) of the variables are specified.

RW\_Holding\_Reg\_Single\_Reg\_Int Unit ID: 255

Access:

Read:  Read Address: 208  
 ReadInputRegisters  
 ReadHoldingRegisters

Write:  Write Address: 208  
 WriteSingleRegister

Number: 1  32-bit swap word order

**Screenshot:** An integer number is defined from the word address 208 of the storage area of the physical outputs.

## Read Holding Registers / Write Multiple Registers

(All data types except Boolean)

**Example: - Fictitious Modbus table (5) - Data type INT16 (integer).**

Register Address 0210<sub>hex</sub> = 528<sub>dez</sub> - Read-Write - Word Access - Physical Outputs, Process image.

9 integer numbers should be defined as OPC tags.

With this setting variables from the ***Holding Registers*** are addressed whose contents reflect individual registers.

These can be individual outputs (analog outputs). The address of the first specified variable and the number of variables is specified.

RW\_Holding\_Register\_Multiple\_Reg\_Int Unit ID: 255

Access:

Read:  Read Address: 528  
 ReadInputRegisters  
 ReadHoldingRegisters

Write:  Write Address: 528  
 WriteMultipleRegisters

Number: 9  32-bit swap word order

**Screenshot:** 9 integer numbers from the word address 528 of the memory area of the physical outputs are defined as OPC tags.

## Read Holding Registers / Write Multiple Registers

(Data types UInt32, Int32, Float)

**Example: - Fictitious Modbus table (9)  
 - Float data type (floating point number).**

Register address 0424<sub>hex</sub> = 1060<sub>dez</sub> - Read-Write - Word access – Watchdog register.

5 floating point numbers are to be defined as OPC tags.

RW\_Holding\_Reg\_Multiple\_Reg\_Float Unit ID: 255

Access:

Read:  Read Address: 1060  
 ReadInputRegisters  
 ReadHoldingRegisters

Write:  Write Address: 1060  
 WriteMultipleRegisters

Number: 5  32-bit swap word order



With this setting, variables from the **holding registers** are addressed, the content of which reflects individual registers.

The address of the first specified variable and the number of variables must be specified.

**Screenshot:** 5 floating point numbers from word address 1120 are defined in the **holding registers** as OPC tags.

**Note:**

**UInt32**, **Int32**, and **Float** numbers occupy two (2) 16-bit words (32-bit). The order of the two 16-bit words can be set.

- **High-Word – Low-Word**      or      **Low-Word – High-Word**

## Read Holding Registers

(All data types except Boolean)

If such a variable is only defined as a read variable, it is defined as an OPC tag with the status **Read**.

**Example: - Fictitious Modbus table (10) - Data type UINT16 unsigned integer number.**

Register address  $0430_{\text{hex}} = 1072_{\text{dez}}$  - Read-Write - word access - special register.

6 unsigned integer numbers should be defined as OPC tags.

With this setting variables from the **Holding Registers** are addressed whose contents reflect individual registers. The address of the first specified variable and the number of variables is specified.

**Screenshot:** 6 numbers (unsigned) from the word address 1104 of the memory area of the special registers are defined as OPC tags.

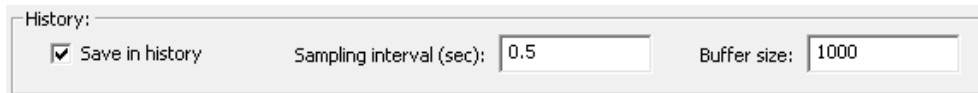
### 1.13.4 Analog-limits

Analog limit values can be specified.

### 1.13.5 History

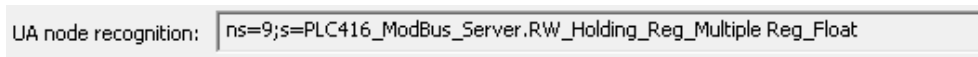
While **OPC Data Access** provides real-time access to data, **OPC Historical Data Access**, also known as OPC HDA, supports access to data already stored.

Activation of a variable as historical data as well as sampling interval and number of values (buffer size) is done via the dialog box.



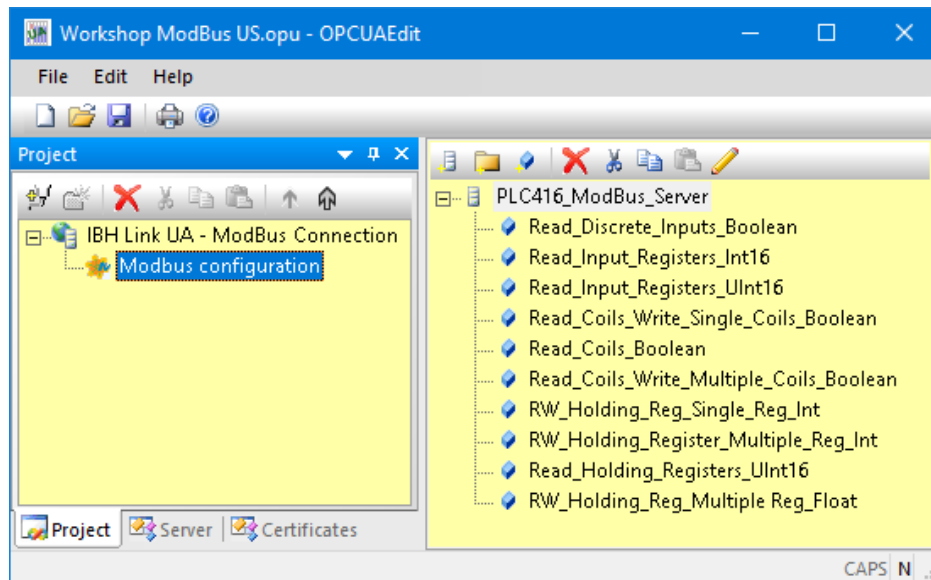
### OPC UA Node detection

The OPC UA Node name of a variable is displayed in the dialog box.



## 1.14 Transfer Modbus configuration to the OPC UA server (IBH Link UA)

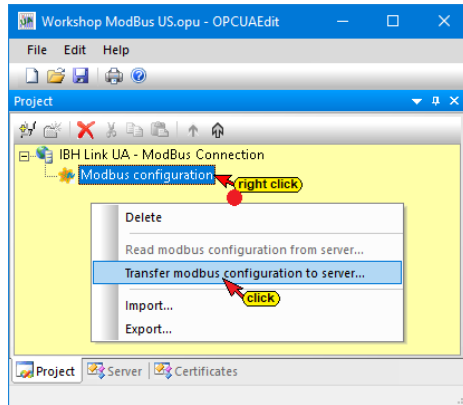
The variables defined as OPC tags are displayed.



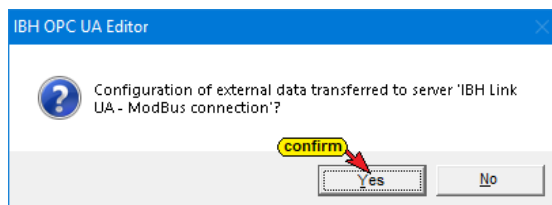
Name	Data type	Access	R address	W address	Number of	Node name
Read_Discrete_Inputs_Boolean	Boolean	R	4368		7	PLC416_ModBus_Server.Read_Discrete_Inputs_Boolean
Read_Input_Registers_Int16	Int16	R	480		4	PLC416_ModBus_Server.Read_Input_Registers_Int16
Read_Input_Registers_UInt16	UInt16	R	1024		10	PLC416_ModBus_Server.Read_Input_Registers_UInt16
Read_Coils_Write_Single_Coils_Boolean	Boolean	RW	256	256	1	PLC416_ModBus_Server.Read_Coils_Write_Single_Coils_Boolean
Read_Coils_Boolean	Boolean	R	320		8	PLC416_ModBus_Server.Read_Coils_Boolean
Read_Coils_Write_Multiple_Coils_Boolean	Boolean	RW	12544	12544	12	PLC416_ModBus_Server.Read_Coils_Write_Multiple_Coils_Boolean
RW_Holding_Reg_Single_Reg_Int	Int16	RW	208	208	1	PLC416_ModBus_Server.RW_Holding_Reg_Single_Reg_Int
RW_Holding_Register_Multiple_Reg_Int	Int16	RW	528	528	9	PLC416_ModBus_Server.RW_Holding_Register_Multiple_Reg_Int
Read_Holding_Registers_UInt16	UInt16	R	1104		6	PLC416_ModBus_Server.Read_Holding_Registers_UInt16
RW_Holding_Reg_Multiple_Reg_Float	Float	RW	1120	1120	5	PLC416_ModBus_Server.RW_Holding_Reg_Multiple_Reg_Float

If all Modbus variables are defined as OPC UA tags, the Modbus configuration can be transferred to the OPC UA server.

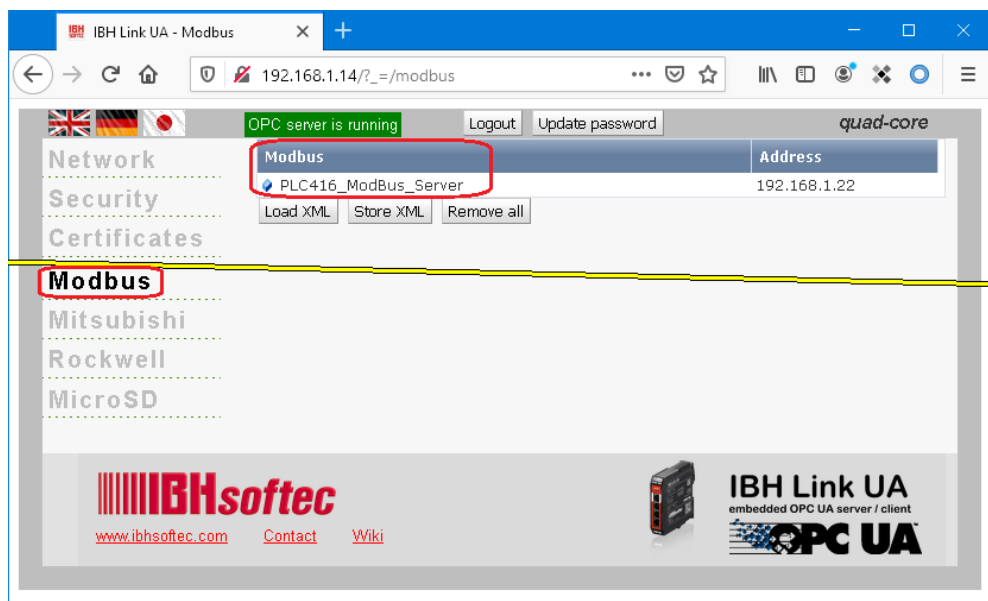
A right-click on **Modbus configuration** opens the context menu.



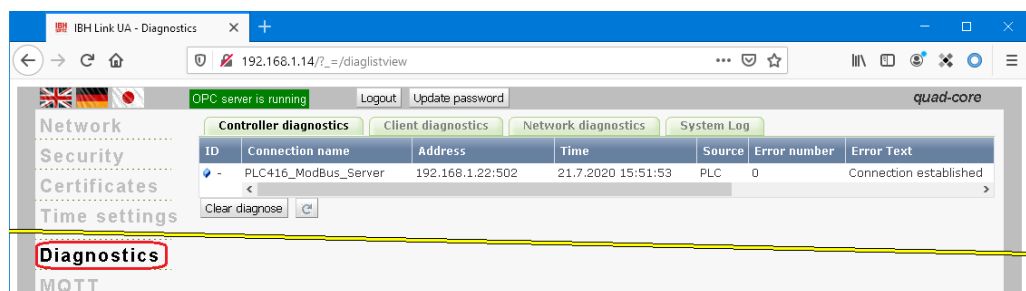
The command to transfer the Modbus configuration must be confirmed.



The transmitted Modbus configuration is displayed under **Modbus** in the IBH Link UA Browser window.



The Modbus device connection is displayed under **Diagnostics** in the IBH Link UA Browser window.

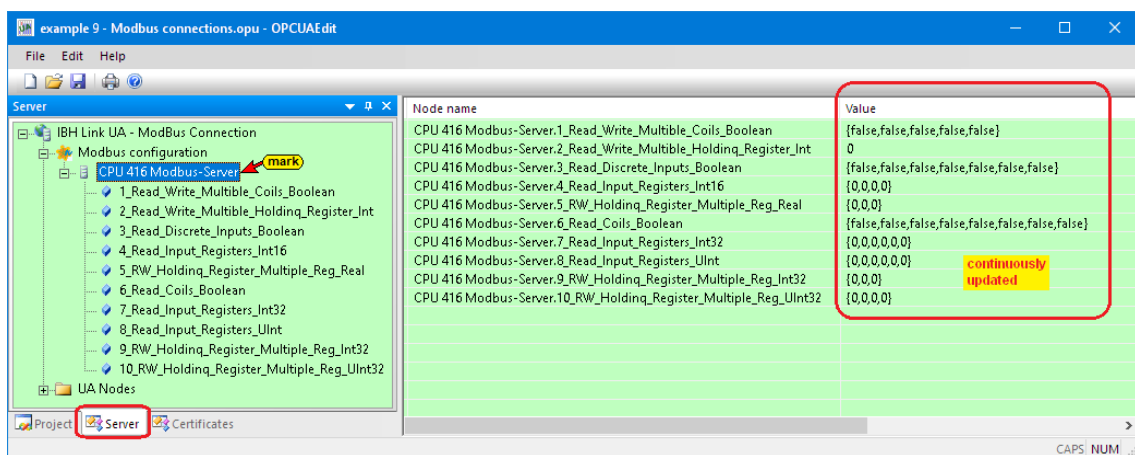
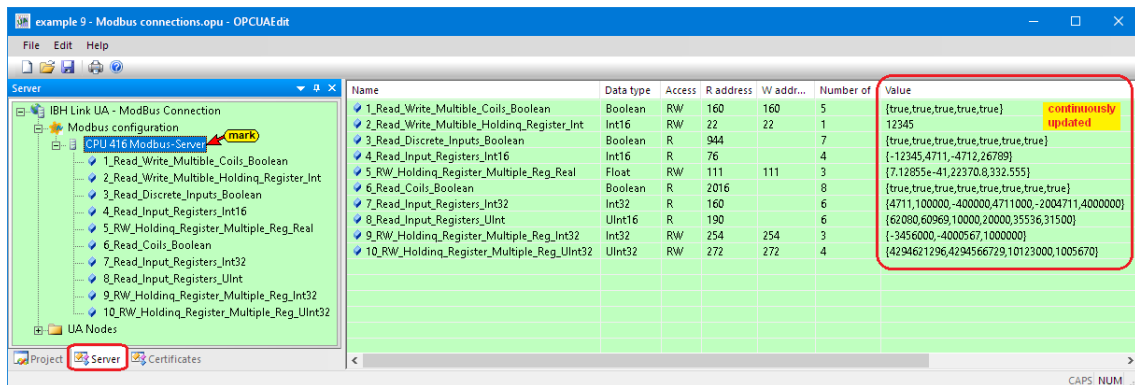


## 1.15 IBH OPC UA Editor Server Window

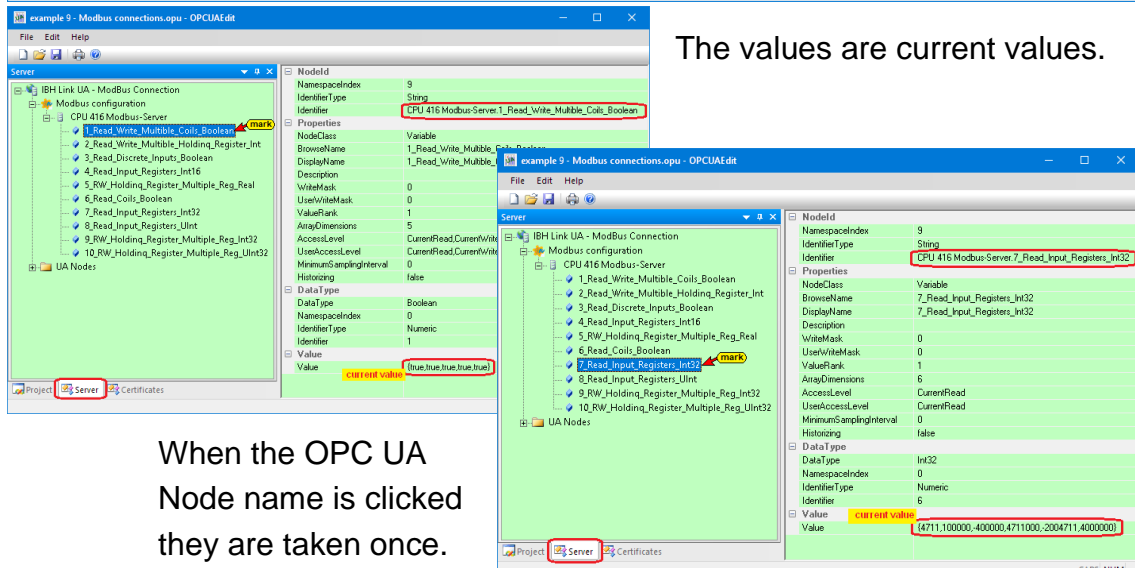
The **Modbus configuration** successfully transmitted to the OPC UA server can be displayed online in the server window.

The variables (Modbus configuration, Modbus device, OPC tag) are listed in the left-hand server window.

Clicking on a variable displays the variable definitions with the status in the right-hand server window. The status of this OPC tag is constantly being updated.



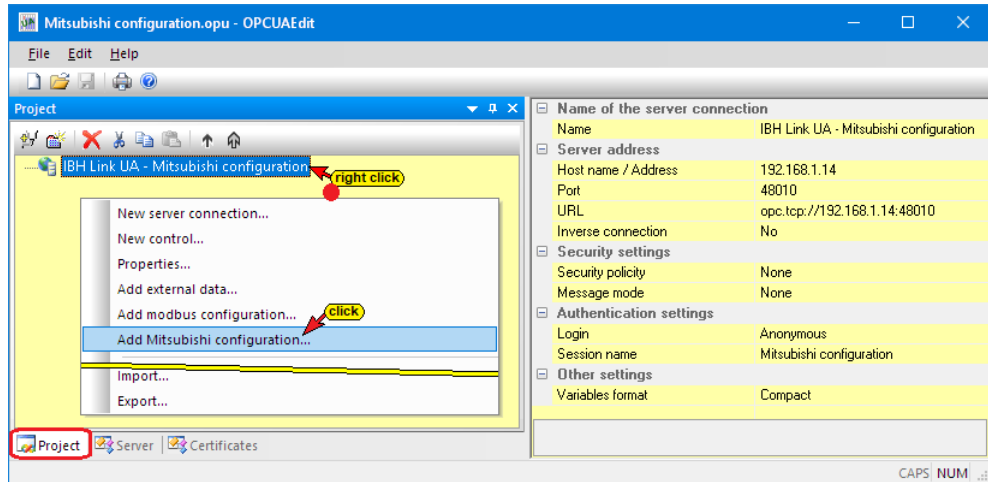
The values are current values.



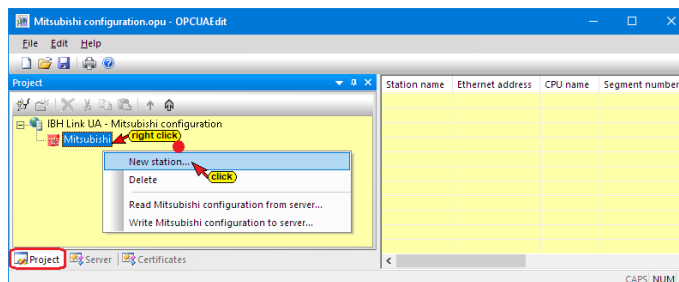
When the OPC UA Node name is clicked they are taken once.

## 1.16 Mitsubishi configuration

If an OPC server connection has been created with the IBH OPC UA Editor, a Mitsubishi configuration can be added. The **Add Mitsubishi configuration** command starts the configuration process.

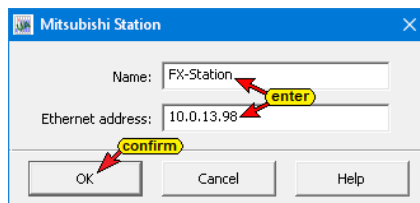


Click **Add Mitsubishi configuration** to open the context menu.

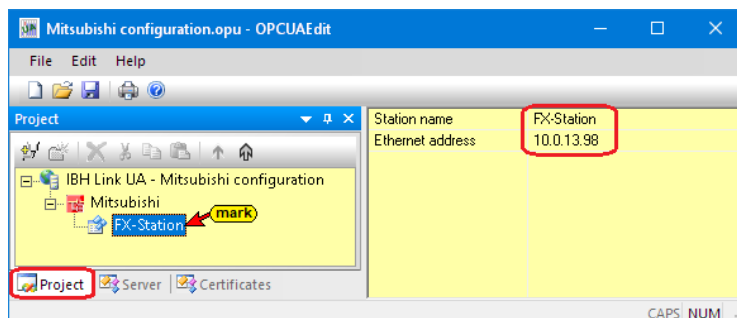


A new Mitsubishi configuration is to be created.

Clicking **New Station ...** opens a dialog box for entering the station name and its Ethernet address.

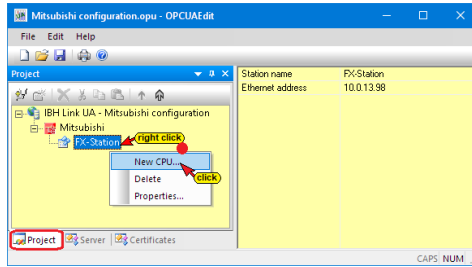


The Mitsubishi FX station is entered as a project

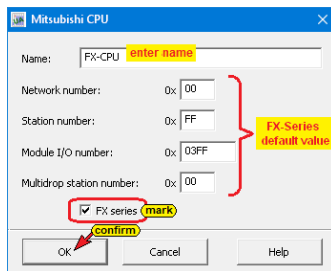


## Insert new CPU

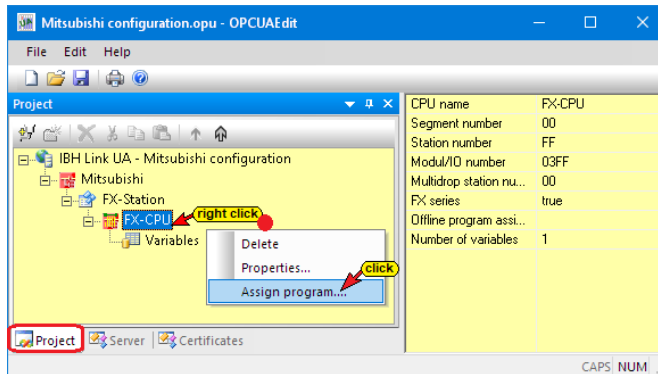
A CPU is to be inserted into the Mitsubishi FX station.



The command **New CPU...** opens the dialog box for defining the CPU.



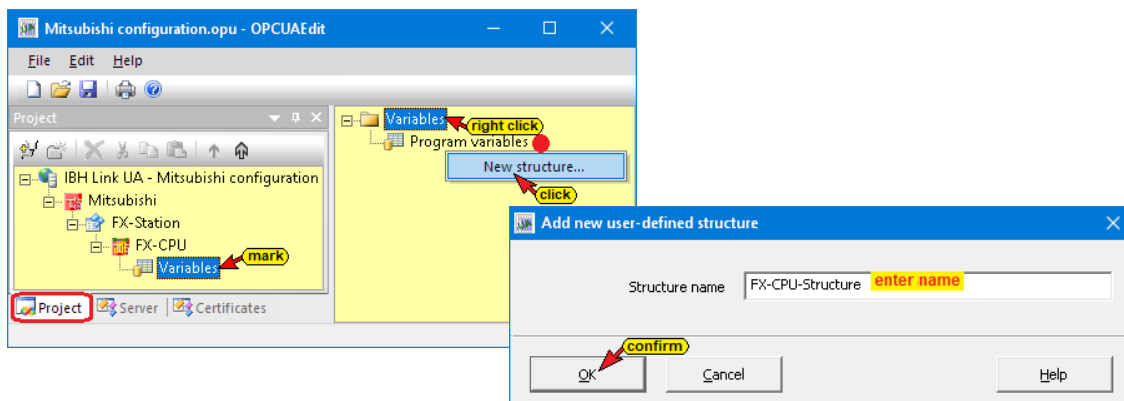
A program can be assigned to the inserted CPU.



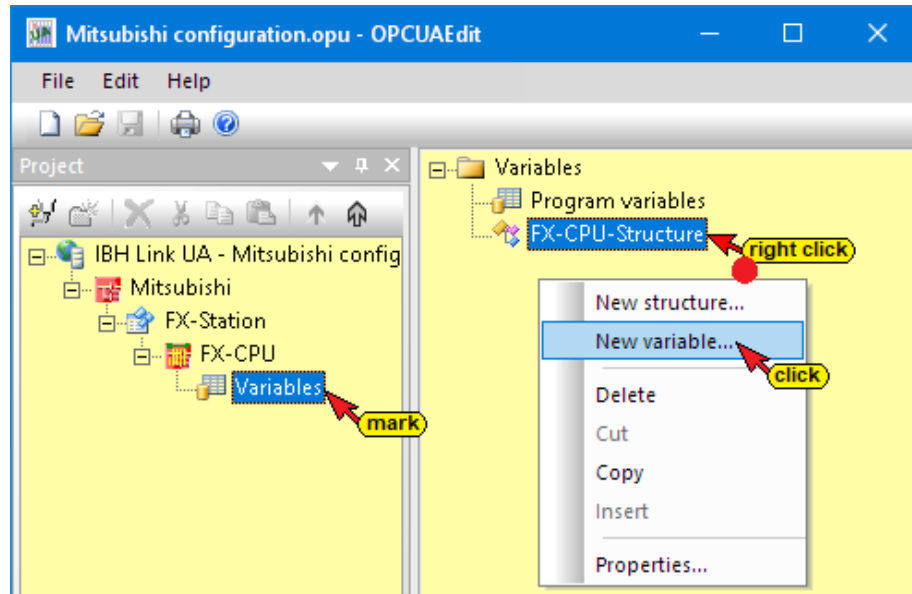
The **Assign program ...** command opens the dialog box for selecting a Mitsubishi project file with the file extension **gxw** or **gx3**.

The variables of the Mitsubishi project may be adopted as OPC tags.

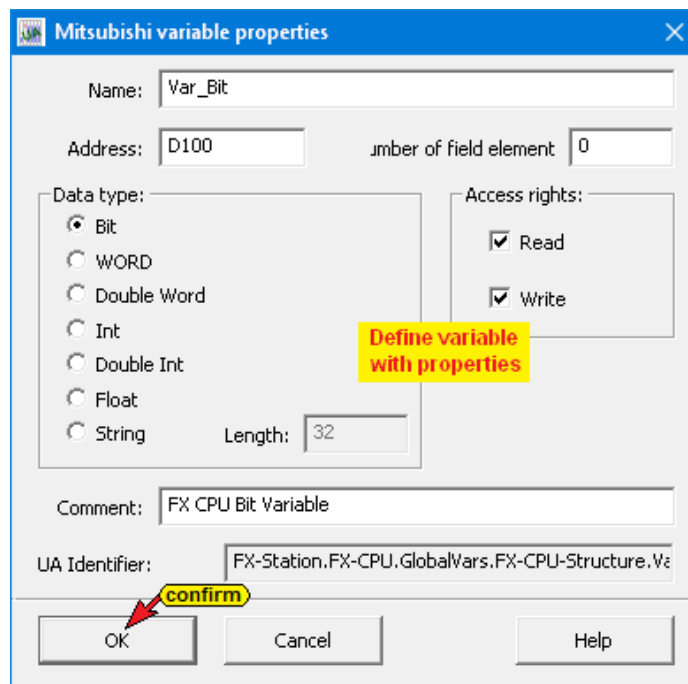
In addition to the program variables, structures can be created additionally or exclusively.



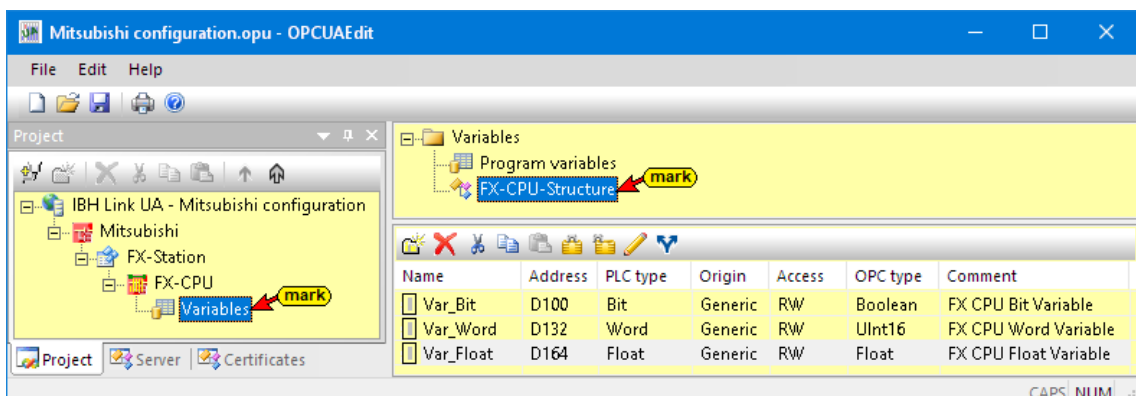
In the created structure, variables can be defined in the.



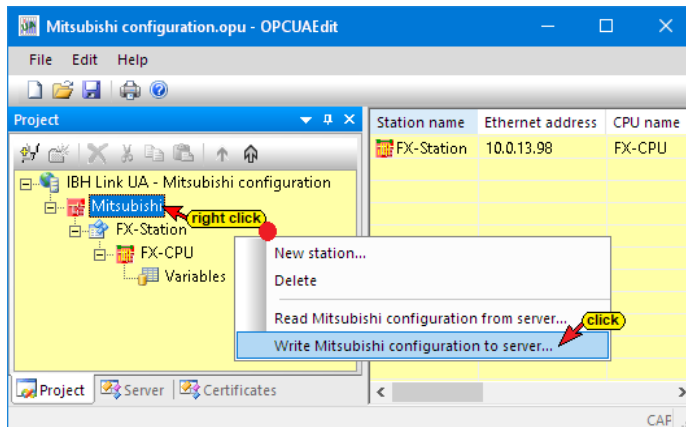
The dialog box for defining the variable opens.



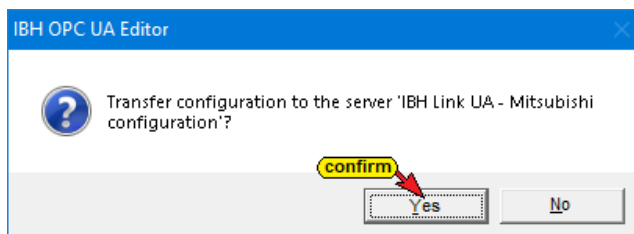
The defined variables (OPC tags) are listed in the right project window.



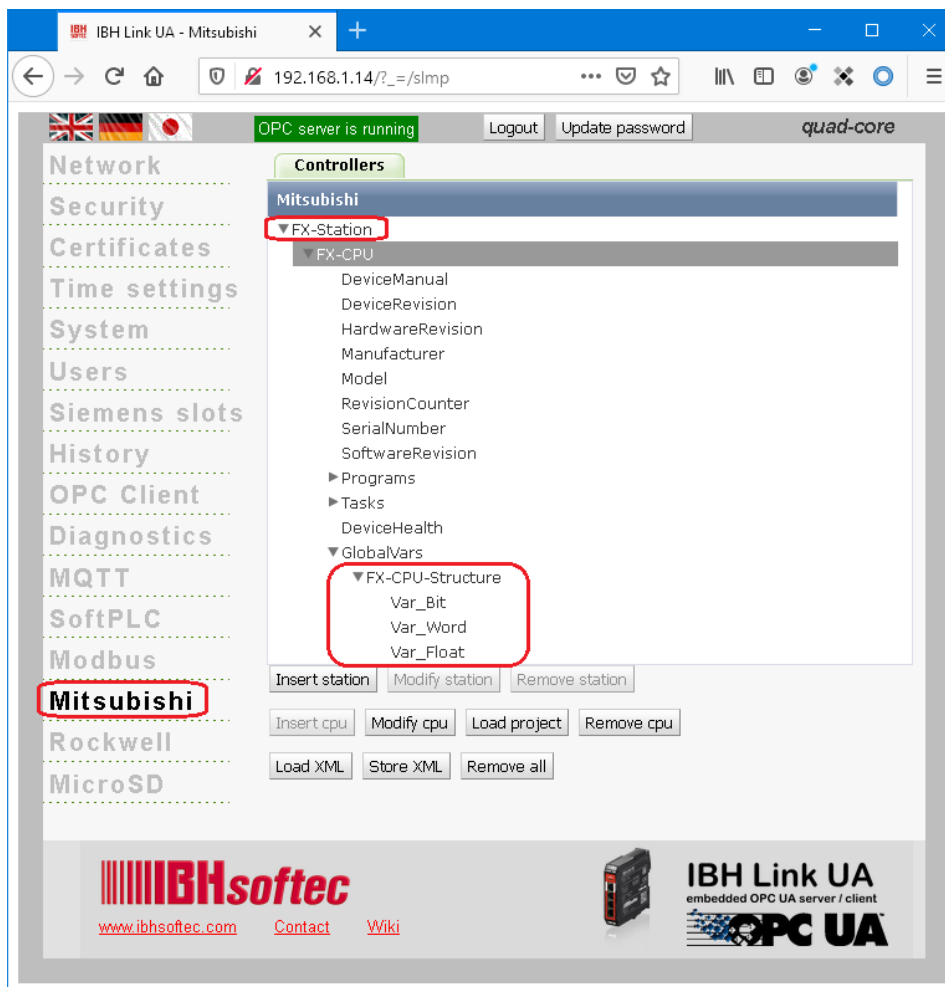
A right-click on **Mitsubishi** opens the context menu.



The command to transfer the Mitsubishi configuration must be confirmed.



The transmitted Mitsubishi configuration is displayed under **Mitsubishi** in the **IBH Link UA Browser** window.





## 2 IBH OPC UA Editor - Configuration Examples

---

The following are examples of how to handle the **IBH OPC UA Editor**. All examples are summarized as one project and are transferred as such to the IBH Link UA. The PLC example projects are in the file **IBH Link UA Editor – Examples**.

### Example 1 - Project: CPU 416 S7

Connect CPU 300 / CPU 400 with TCP / IP port to IBH Link UA. The PLC project is available as a STEP 7 (Simatic Manager) program.

In the example, the SoftPLC CPU 416 is used.

### Example 2 - TANK\_PST.S5D or tank level S5W.s5p:

The S5 CPU 103U PLC program is available as SIMATIC S5 or **S5 for Windows** project. The S5 CPU is connected to the IBH Link UA via an IBH Link S5 ++.

### Example 3 - Project: CPU 300 TIA 16:

Connect CPU 300 / CPU 400 directly to IBH Link UA via IBH Link S7 ++. The PLC program (CPU300 TIA16) is available as a TIA16 project.

### Example 4 - Project: CPU 1200 TIA 16:

Connect CPU 1211C with TCP / IP port to IBH Link UA.

The PLC program (CPU 1200 TIA16) is available as a TIA16 project.

### Example 5 - project: CPU 416 TIA 16 server - server

In the project, the CPU 416, which is available as an OPC UA server, is connected to an air conditioning system that also has an OPC UA server.

Establishing a server connection. The data of an air conditioning system (OPC UA server) is sent to the CPU-416 (OPC UA server).

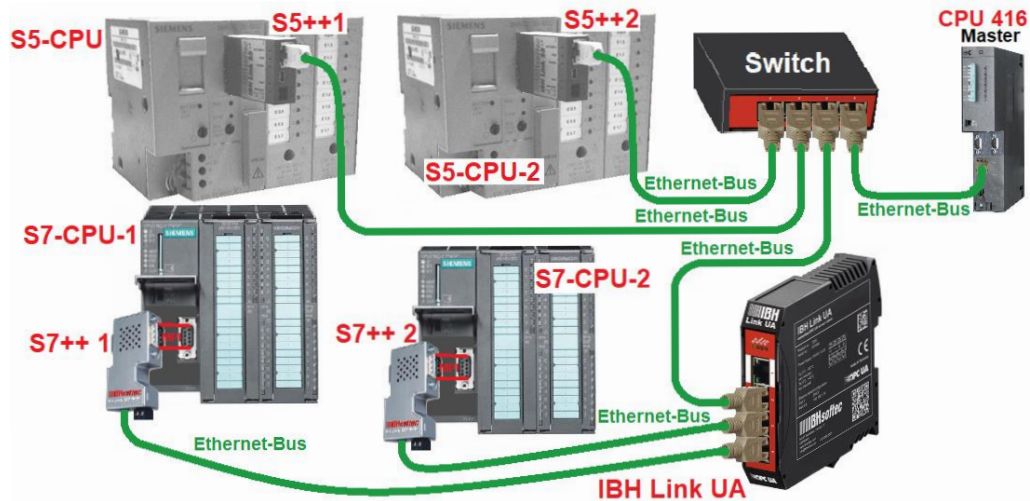
### Example 6 - Project: CPU 1200 connected to CPU 1500 TIA 16

A CPU 1211C with a TCP / IP port is connected to a CPU 1511-1 PN, which is also a TCP / IP port.

### Example 7 - Data exchange between several S7 / S5 CPUs

On a system with three (3) S7 CPU's and two (2) S5 CPU's, data should be exchanged with each other. An S7 CPU that has an

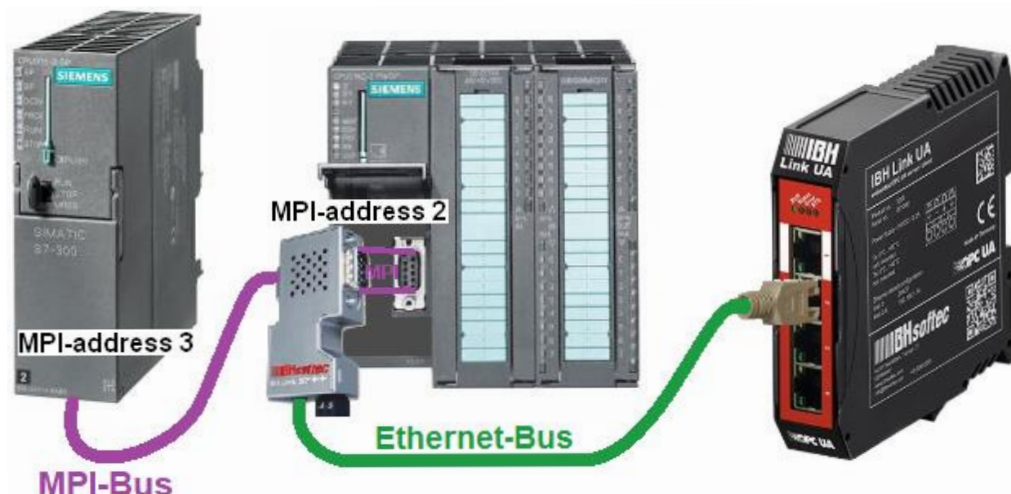
Ethernet connection acts as a master. The other CPUs have no Ethernet interfaces and are connected to the IBH Link UA via IBH Link S7 ++ or IBH Link S5 ++. The master provides data for all CPUs, which provide information to the master.



### Example 8 - Connection of two S7 CPU 300 via an IBH Link S7 ++

S7-300 series CPUs not having a free Ethernet port may be connected to the IBH Link UA via MPI bus via an IBH Link S7 ++ via Ethernet (protocol RFC 1006).

The example shows the creation of a project with the connection of two (2) CPU 312 to one (1) IBH Link UA via IBH Link S7 ++. Instead of the CPU 312, any other S7 CPU 300/400 that does not have a free Ethernet port could be used.

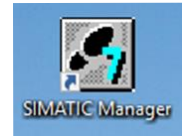


### Example 9 - Modbus

The IBH SoftPLC PLC416 has the possibility of a Modbus connection. In the example, variables are defined as OPC tags. This Modbus configuration is transferred to the IBH Link UA and the variables are displayed in the **UAExpert client program**.

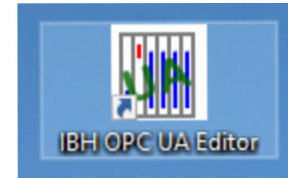
## 2.1 Example 1: CPU 416 S7

CPU 416-3 PN/DP (SoftPLC) - with the program **Counter** (CPU 416) in the project **CPU416 S7** shall be connected to the **IBH Link UA**.

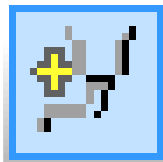
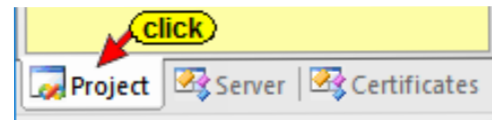


### Calling the IBH OPC UA Editor

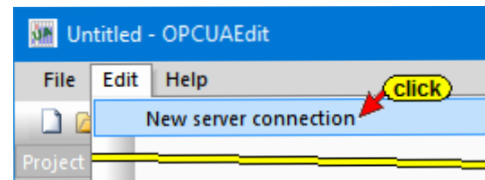
Double-click the **IBH OPC UA Editor** icon to open the program window.



Open the **Project window** by clicking on the **Project** tab.



Open the **New Server Connection** dialog box with the New Server Connection command from the **Edit** menu or by clicking the icon.



### 2.1.1 Server connection

Settings for the connection to the **IBH Link UA** OPC UA Server are done in the **New server connection** dialog box.

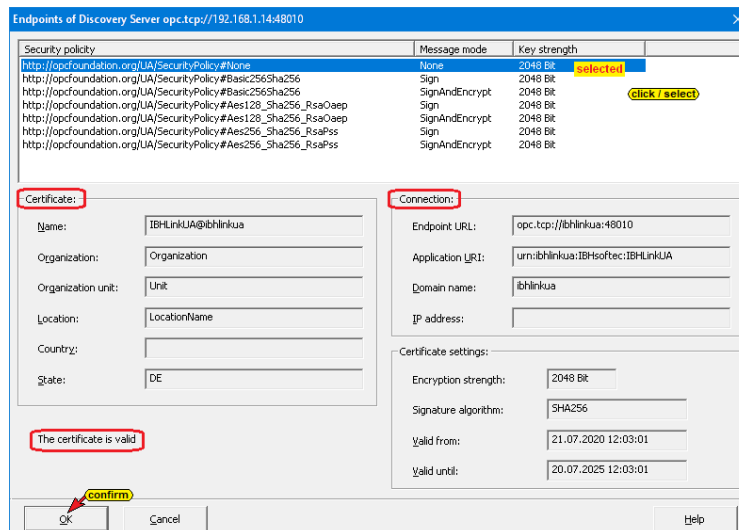
The 'Server connection properties' dialog box is shown with the following settings and annotations:

- Name of the server connection: **IBH\_Link\_UA** (highlighted)
- Server address:
  - Host name or IP address: **192.168.1.14** (highlighted)
  - Port: **48010** (highlighted)
  - URL: **opc.tcp://192.168.1.14:48010** (highlighted)
  - Annotation: **click** (pointing to the 'Select endpoint...' button)
- Security settings:
  - None** (highlighted)
  - Other options: Basic128Rsa15, Basic256, BasicSha256, Aes128Sha256RsaOaep, Aes256Sha256RsaPss
  - Message mode:
    - None (selected)
    - Signature
    - Signature and Encryption
  - Inverse connection:
    - Connect invers (checkbox)
    - Properties... (button)
- Login:
  - Anonymous (selected)
  - User name and password (checkbox)
  - User name: (text field)
  - Password: (text field)
  - Store (checkbox)
- Session Name: **Example 1: CPU 416 S7** (highlighted)
- Variables format: **Compact** (highlighted)
- Annotation: **open / select** (pointing to the dropdown arrow for Session Name)
- Annotations: **confirm** (pointing to the OK button)

## Check connection to the IBH Link UA

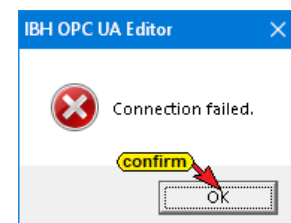
Click **Select endpoint ...**. A connection to the online connected IBH Link UA is established. If the connection is successful, the **Discovery Server Endpoints to opc** dialog box opens.

The possible security settings of the data to be transmitted are displayed here for selection. The existing certificates in the OPC UA server with their settings and the connection path are displayed.

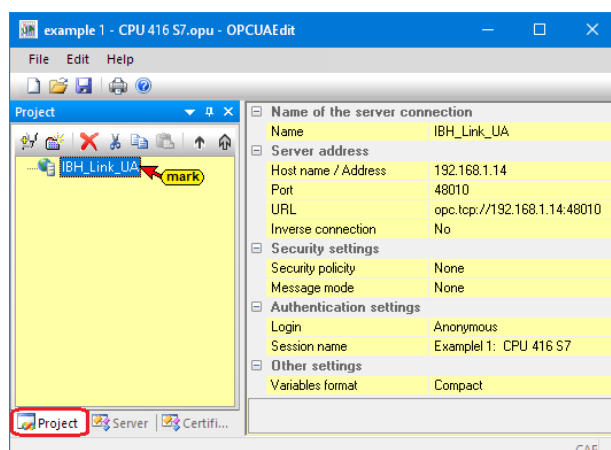


In the example, the security procedure **None** is used. Additional settings are not required. The necessary settings have already been made in the **Server connection properties** dialog box. Click **OK** to close the dialog box.

If a connection to the online connected **IBH Link UA** cannot be established, this will be displayed.

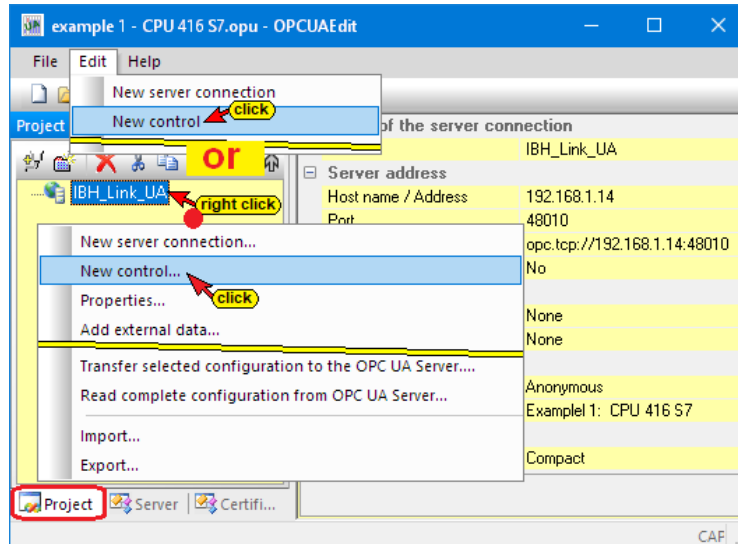


Is the **Server connection properties** dialog box closed, the specified settings for the connection to the **OPC UA server** are displayed in the right part of the project window.

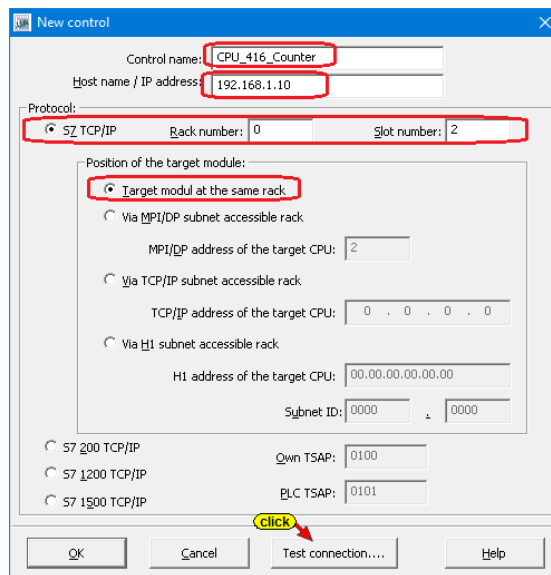


## 2.1.2 Insert new control

The **New Control** command from the context menu (or menu Edit / New control) opens the dialog box **New control** to specify the access to the control (CPU).



### New control dialog box

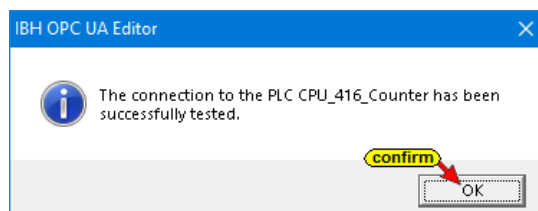


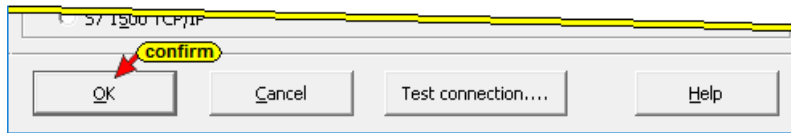
### Test connection

After completing the New Control dialog box, the connection to the online connected CPU can be tested.

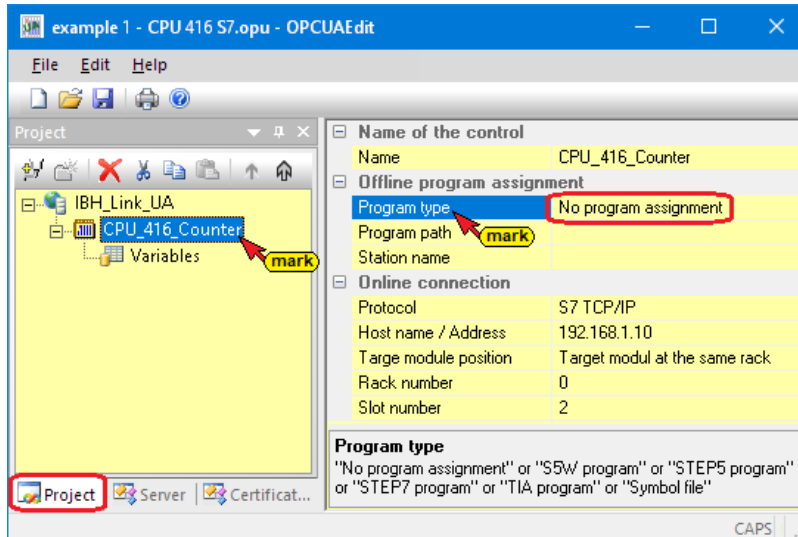


Information about the successful connection is displayed.





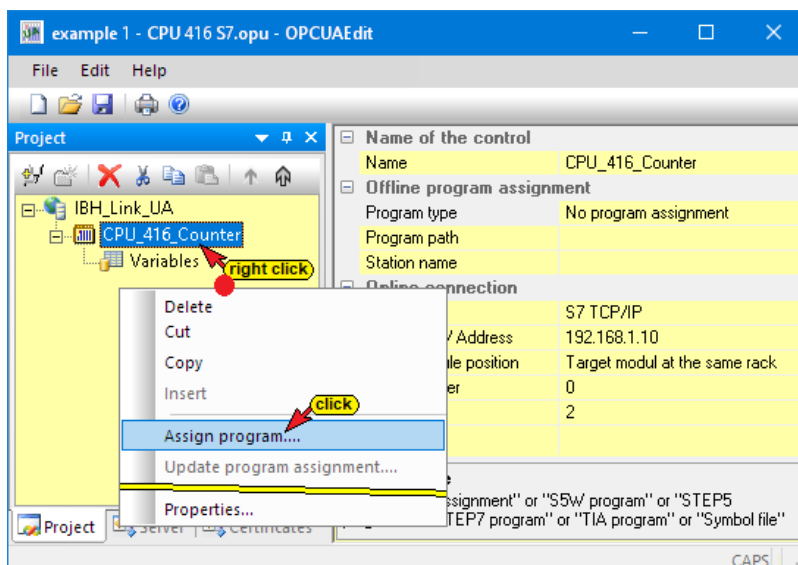
To accept and close the **New control** dialog box settings click on **OK**.



The access data of the **CPU\_416\_Counter** (CPU 416-3 PN / DP SoftSPS with TCP/IP Port) is displayed in the right project window.

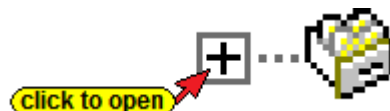
### 2.1.3 Program assignment

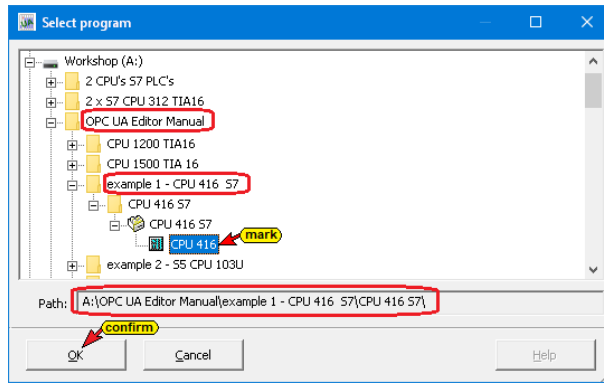
Open the **Program selection** dialog box with the **Assign program** command.



#### **Program selection**

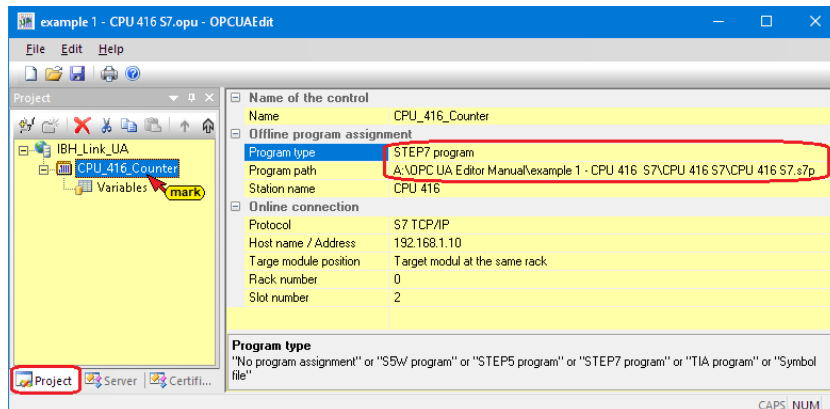
Select the PLC program in the open **Select program** dialog box. By clicking on the **Plus** icon in front of the **CPU 416 S7** project icon, the PLC program (CPUs) is displayed in the project.





Clicking **OK**, the variables, data and program information are transferred to the **OPC UA Editor**.

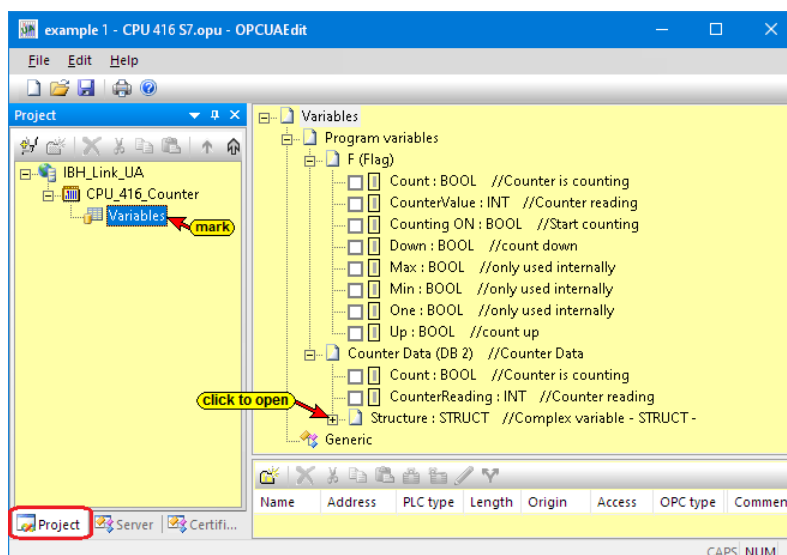
In the right part of the project window the **Offline program assignment** is displayed under.



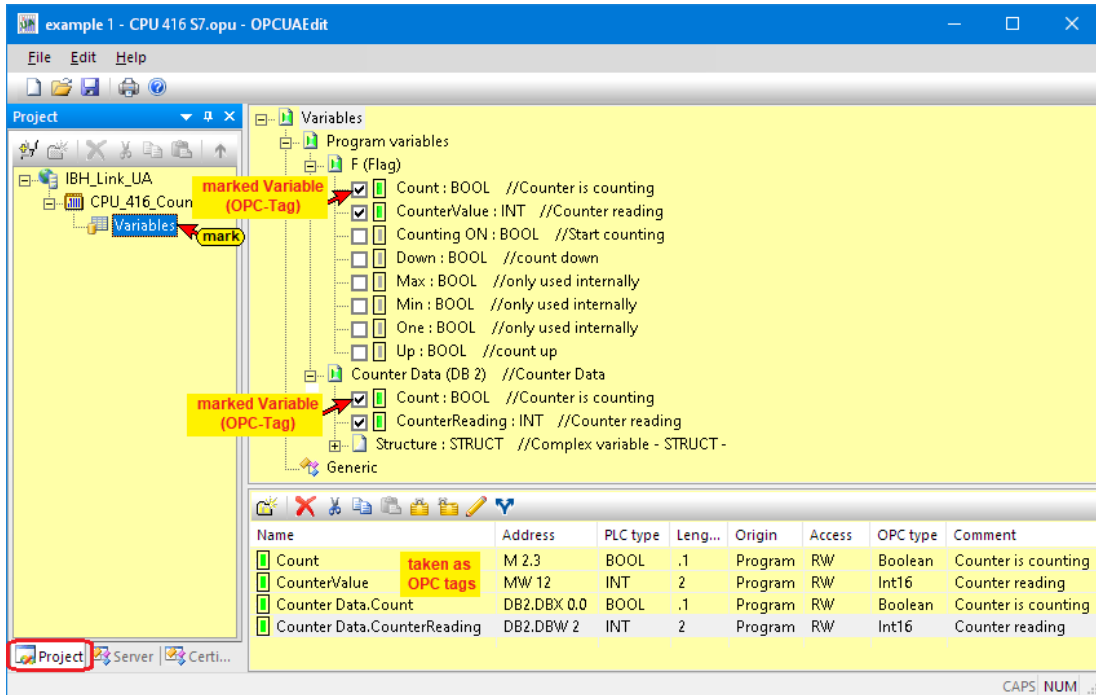
## Define variables as OPC tags

Clicking on **Variables**, these, and data from Data Blocks of the acquired PLC program are listed in the right part of the project window.

Clicking the symbol **Plus** in front of the symbol of the variable area, the existing variables are listed.



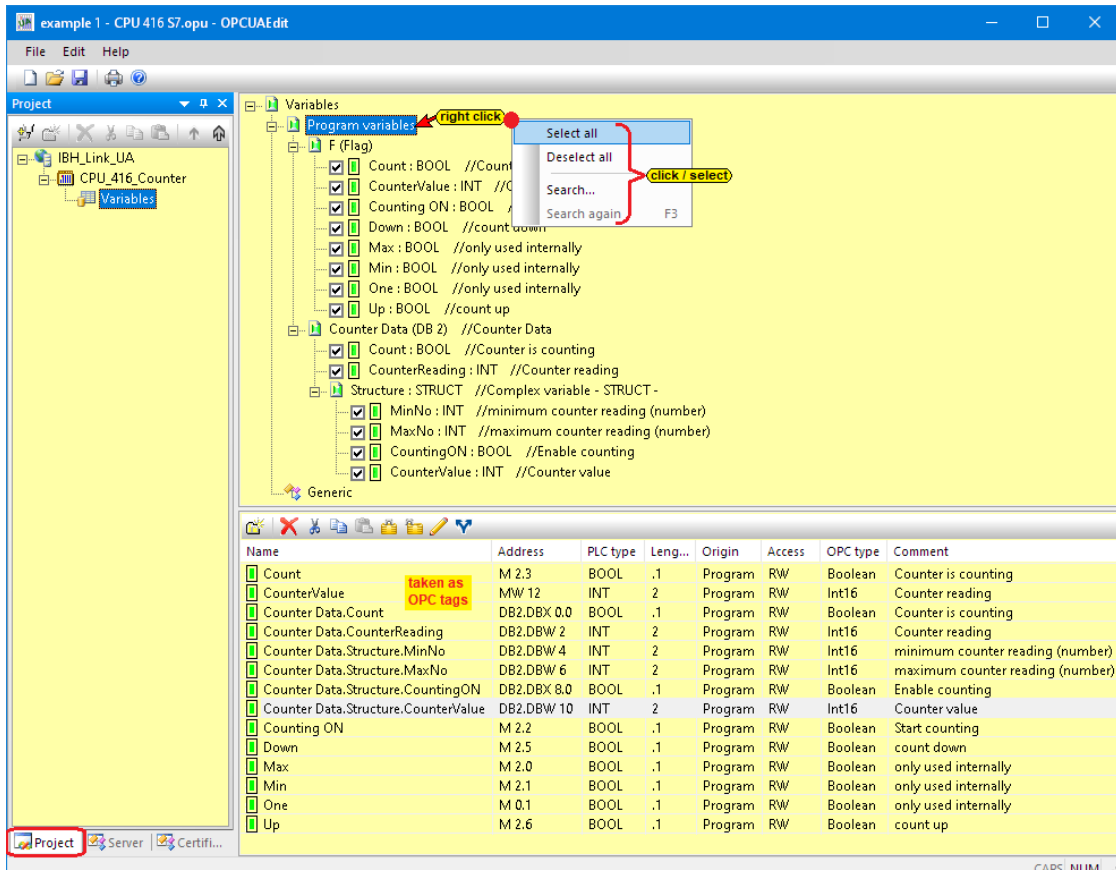
If a variable is selected, this is accepted as an OPC tag and is displayed in the lower part of the window with additional information.



### Select / search OPC tags

The context menu offers the commands to select all tags as OPC tags (select all) or deselect (deselect all).

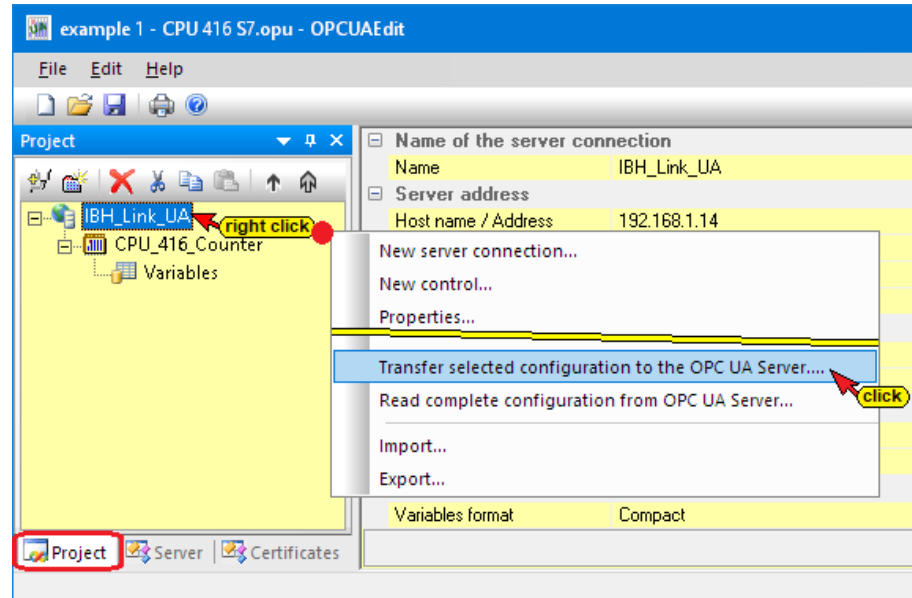
A search function is available to handle extensive variable lists.



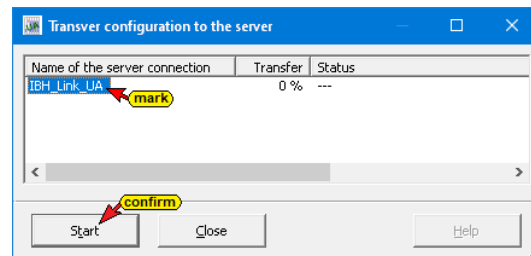


## 2.1.4 Transfer configuration to the OPC UA server (IBH Link UA)

A right-click on the **Server** icon (IBH Link UA) opens the context menu.

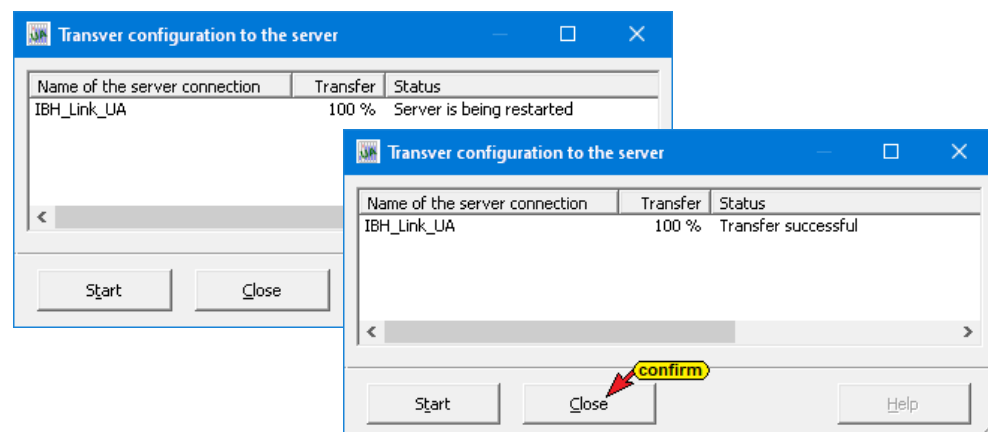


The command **Transfer Selected Configuration to OPC UA Server** command opens the **Transfer Configuration to Server** dialog box.



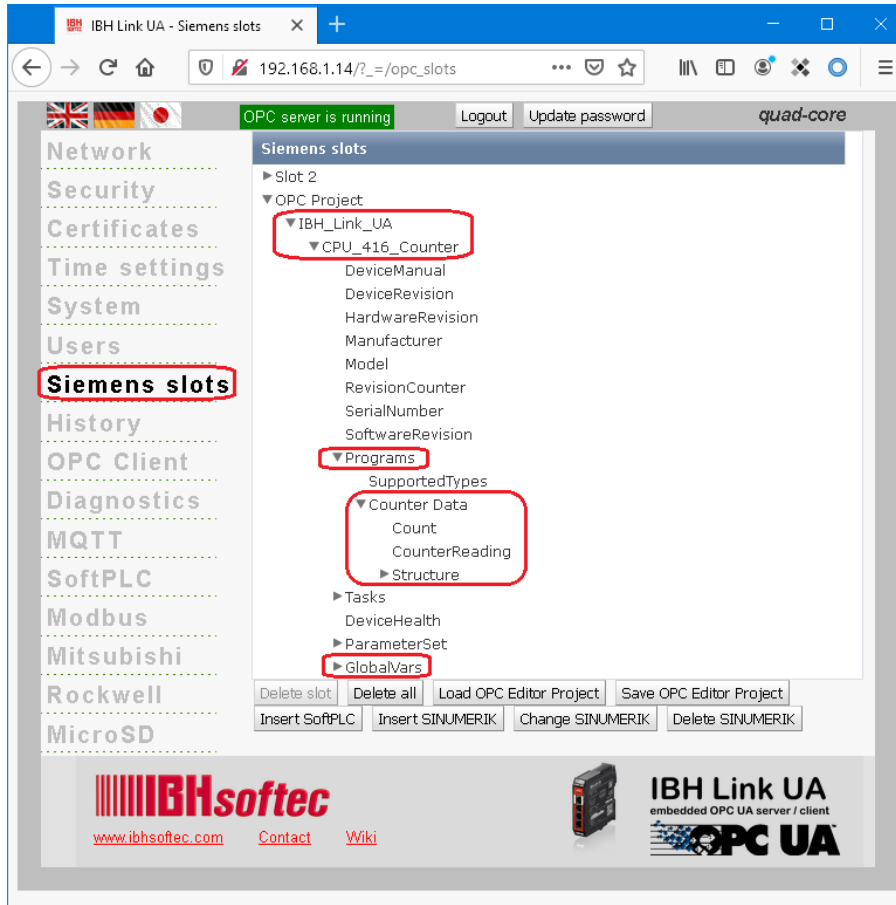
Select the server and then click Start.

If a certified data exchange has been agreed between the **IBH OPC Editor** and the **IBH Link UA**, the exchanged certificates must be confirmed as **trusted** (see Trust certificates; chapter 1 page 1-40).



The successful transfer is displayed.

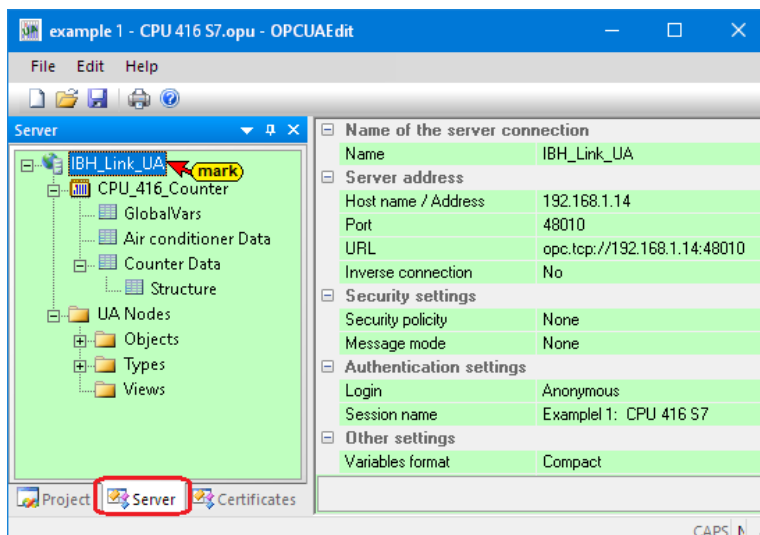
## IBH Link UA - Siemens Slots project CPU 416 Counter



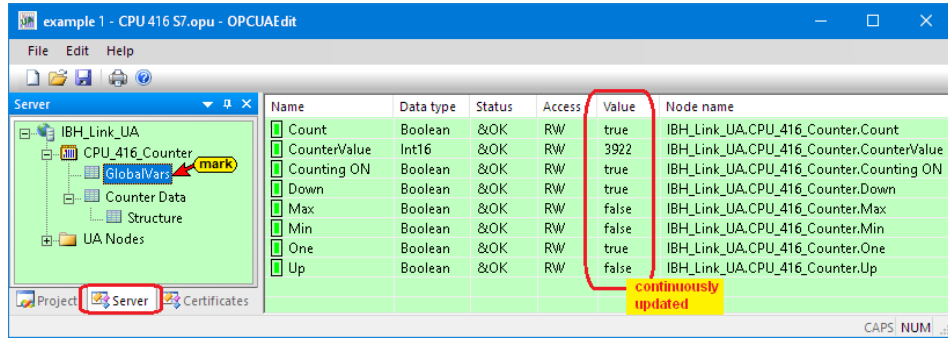
### 2.1.5 Online OPC UA Server Information

Information from the online connected OPC UA server with the online connected CPUs is displayed.

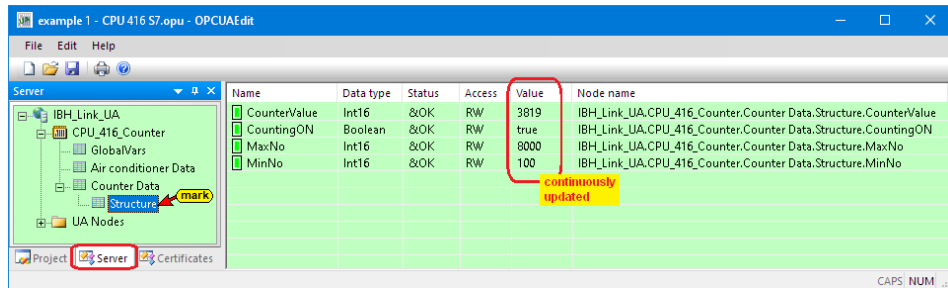
The groups of the variables (GlobalVars, data blocks) are listed in the left server window. By clicking on a group, the individual variables (OPC tags) are displayed in the right server window with their status. The status of the OPC tags is continuously updated.



## GlobalVars



## Data Block DB2 – Counter Data / Structure

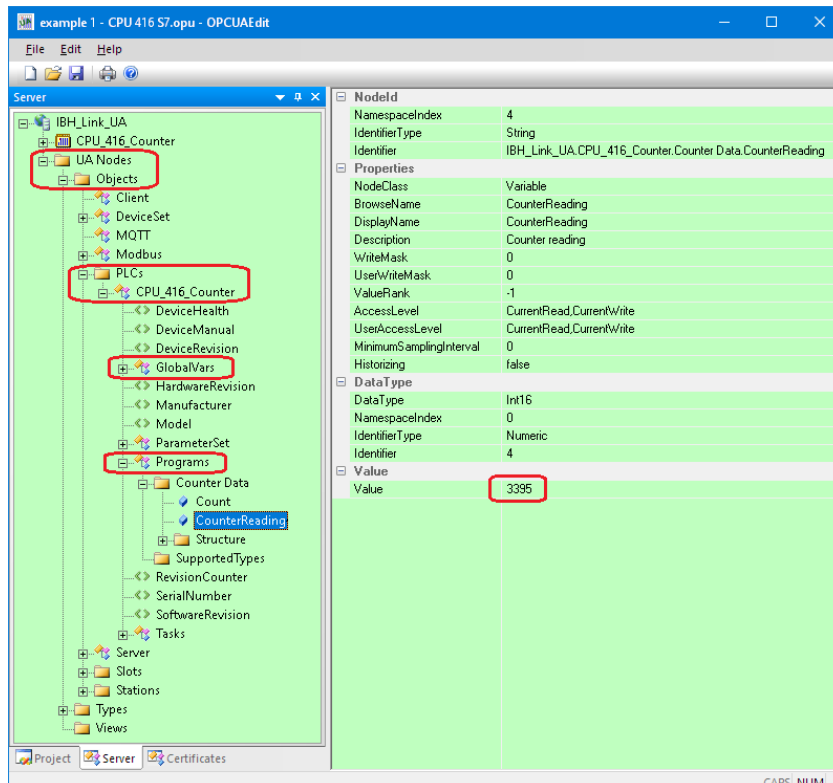


## Show under UA Nodes

The names of the OPC UA Nodes are listed in the left-hand server window (attributes, OPC tags, etc.).

The corresponding values are displayed in the right-hand server window.

The values are current values and are only taken once when the OPC UA Node name is clicked.

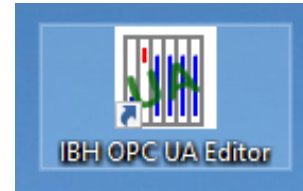


## 2.2 Example 2: S5 CPU 103U

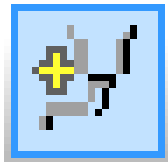
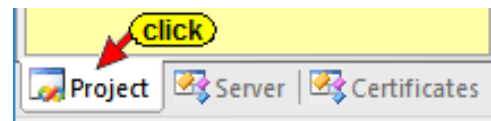
Connect an S5 CPU 103 U to the IBH Link UA using IBH Link S5 ++. The S5 CPU 103U PLC program is available as a SIMATIC S5 (*COUNT\_ST.S5D*) or *S5 for Windows (Counter S5W.s5p)* project

### Calling the IBH OPC UA Editor

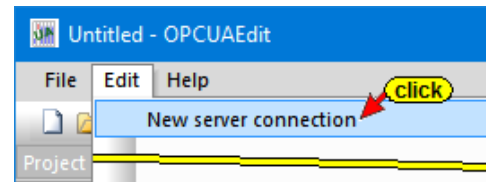
Double-click the *IBH OPC UA Editor* icon to open the program window.



Open the *Project window* by clicking on the *Project* tab.



Open the *New Server Connection* dialog box with the New Server Connection command from the *Edit* menu or by clicking the icon.



The new server connection setup was explained in example 1 (see chapter 2, page 2-3).

Server connection properties

Name of the server connection: IBH\_Link-UA

Server address:

Host name or IP address: 192.168.1.14

Port: 48010

URL: opc.tcp://192.168.1.14:48010

Security settings:

None

Basic128Rsa15

Basic256

BasicSha256

Aes128Sha256RsaOaep

Aes256Sha256RsaPss

Message mode:

Signature

Signature and Encryption

Inverse connection:

Connect invers

Properties...

Login:

Anonymous

User name and password

User name:

Password:

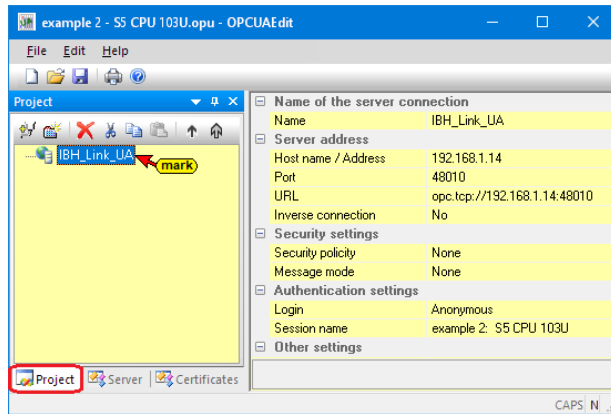
Store

Session Name: example 2: S5 CPU 103U

Variables format: Compact

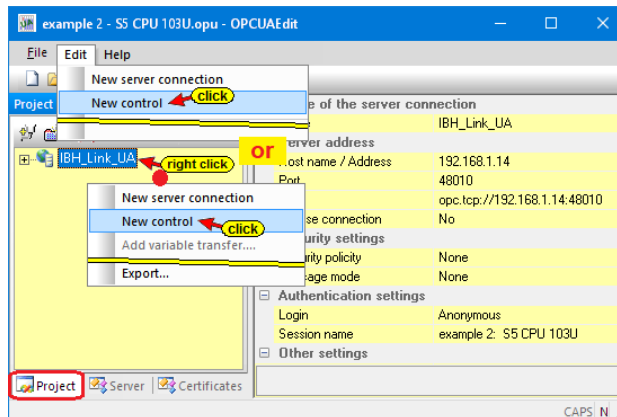
OK Cancel Help

The settings for the connection to the *IBH Link UA* OPC UA server are displayed in the right part of the *project window*.

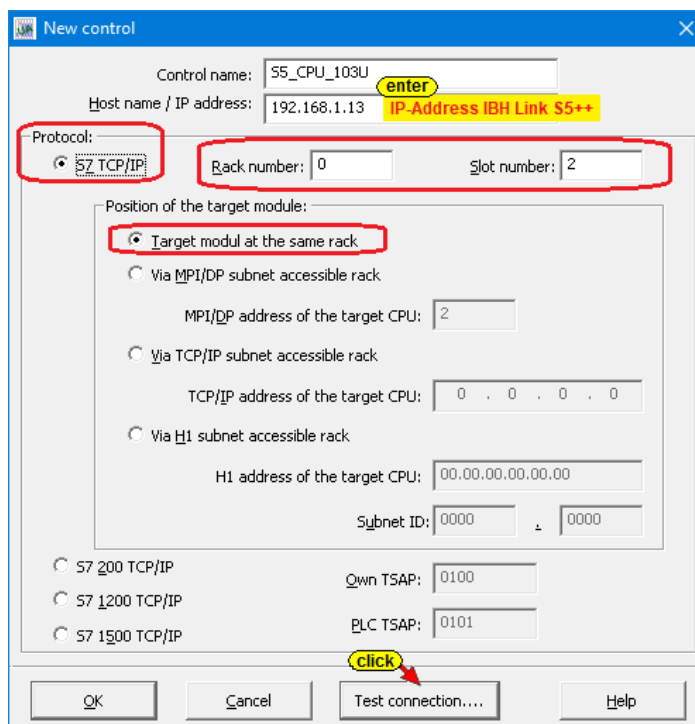


## 2.2.1 Inserting a New control (PLC)

The **New Control** command from the context menu (or menu Edit / New control) opens the dialog box **New control** to specify the access to the control (CPU).



## New control dialog box

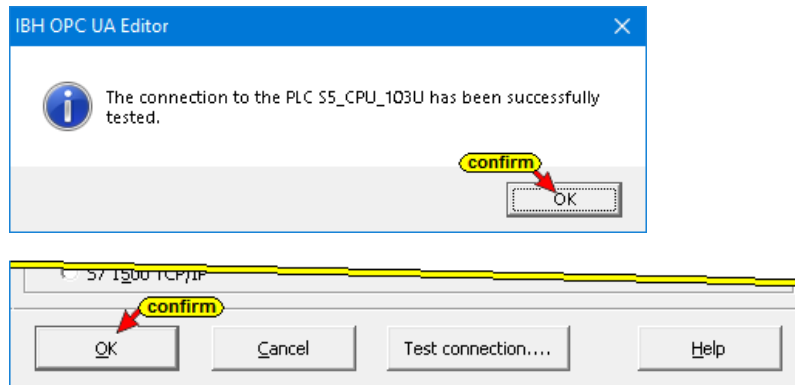


## Test connection

After completing the **New control** dialog box, the connection to the online connected CPU can be tested.

Test connection....

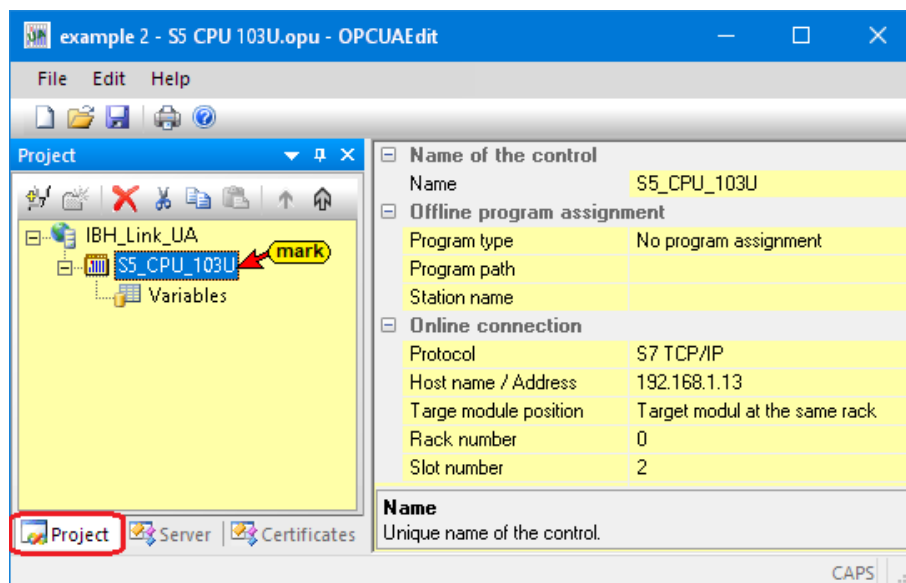
Information about the successful connection is displayed.



To accept and close the **New control** dialog box settings click on **OK**.

## Right project window

The right project window displays the access data for the **S5 CPU 103U** (CPU connected via IBH Link S5 ++ with IBH Link UA).



The selected name of the controller is **S5\_CPU\_103U** (no spaces are allowed in the name). **S7 TCP / IP** was selected as the protocol for the online connection to the PLC.

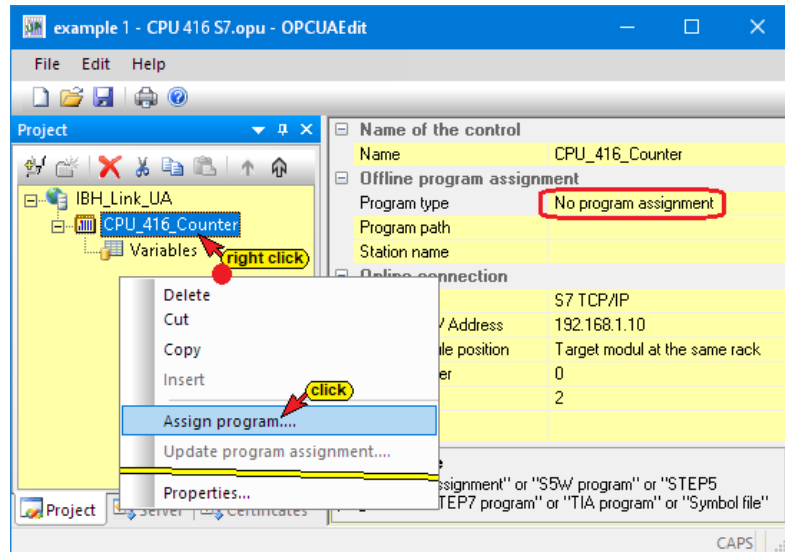
Under Host name / IP address, the IP address of the IBH Link S5 ++ (192.168.1.13) has been specified.

The target module is on the same rack. The rack number and slot number are irrelevant.

An offline program assignment has not been made, since a program assignment is done via a command.

## 2.2.2 Offline program assignment

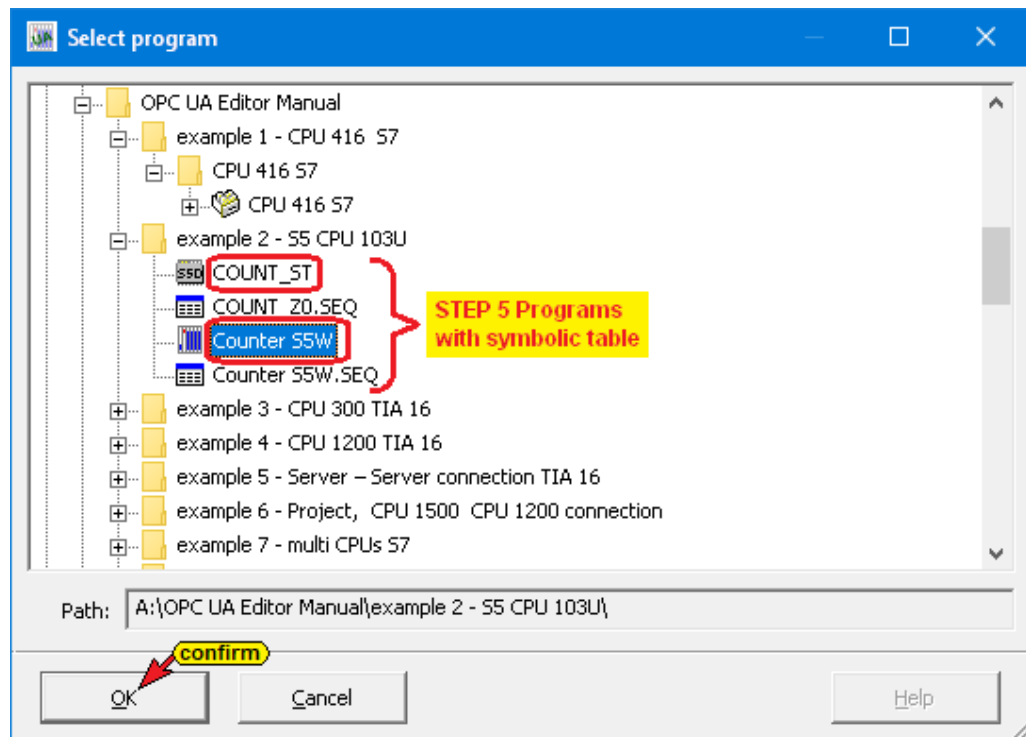
The **Assign Program** command is used to open the **Program Selection** dialog box.



### Program Selection dialog box

Select the PLC program in the Select program dialog box. By clicking the Plus symbol in front of the folder symbol, the PLC program (CPUs) is displayed in the project.

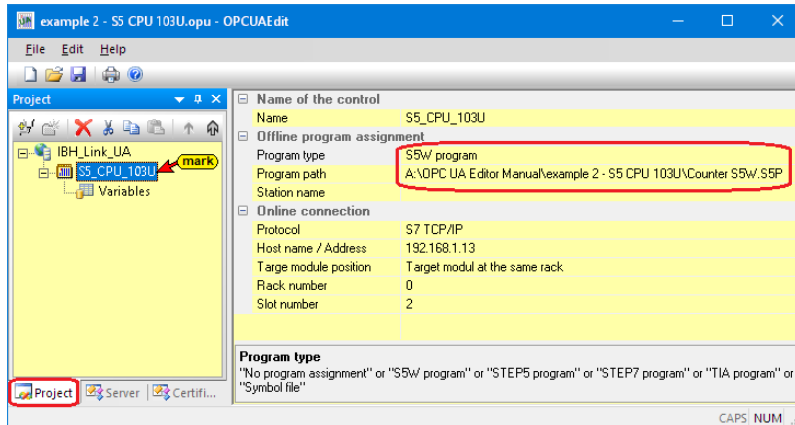
In the **Select program** dialog box mark the **STEP® 5** project **COUNT\_ST** or the **S5 for Windows** project **Counter S5W**.



Click **OK** to transfer the variables, data, and program information to the **OPC UA Editor**.

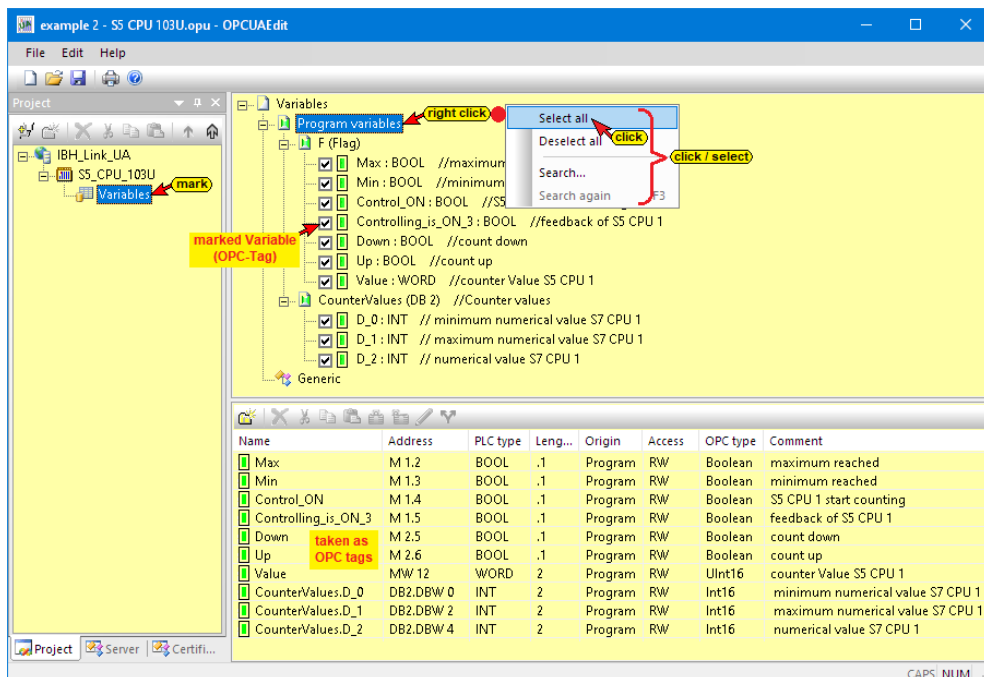
## Project window

In the right part of the project window information about the **Offline program assignment** are displayed. The variables of the PLC program **Counter S5W** were adopted.



### 2.2.3 Define variables as OPC tags

Clicking the name **Variables** will list the variables / data from the S5W PLC project **Counter S5W** in the right part of the project window.



Clicking the Plus symbol in front of the variable area symbol lists the existing variables.

If a variable is selected, it is adopted as an OPC tag and displayed in the lower part of the window with additional information.

The context menu offers the commands to select all variables as OPC tags (select all) or deselect (deselect all).



A search function is available to handle extensive variable lists.

## Add new variable (OPC tag)

With a right-click in a blank line, the Context menu is opened. A click on the **Variable definition** command opens the **Variables properties** dialog box.

Here a new variable (OPC tag) can be created. The drop-down list fields facilitate the definition of a variable.

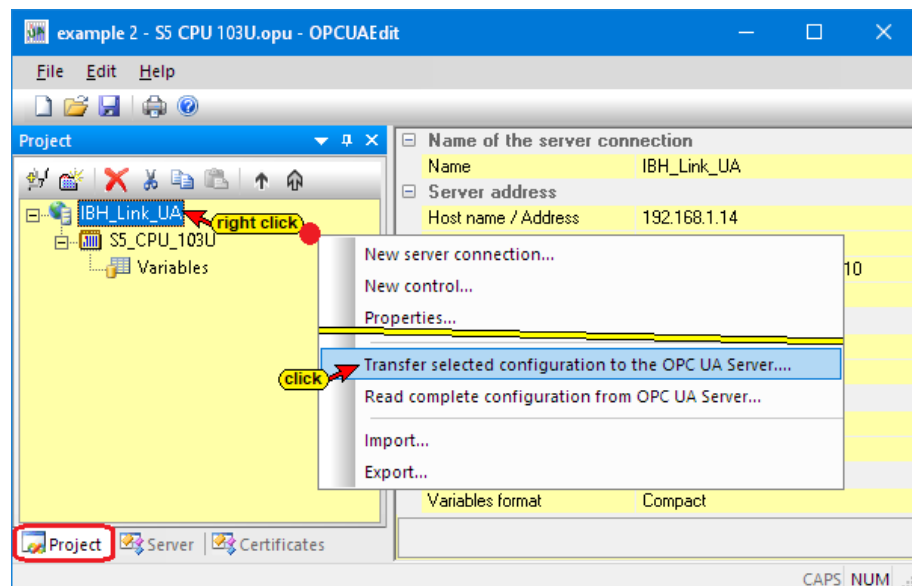
## Change Variable (OPC tag)

A right-click on a line with a variable (OPC tag) opens the context menu with commands for editing this variable.

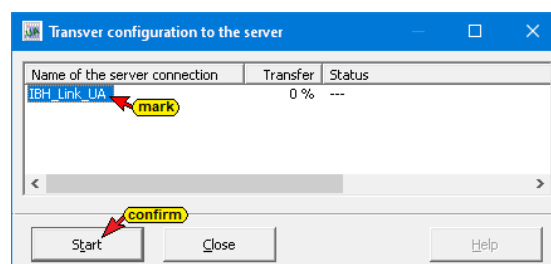
The **Properties** command opens the **Variables Properties** dialog box. The marked variable (OPC tag), can be modified.

### 2.2.4 Transfer configuration to the OPC UA server (IBH Link UA).

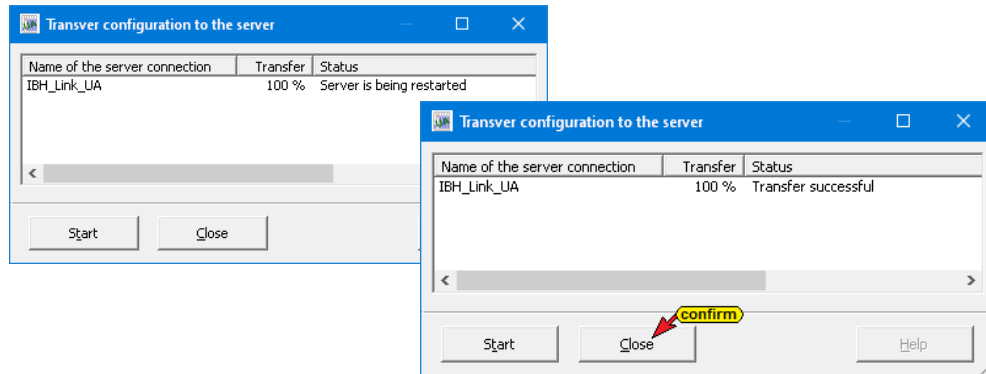
A right-click on the Server icon (IBH Link UA) opens the context menu.



The command **Transfer Selected Configuration to OPC UA Server** command opens the **Transfer Configuration to Server** dialog box.



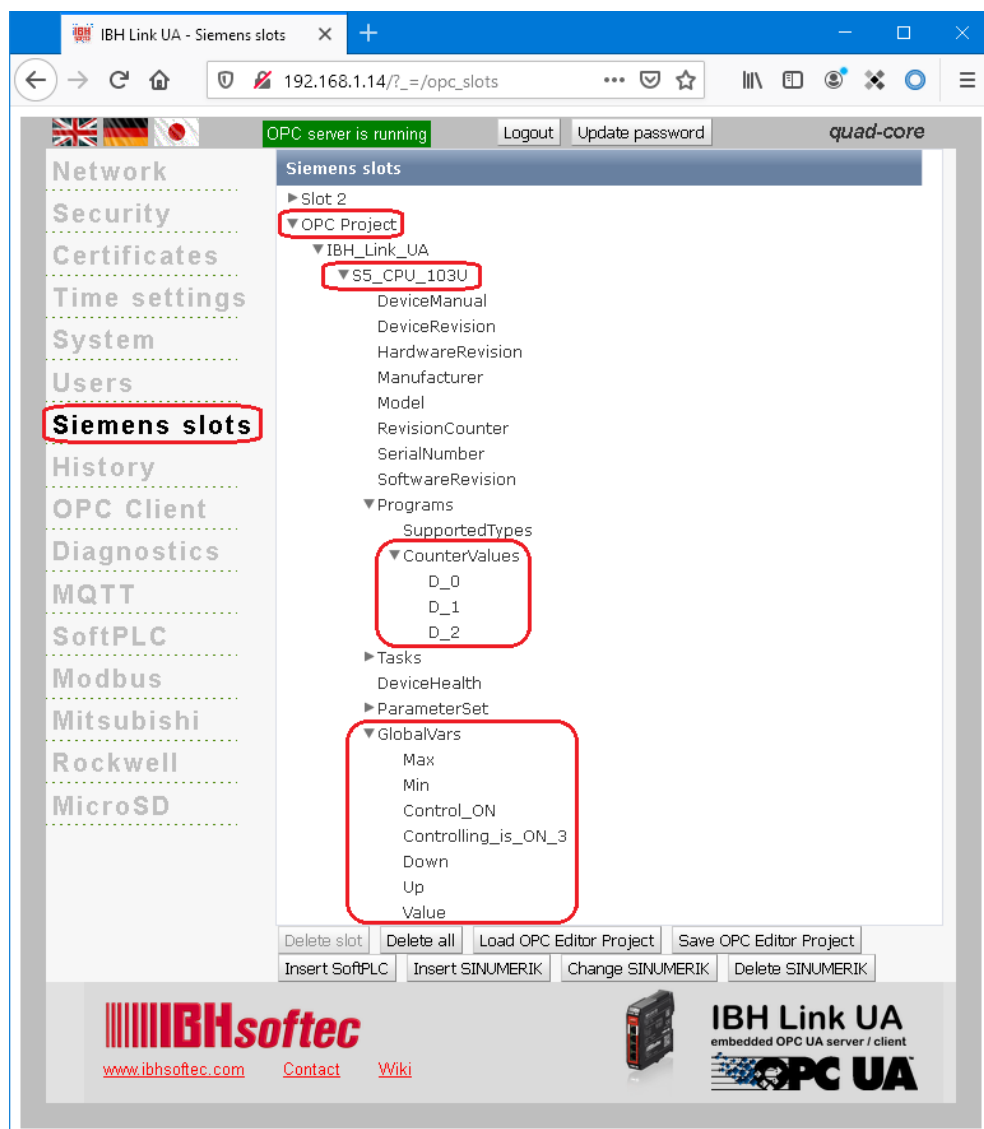
Select the server **IBH Link UA** and then click Start. The configuration of **Counter S5W** the STEP® 5 PLC Program is transferred to the **IBH Link UA**.



The successful transfer is displayed.

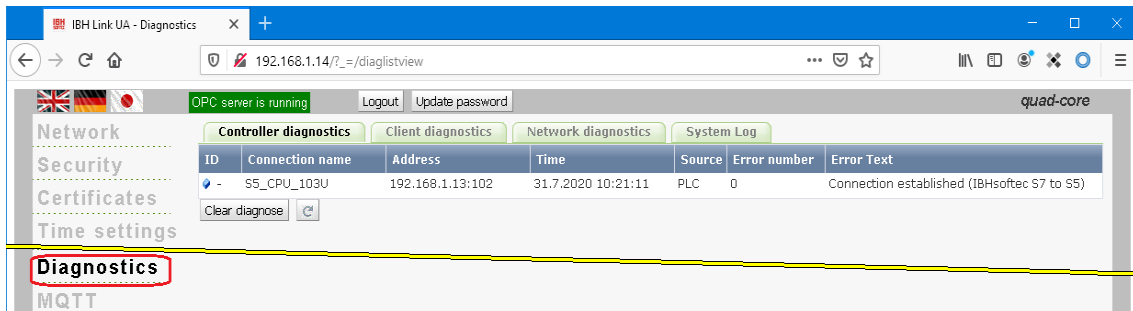
If a certified data exchange has been agreed between the **IBH OPC Editor** and the **IBH Link UA**, the exchanged certificates must be confirmed as **trusted** (see Trust certificates; chapter 1 page 1-40).

## IBH Link UA - Siemens Slots project S5\_CPU\_103U



## IBH Link UA – Browser window Diagnostics

The browser window *Diagnostics* displays the status of the connection **IBH Link UA – PLC / S5\_CPU\_103U**.

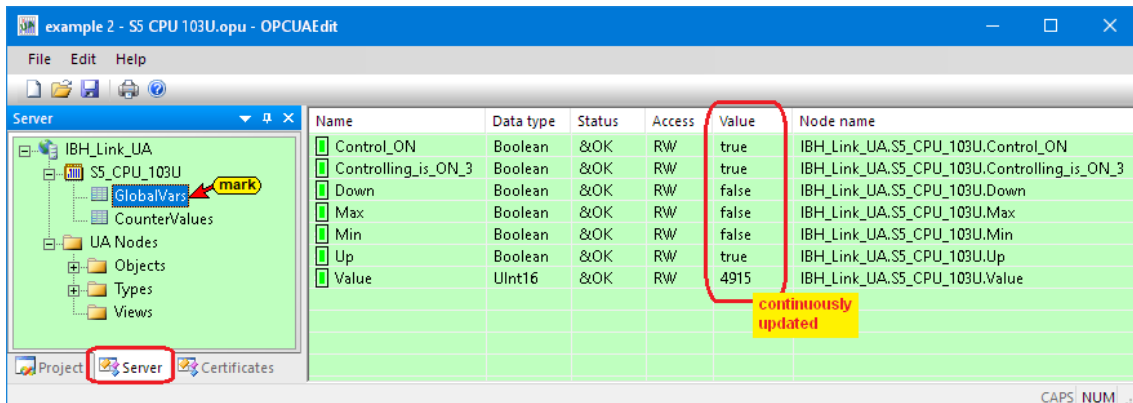


### 2.2.5 Online OPC UA Server Information

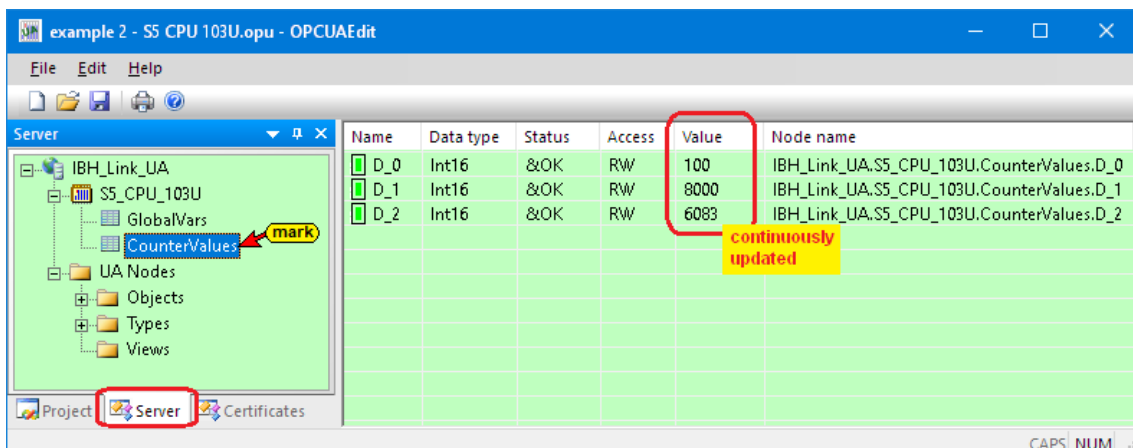
Information from the online connected **OPC UA server** with the online connected PLC **S5 CPU 103U** is displayed.

The groups of the variables (GlobalVars, data blocks) are listed in the left server window. By clicking on a group, the individual variables (OPC tags) are displayed in the right server window with their status. The status of the OPC tags is continuously updated.

#### GlobalVars



#### Data Block DB2 – CounterValues



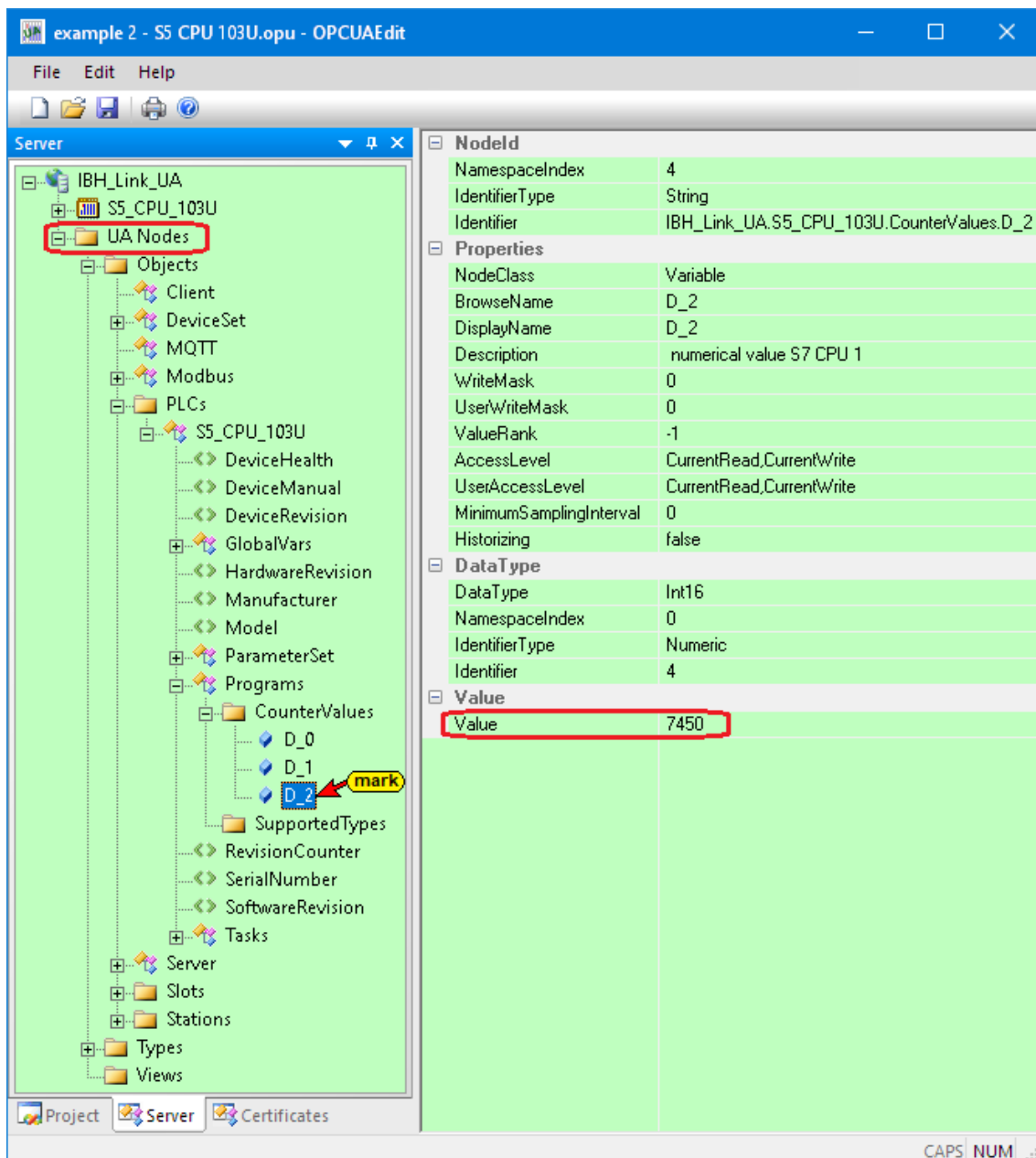
## Show under UA Nodes

The names of the OPC UA Nodes are listed in the left-hand server window (attributes, OPC tags, etc.).

The names of the OPC UA Nodes are listed in the left-hand server window (attributes, OPC tags, etc.).

The corresponding values are displayed in the right-hand server window.

The values are current values and are only taken once when the OPC UA Node name is clicked.



The screenshot shows the OPC UA Editor interface. On the left, a tree view displays the hierarchy of nodes. The 'UA Nodes' folder is highlighted with a red box. Underneath, the 'CounterValues' folder is expanded, showing nodes 'D\_0', 'D\_1', and 'D\_2'. The 'D\_2' node is selected and marked with a red arrow and a yellow 'mark' label. On the right, the properties table for the selected node is displayed. The table has the following structure:

NodeId	
NamespaceIndex	4
IdentifierType	String
Identifier	IBH_Link_UA.S5_CPU_103U.CounterValues.D_2
Properties	
NodeClass	Variable
BrowseName	D_2
DisplayName	D_2
Description	numerical value S7 CPU 1
WriteMask	0
UserWriteMask	0
ValueRank	-1
AccessLevel	CurrentRead,CurrentWrite
UserAccessLevel	CurrentRead,CurrentWrite
MinimumSamplingInterval	0
Historizing	false
DataType	
DataType	Int16
NamespaceIndex	0
IdentifierType	Numeric
Identifier	4
Value	
Value	7450

## 2.2.6 Unified Automation UaExpert - The OPC Unified Architecture Client

The *UaExpert program window* lists the **OPC tags** transferred from the IBH OPC UA Editor and the associated **UA nodes**.

The screenshot shows the 'Unified Automation UaExpert - The OPC Unified Architecture Client - NewProject\*' window. The 'Data Access View' table is as follows:

#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.Control_ON	Control_ON	true	Boolean	12:52:02.483	12:52:03.507	Good
2	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.Controlling_is_ON_3	Controlling_is_ON_3	true	Boolean	12:52:04.938	12:52:05.760	Good
3	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.Down	Down	false	Boolean	12:52:08.065	12:52:08.763	Good
4	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.Max	Max	false	Boolean	12:52:09.456	12:52:10.014	Good
5	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.Min	Min	false	Boolean	12:52:09.456	12:52:10.014	Good
6	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.Up	Up	true	Boolean	12:52:09.456	12:52:10.014	Good
7	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.Value	Value	4849	UInt16	12:52:16.160	12:52:17.018	Good
8	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.CounterValues.D_0	D_0	100	Int16	12:52:18.480	12:52:19.268	Good
9	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.CounterValues.D_1	D_1	8000	Int16	12:52:18.480	12:52:19.268	Good
10	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.CounterValues.D_2	D_2	4995	Int16	12:53:48.547	12:53:49.048	Good

The project tree on the left shows the following structure:

- Root
  - Objects
    - Client
    - DeviceSet
    - MQTT
    - Modbus
    - PLCs
      - S5\_CPU\_103U
        - DeviceHealth
        - DeviceManual
        - DeviceRevision
        - GlobalVars
          - Control\_ON
          - Controlling\_is\_ON\_3
          - Down
          - Max
          - Min
          - Up
          - Value
        - HardwareRevision
        - Manufacturer
        - Model
        - ParameterSet
        - Programs
          - CounterValues
            - D\_0
            - D\_1
            - D\_2
          - SupportedTypes
          - RevisionCounter
          - SerialNumber
          - SoftwareRevision
        - Tasks

A red box highlights the 'GlobalVars' and 'CounterValues' sections in the project tree. A red box highlights a subset of the Data Access View table. A 'Drag & Drop' label with arrows points from the project tree to the highlighted table data.

Display Name	Value	Datatype	Source Timestamp
Control_ON	true	Boolean	12:52:02.483
Controlling_is_ON_3	true	Boolean	12:52:04.938
Down	false	Boolean	12:52:08.065
Max	false	Boolean	12:52:08.065
Min	false	Boolean	12:52:09.456
Up	true	Boolean	12:52:09.456
Value	4019	UInt16	12:52:16.160
D_0	100	Int16	12:52:16.160
D_1	8000	Int16	12:52:18.480
D_2	4164	Int16	12:52:18.480

#	Server	Node Id
1	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.Control_ON
2	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.Controlling_is_ON_3
3	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.Down
4	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.Max
5	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.Min
6	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.Up
7	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.Value
8	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.CounterValues.D_0
9	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.CounterValues.D_1
10	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.S5_CPU_103U.CounterValues.D_2

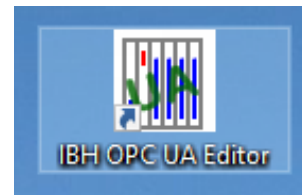
## 2.3 Example 3: CPU 300 TIA16

Connect **CPU 300 / CPU 400** directly to IBH Link UA via **IBH Link S7 ++**. PLC program is available as a TIA 16 project (CPU 300 TIA16).

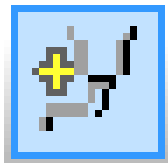
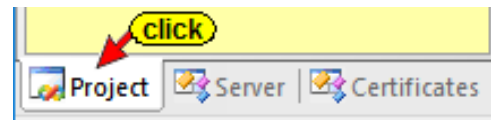
The CPU 312 (6ES7 312-1AE13-0AB0) – within the PLC project **CPU 300 TIA16**- is to be connected to the IBH Link UA using the Ethernet connection via an **IBH Link S7++** to exchange variables (OPC tags).

### Calling the IBH OPC UA Editor

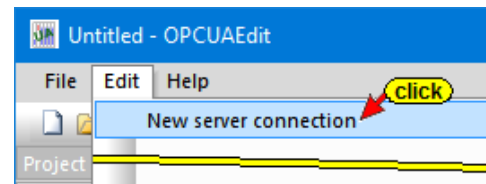
Double-click the **IBH OPC UA Editor** icon to open the program window.



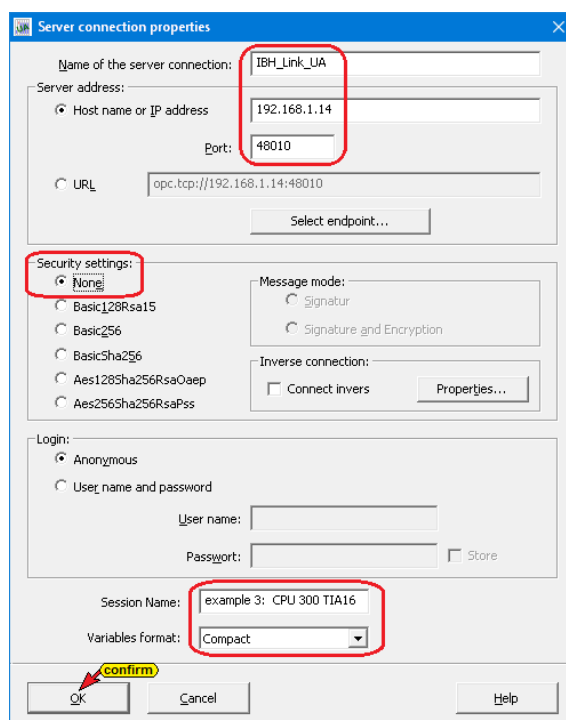
Open the **Project window** by clicking on the **Project** tab.



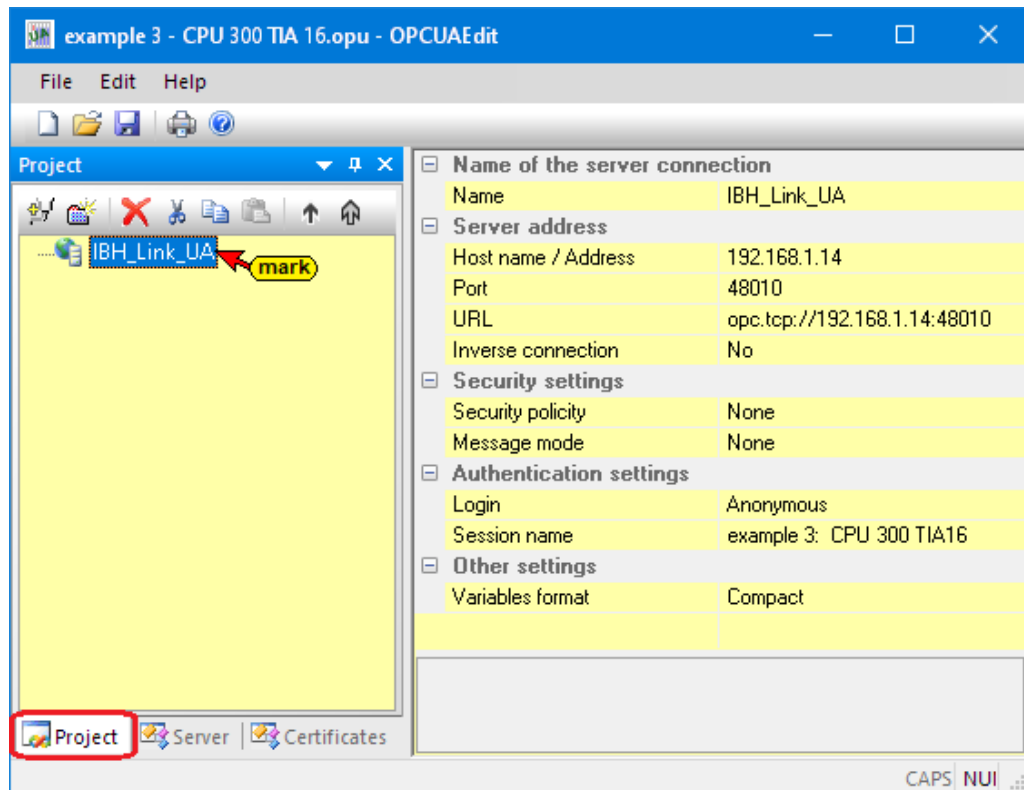
Open the **New Server Connection** dialog box with the New Server Connection command from the **Edit** menu or by clicking the icon.



The new server connection setup was explained in example 1 (see chapter 2, page 2-3).

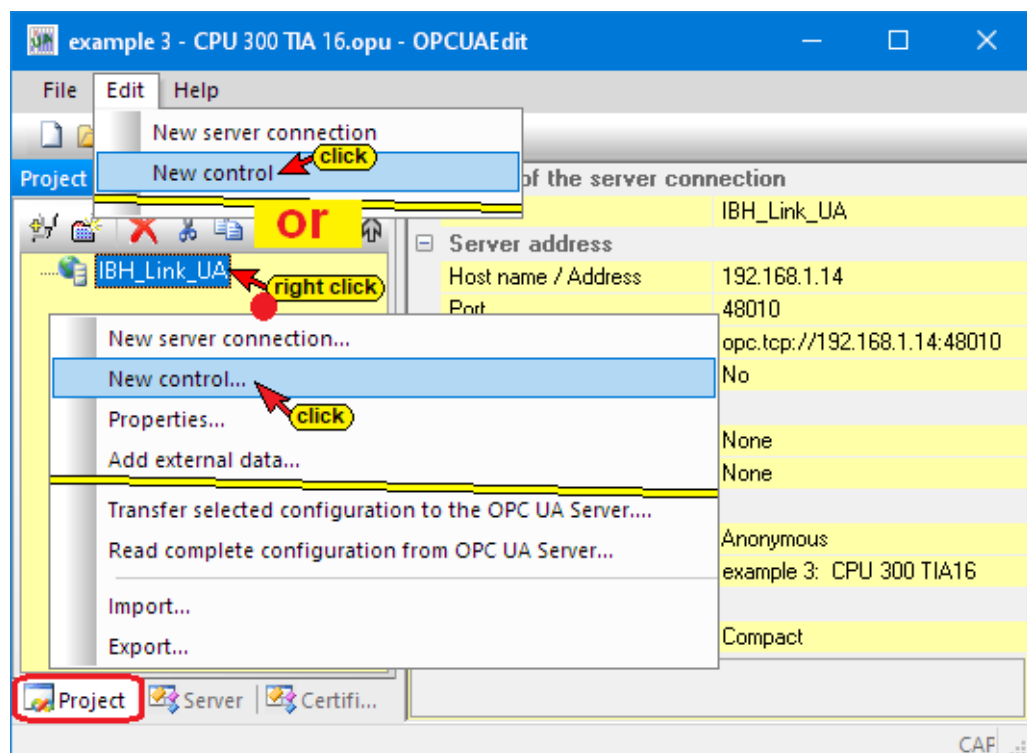


The settings for the connection to the **IBH Link UA** OPC UA server are displayed in the right part of the **project window**.



### 2.3.1 Inserting a New control (PLC)

The **New control** command from the context menu (or menu Edit / New control) opens the dialog box **New control** to specify the access to the control (CPU).



## New control dialog box

Control name: CPU\_300\_TIA\_16 **enter**

Host name / IP address: 192.168.1.12 **IP address IBH Link S7++**

Protocol:  S7 TCP/IP

Rack number: 0 Slot number: 2

Position of the target module:

Target modul at the same rack

Via MPI/DP subnet accessible rack

MPI/DP address of the target CPU: 2

Via TCP/IP subnet accessible rack

TCP/IP address of the target CPU: 0 . 0 . 0 . 0

Via H1 subnet accessible rack

H1 address of the target CPU: 00.00.00.00.00.00

Subnet ID: 0000 . 0000

Own TSAP: 0100

PLC TSAP: 0101

**click**

OK Cancel Test connection.... Help

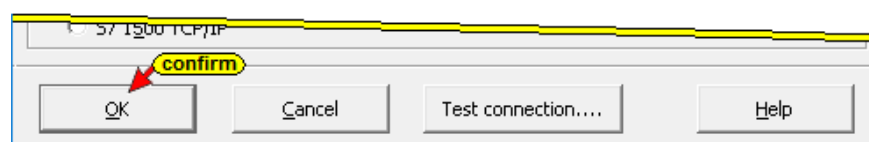
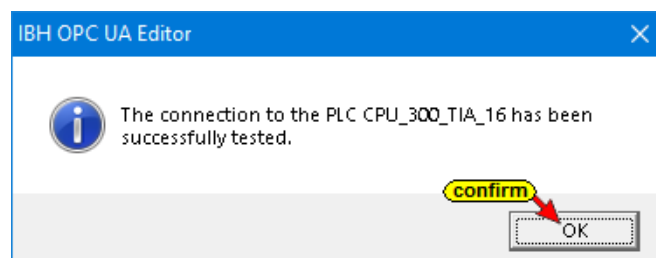
corresponds to MPI address	
Rack	Slot
3 Bit	5 Bit (0-31)
000	00010
0	2

## Test connection

After completing the New Control dialog box, the connection to the online connected CPU can be tested.

Test connection....

Information about the successful connection is displayed.



To accept and close the **New control** dialog box settings click on **OK**.



## 2.3.2 IBH Link S7 ++ setting

If no connection is established from the PC via **IBH Link S7 ++**, the settings must be checked (see chapter 1 page 1-13 – IBH Link UA - S7 CPU 300 / 400 connection via IBH Link S7++).

### Note:

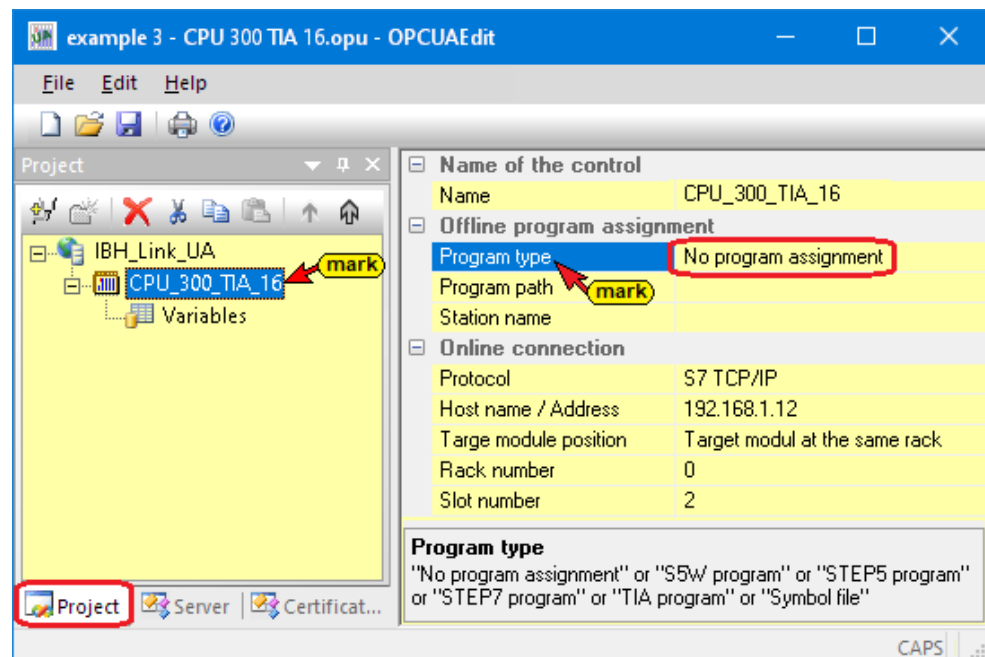


To address the CPU 312 from the **CPU 300 TIA16** project via the **IBH Link S7++**, the routing option (dialog box **IBHLink settings / Network tab**) **Configuration with NetPro** must be deactivated (Apply permanently).

**This applies to all S7 300/400 CPUs with IBH Link S7++ connection.**

### Right project window

The right project window displays the access data for the **CPU 312** (CPU S7-300 / S7-400 via IBH Link S7++).



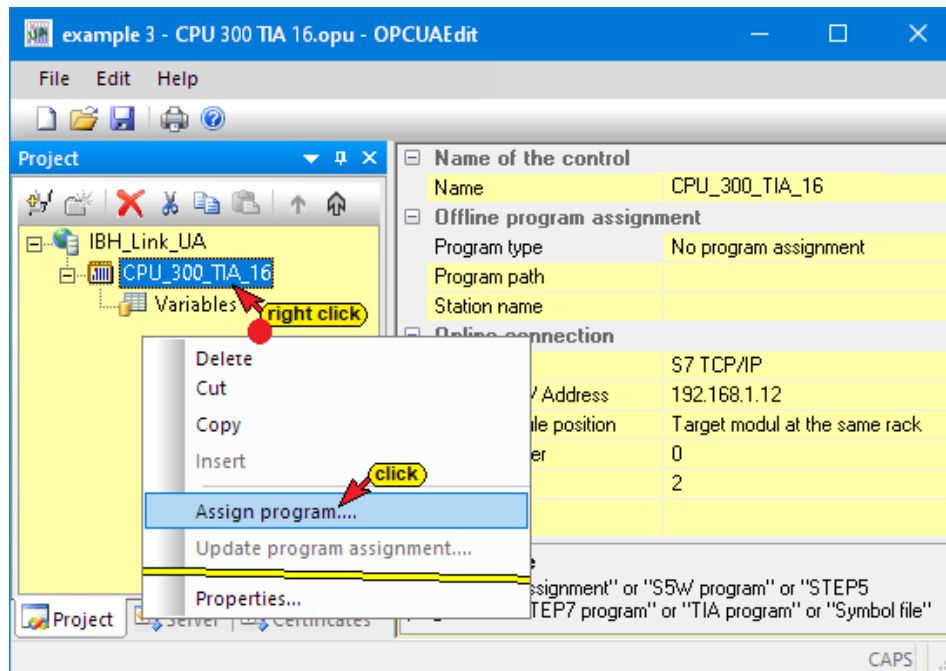
**CPU 300 TIA16** was specified as the name of the controller. S7 TCP / IP was selected as the protocol for the online connection to the PLC. The IP address of the **IBH Link S7++ (192.168.1.12)** has been defined under host name / IP address.

The target module is on the same rack. The rack number and slot number select the **MPI address 2** of the CPU.

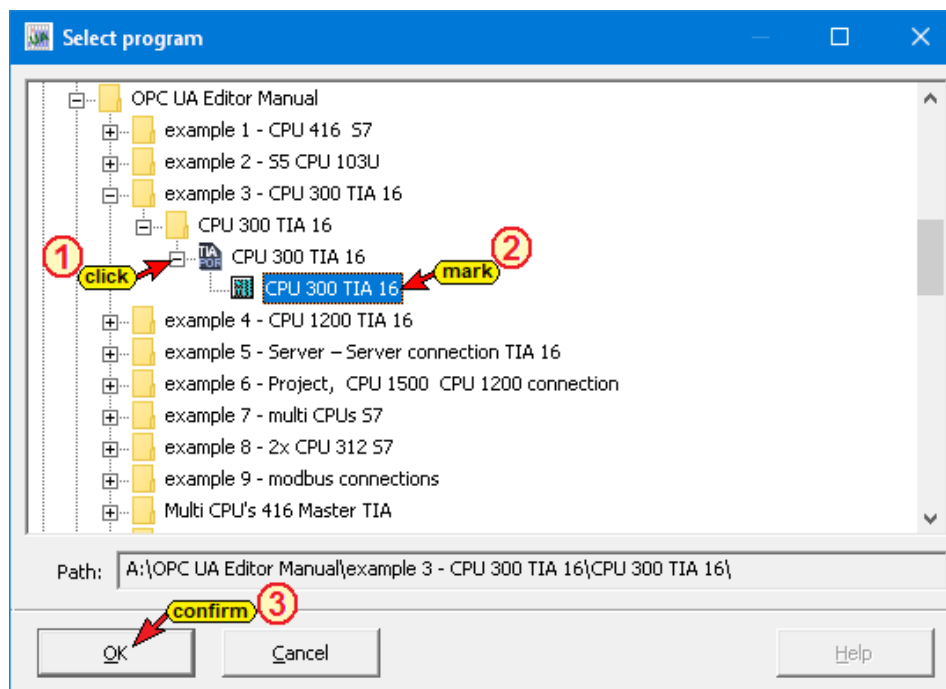
An **offline program assignment** has not yet been made. The program is assigned via a command.

### 2.3.3 Offline program assignment

The **Assign program** command is used to open the **Program Selection** dialog box.

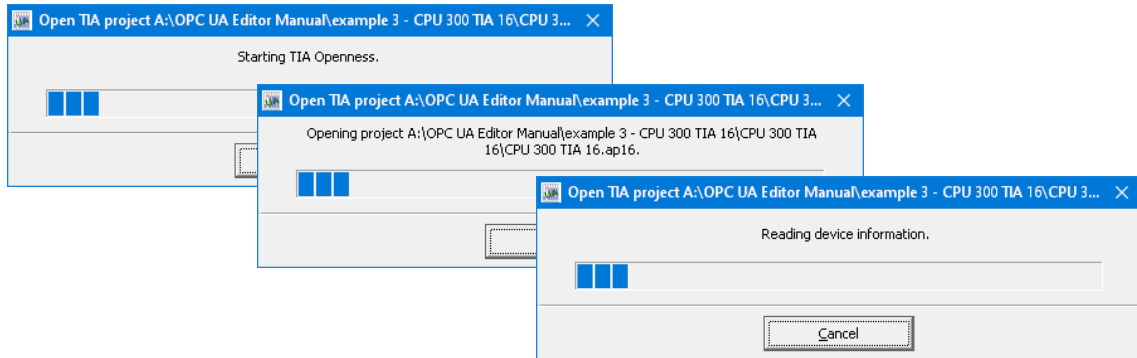


Select the PLC program in the **Select program** dialog box. Clicking the **Plus** symbol in front of the TIA symbol of the PLC project, the PLC program (CPUs) is displayed in the project.



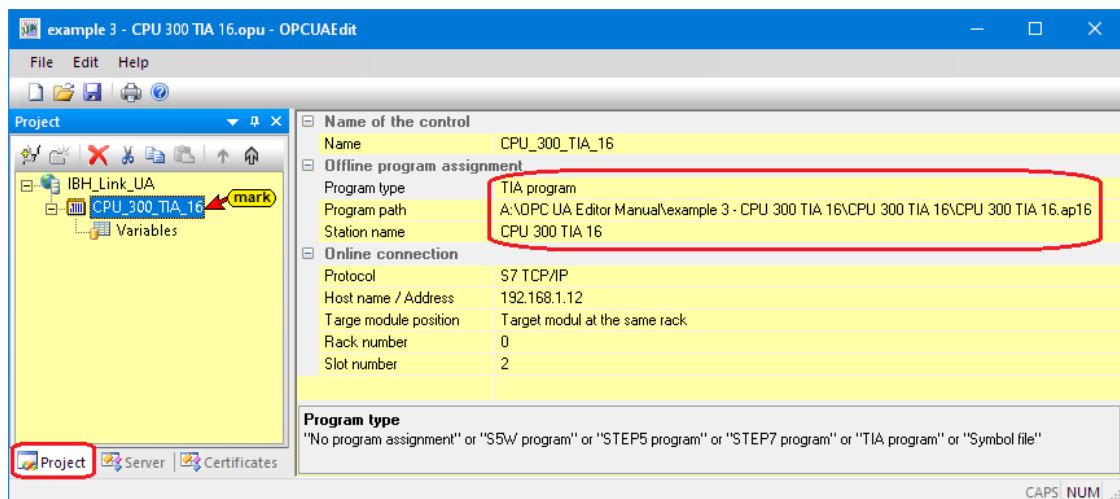
The **SIEMENS support software TIA Openness** is started in the background. If an error occurs see chapter 1, **Special features when selecting TIA projects** page 1-16.

Several notices are displayed.



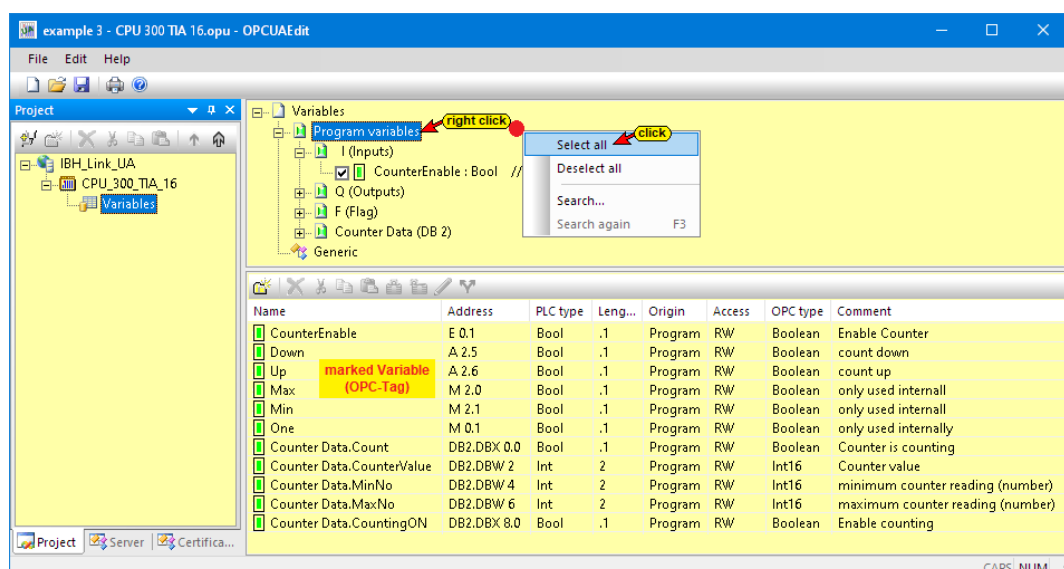
## Listed transferred project

In the right part of the project window information about the **Offline program assignment** are displayed.



### 2.3.4 Define variables as OPC tags

Clicking **Variables** lists the variables / data (data blocks) from the PLC in the right part of the project window.

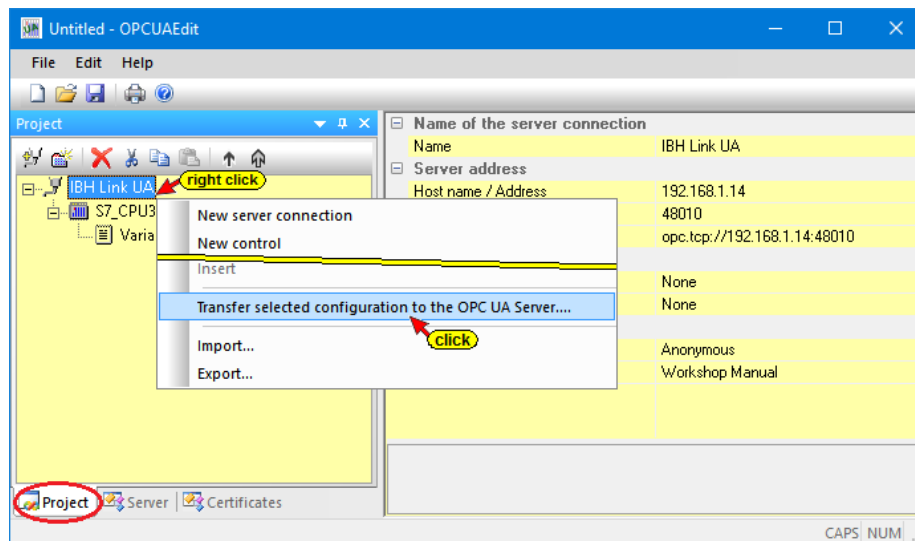


Clicking the **Plus** icon in front of the variable area symbol displays the existing variables.

If a variable is selected, it is adopted as an OPC tag and displayed in the lower part of the window with additional information.

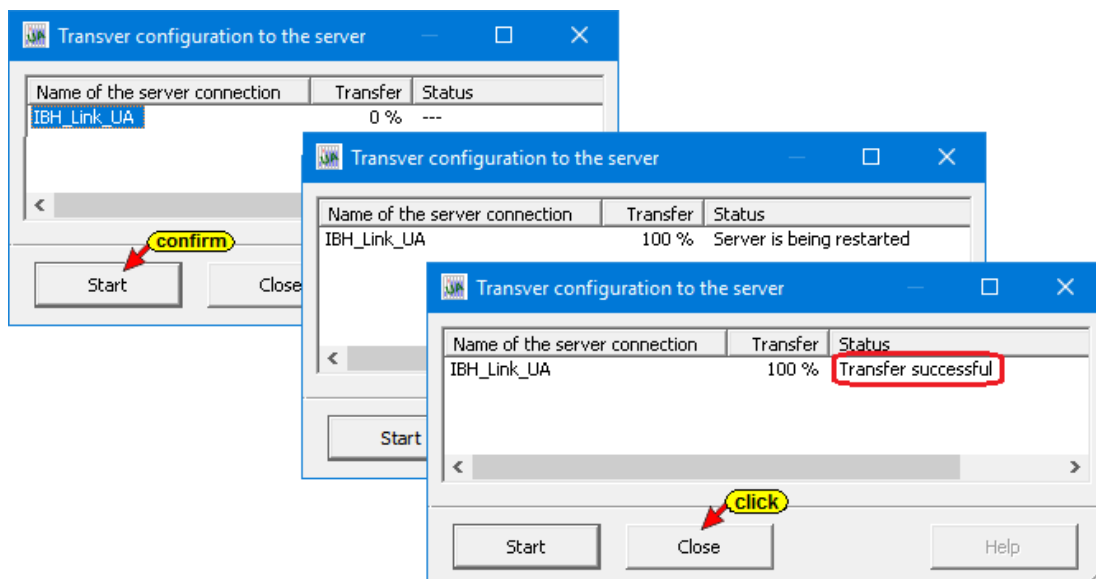
### 2.3.5 Transfer configuration to the OPC UA server (IBH Link UA).

A right-click on the Server icon (IBH Link UA) opens the context menu.



The command **Transfer Selected Configuration to OPC UA Server** command opens the **Transfer Configuration to Server** dialog box.

Select the server **IBH Link UA** and then click Start. The configuration (**CPU\_300\_TIA\_16**) is transferred to the **IBH Link UA**. Successful transfer is displayed.



The successful transfer is displayed.

If a certified data exchange between the IBH OPC Editor and the IBH Link UA has been selected, the exchanged certificates must be trusted (see chapter Trust certificate, Chapter 1, page 1-40).

**Note:**

To address the CPU 312 from the **CPU\_300\_TIA\_16** project via the **IBH Link S7++**, the routing option (dialog box **IBHLink settings / Network tab**) **Configuration with NetPro** must be deactivated (Apply permanently).

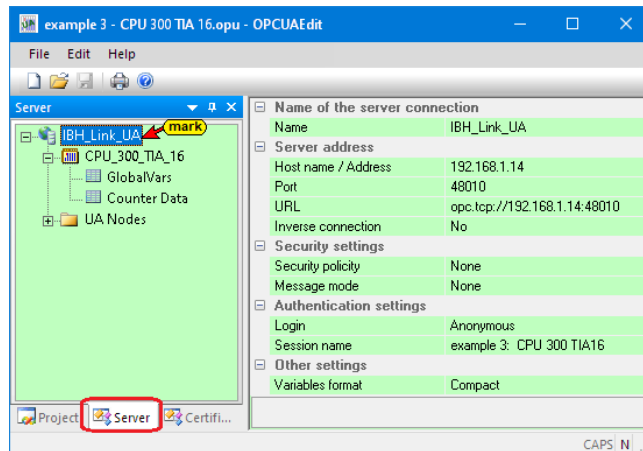
**This applies to all S7 300/400 CPUs with IBH Link S7++ connection.**

## IBH Link UA - Siemens Slots - Project CPU 300 TIA 16

The browser window **Diagnostics** displays the status of the connection **IBH Link UA – PLC / CPU 300 TIA 16**.

## 2.3.6 Online OPC UA Server Information

Information from the online connected **OPC UA server** with the online connected **S7 CPU 312** are displayed.



### View server

The groups of the variables (GlobalVars, data blocks) are listed in the left server window. By clicking on a group, the individual variables (OPC tags) are displayed in the right server window with their status. The status of the OPC tags is updated continuously.

#### CPU 312 – Global Variable

Name	Data type	Status	Access	Value	Node name
CounterEnable	Boolean	&OK	RW	false	IBH_Link_UA.CPU_300_TIA_16.CounterEnable
Down	Boolean	&OK	RW	true	IBH_Link_UA.CPU_300_TIA_16.Down
Max	Boolean	&OK	RW	false	IBH_Link_UA.CPU_300_TIA_16.Max
Min	Boolean	&OK	RW	false	IBH_Link_UA.CPU_300_TIA_16.Min
One	Boolean	&OK	RW	true	IBH_Link_UA.CPU_300_TIA_16.One
Up	Boolean	&OK	RW	false	IBH_Link_UA.CPU_300_TIA_16.Up

A yellow callout box with the text "continuously updated" points to the 'Value' column.

#### CPU 312 Data Block DB2 – Counter Data

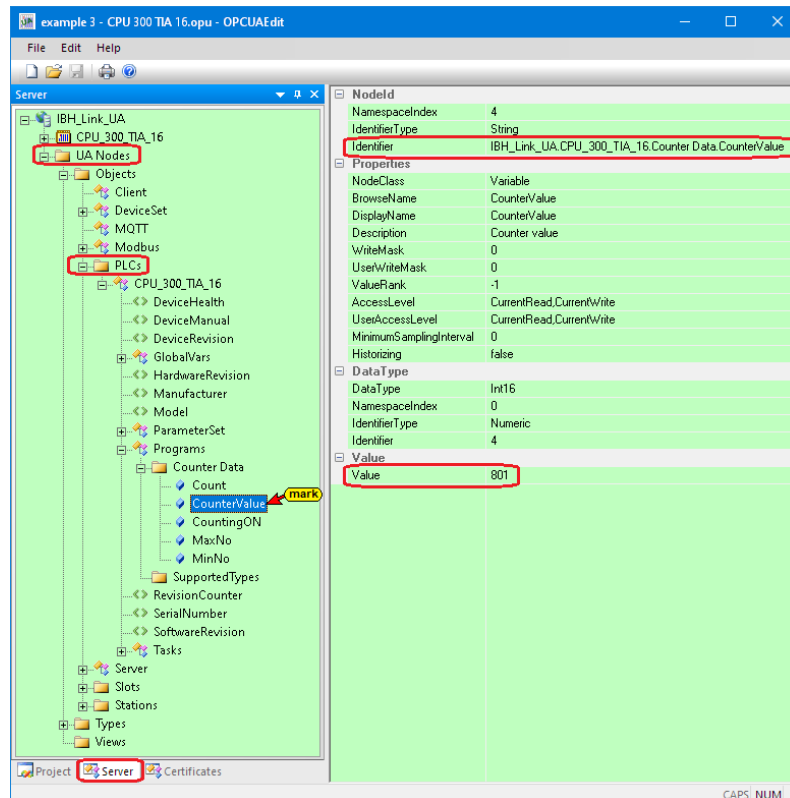
Name	Data type	Status	Access	Value	Node name
Count	Boolean	&OK	RW	true	IBH_Link_UA.CPU_300_TIA_16.Counter Data.Count
CounterValue	Int16	&OK	RW	481	IBH_Link_UA.CPU_300_TIA_16.Counter Data.CounterValue
CountingON	Boolean	&OK	RW	true	IBH_Link_UA.CPU_300_TIA_16.Counter Data.CountingON
MaxNo	Int16	&OK	RW	8000	IBH_Link_UA.CPU_300_TIA_16.Counter Data.MaxNo
MinNo	Int16	&OK	RW	100	IBH_Link_UA.CPU_300_TIA_16.Counter Data.MinNo

A yellow callout box with the text "continuously updated" points to the 'Value' column.

### Show under UA Nodes

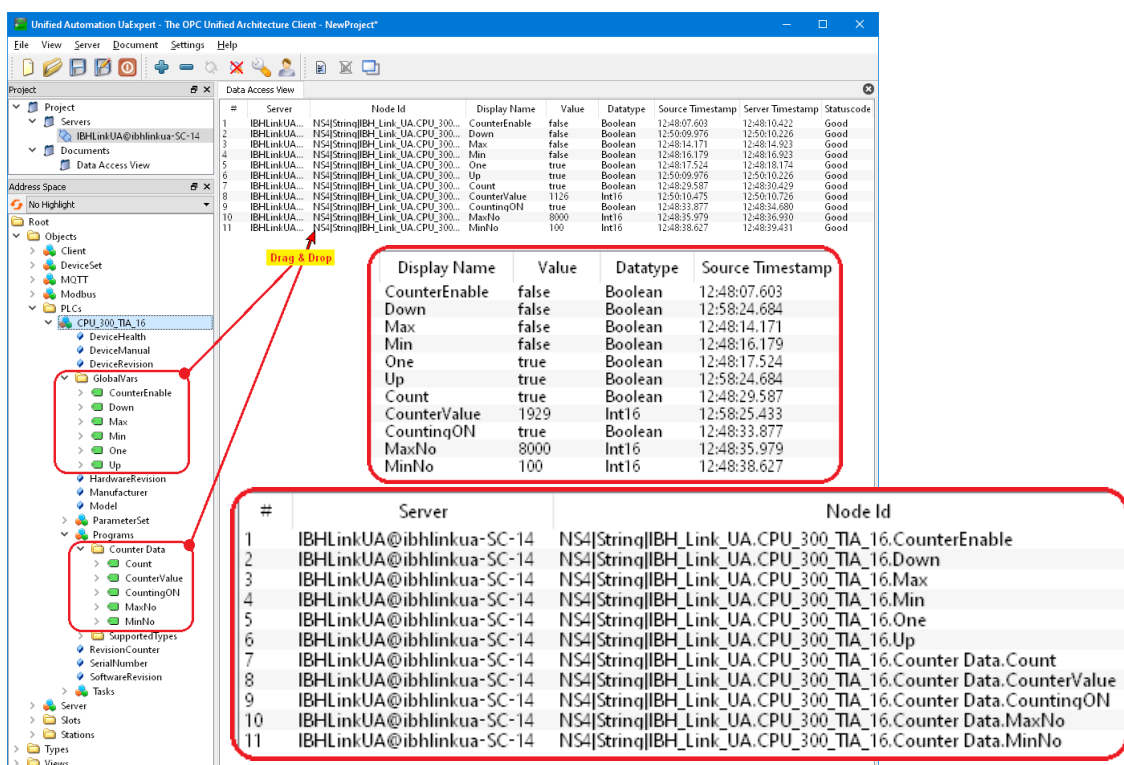
The names of the OPC UA Nodes are listed in the left-hand server window (attributes, OPC tags, etc.). The corresponding values are displayed in the right-hand server window.

The values are current values and are only taken once when the OPC UA Node name is clicked.



### 2.3.7 Unified Automation UaExpert - The OPC Unified Architecture Client

The UaExpert program window lists the OPC tags transferred by the IBH OPC UA Editor and the associated UA nodes.

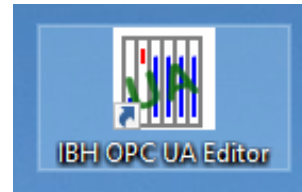


## 2.4 Example 4: CPU 1200 TIA 16

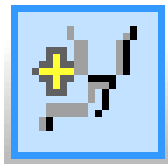
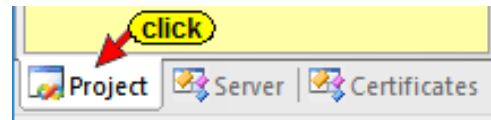
Connect CPU 1211C with TCP / IP port to IBH Link UA.  
 The PLC program (CPU 1200 TIA16) is available as a TIA16 project.

### Calling the IBH OPC UA Editor

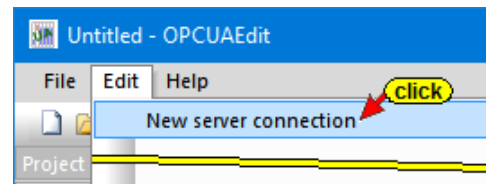
Double-click the **IBH OPC UA Editor** icon to open the program window.



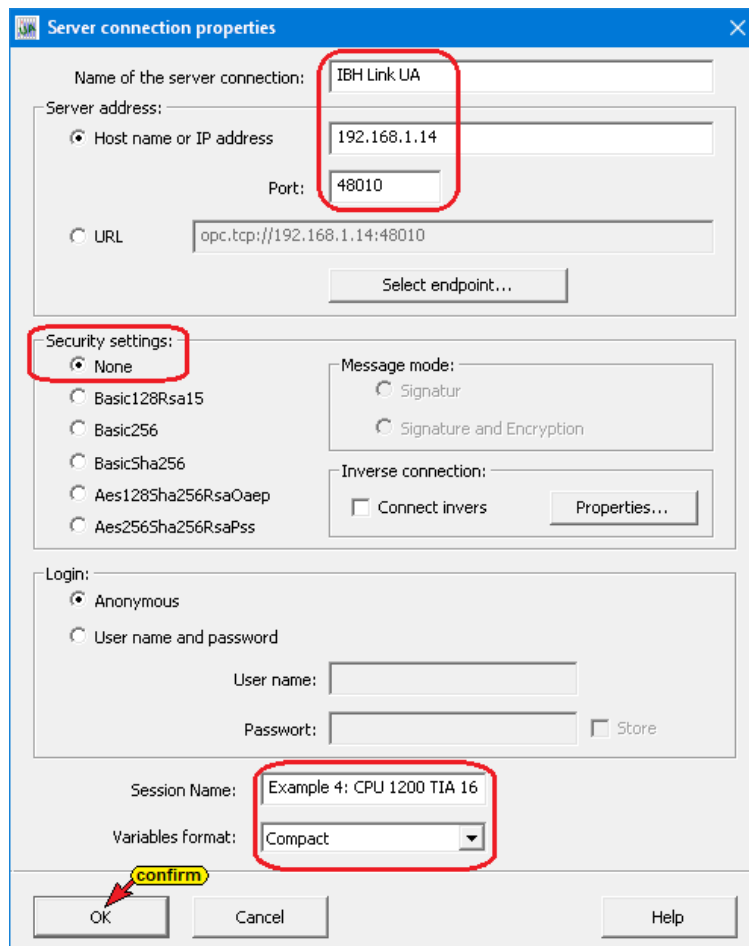
Open the **Project window** by clicking on the **Project** tab.



Open the **New Server Connection** dialog box with the New Server Connection command from the **Edit** menu or by clicking the icon.

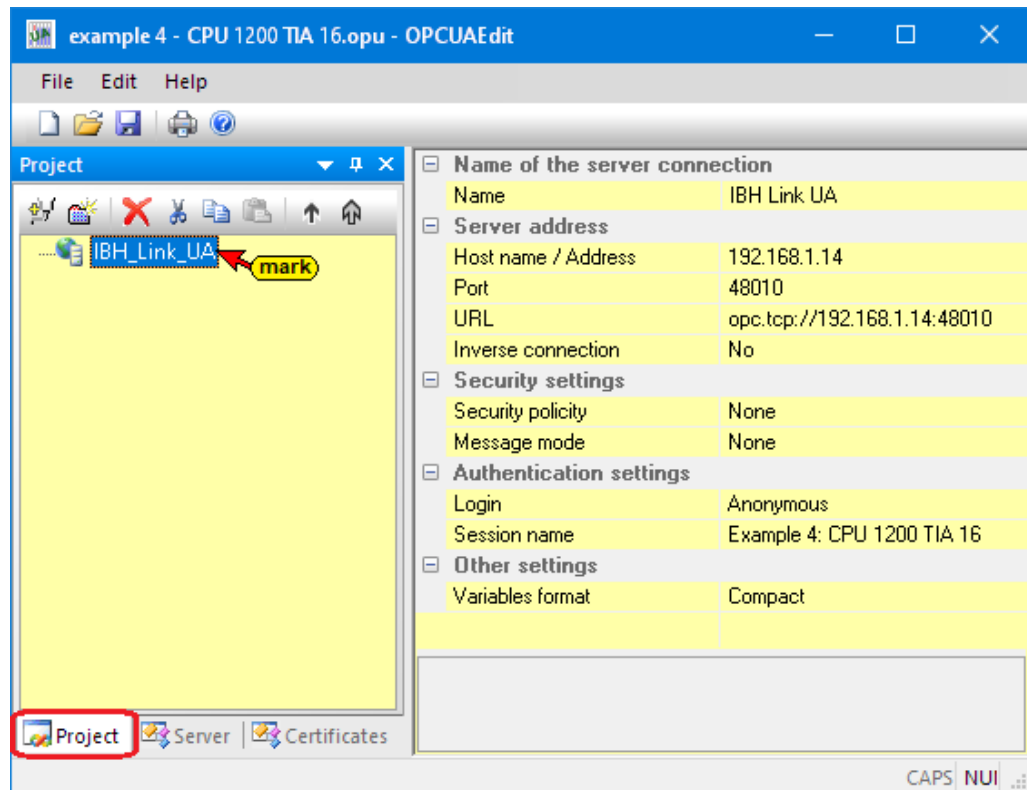


The new server connection setup was explained in example 1 (see chapter 2, page 2-3).



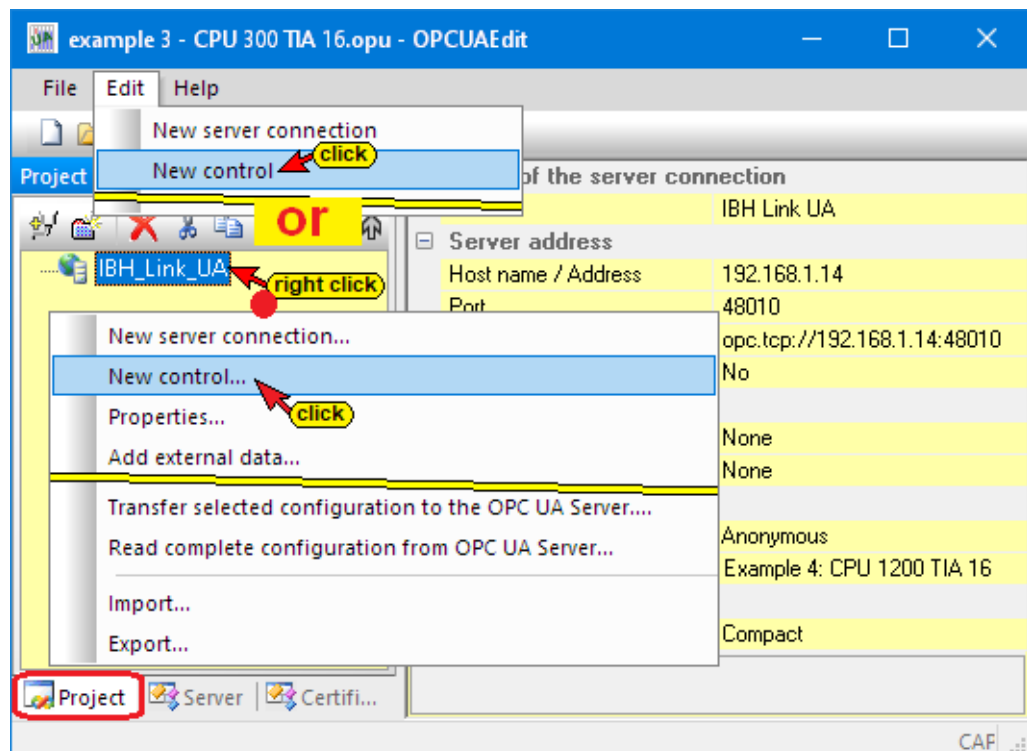


The settings for the connection to the **IBH Link UA** OPC UA server are displayed in the right part of the **project window**.



### 2.4.1 Inserting a New control (PLC)

The **New control** command from the context menu (or menu Edit / New control) opens the dialog box **New control** to specify the access to the control (CPU).



## New control dialog box

## Test connection

After completing the New Control dialog box, the connection to the online connected CPU can be tested.

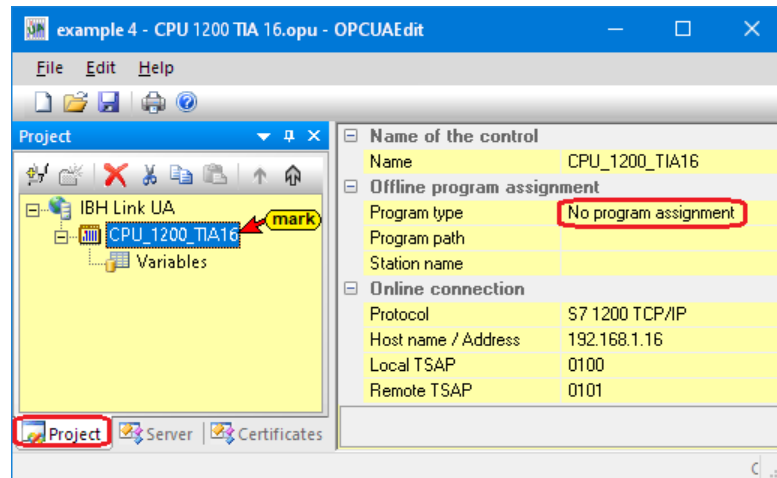
Test connection....

Information about the successful connection is displayed.

To accept and close the **New control** dialog box settings click on **OK**.

## Right project window

The right project window displays the access data for the **CPU 1200**.



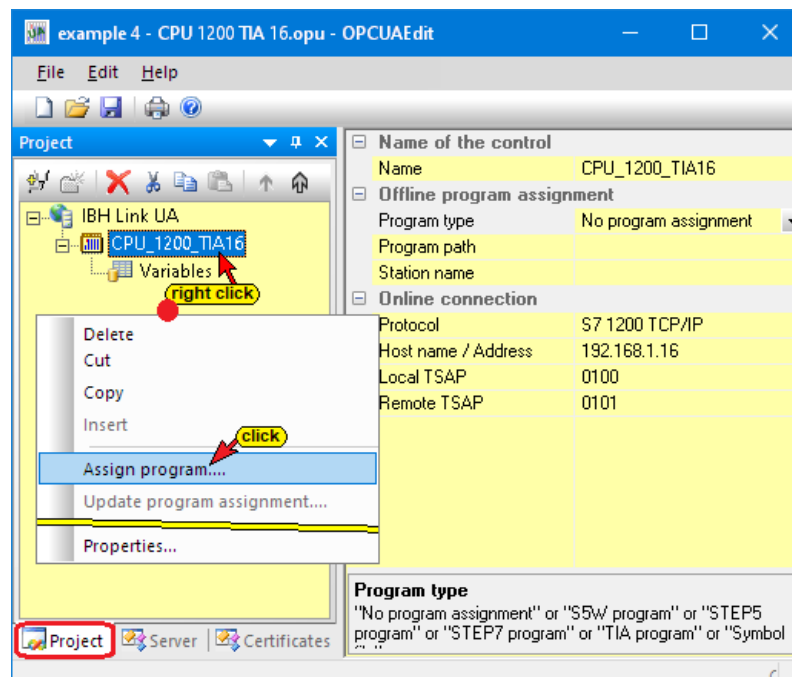
**CPU1200\_TIA16** was specified as the name of the controller. S7 1200 TCP / IP was selected as the protocol for the online connection to the PLC. The IP address of the **CPU1200** (**192.168.1.16**) has been defined under host name / IP address.

The Local TSAP (0100) and the Remote TSAP (0101) have their default value.

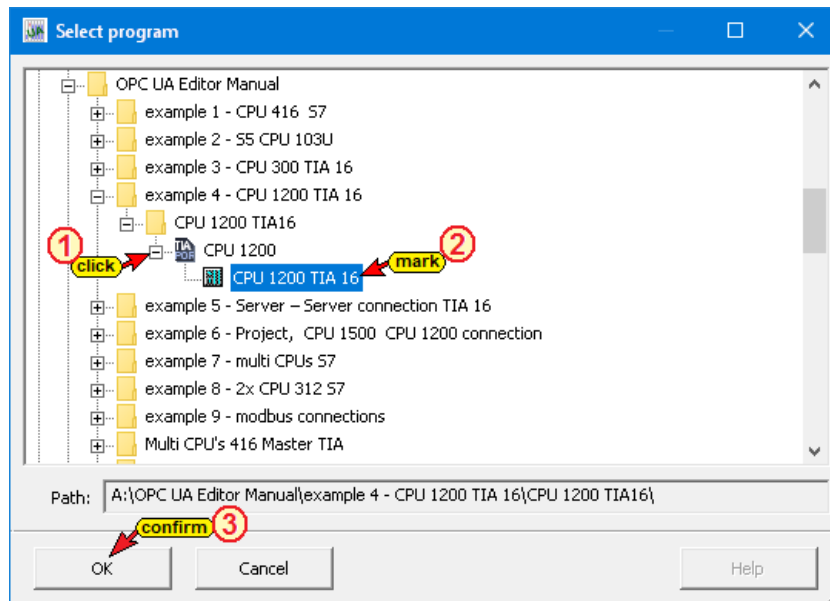
An **offline program assignment** has not yet been made. The program is assigned via a command.

## 2.4.2 Offline program assignment

The **Assign program** command is used to open the **Program Selection** dialog box.

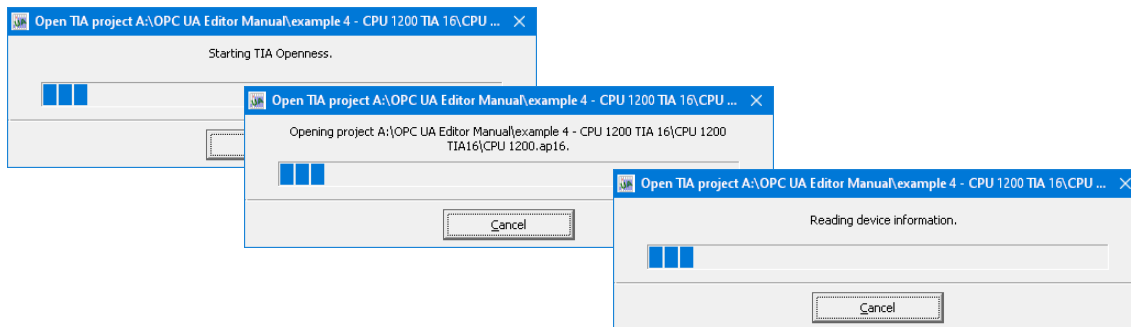


Select the PLC program in the **Select program** dialog box. Clicking the **Plus** symbol in front of the TIA symbol of the PLC project, the PLC program (CPUs) is displayed in the project.



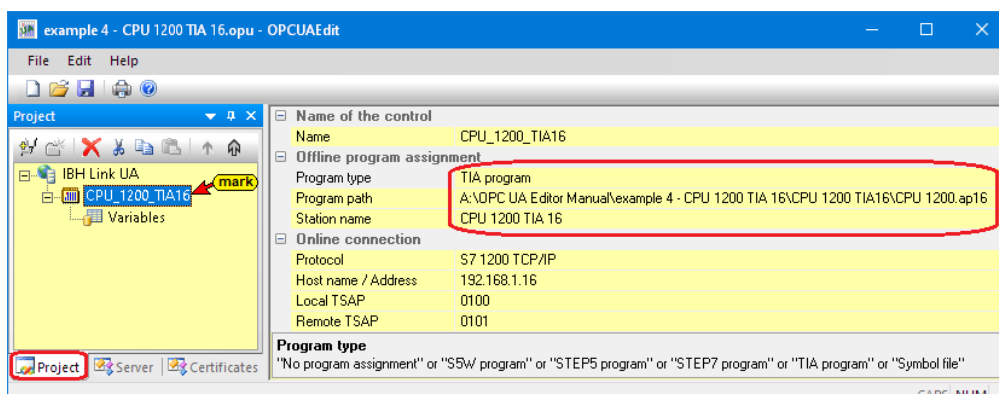
The **SIEMENS support software TIA Openness** is started in the background. If an error occurs see chapter 1, **Special features when selecting TIA projects** page 1-16.

Several notices are displayed.



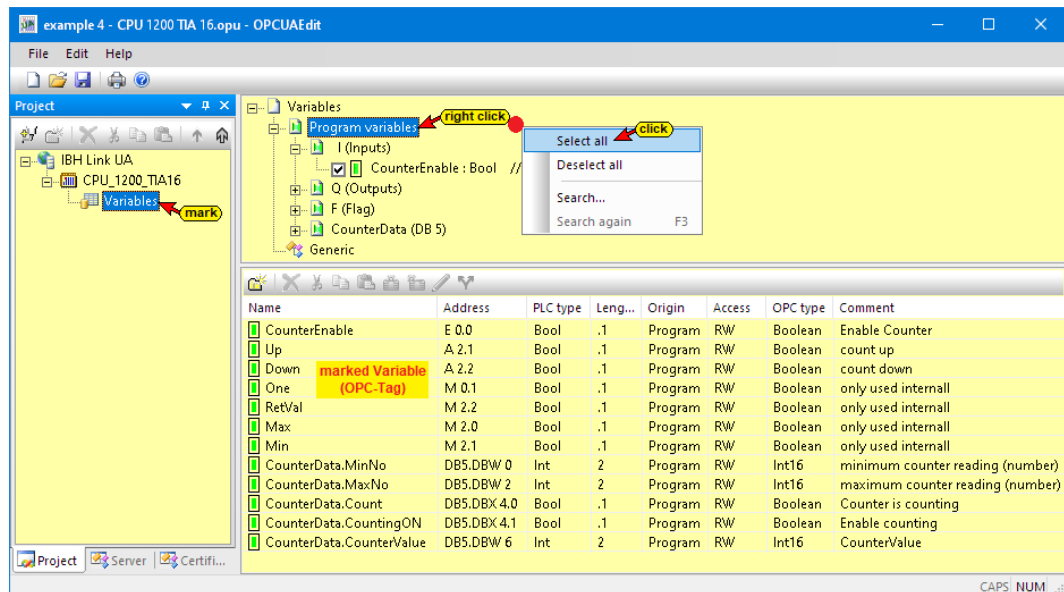
### Listed transferred project

In the right part of the project window information about the **Offline program assignment** are displayed.



### 2.4.3 Define variables as OPC tags

Clicking **Variables** lists the variables / data (data blocks) from the PLC in the right part of the project window.

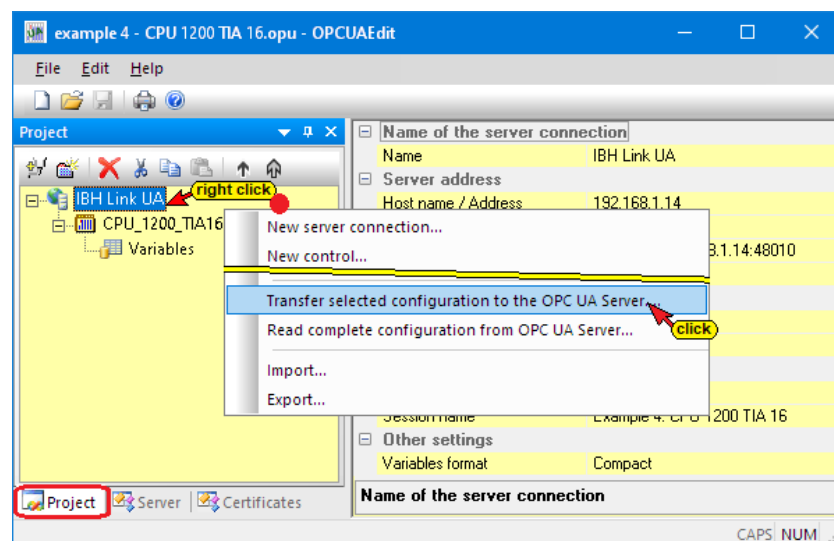


Clicking the **Plus** icon in front of the variable area symbol displays the existing variables.

If a variable is selected, it is adopted as an OPC tag and displayed in the lower part of the window with additional information.

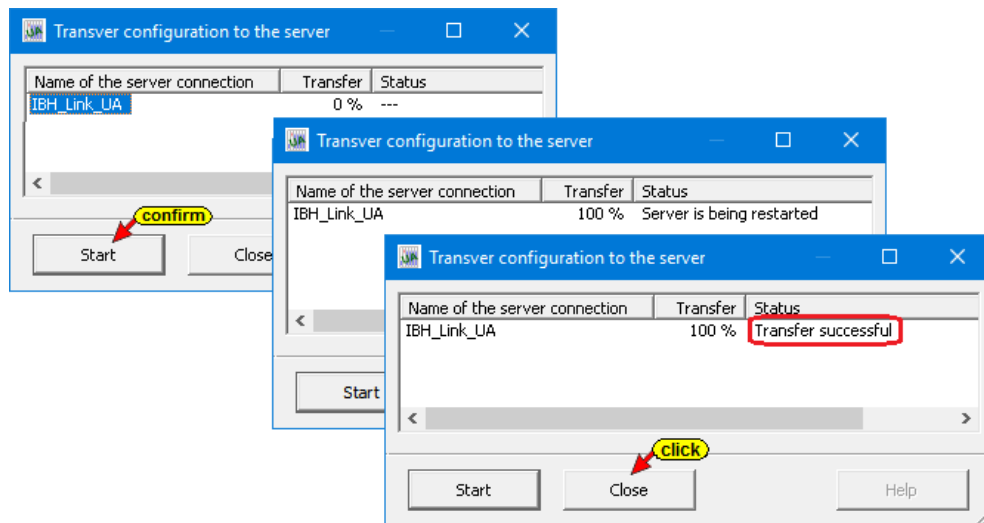
### 2.4.4 Transfer configuration to the OPC UA server (IBH Link UA).

A right-click on the Server icon (IBH Link UA) opens the context menu.



The command **Transfer Selected Configuration to OPC UA Server** command opens the **Transfer Configuration to Server** dialog box.

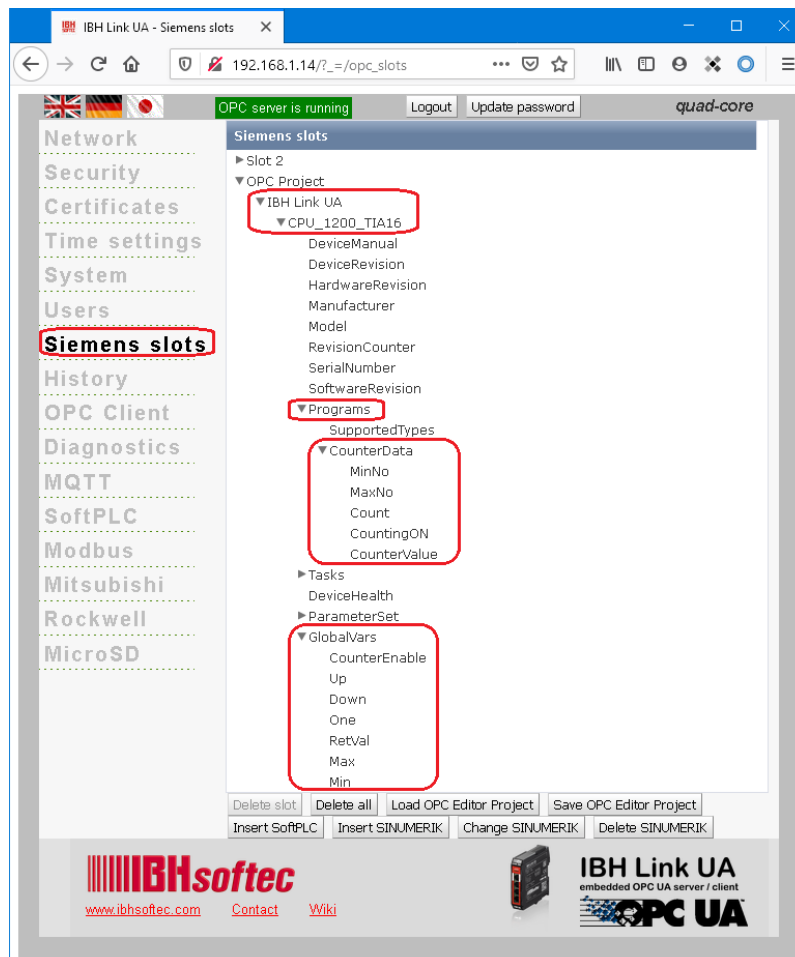
Select the server **IBH Link UA** and then click Start. The configuration (**CPU\_300\_TIA\_16**) is transferred to the **IBH Link UA**. Successful transfer is displayed.



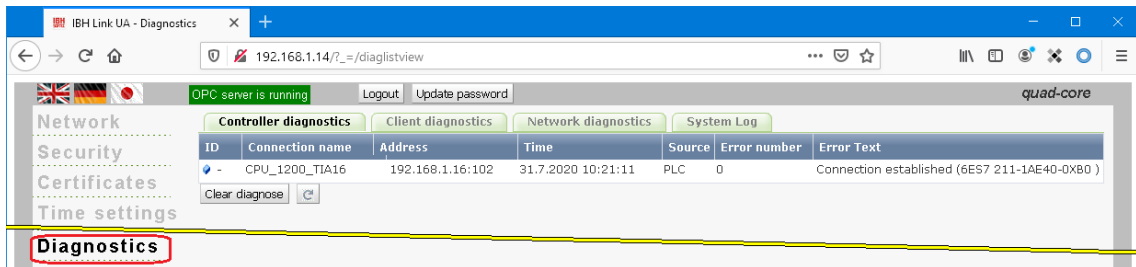
The successful transfer is displayed.

If a **certified data exchange** between the **IBH OPC Editor** and the **IBH Link UA** has been selected, the exchanged certificates must be trusted (see chapter Trust certificate, Chapter 1, page 1-40).

## IBH Link UA - Siemens Slots - Project CPU 1200 TIA 16

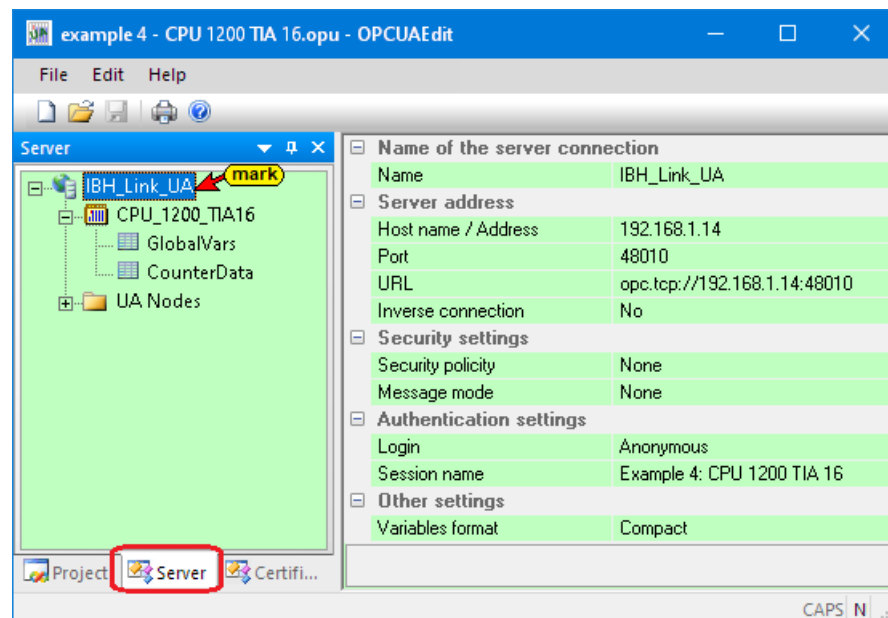


The browser window *Diagnostics* displays the status of the connection **IBH Link UA – PLC / CPU 1200 TIA 16**.



## 2.4.5 Online OPC UA Server Information

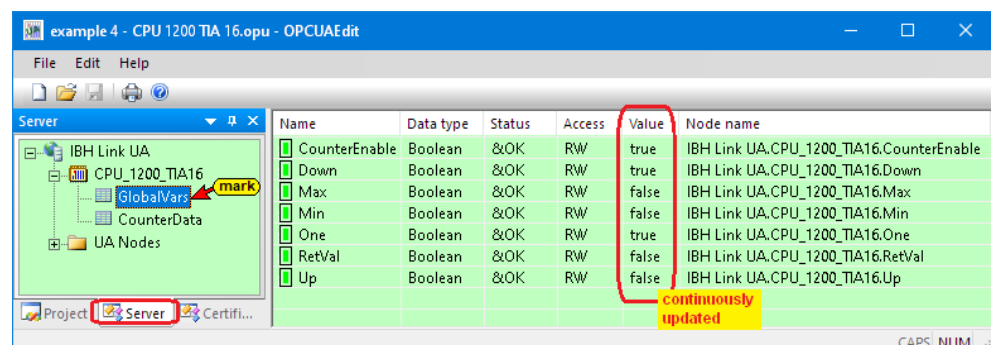
Information from the online connected **OPC UA server** with the online connected **S7 CPU 312** are displayed.



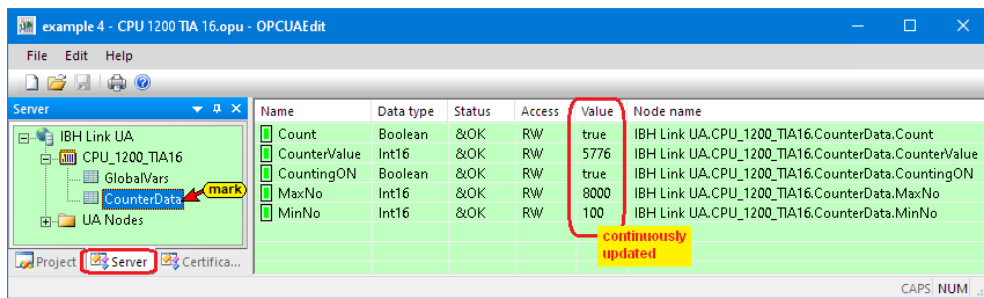
### View server

The groups of the variables (GlobalVars, data blocks) are listed in the left server window. By clicking on a group, the individual variables (OPC tags) are displayed in the right server window with their status. The status of the OPC tags is updated continuously.

### CPU 1200– Global Variable



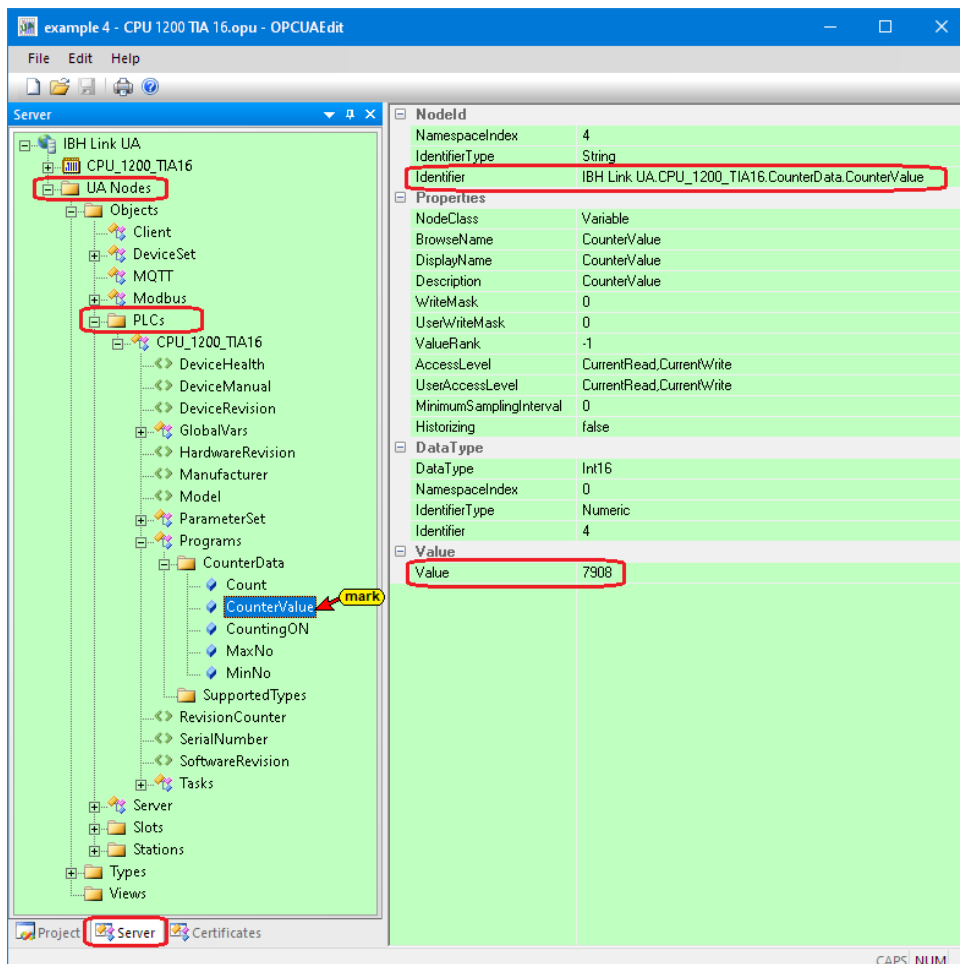
## CPU 1200 Data Block DB5 – CounterData



### Show under UA Nodes

The names of the OPC UA Nodes are listed in the left-hand server window (attributes, OPC tags, etc.). The corresponding values are displayed in the right-hand server window.

The values are current values and are only taken once when the OPC UA Node name is clicked.



### 2.4.6 Unified Automation UaExpert - The OPC Unified Architecture Client

The UaExpert program window lists the OPC tags transferred by the IBH OPC UA Editor and the associated UA nodes.



The UaExpert program window lists the OPC tags transferred by the IBH OPC UA Editor and the associated UA nodes.

The screenshot shows the UaExpert interface with the 'Data Access View' table and the 'Address Space' tree. A red box highlights the 'GlobalVars' and 'CounterData' nodes in the tree, and a callout box shows a detailed view of the data from the Data Access View table.

#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	IBHLinkUA@ib...	NS4 String IBH ...	CounterEnable	true	Boolean	19:33:05.371	19:33:06.266	Good
2	IBHLinkUA@ib...	NS4 String IBH ...	Down	false	Boolean	19:34:01.535	19:34:01.786	Good
3	IBHLinkUA@ib...	NS4 String IBH ...	Max	false	Boolean	19:33:09.017	19:33:09.766	Good
4	IBHLinkUA@ib...	NS4 String IBH ...	Min	false	Boolean	19:33:10.335	19:33:11.016	Good
5	IBHLinkUA@ib...	NS4 String IBH ...	One	true	Boolean	19:33:12.000	19:33:12.766	Good
6	IBHLinkUA@ib...	NS4 String IBH ...	RetVal	false	Boolean	19:33:13.504	19:33:14.268	Good
7	IBHLinkUA@ib...	NS4 String IBH ...	Up	true	Boolean	19:34:01.535	19:34:01.786	Good
8	IBHLinkUA@ib...	NS4 String IBH ...	Count	true	Boolean	19:33:17.296	19:33:18.019	Good
9	IBHLinkUA@ib...	NS4 String IBH ...	CounterValue	4309	Int16	19:34:02.786	19:34:03.036	Good
10	IBHLinkUA@ib...	NS4 String IBH ...	CountingON	true	Boolean	19:33:21.048	19:33:21.771	Good
11	IBHLinkUA@ib...	NS4 String IBH ...	MaxNo	8000	Int16	19:33:23.064	19:33:23.773	Good
12	IBHLinkUA@ib...	NS4 String IBH ...	MinNo	100	Int16	19:33:25.207	19:33:26.023	Good

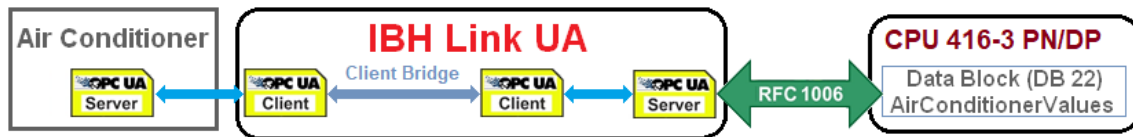
  

Display Name	Value	Datatype	Source Timestamp
CounterEnable	true	Boolean	19:33:05.371
Down	false	Boolean	19:38:03.371
Max	false	Boolean	19:33:09.017
Min	false	Boolean	19:33:10.335
One	true	Boolean	19:33:12.000
RetVal	false	Boolean	19:33:13.504
Up	true	Boolean	19:38:03.371
Count	true	Boolean	19:33:17.296
CounterValue	1767	Int16	19:38:03.871
CountingON	true	Boolean	19:33:21.048
MaxNo	8000	Int16	19:33:23.064
MinNo	100	Int16	19:33:25.207

#	Server	Node Id
1	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.CPU_1200_TIA16.CounterEnable
2	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.CPU_1200_TIA16.Down
3	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.CPU_1200_TIA16.Max
4	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.CPU_1200_TIA16.Min
5	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.CPU_1200_TIA16.One
6	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.CPU_1200_TIA16.RetVal
7	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.CPU_1200_TIA16.Up
8	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.CPU_1200_TIA16.CounterData.Count
9	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.CPU_1200_TIA16.CounterData.CounterValue
10	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.CPU_1200_TIA16.CounterData.CountingON
11	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.CPU_1200_TIA16.CounterData.MaxNo
12	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.CPU_1200_TIA16.CounterData.MinNo

## 2.5 Example 5 - project: CPU 416 TIA 16 server - server

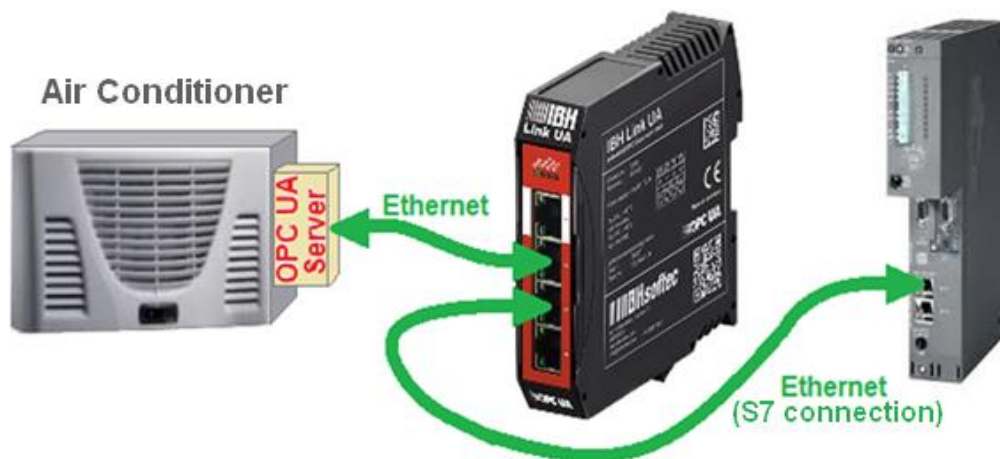
In the project, the CPU 416, which is available as an OPC UA server, is connected to an air conditioning system that also has an OPC UA server.



Establishing a server connection. The data of an air conditioning system (OPC UA server) is sent to the CPU-416 (OPC UA server).

### 2.5.1 Device configuration S7 project CPU 416 TIA 16 server – server

The data of an air conditioner are given to the CPU-416.



The data block (**DB 22 – AirConditioningValues**) stores the data of the air conditioning system for further processing.

... TIA 16 Server - Server > Server - Server [CPU 416-3 PN/DP] > Program blocks > AirConditionerValues [DB22]

AirConditionerValues					
	Name	Data type	Offset	Start value	Comment
1	Static				
2	Temperature	Real	0.0	0.0	Temperature external OPC UA Server
3	TimeStamp_Temp	Date_And_Time	4.0	DT#1990-01-01-00:00:00	Temperature Time Stamp
4	Status_Temp	DWord	12.0	16#0	Temperature Status
5	TemperatureSP	Real	16.0	0.0	Temperature Set Point external OPC UA Server
6	TimeStampSP	Date_And_Time	20.0	DT#1990-01-01-00:00:00	Temperature Set Point Time Stamp
7	StatusSP	DWord	28.0	16#0	Temperature Set Point Status
8	CurrentTime_TimeStamp	Date_And_Time	32.0	DT#1990-01-01-00:00:00	External Server Current Time
9	CurrentTime_Status	DWord	40.0	16#0	External Server Current Time Status

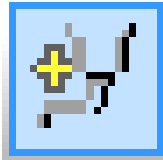
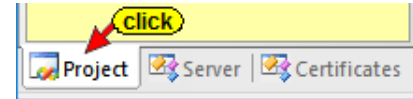
The temperature / temperature set point with time stamp and status are to be read as OPC tags from an air conditioning system. Server time and status should also be transferred for monitoring purposes.

## 2.5.2 Calling the IBH OPC UA Editor

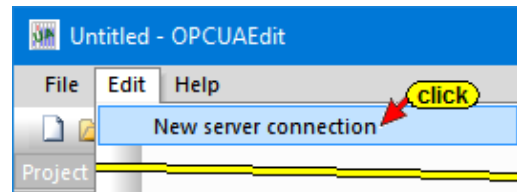
Double-click the **IBH OPC UA Editor** icon to open the program window.



Open the **Project window** by clicking on the **Project** tab.



Open the **New Server Connection** dialog box with the New Server Connection command from the **Edit** menu or by clicking the icon.



The new server connection setup was explained in example 1 (see chapter 2, page 2-3).

**Server connection properties**

Name of the server connection: IBH\_Link\_UA

Server address:

Host name or IP address: 192.168.1.14

Port: 48010

URL: opc.tcp://192.168.1.14:48010

Select endpoint...

Security settings:

None

Basic128Rsa15

Basic256

BasicSha256

Aes128Sha256RsaOaep

Aes256Sha256RsaPss

Message mode:

Signatur

Signature and Encryption

Inverse connection:

Connect invers

Properties...

Login:

Anonymous

User name and password

User name: \_\_\_\_\_

Password: \_\_\_\_\_  Store

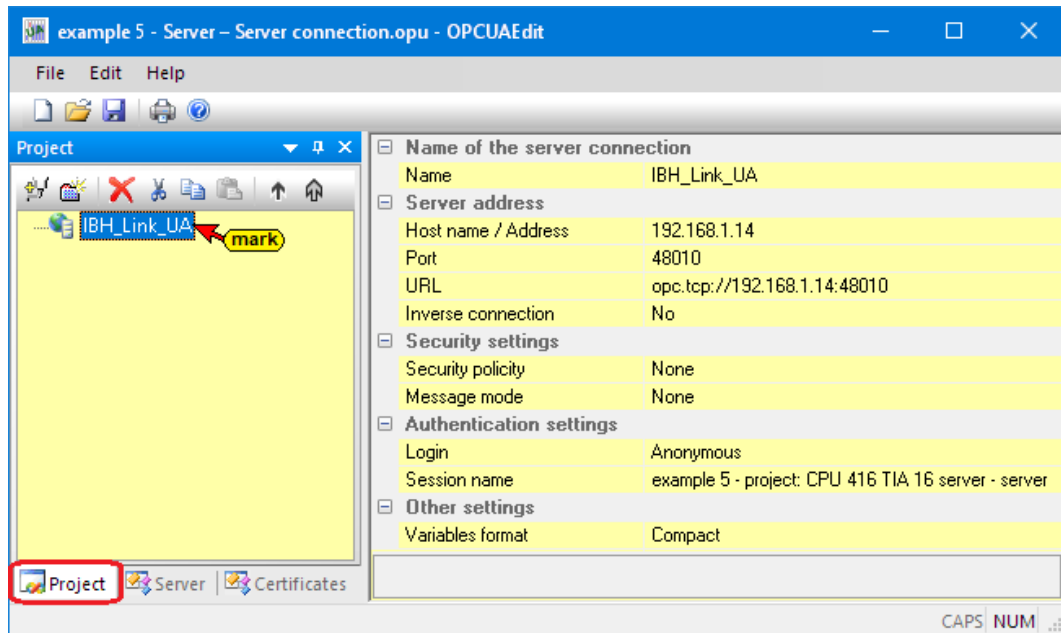
Session Name: example 5 - project: CPU 416

Variables format: Compact

confirm

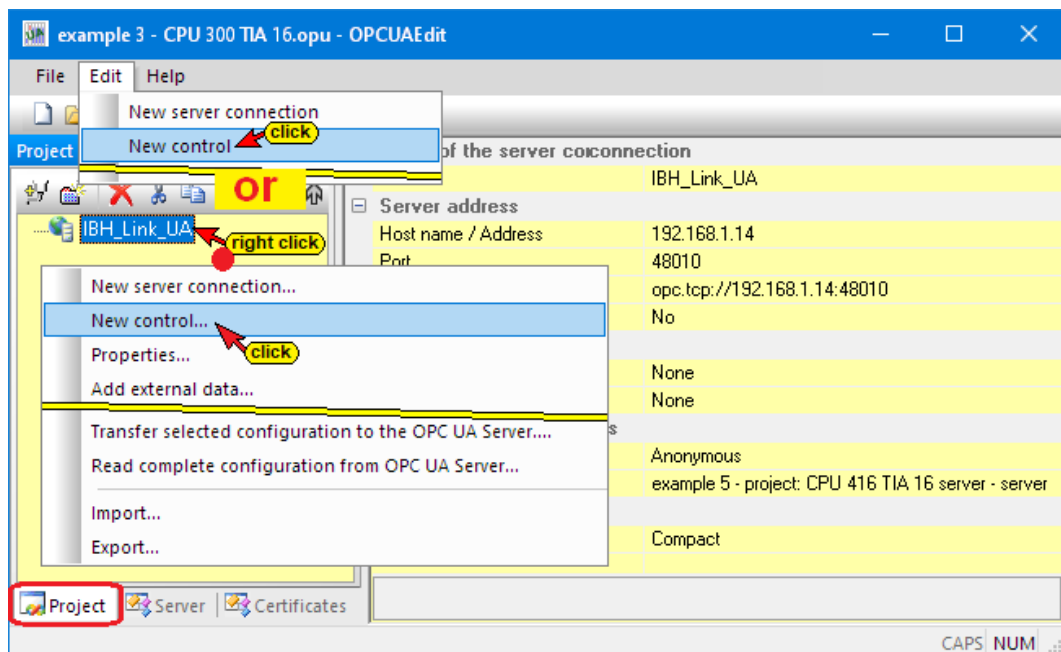
OK Cancel Help

The settings for the connection to the **IBH Link UA** OPC UA server are displayed in the right part of the **project window**.

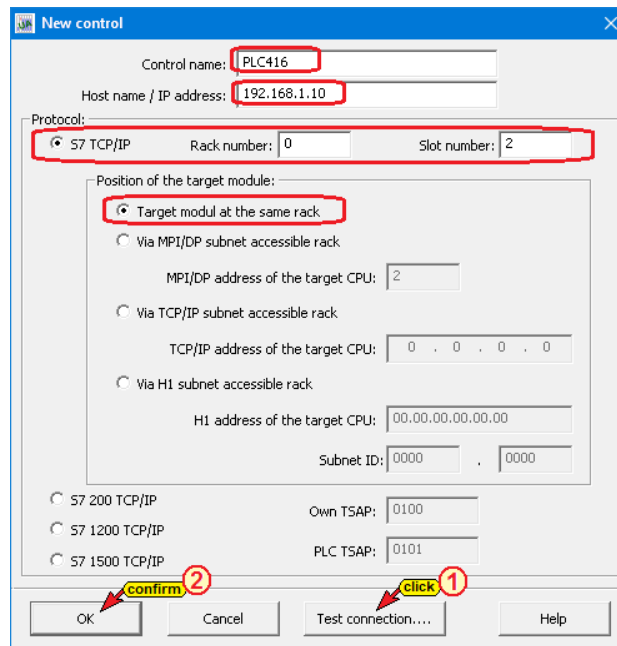


### 2.5.3 Inserting a New control (PLC)

The **New control** command from the context menu (or menu Edit / New control) opens the dialog box **New control** to specify the access to the control (CPU).



## New control dialog box

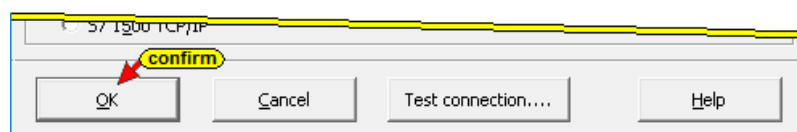
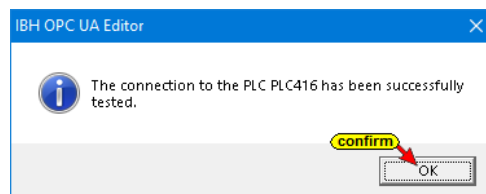


## Test connection

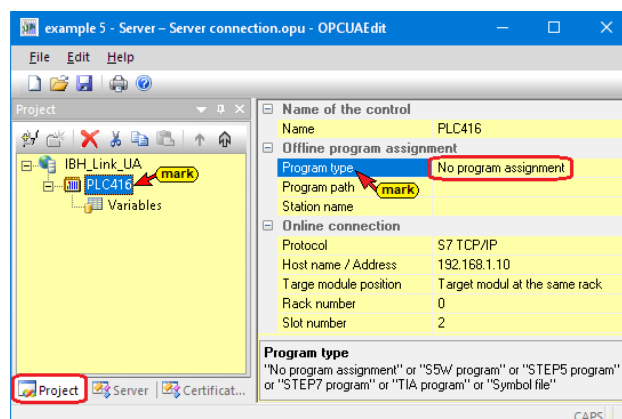
After completing the New Control dialog box, the connection to the online connected CPU can be tested.

Test connection....

Information about the successful connection is displayed.



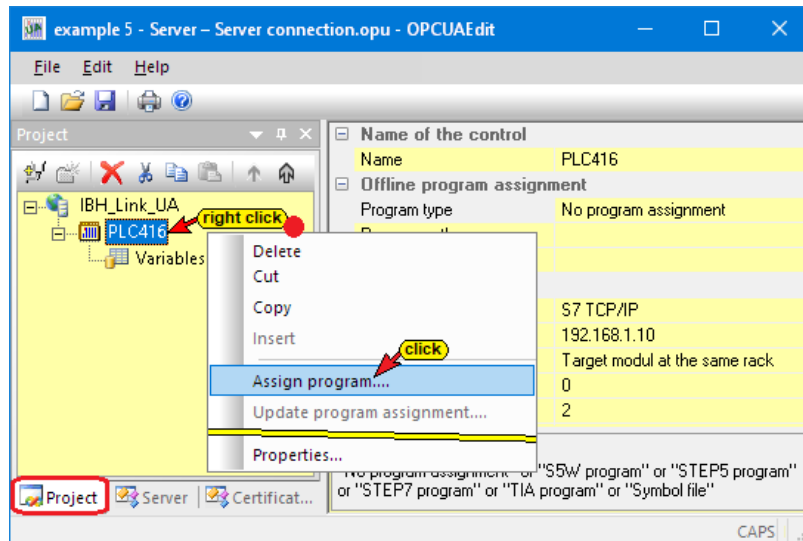
To accept and close the **New control** dialog box settings click on **OK**.



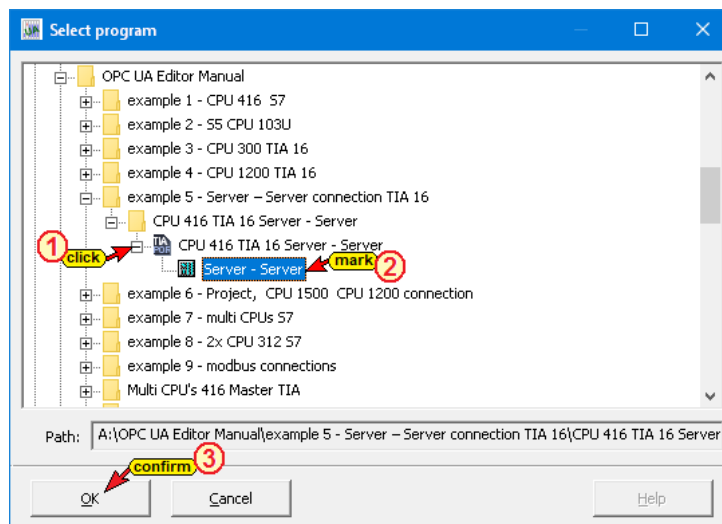
The access data of the **PLC416** (CPU 416-3 PN / DP SoftSPS with TCP/IP Port) is displayed in the right project window.

## 2.5.4 Offline program assignment

The **Assign program** command is used to open the **Program Selection** dialog box.



Select the PLC program in the **Select program** dialog box. Clicking the **Plus** symbol in front of the TIA symbol of the PLC project, the PLC program (CPUs) is displayed in the project.



### Note:



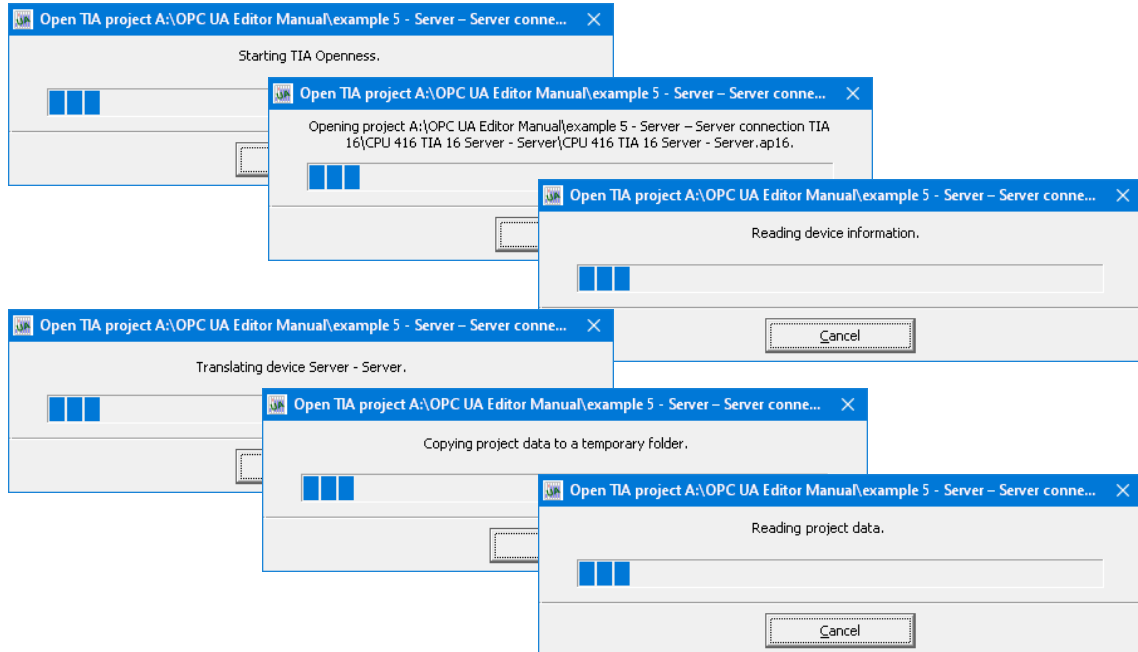
Opening a TIA project can take some time. The TIA project must be opened in the background with the SIEMENS support software TIA Openness.

To adopt the PLC program, the TIA software with the TIA Openness support software must be installed on the PC and the user of the PC must be member of the administrator in the group.

**It is essential to ensure that the software versions match.**

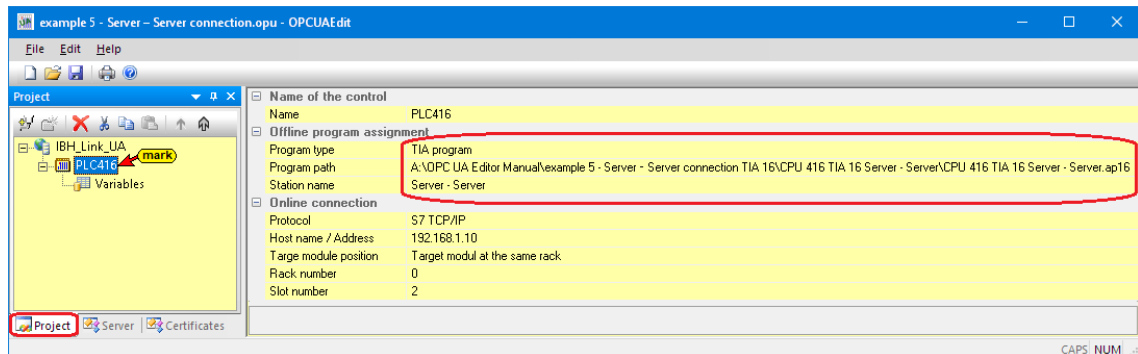
The **SIEMENS support software TIA Openness** is started in the background. If an error occurs see chapter 1, **Special features when selecting TIA projects** page 1-16.

Several notices are displayed.



## Listed transferred program

In the right part of the project window information about the **Offline program assignment** are displayed.



## 2.5.5 Define variables as OPC tags

Click Variable to list the variables / data (data blocks) of the PLC program in the right part of the project window.

The existing variables are displayed by clicking the plus symbol in front of the variable area symbol.

The variables of the data block **AirConditionerValues [DB 22]** should be selected as OPC tags by clicking.

If a variable is selected, it is adopted as an OPC tag and displayed in the lower part of the window with additional information.

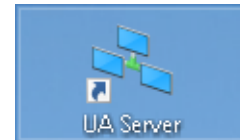
## AirConditionerValues [DB 22] selected OPC tags

The screenshot shows the IBH OPC UA Editor interface. The 'Variables' tree on the left shows 'AirConditionerValues (DB 22)' selected. The main area displays the variable declaration for 'AirConditionerValues (DB 22)' with several tags checked. A red box highlights the 'marked variable (OPC tags)' label. The table below lists the details for each tag:

Name	Address	PLC type	Leng...	Origin	Access	OPC type	Comment
AirConditionerValues.Temperature	DB22.DBD 0	Real	4	Program	RW	Float	Temperature external OPC UA Server
AirConditionerValues.Timestamp_Temp	DB22.DBX 4.0	Date_And_Time	8	Program	RW	DateTime	Temperature Time Stamp
AirConditionerValues.Status_Temp	DB22.DBD 12	DWord	4	Program	RW	UInt32	Temperature Status
AirConditionerValues.TemperatureSP	DB22.DBD 16	Real	4	Program	RW	Float	Temperature Set Point external OPC UA Server
AirConditionerValues.TimestampSP	DB22.DBX 20.0	Date_And_Time	8	Program	RW	DateTime	Temperature Set Point Time Stamp
AirConditionerValues.StatusSP	DB22.DBD 28	DWord	4	Program	RW	UInt32	Temperature Set Point Status
AirConditionerValues.CurrentTime_TimeStamp	DB22.DBX 32.0	Date_And_Time	8	Program	RW	DateTime	External Server Current Time
AirConditionerValues.CurrentTime_Status	DB22.DBD 40	DWord	4	Program	RW	UInt32	External Server Current Time Status

### 2.5.6 Start external OPC UA server

Double-click the UA Server icon to start the external **OPC UA server** program (from **UnitedAutomation**). It simulates several air conditioning and heating systems (air conditioner, furnas) and provides operating data (temperature, time value, etc.) as **OPC UA tags**.



### External OPC UA server (air conditioning and heating systems)

```

UA Server
I/O warning : failed to load external entity "file:///
C:/UA_Server_Beispiel/uanodesetimport.xml"
*****
Server opened endpoints for following URLs:
opc.tcp://TTi-Yellow:48011
*****
Press x to shutdown server
*****

```

The external OPC UA server has the endpoint URL:

**opc.tcp://TTi-Yellow:48011**

Since there is no name server (DNS server), the absolute endpoint URL must be used:

**opc.tcp://192.168.1.10: 48011**


### 2.5.7 Adding a server

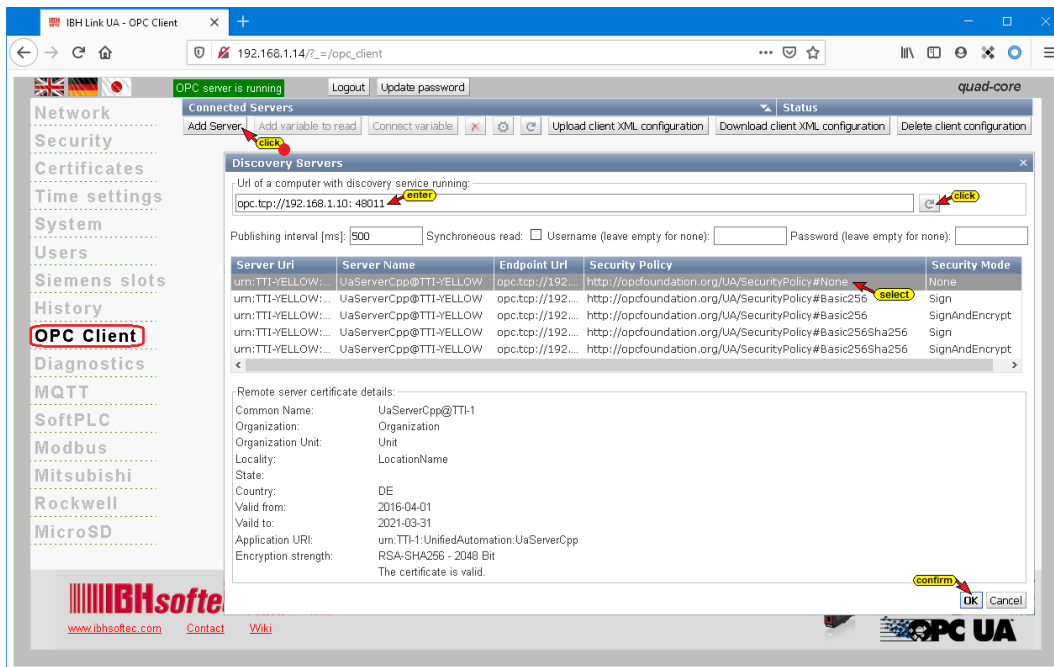
There are two options to add an external server. This can be done via the **IBH Link UA** or using the **IBH OPC UA Editor**.



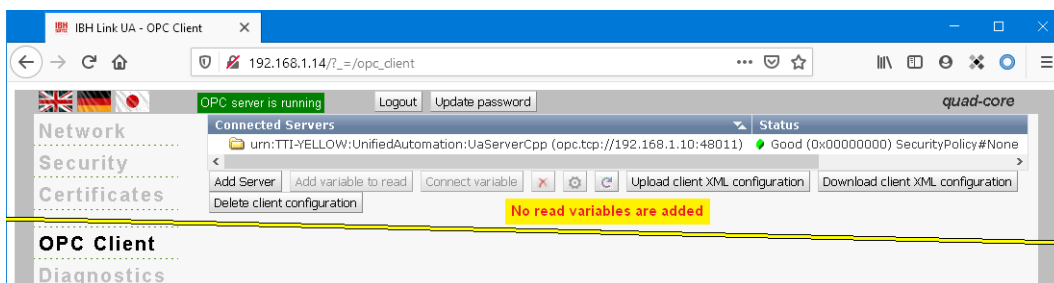
## 2.5.8 Option 1: Adding a server via the IBH Link UA

In the IBH Link UA web browser window **OPC Client** click **Add Server**. Enter the endpoint URL **opc.tcp://192.168.1.10:48011**.

 Click the icon and select the Security Policy.



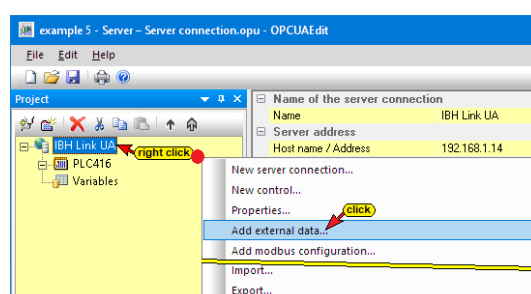
For the data transmission **Security Policy None** were selected.



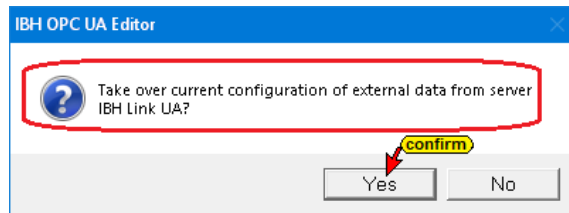
The server - **UaServerCpp (opc.tcp://192.168.1.10: 48011)** is connected. The status of the external OPC UA server is good. No read variables are added.

## Add external data

Right-click on **IBH Link UA** in the IBH OPC UA Editor and execute the **Add external data ...** command.

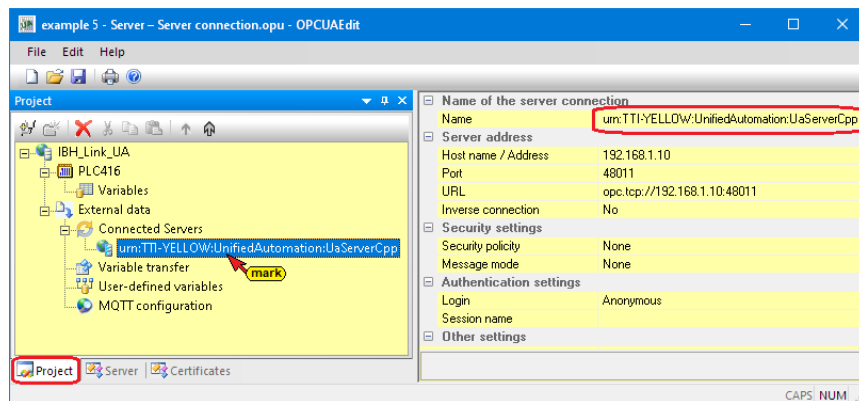


Clicking on **Add External Data...** opens the IBH OPC UA Editor dialog box.



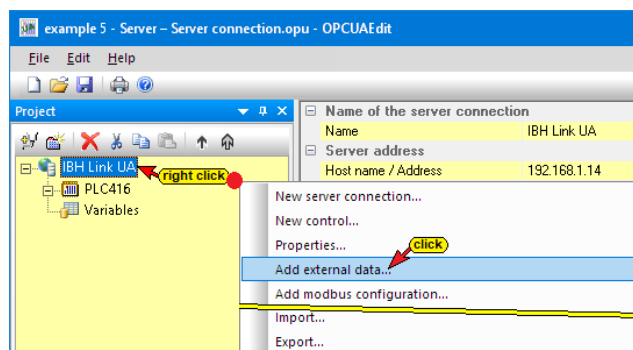
If the external server **UaServerCpp (opc.tcp: //192.168.1.10: 48011)** has already exists in the IBH Link UA, you can click the **Yes** button in the opened dialog box. The configuration of the external server (OPC tags, names etc.) is transferred into the IBH OPC UA Editor.

## OPC server adopted as external data

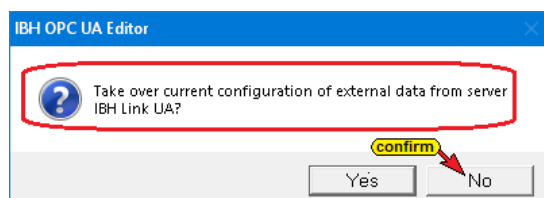


### 2.5.9 Option 2: Add server via the IBH OPC UA Editor

The server - **UaServerCpp (opc.tcp: //192.168.1.10: 48011)** - does not exist in the IBH Link UA web browser window / OPC client.

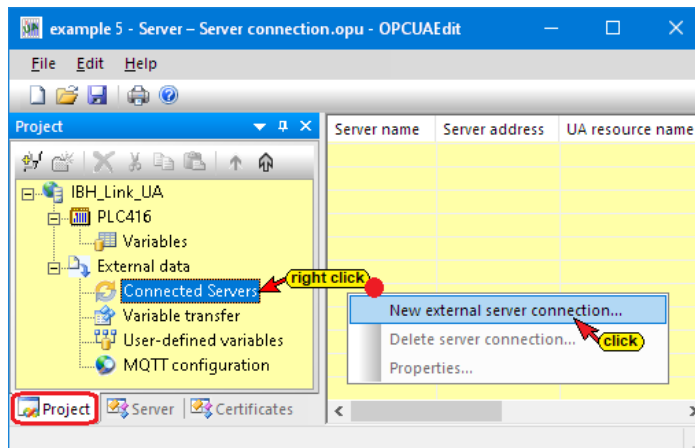


Clicking on **Add External Data...** opens the IBH OPC UA Editor dialog box. Click the **No** button to close the dialog box.

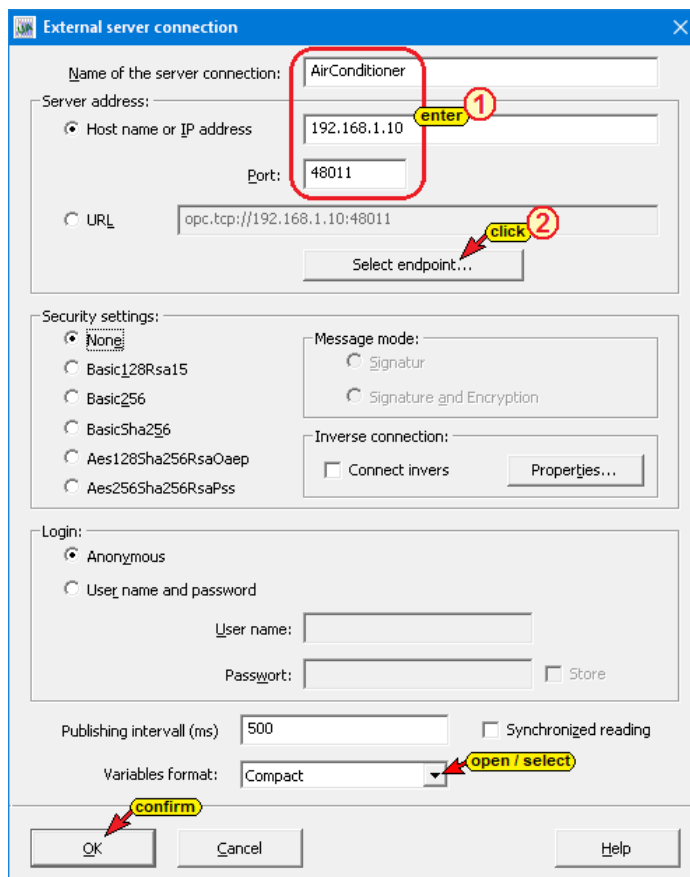


## New external server connection – configuration

The **New external server connection** command opens the dialog box to configure an OPC UA server to reading variables.

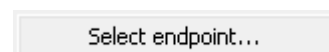


The connection data to the external server – **Air Conditioner** – (IP address 192.168.1.10: 48011) must be specified.

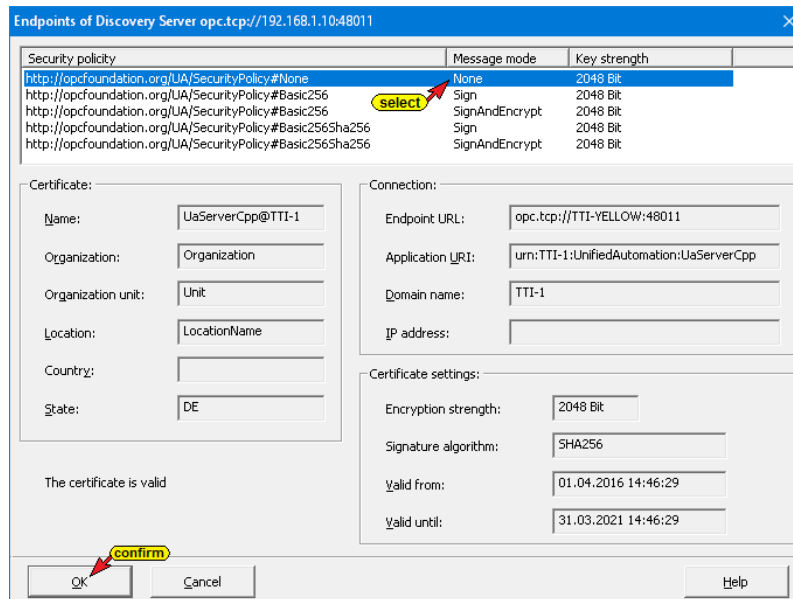


### Select end point

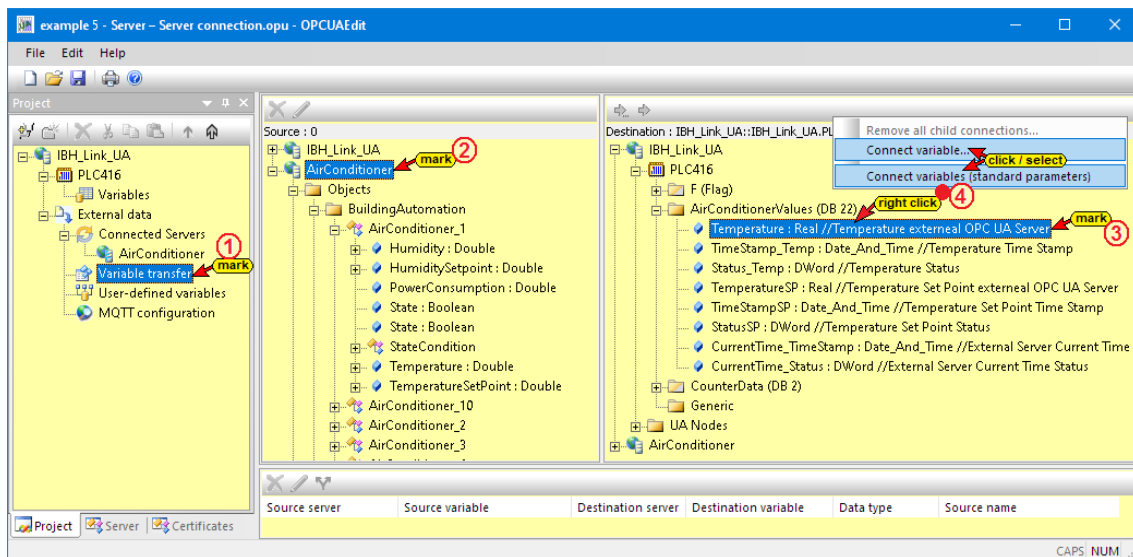
By clicking the **Select endpoint** button, a connection to the specified OPC UA server is established. If the connection is successful, possible encryptions of the data to be transmitted are displayed for selection in the opened dialog box.



The desired security procedure for the data exchange between the **IBH OPC Editor** and the **OPC UA server** (IBH Link UA) must be specified.



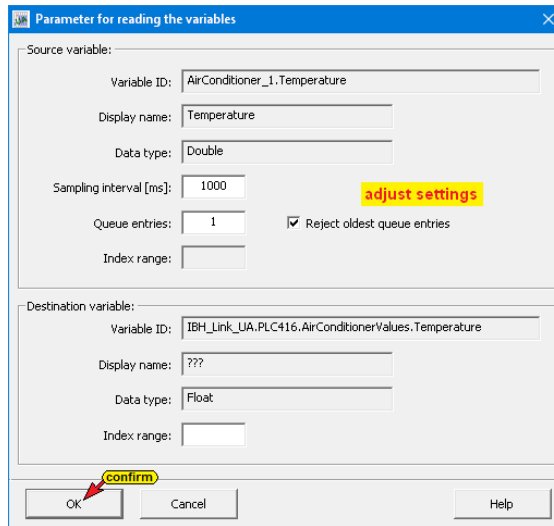
## 2.5.10 Variable transfer - define source and target variables



The OPC UA variable connection is adopted by marking the source and target variables and then clicking the **Connect variable** command (standard parameters). The connection is displayed.

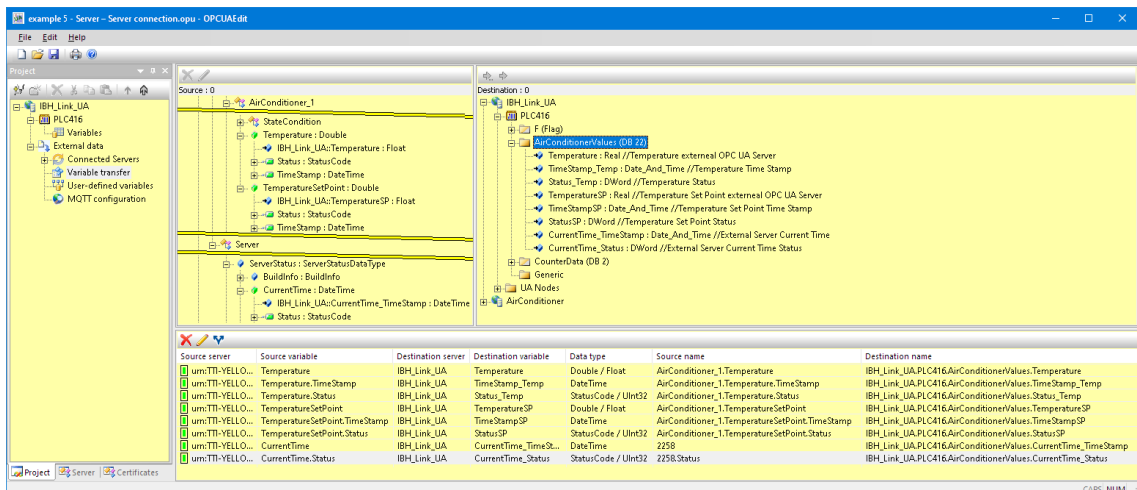
Source server	Source variable	Destination server	Destination variable	Data type	Source name	Destination name	Status
IBH_Link_UA	AirConditioner	IBH_Link_UA	Temperature	Double / Float	AirConditioner_1.Temperature	IBH_Link_UA.PLC416.AirConditionerValues.Temperature	Connected variable

The **Connect variable...** command opens the **Parameters for reading the variables** dialog box. The ID of the variables, the names and data types are displayed here. Sampling interval, queue entries and possibly the index area can be adjusted.



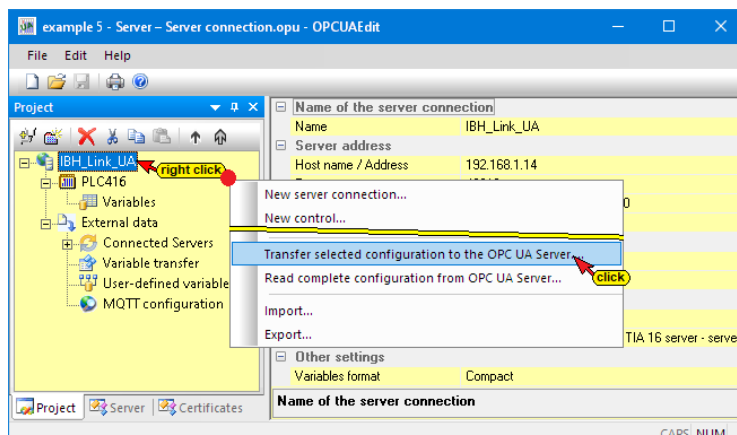
If the connection of a variable (value) has been done, the time stamp and status of the source variable are offered.

## Variable links



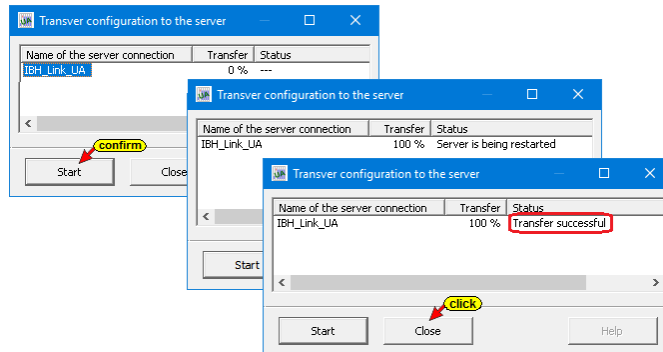
### 2.5.11 Transferring the configuration to the OPC UA server (IBH Link UA)

A right click on the Server icon (IBH Link UA) opens the context menu.



The command **Transfer Selected Configuration to OPC UA Server** command opens the **Transfer Configuration to Server** dialog box.

Select the server **IBH Link UA** and then click Start. The configuration is transferred to the **IBH Link UA**. Successful transfer is displayed.

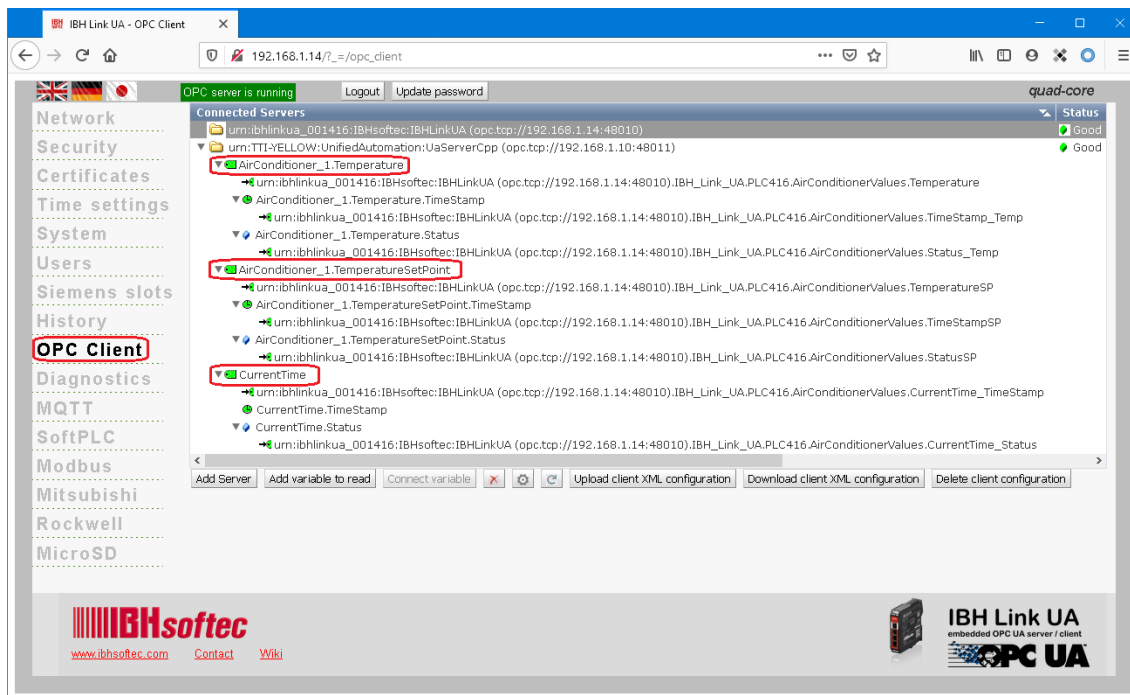


The successful transfer is displayed.

If a certified data exchange between the **IBH OPC Editor** and the **IBH Link UA** has been selected, the exchanged certificates must be trusted (see chapter Trust certificate, Chapter 1, page 1-40).

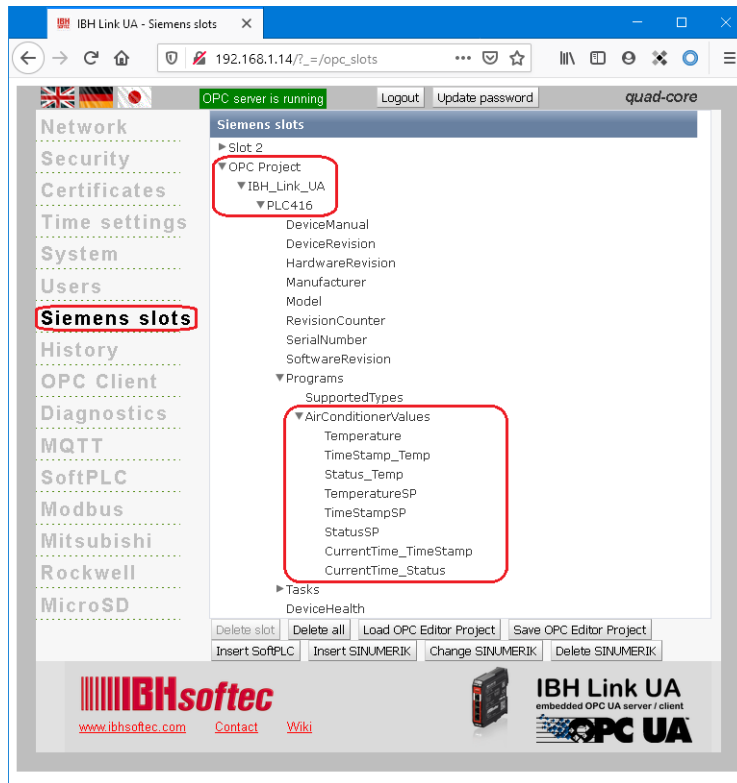
## 2.5.12 Displaying the links

The connections are displayed in the IBH Link UA web browser window OPC Client.

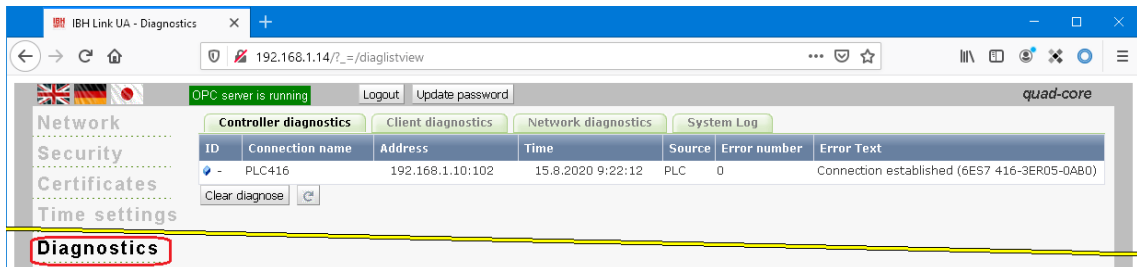


In the IBH Link UA web browser window **Siemens Slots**, the CPU (PLC416) connected via the IBH Link UA is listed with the **OPC tags** data block **AirConditionerValues [DB22]** selected in the IBH OPC UA Editor.

## Siemens slots - Project CPU 416 TIA 16 server - server



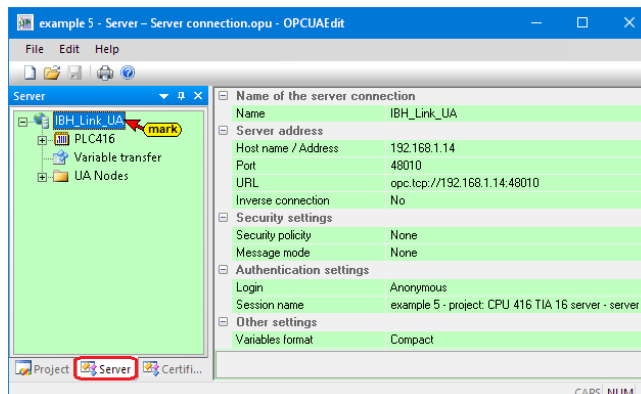
The browser window *Diagnostics* displays the status of the connection *IBH Link UA – PLC / CPU 300 TIA 16*.



### 2.5.13 OPC UA server information online

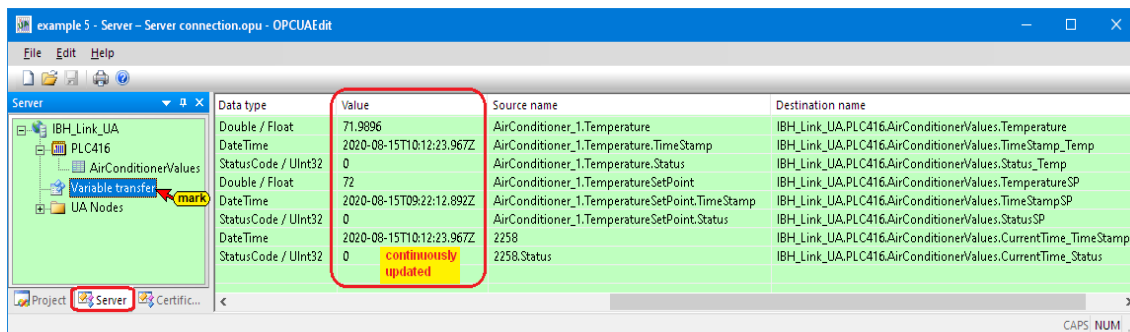
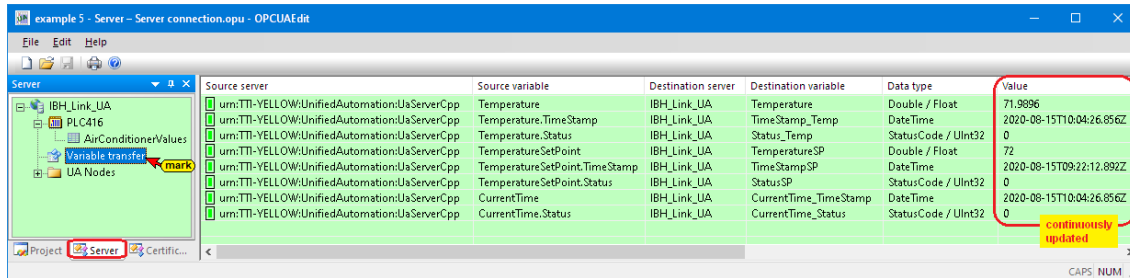
Information from the OPC UA server connected online with the PLC416 are displayed.

#### IBH Link UA

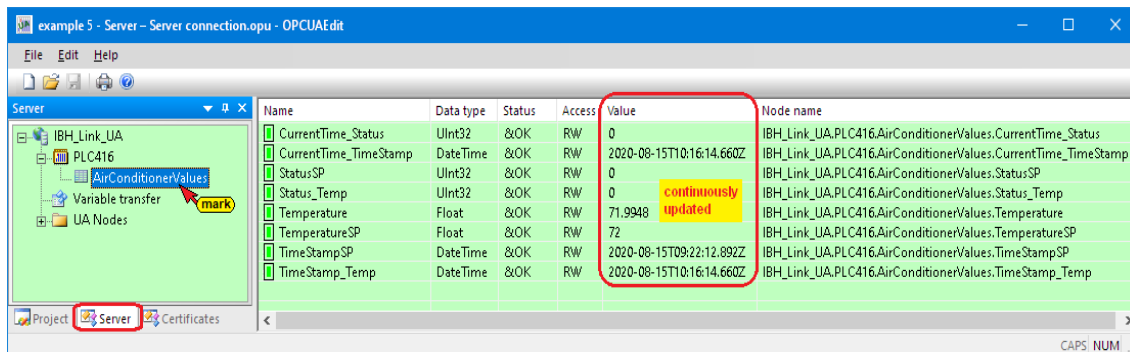


## Show variable transfer

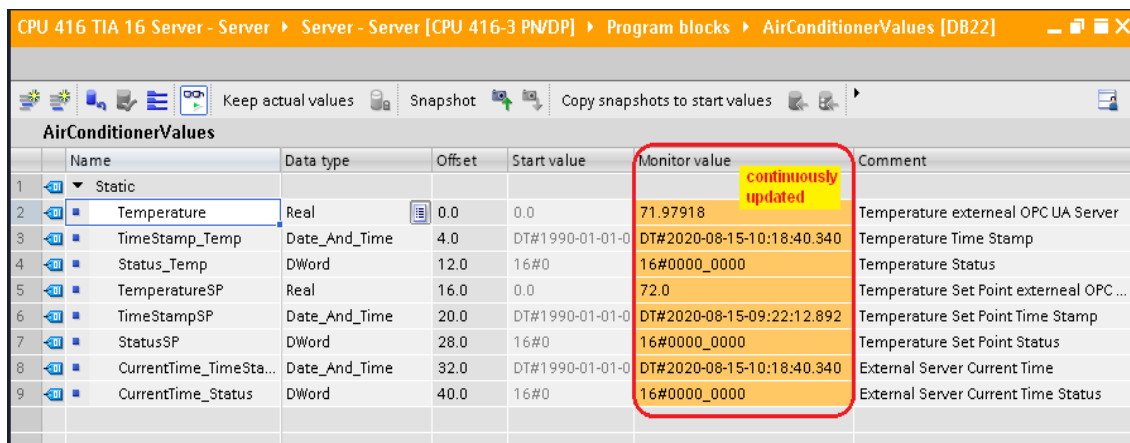
Mark Variable transfer to display the status of the OPC tags in the right server window. The status of the OPC tags is updated continuously.



## Status – Data block AirConditionerValue [DB 22]



### 2.5.14 Status – AirConditionerValue [DB 22] – CPU 416 TIA 16 Server - Server





### 2.5.15 UaExpert - Data Access View

With the external OPC server connected information about the external OPC server are displayed in the **UaExpert** program window **Data Access View**.

Use Drag & Drop to display the variables CurrentTime, Temperature, TemperatureSetPoint etc. in the Data Access Viewer window.

#### Server – IBH Link UA

#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	IBHLinkUA@ib...	NS4 String IBH...	CurrentTime_Status	0	UInt32	12:40:43.151	12:40:43.704	Good
2	IBHLinkUA@ib...	NS4 String IBH...	CurrentTime_TimeStamp	2020-08-15T10:42:58.141Z	DateTime	12:42:58.119	12:42:58.254	Good
3	IBHLinkUA@ib...	NS4 String IBH...	Status_SP	0	UInt32	12:40:46.725	12:40:47.305	Good
4	IBHLinkUA@ib...	NS4 String IBH...	Status_Temp	0	UInt32	12:40:47.972	12:40:48.456	Good
5	IBHLinkUA@ib...	NS4 String IBH...	Temperature	71.9944	Float	12:42:58.119	12:42:58.254	Good
6	IBHLinkUA@ib...	NS4 String IBH...	TemperatureSP	72	Float	12:40:51.421	12:40:51.708	Good
7	IBHLinkUA@ib...	NS4 String IBH...	Time StampSP	2020-08-15T09:22:12.892Z	DateTime	12:40:53.279	12:40:53.459	Good
8	IBHLinkUA@ib...	NS4 String IBH...	Time Stamp_Temp	2020-08-15T10:42:58.141Z	DateTime	12:42:58.119	12:42:58.254	Good

#	Display Name	Value	Datatype	Source Timestamp
1	CurrentTime_Status	0	UInt32	13:15:26.036
2	CurrentTime_TimeStamp	2020-08-15T11:21:30.500Z	DateTime	13:21:30.440
3	Status_SP	0	UInt32	13:15:28.872
4	Status_Temp	0	UInt32	13:15:30.735
5	Temperature	71.9948	Float	13:21:30.440
6	TemperatureSP	72	Float	13:15:33.587
7	Time StampSP	2020-08-15T09:22:12.892Z	DateTime	13:15:34.558
8	Time Stamp_Temp	2020-08-15T11:21:30.500Z	DateTime	13:21:30.440

#	Server	Node Id	Display Name
1	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.PLC416.AirConditionerValues.CurrentTime_Status	CurrentTime_Status
2	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.PLC416.AirConditionerValues.CurrentTime_TimeStamp	CurrentTime_TimeStamp
3	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.PLC416.AirConditionerValues.Status_SP	Status_SP
4	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.PLC416.AirConditionerValues.Status_Temp	Status_Temp
5	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.PLC416.AirConditionerValues.Temperature	Temperature
6	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.PLC416.AirConditionerValues.TemperatureSP	TemperatureSP
7	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.PLC416.AirConditionerValues.TimeStampSP	Time StampSP
8	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH_Link_UA.PLC416.AirConditionerValues.TimeStamp_Temp	Time Stamp_Temp

#### Server – UaServerCpp (Air conditioner)

#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	UaServerCpp@TTI-YELLOW	NS3 String AirCond...	Temperature	71.973975	Double	12:49:00.774	12:49:00.774	Good
2	UaServerCpp@TTI-YELLOW	NS3 String AirCond...	TemperatureSetPoint	72	Double	11:22:12.892	12:47:25.961	Good
3	UaServerCpp@TTI-YELLOW	NS0 Numenc 2258	CurrentTime	2020-08-15T10:49:01.408Z	DateTime	12:49:01.408	12:49:01.408	Good

#	Server	Node Id	Display Name
1	UaServerCpp@TTI-YELLOW	NS3 String AirConditioner_1.Temperature	Temperature
2	UaServerCpp@TTI-YELLOW	NS3 String AirConditioner_1.TemperatureSetPoint	TemperatureSetPoint
3	UaServerCpp@TTI-YELLOW	NS0 Numenc 2258	CurrentTime

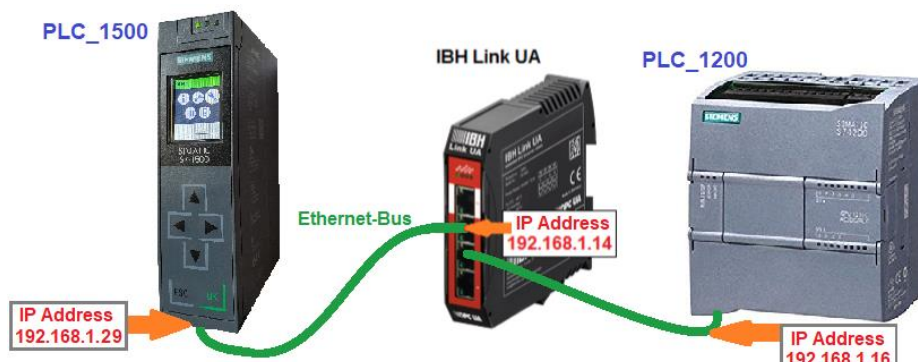
#	Display Name	Value	Datatype	Source Timestamp
1	Temperature	71.98959	Double	13:11:48.662
2	TemperatureSetPoint	72	Double	11:22:12.892
3	CurrentTime	2020-08-15T11:11:48.662Z	DateTime	13:11:48.662

## 2.6 Example 6 - CPU 1500 / CPU 1200 (Server – Server Connection)

A CPU 1211C with a TCP / IP port is connected to a CPU 1511-1 PN, which is also a TCP / IP port.

### 2.6.1 Device configuration project CPU 1500 - CPU 1200

The data from a CPU 1500 and CPU 1200 are exchanged.



### 2.6.2 Exchanging variables

Variables of the **PLC 1200** data block **CounterValues [DB 5]** are transferred into the **PLC 1500** data block **ReceivedValues [DB 10]** for further processing.

Variables of the **PLC 1500** data block **CounterValues [DB 5]** are transferred into the **PLC 1200** data block **ReceivedValues [DB 10]** for further processing.

### PLC 1200 data block ReceivedValues [DB 10]

... PLC_1200 [CPU 1211C DC/DC/DC] ▶ Program blocks ▶ ReceivedValues [DB10]						
ReceivedValues						
	Name	Data type	Offset	Start value	Comment	
1	Static					
2	ON_1500	Bool	0.0	false	Counter is counting 1500	
3	MinValue1500	Int	2.0	0	Minimum count 1500	
4	MaxValue1500	Int	4.0	0	Maximum count 1500	
5	CounterValue1500	Int	6.0	0	Counter value1500	
6	CounterStatus1500	DWord	8.0	16#0	Counter status1500	
7	CounterTimeStamp	String	12.0	"	Counter Time Stamp	

The values are read from the PLC 1500 as OPC tags. The status of **CounterStatus1500** should also be transferred for monitoring purposes.

The TimeStamp of CounterValue1500 cannot be accepted directly. A CPU 1200 does not recognize the data type **Date\_And\_Time**.

A conversion of **Date\_And\_Time** into data types that the CPU 1200 understands could take place in the PLC integrated in the IBH Link UA.

## PLC 1500 data block ReceivedValues [DB 10]

	Name	Data type	Offset	Start value	Comment
1	Static				
2	ON1200	Bool	0.0	false	Counter is counting 1200
3	MinValue1200	Int	2.0	0	Minimum count 1200
4	MaxValue1200	Int	4.0	0	Maximum count 1200
5	CounterValue1200	Int	6.0	0	Counter value 1200
6	CounterStatus1200	DWord	8.0	16#0	Counter status 1200
7	CounterTimeStamp1200	Date_And_Time	12.0	DT#1990-01-01-00:00:00	Counter Time Stamp 1200

The values are read from the PLC 1200 as OPC tags. The **status** and the **TimeStamp** of **CounterStatus1500** should also be transferred for monitoring purposes.

### Note:

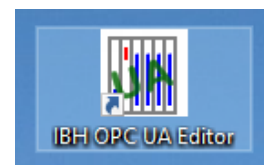


The S7 communication - **GET and PUT** (allow access via PUT / GET by remote partner) in the program of the CPU1200 and the CPU 1500 must be activated.

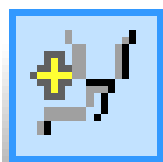
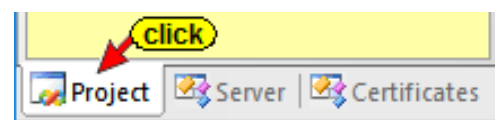
**The IBH Link UA Server cannot access DB variables in an optimized data block (DB) of an S7-1200 CPU / S7-1500 CPU.**

## 2.6.3 Calling the IBH OPC UA Editor

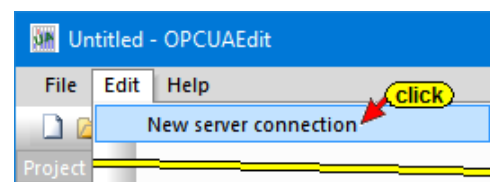
Double-click the **IBH OPC UA Editor** icon to open the program window.



Open the **Project window** by clicking on the **Project** tab.

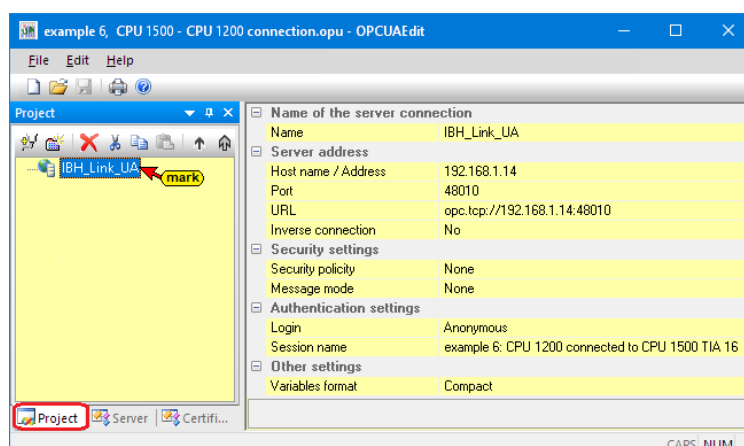


Open the **New Server Connection** dialog box with the New Server Connection command from the **Edit** menu or by clicking the icon.



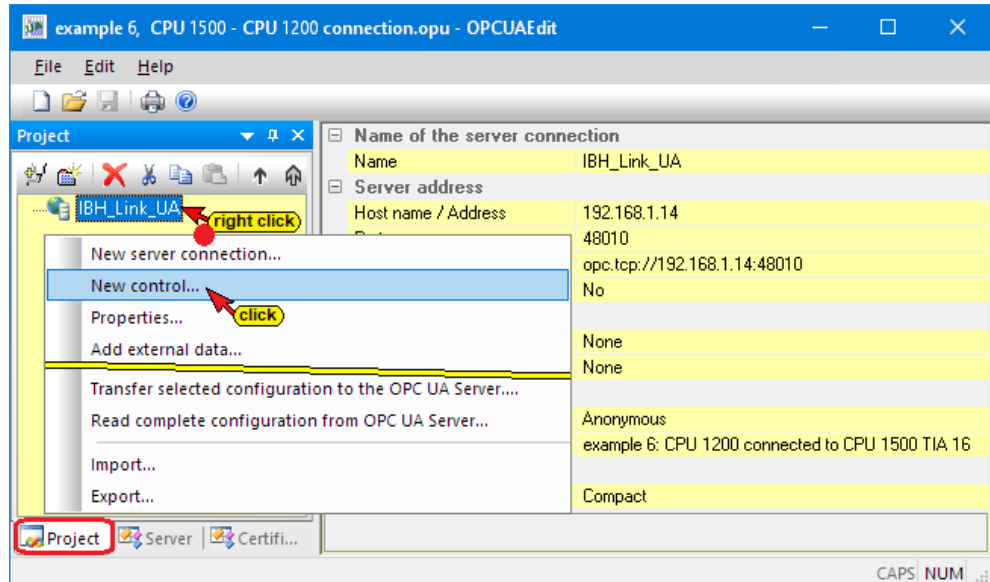
The new server connection setup was explained in example 1 (see chapter 2, page 2-3).

The settings for the connection to the **IBH Link UA** OPC UA server are displayed in the right part of the **project window**.

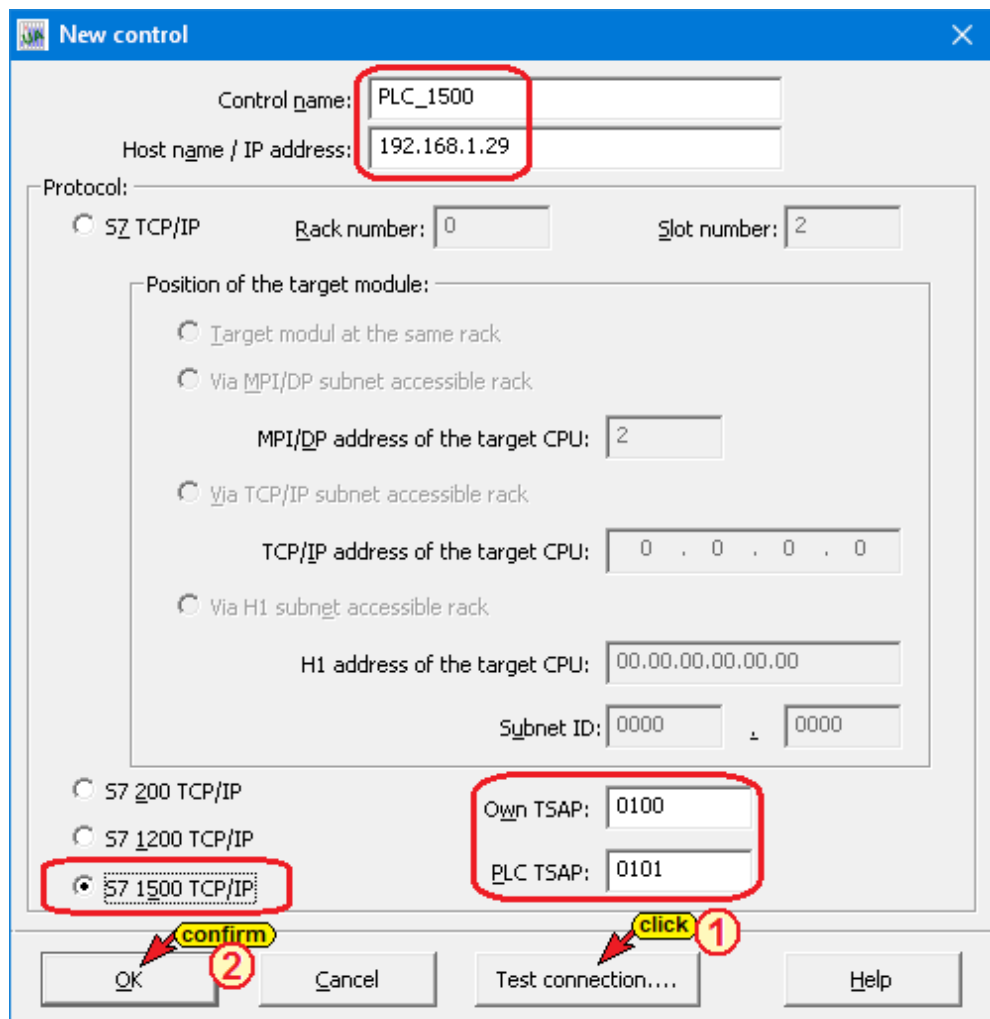


## 2.6.4 Inserting a New control (PLC)

The **New control** command from the context menu (or menu Edit / New control) opens the dialog box **New control** to specify the access to the control (CPU).

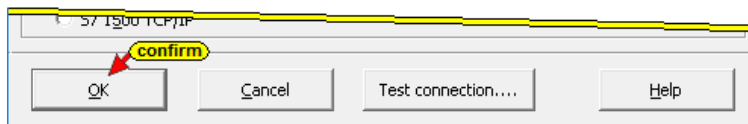
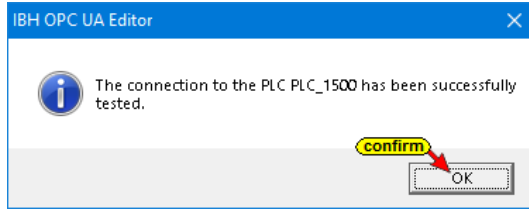


### New control dialog box – PLC\_1500



**Test connection**

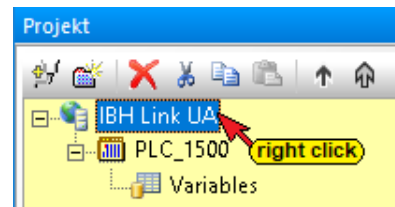
After completing the New Control dialog box, the connection to the online connected CPU can be tested. Information about the successful connection is displayed.



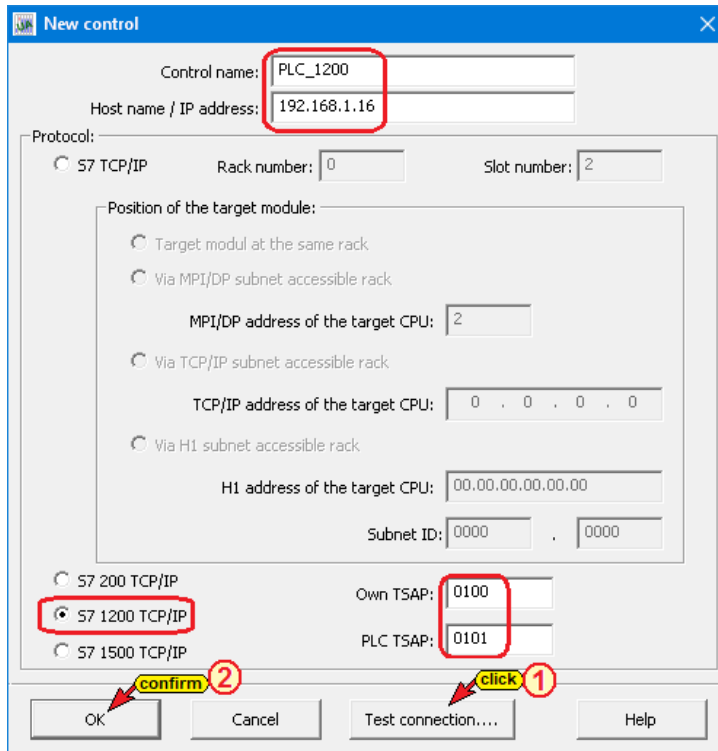
To accept and close the **New control** dialog box settings click on **OK**.

Two CPUs in the project.

Right-click on the **Server ...** icon (IBH Link UA) and select from the context menu the New Control command to open the New Control dialog box again.



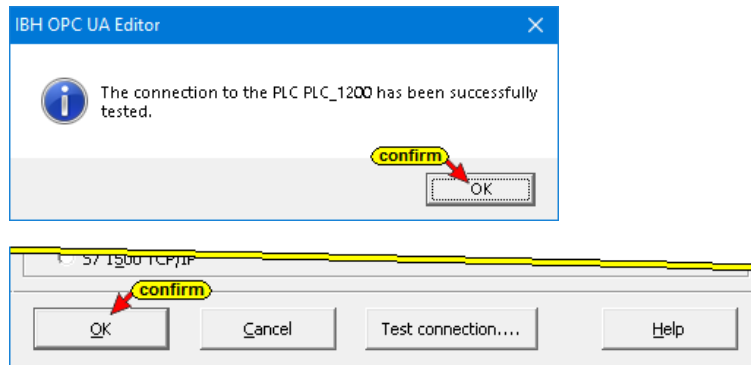
**New control dialog box – PLC\_1200**



**Test connection**

After completing the New Control dialog box, the connection to the online connected CPU can be tested. Information about the successful connection is displayed.





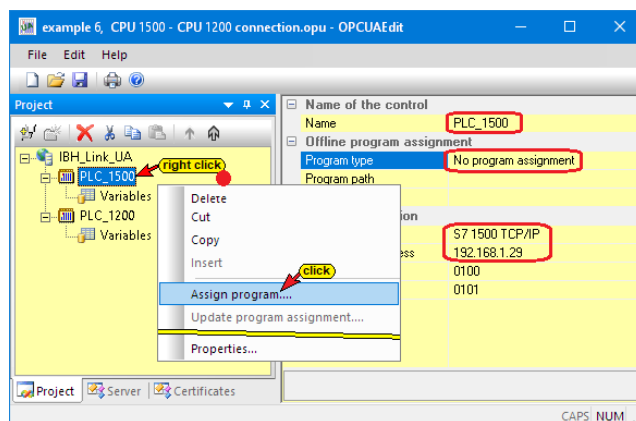
To accept and close the **New control** dialog box settings click on **OK**.

## 2.6.5 Program assignment

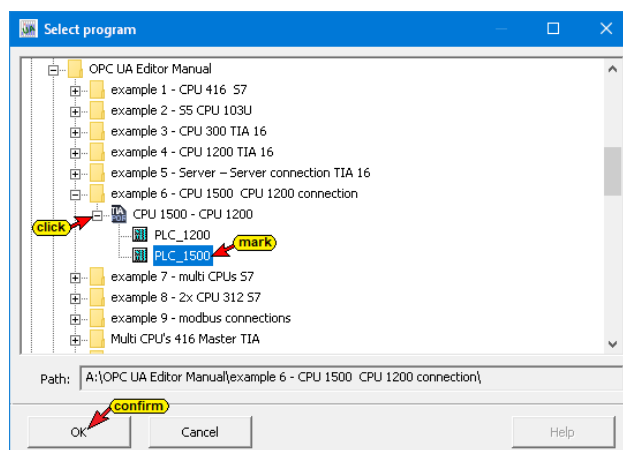
The OPC variables from the project are to be assigned to the two CPUs.

### Program assignment PLC 1500

The Assign program command from the context menu opens the Program selection dialog box.

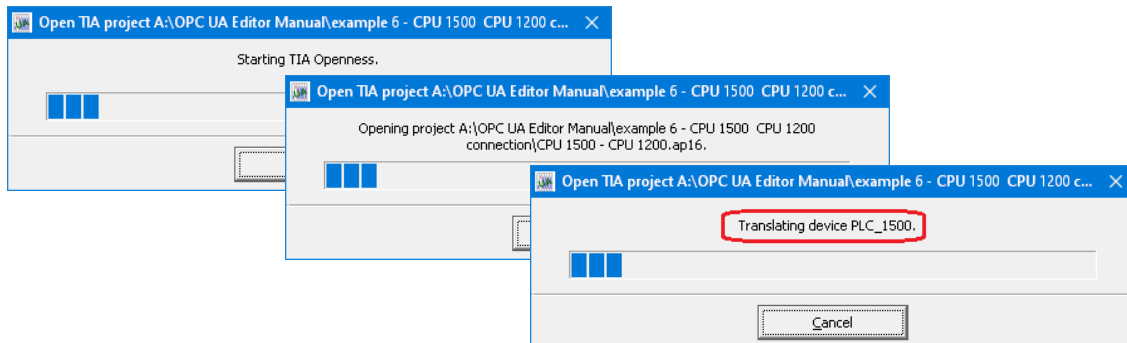


Select the PLC program in the **Select program** dialog box. Clicking the **Plus** symbol in front of the TIA symbol of the PLC project, the PLC program (CPUs) is displayed in the project.



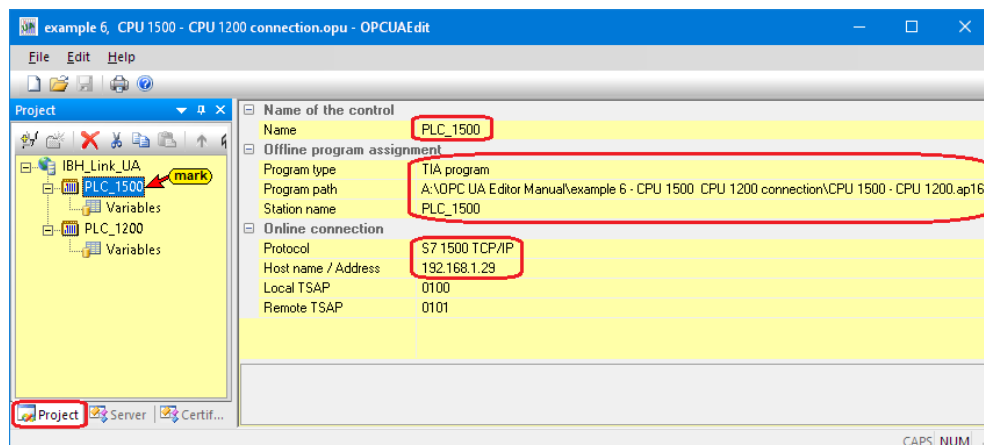
The **SIEMENS support software TIA Openness** is started in the background. If an error occurs see chapter 1, **Special features when selecting TIA projects** page 1-16.

Several notices are displayed.



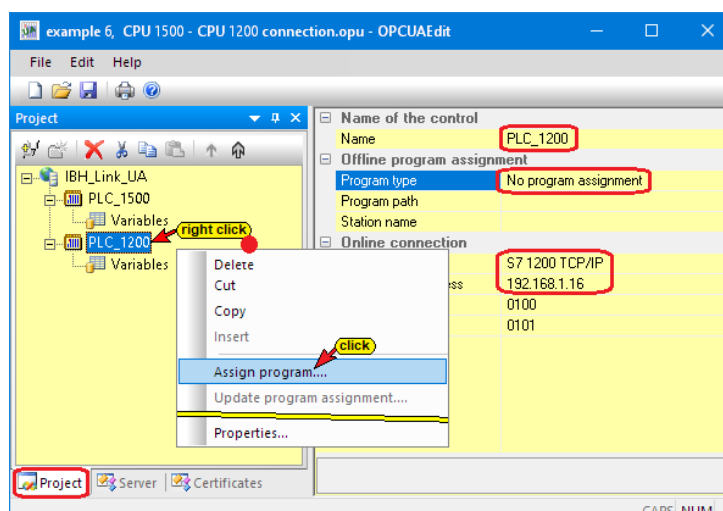
## Transferred 1500 PLC program

In the right part of the project window information about the **Offline program assignment** are displayed.



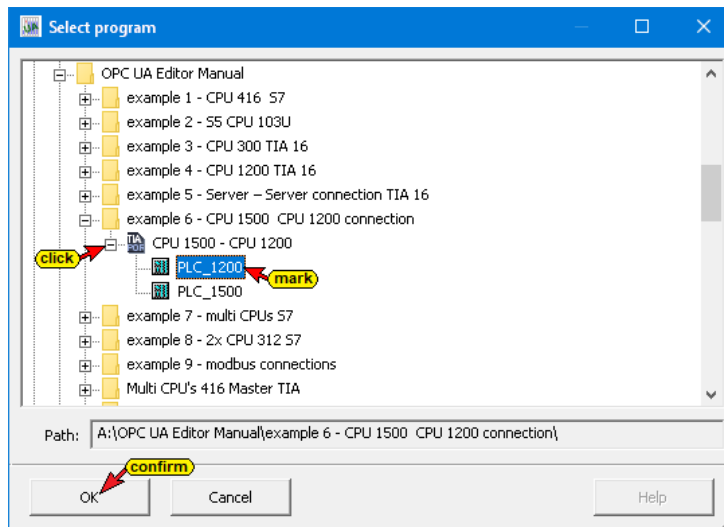
## Program assignment PLC 1200

The Assign program command from the context menu opens the Program selection dialog box.



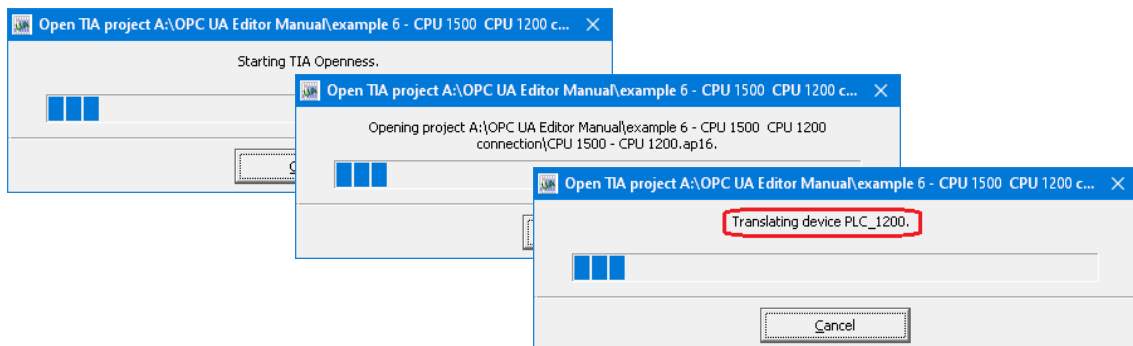


Select the PLC program in the **Select program** dialog box. Clicking the **Plus** symbol in front of the TIA symbol of the PLC project, the PLC program (CPUs) is displayed in the project.



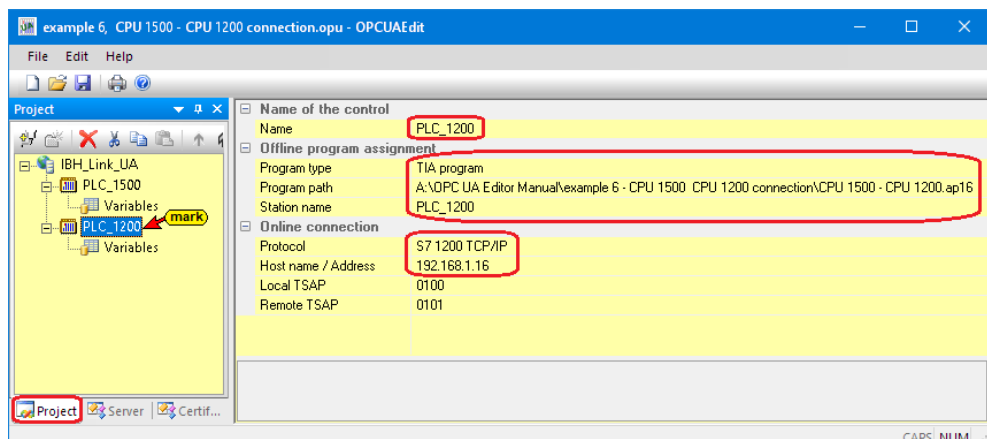
The **SIEMENS support software TIA Openness** is started in the background. If an error occurs see chapter 1, **Special features when selecting TIA projects** page 1-16.

Several notices are displayed.



## Transferred 1200 PLC program

In the right part of the project window information about the **Offline program assignment** are displayed.



## 2.6.6 Define variables as OPC tags

Clicking **Variables** lists the variables / data (data blocks) from the PLC in the right part of the project window.

Clicking the **Plus** icon in front of the variable area symbol displays the existing variables.

If a variable is selected, it is adopted as an OPC tag and displayed in the lower part of the window with additional information.

### Define PLC\_1500 variables as OPC tags

Clicking **Variables** lists the variables / data (data blocks) from the PLC in the right part of the project window.

Not all variables were defined as OPC tags.

Name	Address	PLC type	Leng...	Origin	Access	OPC type	Comment
CounterData.MinNo1500	DB5.DBW 0	Int	2	Program	RW	Int16	minimum counter reading (number)
CounterData.MaxNo1500	DB5.DBW 2	Int	2	Program	RW	Int16	maximum counter reading (number)
CounterData.Count1500	DB5.DBX 4.0	Bool	.1	Program	RW	Boolean	Counter is counting
CounterData.CountingON1500	DB5.DBX 4.1	Bool	.1	Program	RW	Boolean	Enable counting
CounterData.CounterValue1500	DB5.DBW 6	Int	2	Program	RW	Int16	CounterValue
ReceivedValues.ON1200	DB10.DBX 0.0	Bool	.1	Program	RW	Boolean	Counter is counting 1200
ReceivedValues.MinValue1200	DB10.DBW 2	Int	2	Program	RW	Int16	Minimum count 1200
ReceivedValues.MaxValue1200	DB10.DBW 4	Int	2	Program	RW	Int16	Maximum count 1200
ReceivedValues.CounterValue1200	DB10.DBW 6	Int	2	Program	RW	Int16	Counter value 1200
ReceivedValues.CounterStatus1200	DB10.DBD 8	DWord	4	Program	RW	UInt32	Counter status 1200
ReceivedValues.CounterTimeStamp1200	DB10.DBX 12.0	Date_And_Time	8	Program	RW	DateTime	Counter Time Stamp 1200

### Define PLC\_1200 variables as OPC tags

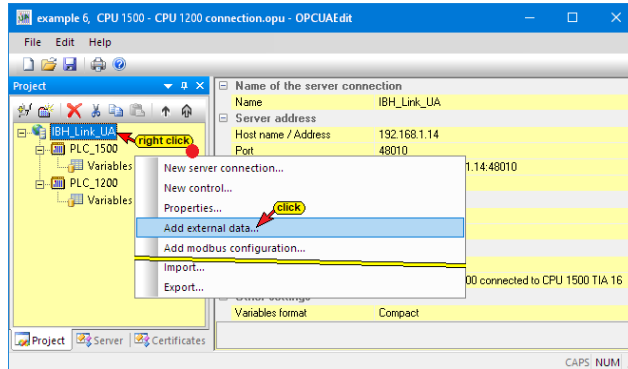
Not all variables were defined as OPC tags.

Name	Address	PLC type	Leng...	Origin	Access	OPC type	Comment
CounterData.MinNo1200	DB5.DBW 0	Int	2	Program	RW	Int16	minimum counter reading (number)
CounterData.MaxNo1200	DB5.DBW 2	Int	2	Program	RW	Int16	maximum counter reading (number)
CounterData.Count1200	DB5.DBX 4.0	Bool	.1	Program	RW	Boolean	Counter is counting
CounterData.CountingON1200	DB5.DBX 4.1	Bool	.1	Program	RW	Boolean	Enable counting
CounterData.CounterValue1200	DB5.DBW 6	Int	2	Program	RW	Int16	CounterValue
ReceivedValues.ON_1500	DB10.DBX 0.0	Bool	.1	Program	RW	Boolean	Counter is counting 1500
ReceivedValues.MinValue1500	DB10.DBW 2	Int	2	Program	RW	Int16	Minimum count 1500
ReceivedValues.MaxValue1500	DB10.DBW 4	Int	2	Program	RW	Int16	Maximum count 1500
ReceivedValues.CounterValue1500	DB10.DBW 6	Int	2	Program	RW	Int16	Counter value1500
ReceivedValues.CounterStatus1500	DB10.DBD 8	DWord	4	Program	RW	UInt32	Counter status1500
ReceivedValues.CounterTimeStamp	DB10.DBX 12.0	String	256	Program	RW	String	Counter Time Stamp

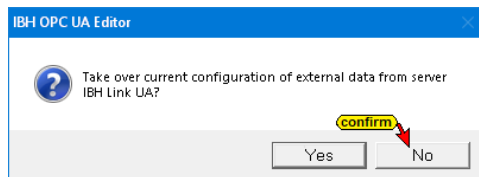
## 2.6.7 Add external data

With the next steps the variables of the PLC\_1200 and PLC\_1500 defined as OPC tags are connected...

Right-click on IBH Link UA in the IBH OPC UA Editor and execute the **Add external data ...** command.



Clicking on **Add External Data...** opens the IBH OPC UA Editor dialog box.



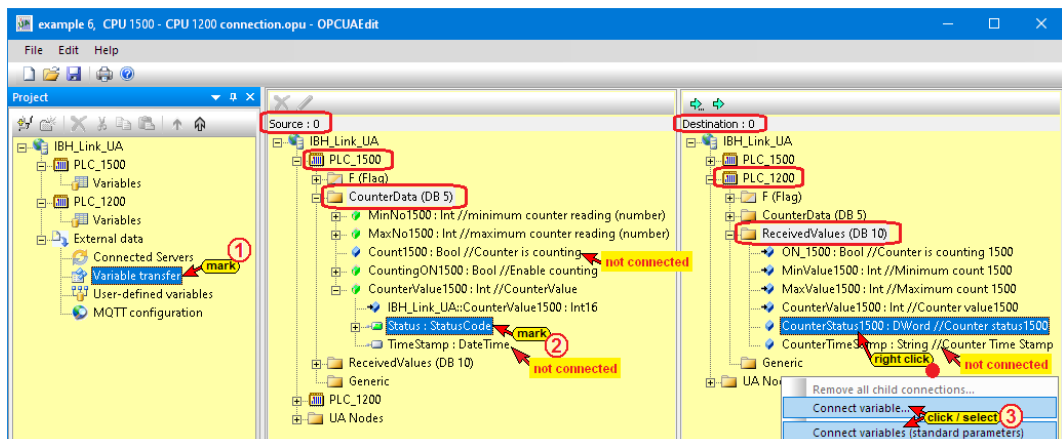
Only the OPC tags defined in the IBH OPC UA Editor (PLC\_1500 and PLV\_1200) are to be used, the opened dialog box must be confirmed with **No**.



## 2.6.8 Variable transfer - define source and target variables

The OPC UA variable connection is adopted by marking the source and target variables and then clicking the Connect variable command (standard parameters).

**Variable links: source – PLC\_1500 ⇒ target – PLC\_1200**

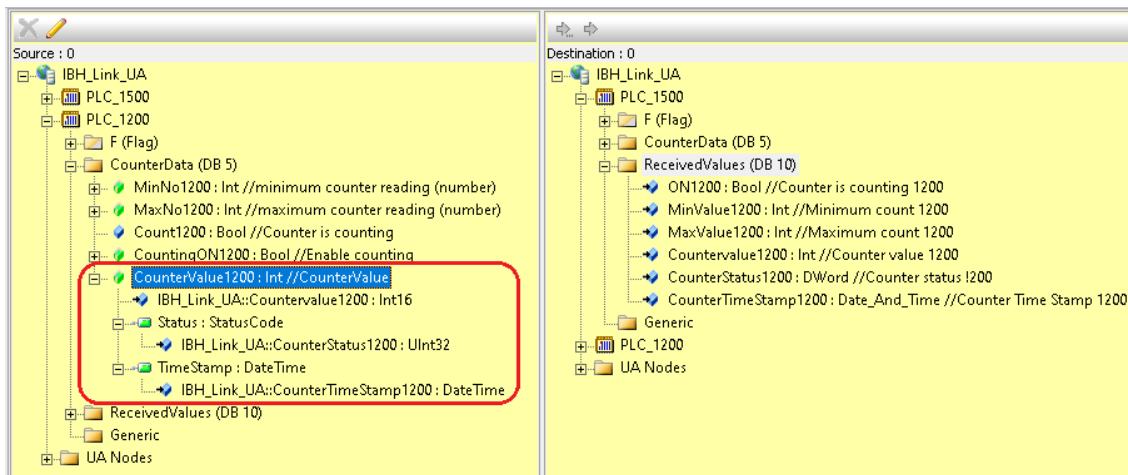


The **Connect variable... (standard parameters)** command takes over the connection of the marked variables directly.

The command **Connect variable...** opens the dialog box **Parameter for reading the variables**. The name, type and ID of the variable are displayed here. Sampling interval, queue entries and index area can be influenced.

OPC tags from the **PLC\_1500** data block **CounterData [DB5]** are transferred to the **PLC\_1200** data block **ReceivedValues [DB10]**.

If a variable (value) has been connected, the status and the time stamp belonging to the source variable are offered to be connected.

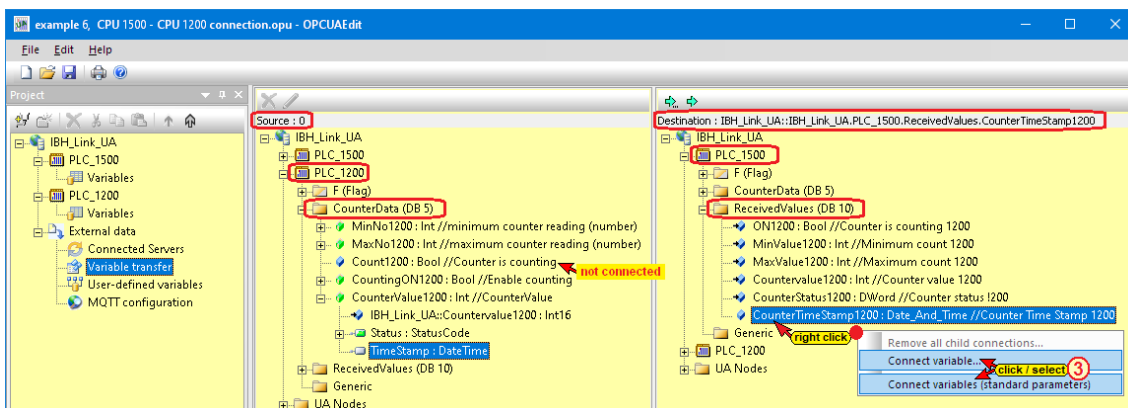


The source OPC tags **CounterValue1500/TimeStamp** and **Count1500** are not connected. The connections are displayed in the lower part of the right project window.

Source server	Source variable	Destination server	Destination variable	Data type	Source name	Destination name
IBH_Link_UA	CountingON1500	IBH_Link_UA	ON_1500	Boolean	IBH_Link_UA.PLC_1500.CounterData.CountingON1500	IBH_Link_UA.PLC_1200.ReceivedValues.ON_1500
IBH_Link_UA	MinNo1500	IBH_Link_UA	MinValue1500	Int16	IBH_Link_UA.PLC_1500.CounterData.MinNo1500	IBH_Link_UA.PLC_1200.ReceivedValues.MinValue1500
IBH_Link_UA	MaxNo1500	IBH_Link_UA	MaxValue1500	Int16	IBH_Link_UA.PLC_1500.CounterData.MaxNo1500	IBH_Link_UA.PLC_1200.ReceivedValues.MaxValue1500
IBH_Link_UA	CounterValue1500	IBH_Link_UA	CounterValue1500	Int16	IBH_Link_UA.PLC_1500.CounterData.CounterValue1500	IBH_Link_UA.PLC_1200.ReceivedValues.CounterValue1500
IBH_Link_UA	CounterValue1500.Status	IBH_Link_UA	CounterStatus1500	StatusC...	IBH_Link_UA.PLC_1500.CounterData.CounterValue1500.Status	IBH_Link_UA.PLC_1200.ReceivedValues.CounterStatus1500

### Variable links: source – PLC\_1200 ➔ target – PLC\_1500

OPC tags from the **PLC\_1200** data block **CounterData [DB5]** are transferred to the **PLC\_1500** data block **ReceivedValues [DB10]**.

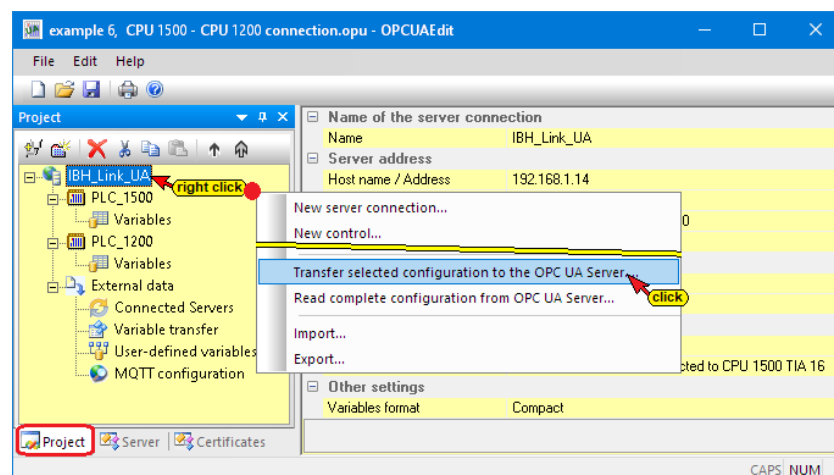


The source OPC tag **Count1500** is not connected. The connections are displayed in the lower part of the right project window.

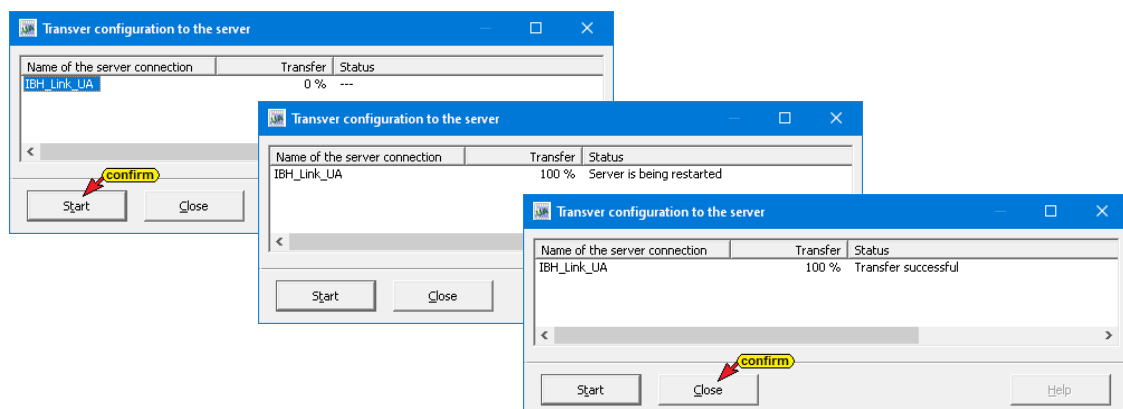
Source variable	Destination server	Destination variable	Data type	Source name	Destination name
CountingON1500	IBH_Link_UA	ON_1500	Boolean	IBH_Link_UA.PLC_1500.CounterData.CountingON1500	IBH_Link_UA.PLC_1200.ReceivedValues.ON_1500
MinNo1500	IBH_Link_UA	MinValue1500	Int16	IBH_Link_UA.PLC_1500.CounterData.MinNo1500	IBH_Link_UA.PLC_1200.ReceivedValues.MinValue1500
MaxNo1500	IBH_Link_UA	MaxValue1500	Int16	IBH_Link_UA.PLC_1500.CounterData.MaxNo1500	IBH_Link_UA.PLC_1200.ReceivedValues.MaxValue1500
CounterValue1500	IBH_Link_UA	CounterStatus1500	Int16	IBH_Link_UA.PLC_1500.CounterData.CounterValue1500	IBH_Link_UA.PLC_1200.ReceivedValues.CounterValue1500
CounterValue1500.Status	IBH_Link_UA	CounterStatus1500	StatusC...	IBH_Link_UA.PLC_1500.CounterData.CounterValue1500.Status	IBH_Link_UA.PLC_1200.ReceivedValues.CounterStatus1500
CountingON1200	IBH_Link_UA	ON1200	Boolean	IBH_Link_UA.PLC_1200.CounterData.CountingON1200	IBH_Link_UA.PLC_1500.ReceivedValues.ON1200
MinNo1200	IBH_Link_UA	MinValue1200	Int16	IBH_Link_UA.PLC_1200.CounterData.MinNo1200	IBH_Link_UA.PLC_1500.ReceivedValues.MinValue1200
MaxNo1200	IBH_Link_UA	MaxValue1200	Int16	IBH_Link_UA.PLC_1200.CounterData.MaxNo1200	IBH_Link_UA.PLC_1500.ReceivedValues.MaxValue1200
CounterValue1200	IBH_Link_UA	CounterValue1200	Int16	IBH_Link_UA.PLC_1200.CounterData.CounterValue1200	IBH_Link_UA.PLC_1500.ReceivedValues.CounterValue1200
CounterValue1200.Status	IBH_Link_UA	CounterStatus1200	StatusC...	IBH_Link_UA.PLC_1200.CounterData.CounterValue1200.Status	IBH_Link_UA.PLC_1500.ReceivedValues.CounterStatus1200
CounterValue1200.TimeStamp	IBH_Link_UA	CounterTimeStamp1200	DateTime	IBH_Link_UA.PLC_1200.CounterData.CounterValue1200.TimeStamp	IBH_Link_UA.PLC_1500.ReceivedValues.CounterTimeStamp1200

## 2.6.9 Transfer configuration to the OPC UA server (IBH Link UA).

A right-click on the Server icon (IBH Link UA) opens the context menu.



The command **Transfer Selected Configuration to OPC UA Server** command opens the **Transfer Configuration to Server** dialog box.



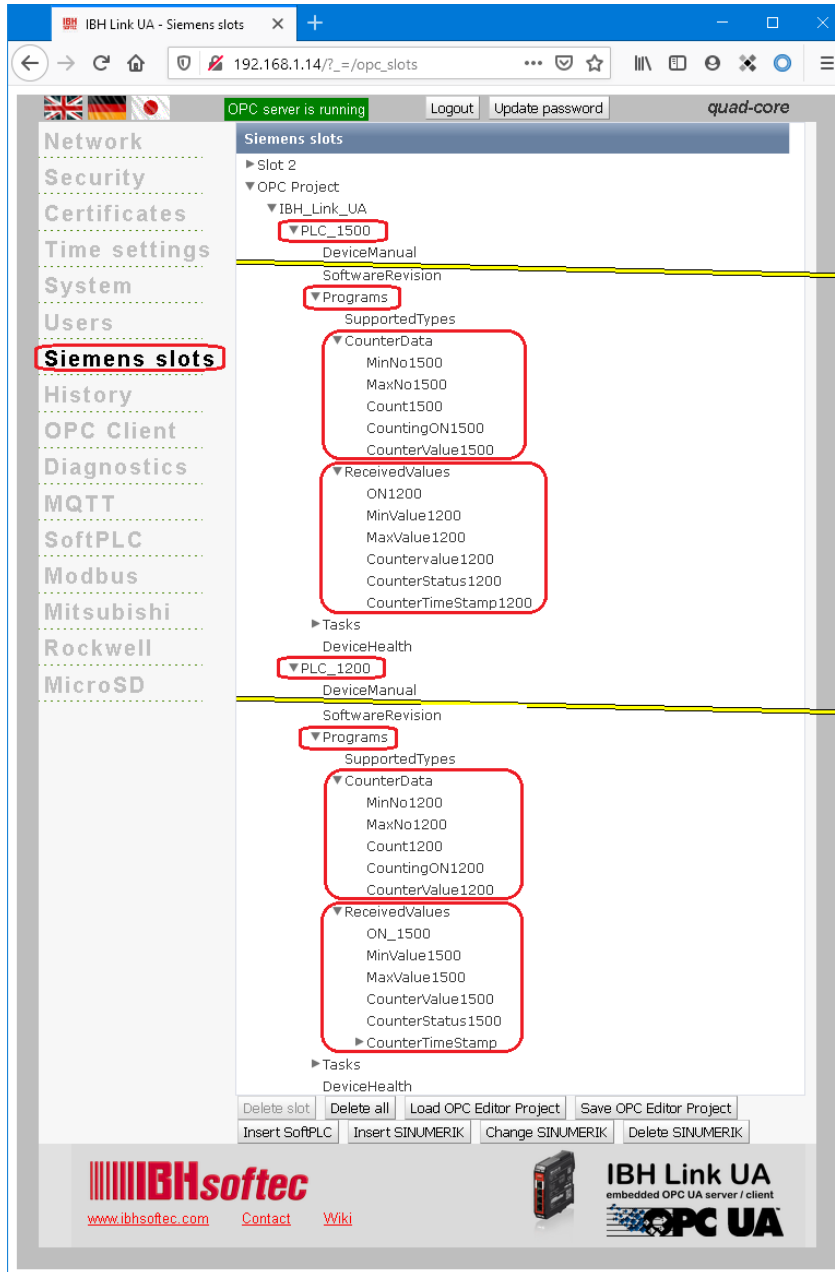
Select the server **IBH Link UA** and then click Start. The configurations (**PLC\_1500 / PLC\_1200**) are transferred to the **IBH Link UA**. Successful transfer is displayed.

If a certified data exchange between the IBH OPC Editor and the IBH Link UA has been selected, the exchanged certificates must be trusted (see chapter Trust certificate, Chapter 1, page 1-40).

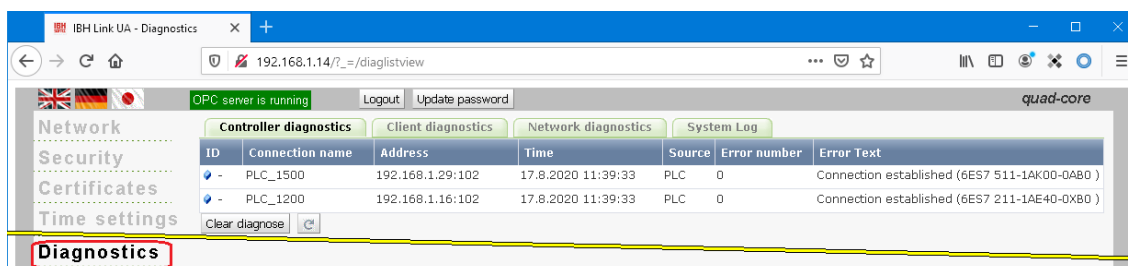
### 2.6.10 IBH Link UA web browser window

The connections are displayed in the IBH Link UA web browser window OPC Client.

#### IBH Link UA - Siemens Slots – PLC\_1500 / PLC\_1200

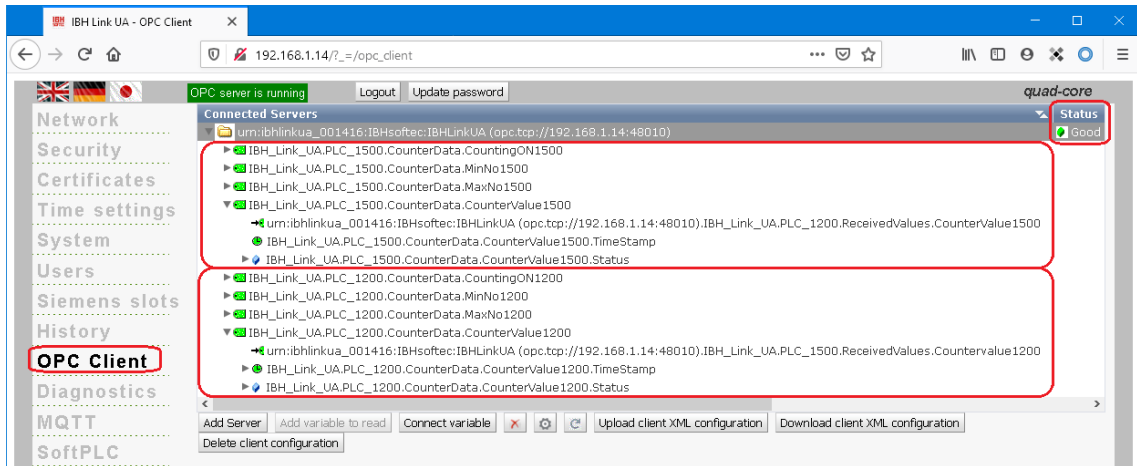


The browser window **Diagnosics** displays the status of the connections **IBH Link UA – PLC\_1500 and PLC\_1200**.



## IBH Link UA - browser window OPC Client

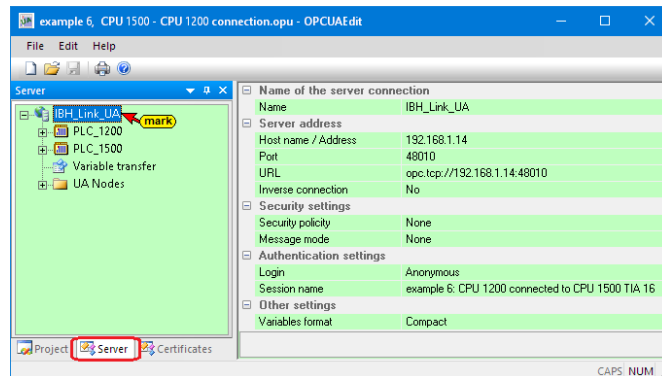
Connections specified the IBH UA Editor are displayed in the web browser window OPC Client.



### 2.6.11 Online OPC UA Server Information

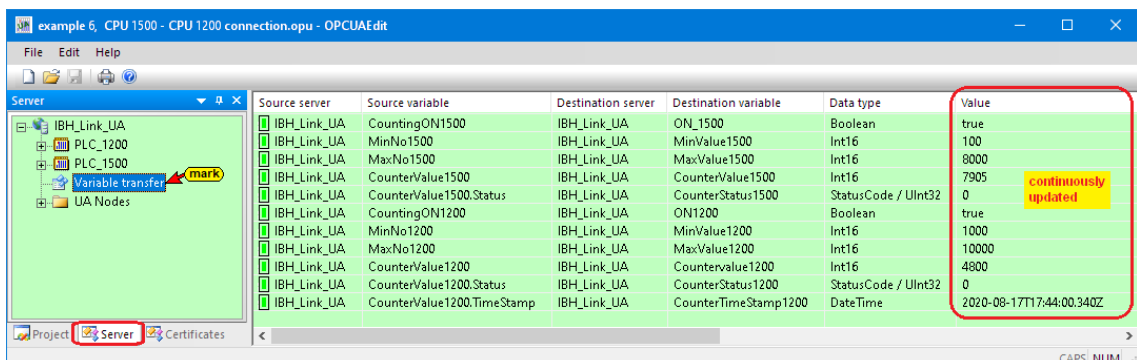
Information from the online connected OPC UA server with the online connected CPUs are displayed.

### IBH Link UA – PLC\_1500 – PLC\_1200

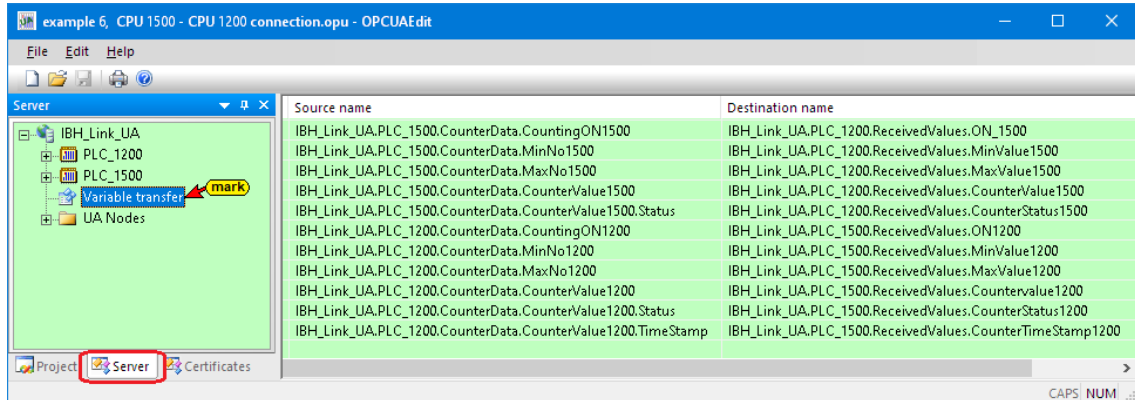


### Variable transfer – PLC\_1500 – PLC\_1200

Click Variable transfer. The variables (OPC tags) are displayed in the right server window with their status. The status of the OPC tags is updated continuously.

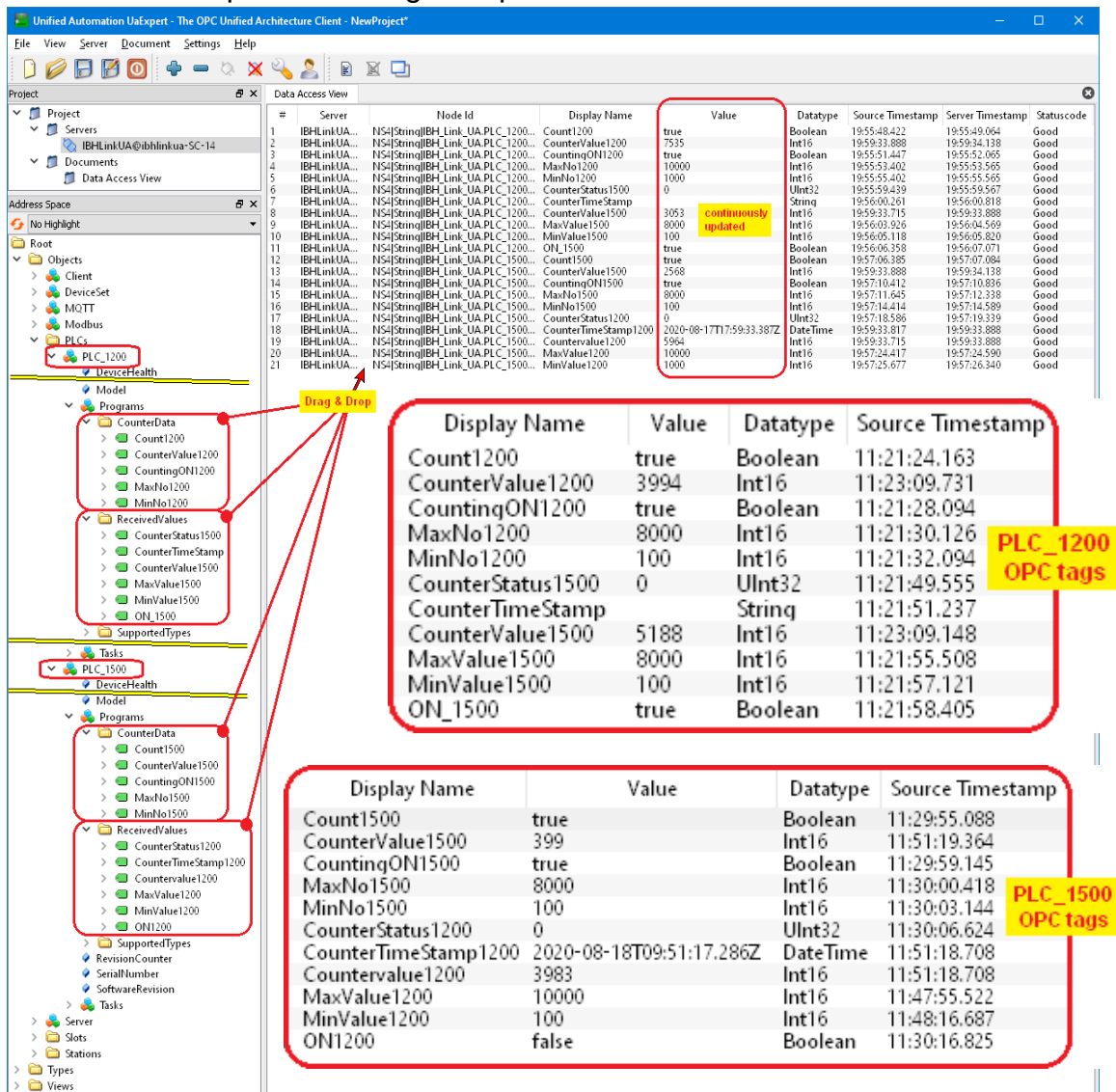


The source names and the target names of the variable transfer are also displayed.



### 2.6.12 UaExpert - Data Access View

With the IBH Link UA OPC and the PLCs PLC\_1200 / PLC\_1500 connected, information about the OPC tags are displayed. With drag & drop the OPC tags are pulled into the Data Access Viewer window.





## 2.6.13 PLC-Status

### PLC\_1200 data block *CounterData* [DB5]

CPU 1500 - CPU 1200 ▶ PLC\_1200 [CPU 1211C DG/DC/DC] ▶ Program blocks ▶ CounterData [DB5]

Keep actual values Snapshot Copy snapshots to start values

CounterData						
	Name	Data type	Offset	Start value	Monitor value	Comment
1	Static					
2	MinNo1200	Int	0.0	1000	100	minimum counter reading (number)
3	MaxNo1200	Int	2.0	10000	8000	maximum counter reading (number)
4	Count1200	Bool	4.0	false	TRUE	Counter is counting
5	CountingON1200	Bool	4.1	false	TRUE	Enable counting
6	CounterValue1200	Int	6.0	0	1396	CounterValue

### PLC\_1200 data block *ReceivedValues* [DB10]

CPU 1500 - CPU 1200 ▶ PLC\_1200 [CPU 1211C DG/DC/DC] ▶ Program blocks ▶ ReceivedValues [DB10]

Keep actual values Snapshot Copy snapshots to start values

ReceivedValues						
	Name	Data type	Offset	Start value	Monitor value	Comment
1	Static					
2	ON_1500	Bool	0.0	false	TRUE	Counter is counting 1500
3	MinValue1500	Int	2.0	0	100	Minimum count 1500
4	MaxValue1500	Int	4.0	0	8000	Maximum count 1500
5	CounterValue1500	Int	6.0	0	3594	Counter value1500
6	CounterStatus1500	DWord	8.0	16#0	16#0000_0000	Counter status1500
7	CounterTimeStamp	String	12.0	"	"	Counter Time Stamp

### PLC\_1500 data block *CounterData* [DB5]

CPU 1500 - CPU 1200 ▶ PLC\_1500 [CPU 1511-1 PN] ▶ Program blocks ▶ CounterData [DB5]

Keep actual values Snapshot Copy snapshots to start values

CounterData						
	Name	Data type	Offset	Start value	Monitor value	Comment
1	Static					
2	MinNo1500	Int	0.0	1000	100	minimum counter reading (number)
3	MaxNo1500	Int	2.0	10000	8000	maximum counter reading (number)
4	Count1500	Bool	4.0	false	TRUE	Counter is counting
5	CountingON1500	Bool	4.1	false	TRUE	Enable counting
6	CounterValue1500	Int	6.0	0	7435	CounterValue

### PLC\_1500 data block *ReceivedValues* [DB10]

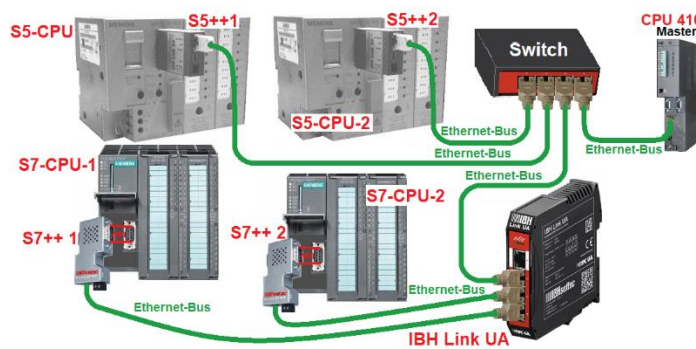
CPU 1500 - CPU 1200 ▶ PLC\_1500 [CPU 1511-1 PN] ▶ Program blocks ▶ ReceivedValues [DB10]

Keep actual values Snapshot Copy snapshots to start values

ReceivedValues						
	Name	Data type	Offset	Start value	Monitor value	Comment
1	Static					
2	ON1200	Bool	0.0	false	FALSE	Counter is counting 1200
3	MinValue1200	Int	2.0	0	100	Minimum count 1200
4	MaxValue1200	Int	4.0	0	10000	Maximum count 1200
5	CounterValue1200	Int	6.0	0	4888	Counter value 1200
6	CounterStatus1200	DWord	8.0	16#0	16#0000_0000	Counter status 1200
7	CounterTimeStamp1200	Date_And_Time	12.0	DT#1990-01-01-00:00:00	DT#2020-08-18-09:59:16.331	Counter Time Stamp 1200

## 2.7 Example 7 – Exchanging data between several S7 / S5 CPUs

On a system with three (3) S7 CPUs and two (2) S5 CPUs, data should be exchanged with one another. An S7 CPU with an Ethernet connection acts as a master and is connected to the IBH Link UA via a switch. The other CPUs have no Ethernet interfaces and are connected to the IBH Link UA via IBH Link S7 ++ or IBH Link S5 ++, via the switch. The master makes data available for all CPUs, these give information to the master.



### 2.7.1 CPU 312 connection via IBH Link S7++

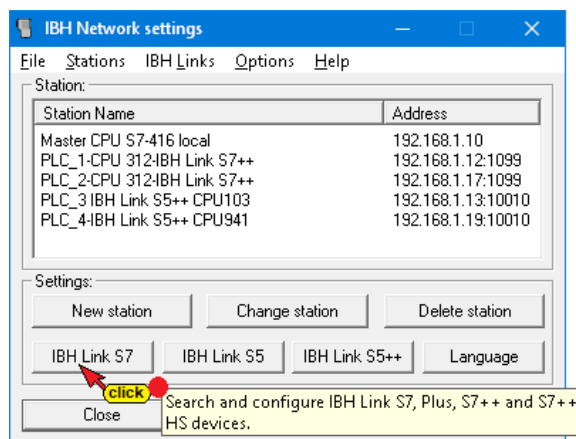
The **IBH Link S7++** is an Ethernet converter. The standard TCP / IP protocol is used. The user benefits from all the advantages of Ethernet without any problems.

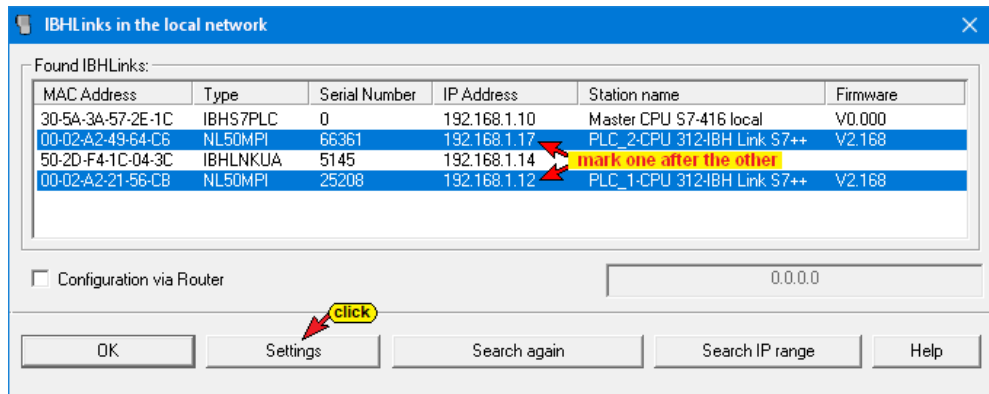


The use of **Simatic Net** and the use of a CP communication processor is not necessary on either the PC or the PLC side.

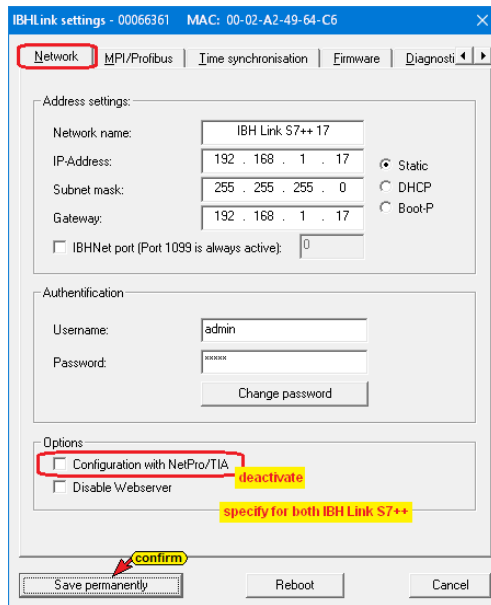
### Configuration of the IBH Link S7 ++

Open the IBH Network settings dialog box to deactivate the option Configuration with **NetPro/TIA**.



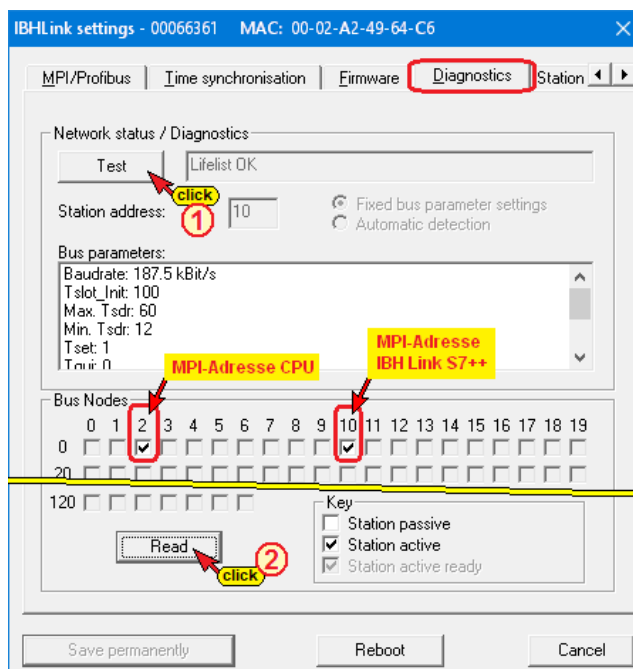


### Deactivate the option Configuration with NetPro / TIA



### IBH Link S7++ MPI addresses

Diagnostics are available to show the MPI bus parameters and the MPI addresses in use.



## 2.7.2 PLC programs

The S5 / S7 CPUs programs count value until **MaxValue** is reached. Then the **Value** is counted down until **MinValue** is reached. This up and down counting is repeated continuously.

The master PLC OPC tag **Control\_ON** enables the counting. The master PLC specifies the numerical values **MaxValue** and **MinValue** for each CPU. Each CPU reports back the current counter **Value** and counting active (**Controlling\_is\_ON**) to the master PLC.

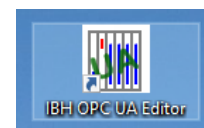
The programs are in the **STEP® 7 - SIMATIC Manager** format for the **S7 CPUs** and in **S5 for Windows® (STEP® 5)** for the **S5 CPUs**.

### IP addresses / PLC programs of the devices used

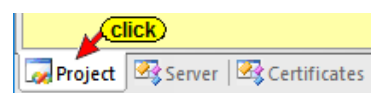
Device	IP / MPI address
IBH Link UA	192.168.1.14
CPU 416 Master PLC project / program: Multi CPU's 416 Master / CPU 416 Master	192.168.1.10
S7 PLC 1 – [CPU 312] PLC project / program: Multi CPU's S7 PLC_1 / S7 CPU 1	IBH Link S7++ on the S7 PLC 1 192.168.1.12 MPI- address IBH Link S7++: 10 MPI- address CPU: 2
S7 PLC 2 – [CPU 312] PLC project / program: Multi CPU's S7 PLC_2 / S7 CPU 2	IBH Link S7++ on the S7 PLC 2 192.168.1.17 MPI- address IBH Link S7++: 10 MPI-address CPU: 2
S5 PLC 3 – [S5 CPU103] PLC program: Multi CPUs S5 PLC_3.S5P	IBH Link S5++ on the S5 PLC 3 192.168.1.13
S5 PLC 4 – [S5 CPU941] PLC program: Multi CPUs S5 PLC_4.S5P	IBH Link S5++ on the S5 PLC 4 192.168.1.19

## 2.7.3 Calling the IBH OPC UA Editor

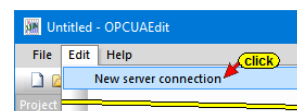
Double-click the **IBH OPC UA Editor** icon to open the program window.



Open the **Project window** by clicking on the **Project** tab.



Open the **Server Connection** dialog box with the New Server Connection command from the **Edit** menu or by clicking the icon.



The new server connection setup was explained in example 1 (see chapter 2, page 2-3).

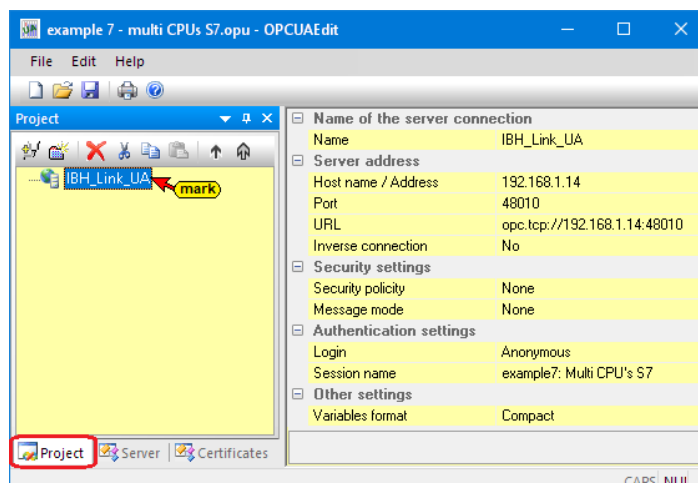
## Server Connection dialog box

The screenshot shows the 'Server connection properties' dialog box with the following settings:

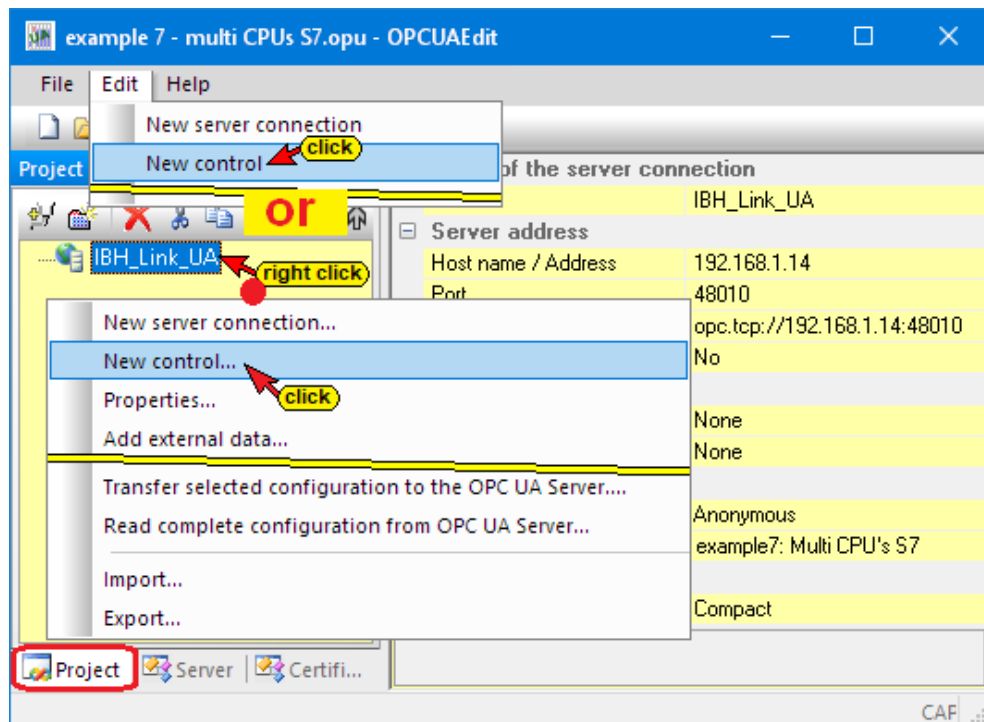
- Name of the server connection: IBH Link UA
- Server address:
  - Host name or IP address: 192.168.1.14
  - Port: 48010
  - URL: opc.tcp://192.168.1.14:48010
- Security settings:
  - None (selected)
  - Basic128Rsa15
  - Basic256
  - BasicSha256
  - Aes128Sha256RsaOaep
  - Aes256Sha256RsaPss
- Message mode:
  - Signatur
  - Signature and Encryption
- Inverse connection:
  - Connect invers (unchecked)
- Login:
  - Anonymous (selected)
  - User name and password
- Session Name: example7: Multi CPU's S7
- Variables format: Compact

All necessary settings are shown. The possible encryptions of the data to be transmitted are displayed. The example is using the security method **None**. By clicking **OK** the content of the dialog box is saved and closed.

The settings for the connection to the **IBH Link UA** OPC UA server are displayed in the right part of the **project window**.

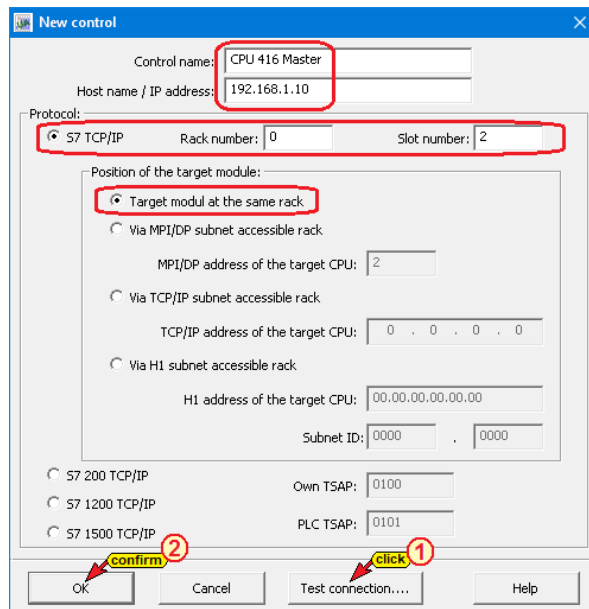


## 2.7.4 Inserting New controls (PLCs)



The **New control** command from the context menu (or menu Edit / New control) opens the dialog box **New control** to specify the access to the control (CPU).

### Dialog box New controller - CPU 416 master

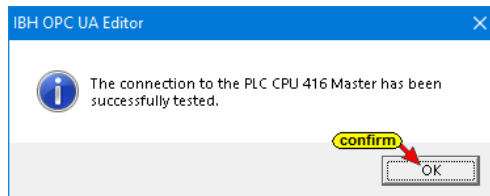


#### Test connection

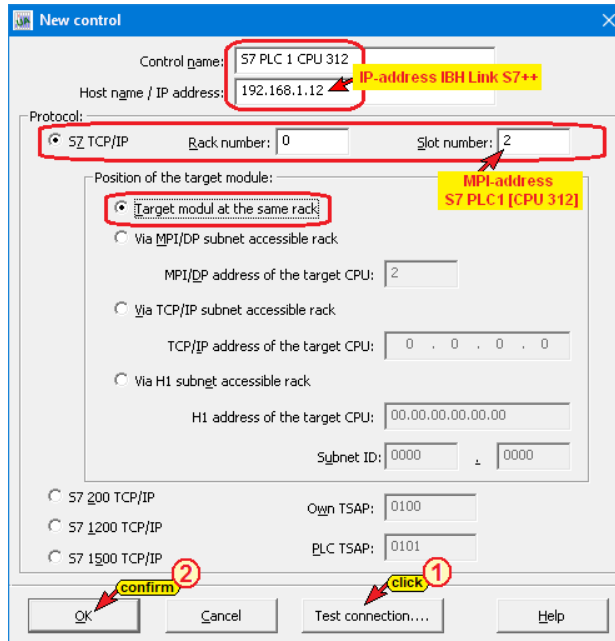
After completing the New Control dialog box, the connection to the online connected CPU can be tested.

Information about the successful connection is displayed.

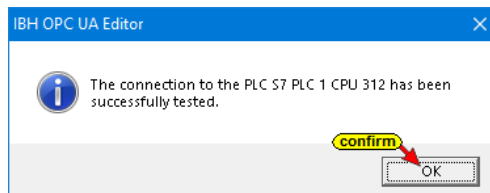
Test connection....



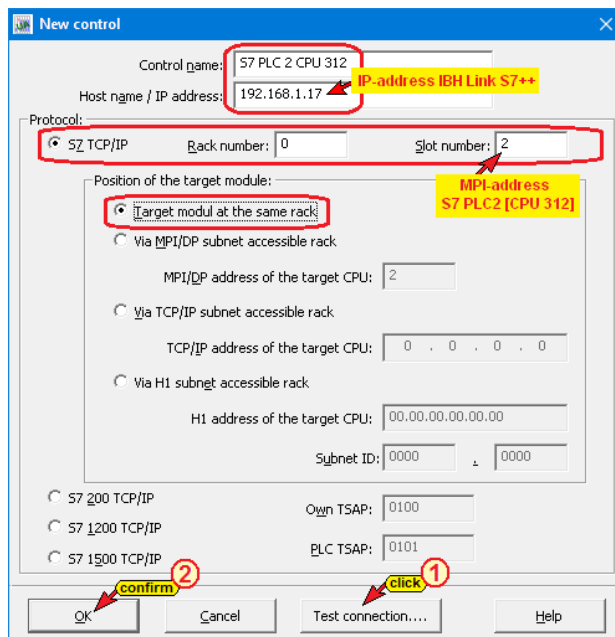
## Dialog box New controller - S7 PLC 1 CPU 312



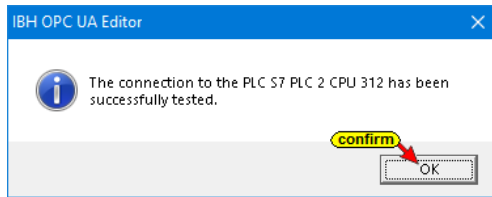
The connection to the online connected **S7 PLC 1** can be tested.



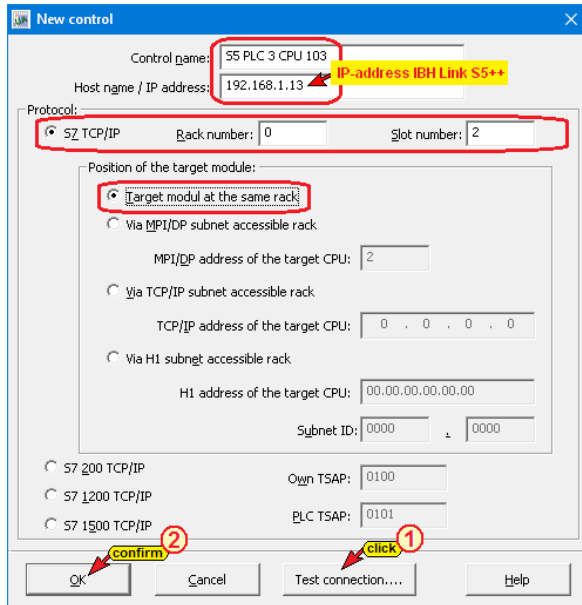
## Dialog box New controller - S7 PLC 2 CPU 312



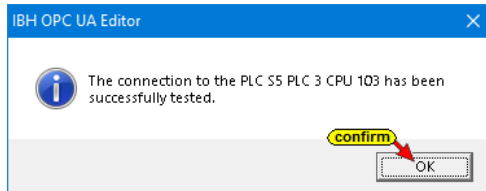
The connection to the online connected **S7 PLC 2** can be tested.



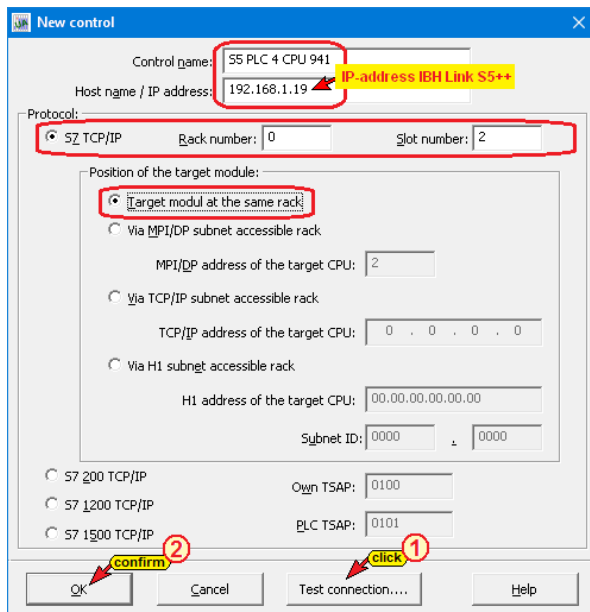
### Dialog box New controller – S5 PLC 3 CPU 103



The connection to the online connected **S5 PLC 3** can be tested.

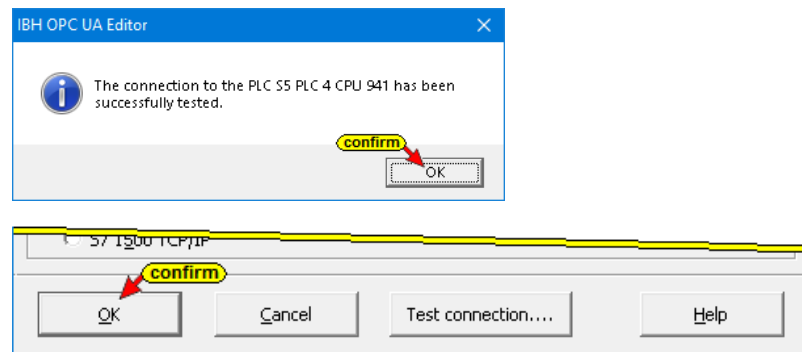


### Dialog box New controller – S5 PLC 3 CPU 103



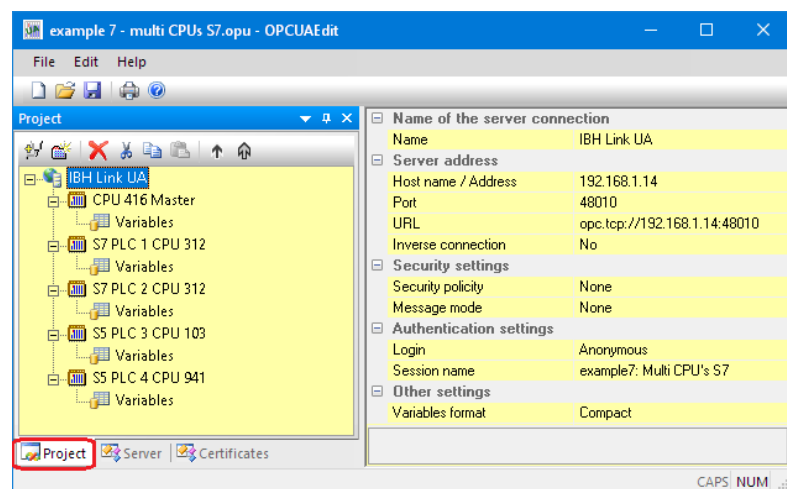


The connection to the online connected **S5 PLC 4** can be tested.



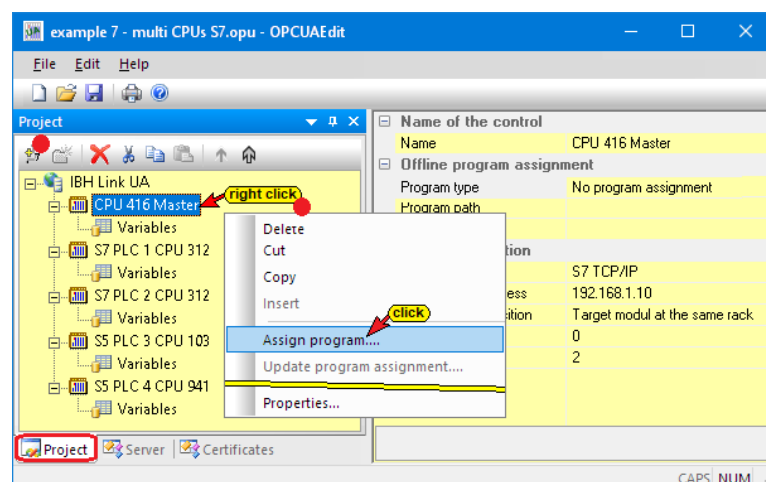
To accept and close the **New control** dialog box settings click on **OK**.

The inserted PLC controls are displayed in the left project window.



## 2.7.5 Program assignment

The **Assign program** command is used to open the **Program Selection** dialog box.

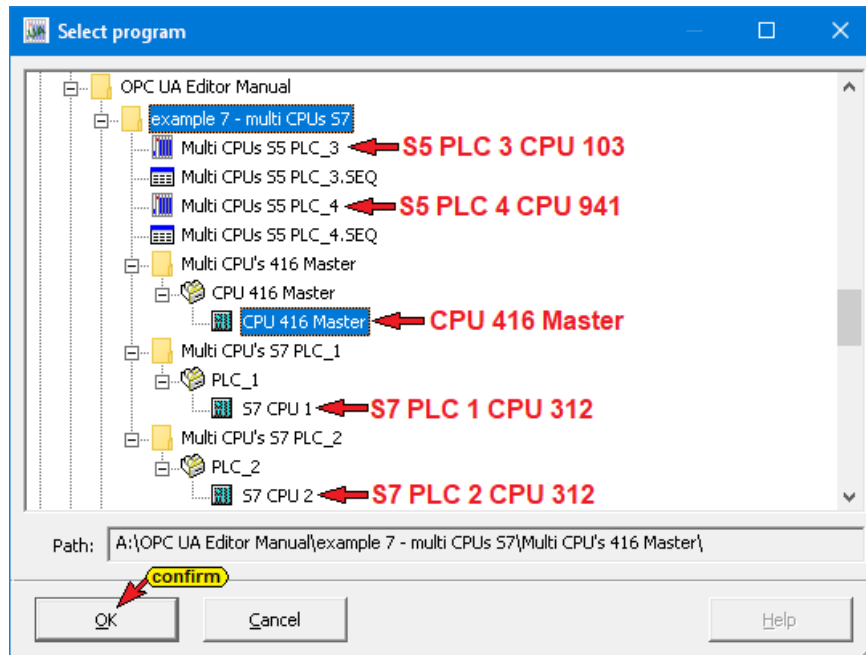


### Select Program dialog box

Select the PLC program in the **Select program** dialog box. Clicking the **Plus** symbol in front of the PLC project

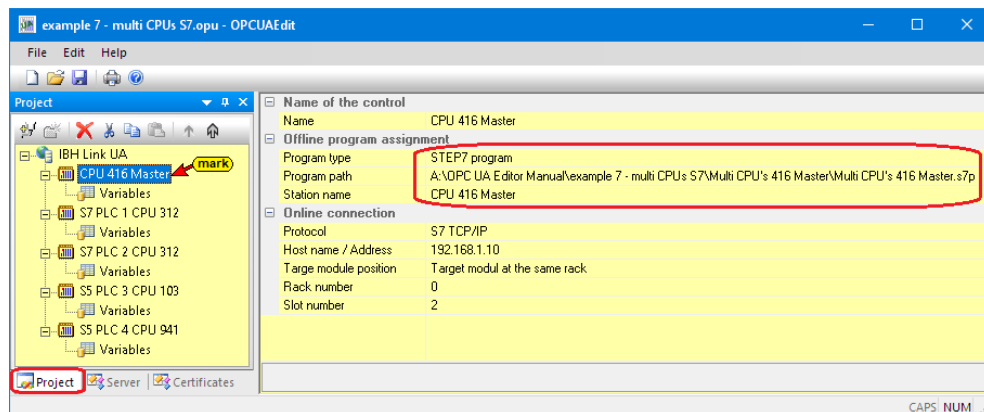


name. Mark the PLC program (CPU) and click OK to assign the PLC program.



## Assigned PLC program

In the right part of the project window information about the **program assignment** are displayed.



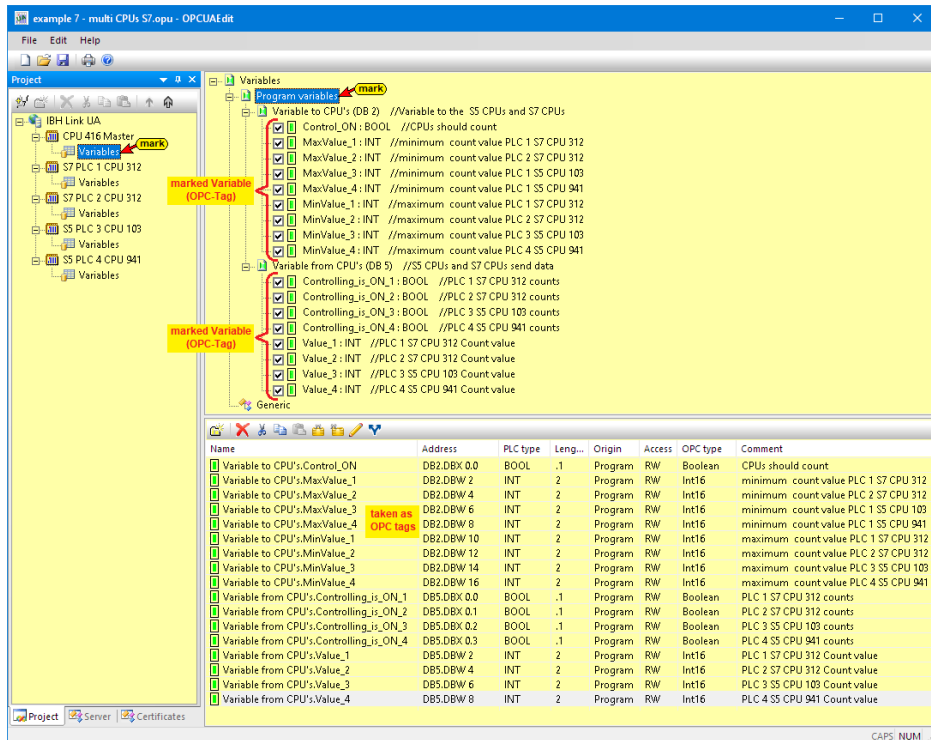
The corresponding PLC program must be assigned to each of the five (5) PLC controls.

### 2.7.6 Define variables as OPC tags

Clicking **Variables** lists the variables / data (data blocks) from the PLC in the right part of the project window. The selected **OPC tags** are listed in the lower part of the right window.

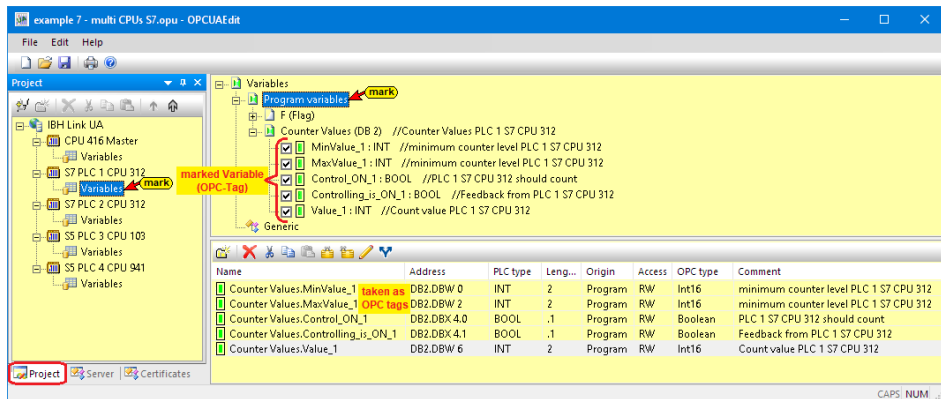
#### CPU 416 master – OPC tags

All variables of the data blocks **Variable to CPU's [DB2]** and **Variable from CPU's [DB5]** are defined as OPC tags.



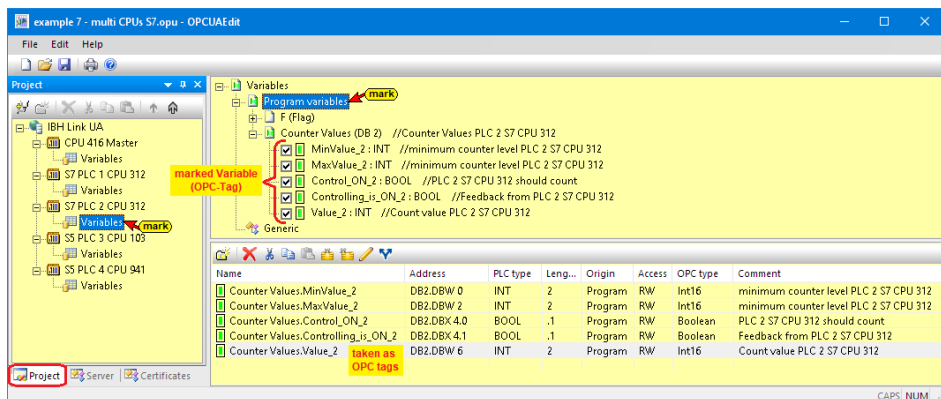
### S7 PLC 1 CPU 312 – OPC tags

Only the five (5) variables of the data block **Counter Values [DB2]** are defined as OPC tags.



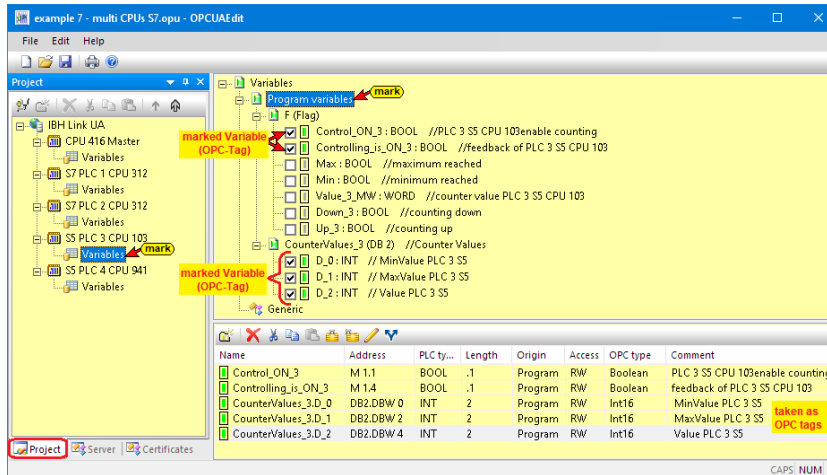
### S7 PLC 2 CPU 312 – OPC tags

Only the five (5) variables of the data block **Counter Values [DB2]** are defined as OPC tags.



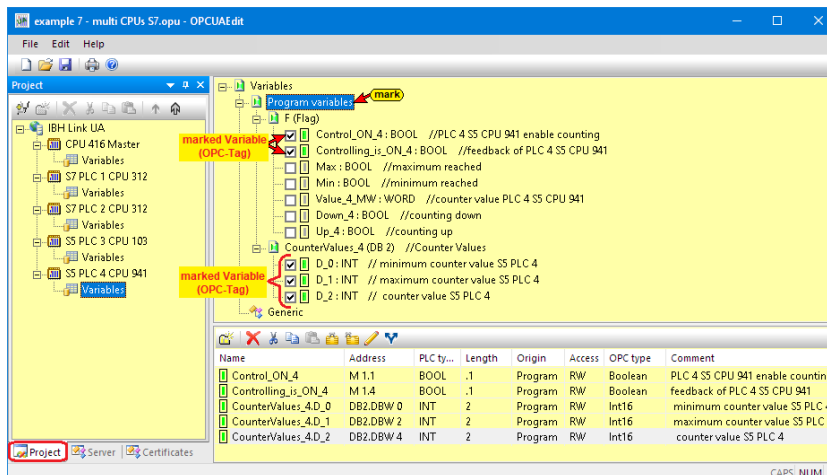
## S5 PLC 3 CPU 103 – OPC tags

Two (2) flag tags and all (3) data of the data block **Counter Values\_3 [DB2]** are defined as OPC tags.



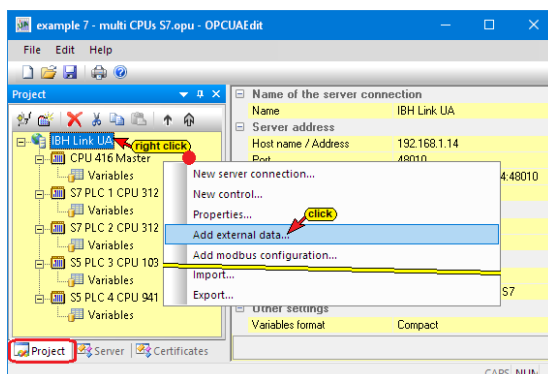
## S5 PLC 4 CPU 941 – OPC tags

Two (2) flag tags and all (3) data of the data block **Counter Values\_3 [DB2]** are defined as OPC tags.



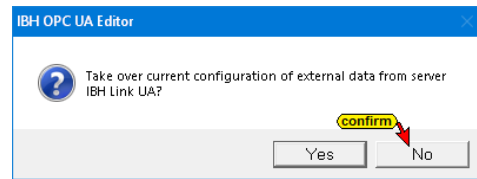
### 2.7.7 Add external data

Right-click on **IBH Link UA** and execute the **Add External Data ...** command.



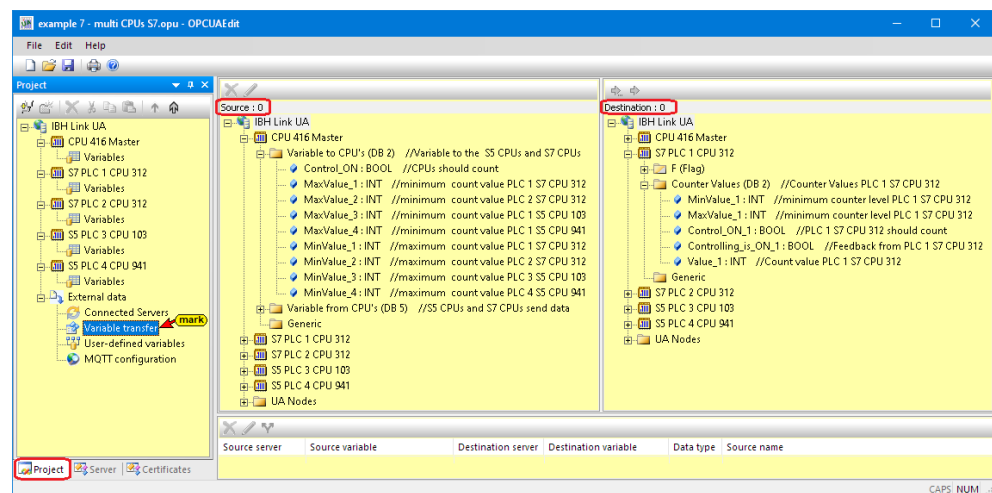
The OPC tags defined in the IBH OPC UA Editor are to be used. Confirm the dialog box with **No**.

No



## 2.7.8 Variable transfer – define source and destination OPC tags

**External data** with additional commands was inserted in the left part of the project window.



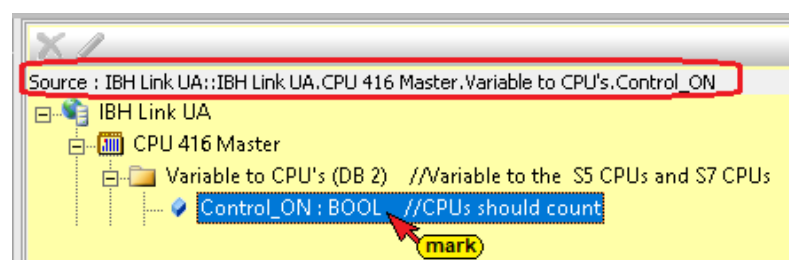
Mark **Variable transfer** to define the source and destination OPC tags. The right project window is divided into two parts. The **Source** window is on the left and the **Destination** window on the right. The Source window and the Destination window lists the OPC tags of the OPC server and the PLC Controllers.

The **OPC tags** to be read (read variables) are specified in the **Source** window. The **OPC tag** to be linked with the variable read is specified in the **Destination** window.

### Variable connection

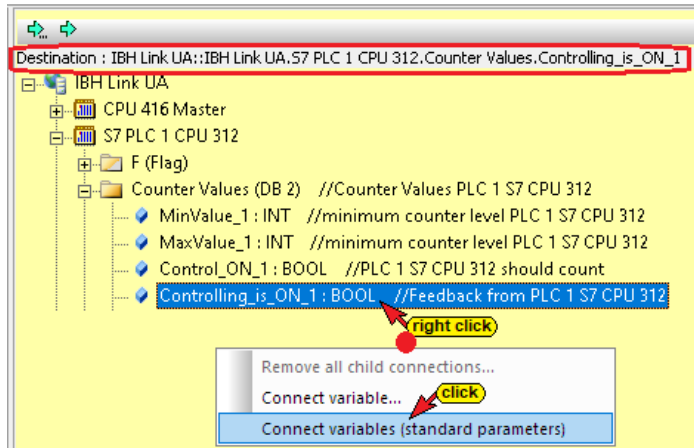
Mark the source OPC tag and right-click the destination OPC tag.

#### Source window



## Destination window

The command **Connect variable (standard parameters)** finalizes the definition. The connected OPC tags are listed in the lower part of the right project window.

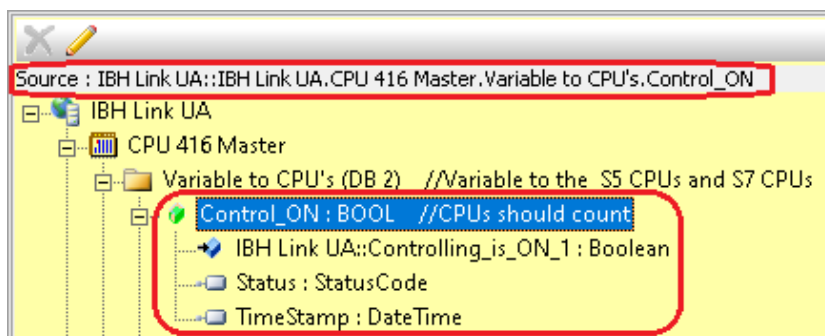


## Established connection

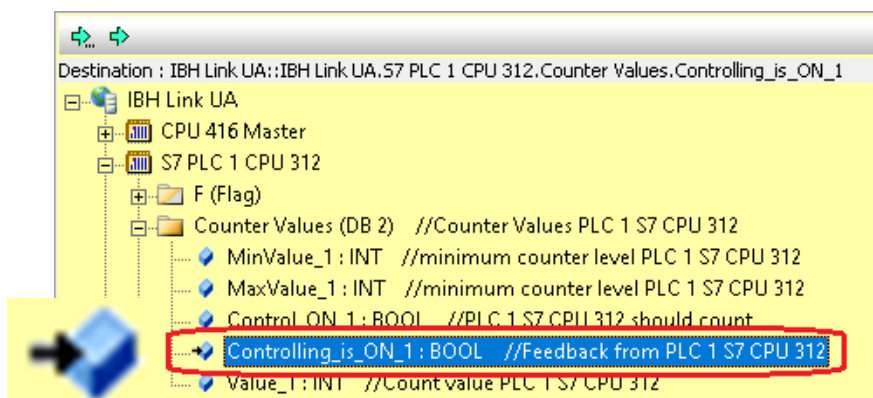
Source server	Source variable	Destination server	Destination variable	Data type	Source name	Destination name	Status
IBH Link UA	Control_ON	IBH Link UA	Controlling_is_ON_1	Boolean	IBH Link UA.CPU 416 Master.Variable to CPU's.Control_ON	IBH Link UA.S7 PLC 1 CPU 312.Counter Values.Controlling_is_ON_1	

Once a connection has been established, the symbols in front of the OPC tags changes. In addition to the **value**, the source OPC tag also offers the **time stamp** and the **status** of the OPC tag. To use these OPC tags, the corresponding destination variables must be available.

## Source variables connection established



## Destination variable connection established

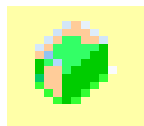


## Defined Connections

Source	Destination
<b>CPU 416 Master; Variable to CPU's [DB2]:</b>	<b>S7 PLC 1 CPU 312; Counter Values (DB2):</b>
Control_ON	Control_ON_1
MaxValue_1	MaxValue_1
MinValue_1	MinValue_1
	<b>S7 PLC 2 CPU 312; Counter Values (DB2):</b>
Control_ON	Control_ON_2
MaxValue_2	MaxValue_2
MinValue_2	MinValue_2
	<b>S5 PLC 3 CPU 103; F (Flags)</b>
Control_ON	Control_ON_3
	<b>Counter Values (DB2):</b>
MaxValue_3	MaxValue_3
MinValue_3	MinValue_3
	<b>S5 PLC 4 CPU 941; F (Flags)</b>
Control_ON_4	Control_ON_4
	<b>Counter Values (DB2):</b>
MaxValue_4	MaxValue_4
MinValue_4	MinValue_4
<b>S7 PLC 1 CPU 312; Counter Values (DB2):</b>	<b>CPU 416 Master; Variable from CPU's [DB5]:</b>
Controlling_is_ON_1	Controlling_is_ON_1
Value_1	Value_1
<b>S7 PLC 2 CPU 312; Counter Values (DB2):</b>	
Controlling_is_ON_2	Controlling_is_ON_2
Value_2	Value_2
<b>S5 PLC 3 CPU 103; F (Flags)</b>	<b>CPU 416 Master; Variable from CPU's [DB5]:</b>
Controlling_is_ON_3	Controlling_is_ON_3
<b>Counter Values (DB2):</b>	
Value_3	Value_3
<b>S5 PLC 4 CPU 941; F (Flags)</b>	
Controlling_is_ON_4	Controlling_is_ON_4
<b>Counter Values (DB2):</b>	
Value_4	Value_4

The connections are displayed in the lower part of the right project window.

Source server	Source variable	Destination server	Destination variable	Data type	Source name	Destination name
IBH Link UA	Control_ON	IBH Link UA	Controlling_is_ON_1	Boolean	IBH Link UA.CPU 416 Master.Variable to CPU's:Control_ON	IBH Link UA.S7 PLC 1 CPU 312.Counter Values:Controlling_is_ON_1
IBH Link UA	Control_ON	IBH Link UA	Control_ON_2	Boolean	IBH Link UA.CPU 416 Master.Variable to CPU's:Control_ON	IBH Link UA.S7 PLC 2 CPU 312.Counter Values:Control_ON_2
IBH Link UA	Control_ON	IBH Link UA	Control_ON_3	Boolean	IBH Link UA.CPU 416 Master.Variable to CPU's:Control_ON	IBH Link UA.S5 PLC 3 CPU 103.Counter_ON_3
IBH Link UA	Control_ON	IBH Link UA	Control_ON_4	Boolean	IBH Link UA.CPU 416 Master.Variable to CPU's:Control_ON	IBH Link UA.S5 PLC 4 CPU 941.Control_ON_4
IBH Link UA	MaxValue_1	IBH Link UA	MaxValue_1	Int16	IBH Link UA.CPU 416 Master.Variable to CPU's:MaxValue_1	IBH Link UA.S7 PLC 1 CPU 312.Counter Values:MaxValue_1
IBH Link UA	MinValue_1	IBH Link UA	MinValue_1	Int16	IBH Link UA.CPU 416 Master.Variable to CPU's:MinValue_1	IBH Link UA.S7 PLC 1 CPU 312.Counter Values:MinValue_1
IBH Link UA	MaxValue_2	IBH Link UA	MaxValue_2	Int16	IBH Link UA.CPU 416 Master.Variable to CPU's:MaxValue_2	IBH Link UA.S7 PLC 2 CPU 312.Counter Values:MaxValue_2
IBH Link UA	MinValue_2	IBH Link UA	MinValue_2	Int16	IBH Link UA.CPU 416 Master.Variable to CPU's:MinValue_2	IBH Link UA.S7 PLC 2 CPU 312.Counter Values:MinValue_2
IBH Link UA	MaxValue_3	IBH Link UA	D_1	Int16	IBH Link UA.CPU 416 Master.Variable to CPU's:MaxValue_3	IBH Link UA.S5 PLC 3 CPU 103.Counter Values:3_D_1
IBH Link UA	MinValue_3	IBH Link UA	D_0	Int16	IBH Link UA.CPU 416 Master.Variable to CPU's:MinValue_3	IBH Link UA.S5 PLC 3 CPU 103.Counter Values:3_D_0
IBH Link UA	MaxValue_4	IBH Link UA	D_1	Int16	IBH Link UA.CPU 416 Master.Variable to CPU's:MaxValue_4	IBH Link UA.S5 PLC 4 CPU 941.Counter Values:4_D_1
IBH Link UA	MinValue_4	IBH Link UA	D_0	Int16	IBH Link UA.CPU 416 Master.Variable to CPU's:MinValue_4	IBH Link UA.S5 PLC 4 CPU 941.Counter Values:4_D_0
IBH Link UA	Controlling_is_ON_1	IBH Link UA	Controlling_is_ON_1	Boolean	IBH Link UA.S7 PLC 1 CPU 312.Counter Values:Controlling_is_ON_1	IBH Link UA.CPU 416 Master.Variable from CPU's:Controlling_is_ON_1
IBH Link UA	Value_1	IBH Link UA	Value_1	Int16	IBH Link UA.S7 PLC 1 CPU 312.Counter Values:Value_1	IBH Link UA.CPU 416 Master.Variable from CPU's:Value_1
IBH Link UA	Controlling_is_ON_2	IBH Link UA	Controlling_is_ON_2	Boolean	IBH Link UA.S7 PLC 2 CPU 312.Counter Values:Controlling_is_ON_2	IBH Link UA.CPU 416 Master.Variable from CPU's:Controlling_is_ON_2
IBH Link UA	Value_2	IBH Link UA	Value_2	Int16	IBH Link UA.S7 PLC 2 CPU 312.Counter Values:Value_2	IBH Link UA.CPU 416 Master.Variable from CPU's:Value_2
IBH Link UA	Controlling_is_ON_3	IBH Link UA	Controlling_is_ON_3	Boolean	IBH Link UA.S5 PLC 3 CPU 103.Counter Values:Controlling_is_ON_3	IBH Link UA.CPU 416 Master.Variable from CPU's:Controlling_is_ON_3
IBH Link UA	D_2	IBH Link UA	Value_3	Int16	IBH Link UA.S5 PLC 3 CPU 103.Counter Values:3_D_2	IBH Link UA.CPU 416 Master.Variable from CPU's:Value_3
IBH Link UA	Controlling_is_ON_4	IBH Link UA	Controlling_is_ON_4	Boolean	IBH Link UA.S5 PLC 4 CPU 941.Counter Values:Controlling_is_ON_4	IBH Link UA.CPU 416 Master.Variable from CPU's:Controlling_is_ON_4
IBH Link UA	D_2	IBH Link UA	Value_4	Int16	IBH Link UA.S5 PLC 4 CPU 941.Counter Values:4_D_2	IBH Link UA.CPU 416 Master.Variable from CPU's:Value_4



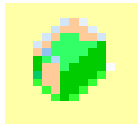
Linked read variables have this green symbol.

Source : 0

IBH Link UA

- CPU 416 Master
  - Variable to CPU's (DB 2) //Variable to the S5 CPUs and S7 CPUs
    - Control\_ON : BOOL //CPUs should count
    - MaxValue\_1 : INT //minimum count value PLC 1 S7 CPU 312
    - MaxValue\_2 : INT //minimum count value PLC 2 S7 CPU 312
    - MaxValue\_3 : INT //minimum count value PLC 1 S5 CPU 103
    - MaxValue\_4 : INT //minimum count value PLC 1 S5 CPU 941
    - MinValue\_1 : INT //maximum count value PLC 1 S7 CPU 312
    - MinValue\_2 : INT //maximum count value PLC 2 S7 CPU 312
    - MinValue\_3 : INT //maximum count value PLC 3 S5 CPU 103
    - MinValue\_4 : INT //maximum count value PLC 4 S5 CPU 941
  - Variable from CPU's (DB 5) //S5 CPUs and S7 CPUs send data
    - Generic
- S7 PLC 1 CPU 312
  - F (Flag)
  - Counter Values (DB 2) //Counter Values PLC 1 S7 CPU 312
    - MinValue\_1 : INT //minimum counter level PLC 1 S7 CPU 312
    - MaxValue\_1 : INT //minimum counter level PLC 1 S7 CPU 312
    - Control\_ON\_1 : BOOL //PLC 1 S7 CPU 312 should count
    - Controlling\_is\_ON\_1 : BOOL //Feedback from PLC 1 S7 CPU 312
    - Value\_1 : INT //Count value PLC 1 S7 CPU 312
  - Generic
- S7 PLC 2 CPU 312
  - F (Flag)
  - Counter Values (DB 2) //Counter Values PLC 2 S7 CPU 312
    - MinValue\_2 : INT //minimum counter level PLC 2 S7 CPU 312
    - MaxValue\_2 : INT //minimum counter level PLC 2 S7 CPU 312
    - Control\_ON\_2 : BOOL //PLC 2 S7 CPU 312 should count
    - Controlling\_is\_ON\_2 : BOOL //Feedback from PLC 2 S7 CPU 312
    - Value\_2 : INT //Count value PLC 2 S7 CPU 312
  - Generic
- S5 PLC 3 CPU 103
  - F (Flag)
  - Control\_ON\_3 : BOOL //PLC 3 S5 CPU 103enable counting
  - Controlling\_is\_ON\_3 : BOOL //feedback of PLC 3 S5 CPU 103
  - Max : BOOL //maximum reached
  - Min : BOOL //minimum reached
  - Value\_3\_MW : WORD //counter value PLC 3 S5 CPU 103
  - Down\_3 : BOOL //counting down
  - Up\_3 : BOOL //counting up
  - Counter Values\_3 (DB 2) //Counter Values
    - D\_0 : INT // MinValue PLC 3 S5
    - D\_1 : INT // MaxValue PLC 3 S5
    - D\_2 : INT // Value PLC 3 S5
  - Generic





Linked read variables have this green symbol.

Source : 0

IBH Link UA

- CPU 416 Master
- S7 PLC 1 CPU 312
- S7 PLC 2 CPU 312
- S5 PLC 3 CPU 103
- S5 PLC 4 CPU 941
  - F (Flag)
    - Control\_ON\_4 : BOOL //PLC 4 S5 CPU 941 enable counting
    - Controlling\_is\_ON\_4 : BOOL //feedback of PLC 4 S5 CPU 941
    - Max : BOOL //maximum reached
    - Min : BOOL //minimum reached
    - Value\_4\_MW : WORD //counter value PLC 4 S5 CPU 941
    - Down\_4 : BOOL //counting down
    - Up\_4 : BOOL //counting up
  - CounterValues\_4 (DB 2) //Counter Values
    - D\_0 : INT // minimum counter value S5 PLC 4
    - D\_1 : INT // maximum counter value S5 PLC 4
    - D\_2 : INT // counter value S5 PLC 4
  - Generic
  - UA Nodes

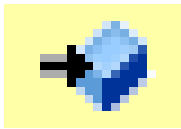


Linked target variables have this blue symbol.

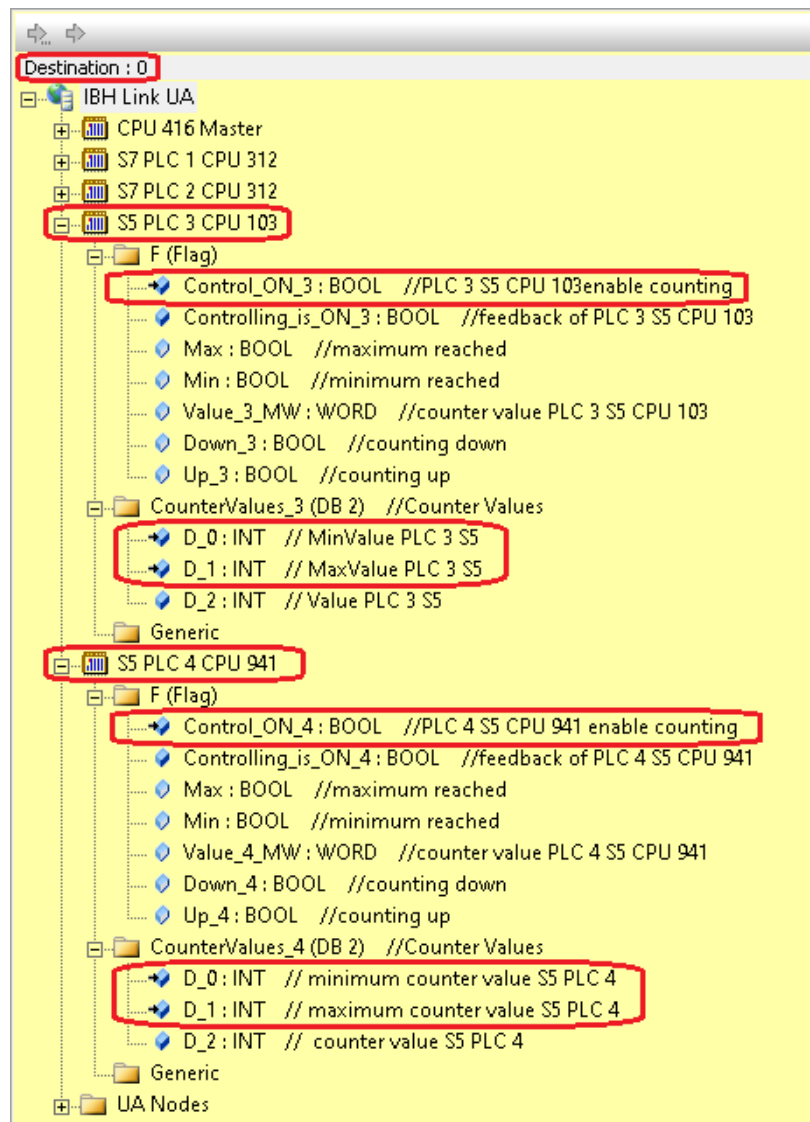
Destination : 0

IBH Link UA

- CPU 416 Master
  - Variable to CPU's (DB 2) //Variable to the S5 CPUs and S7 CPUs
  - Variable from CPU's (DB 5) //S5 CPUs and S7 CPUs send data
    - Controlling\_is\_ON\_1 : BOOL //PLC 1 S7 CPU 312 counts
    - Controlling\_is\_ON\_2 : BOOL //PLC 2 S7 CPU 312 counts
    - Controlling\_is\_ON\_3 : BOOL //PLC 3 S5 CPU 103 counts
    - Controlling\_is\_ON\_4 : BOOL //PLC 4 S5 CPU 941 counts
    - Value\_1 : INT //PLC 1 S7 CPU 312 Count value
    - Value\_2 : INT //PLC 2 S7 CPU 312 Count value
    - Value\_3 : INT //PLC 3 S5 CPU 103 Count value
    - Value\_4 : INT //PLC 4 S5 CPU 941 Count value
  - Generic
- S7 PLC 1 CPU 312
  - F (Flag)
  - Counter Values (DB 2) //Counter Values PLC 1 S7 CPU 312
    - MinValue\_1 : INT //minimum counter level PLC 1 S7 CPU 312
    - MaxValue\_1 : INT //minimum counter level PLC 1 S7 CPU 312
    - Control\_ON\_1 : BOOL //PLC 1 S7 CPU 312 should count
    - Controlling\_is\_ON\_1 : BOOL //Feedback from PLC 1 S7 CPU 312
    - Value\_1 : INT //Count value PLC 1 S7 CPU 312
  - Generic
- S7 PLC 2 CPU 312
  - F (Flag)
  - Counter Values (DB 2) //Counter Values PLC 2 S7 CPU 312
    - MinValue\_2 : INT //minimum counter level PLC 2 S7 CPU 312
    - MaxValue\_2 : INT //minimum counter level PLC 2 S7 CPU 312
    - Control\_ON\_2 : BOOL //PLC 2 S7 CPU 312 should count
    - Controlling\_is\_ON\_2 : BOOL //Feedback from PLC 2 S7 CPU 312
    - Value\_2 : INT //Count value PLC 2 S7 CPU 312
  - Generic

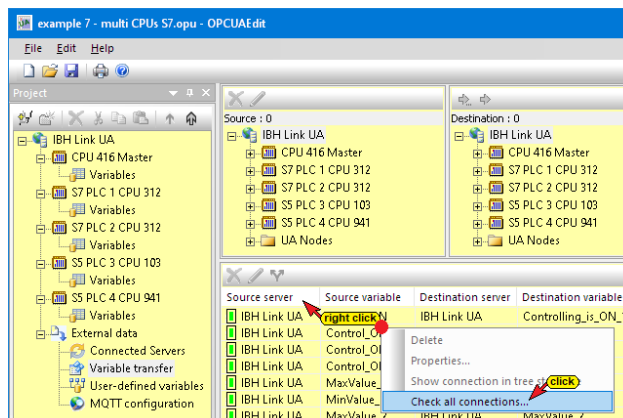


Linked target variables have this blue symbol.



## 2.7.9 Check connections

Is the configuration transferred to the IBH Link UA and online connection to the individual CPUs are present, a right-click on the heading opens the context menu with the command **Check all connections**.

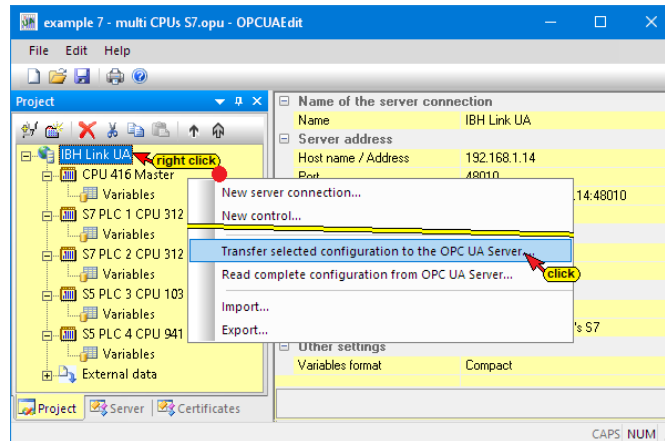


Click the command **Check all connections**. A defective connection changes its green status icon to red.



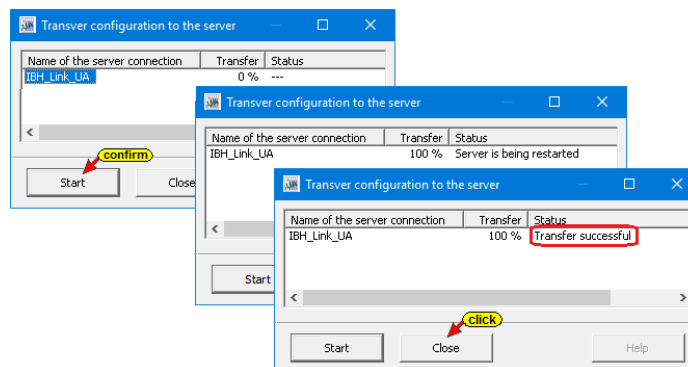
## 2.7.10 Transfer configuration to the OPC UA server (IBH Link UA).

A right-click on the Server icon (IBH Link UA) opens the context menu.



The command **Transfer Selected Configuration to OPC UA Server** command opens the **Transfer Configuration to Server** dialog box.

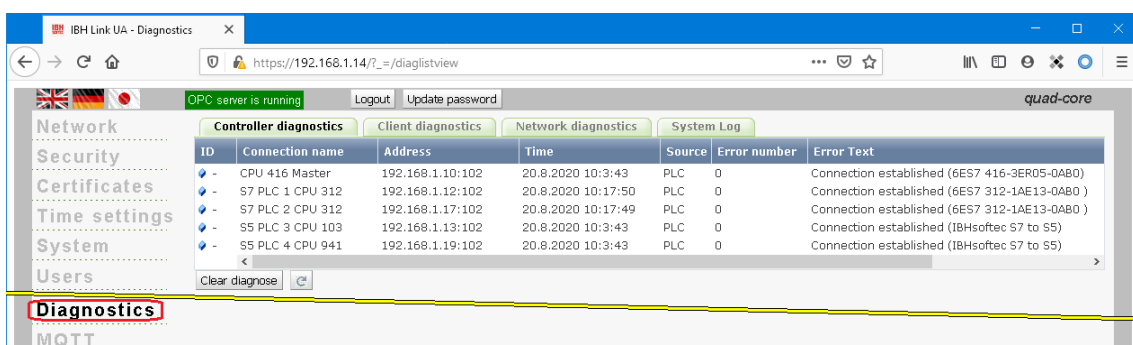
Select the server **IBH Link UA** and then click Start. The configuration is transferred to the **IBH Link UA**.



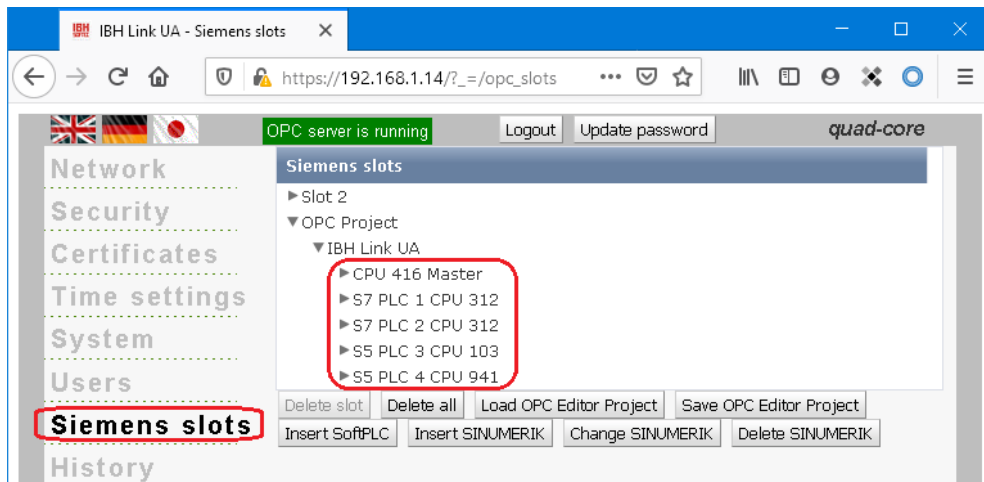
The successful transfer is displayed.

## 2.7.11 IBH Link UA browser windows

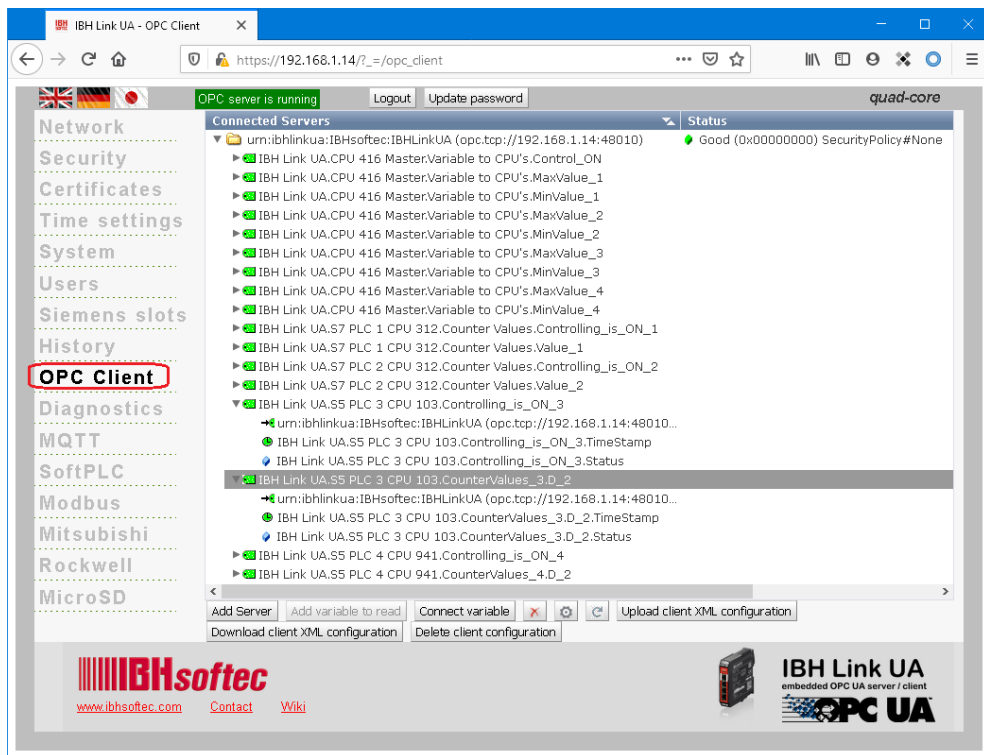
The browser window **Diagnostics** displays the status of the connection **IBH Link UA – PLCs**.



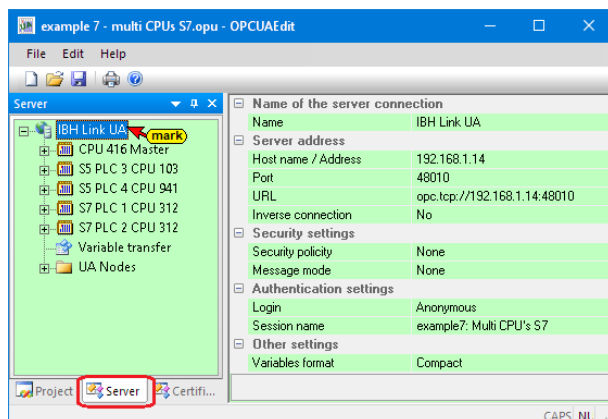
## IBH Link UA – Siemens Slots – OPC Project



## IBH Link UA – OPC Client



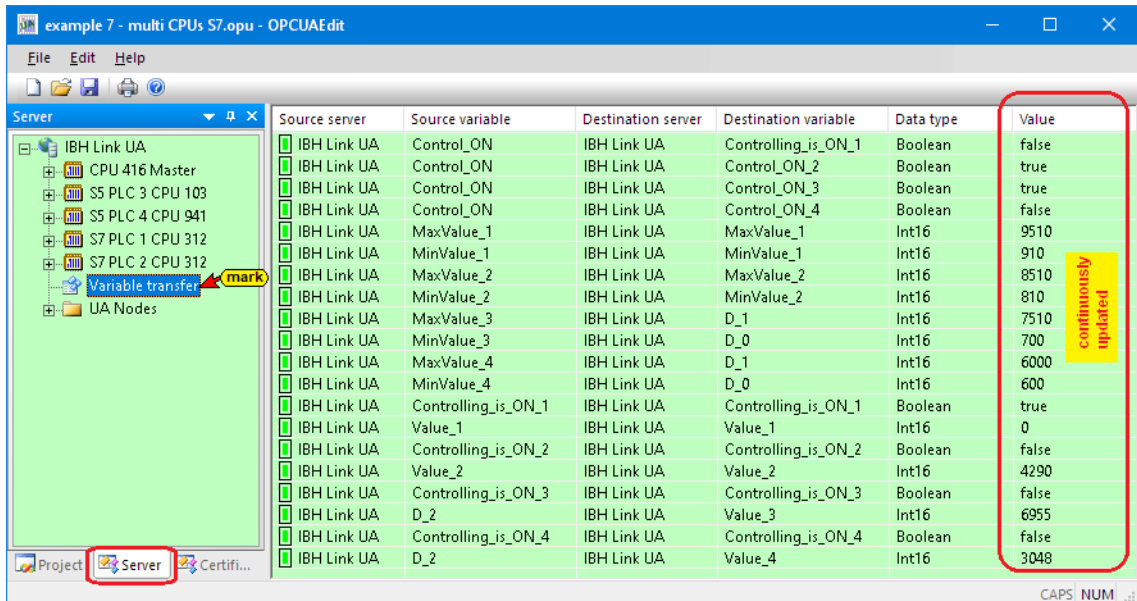
### 2.7.12 Online OPC UA Server Information



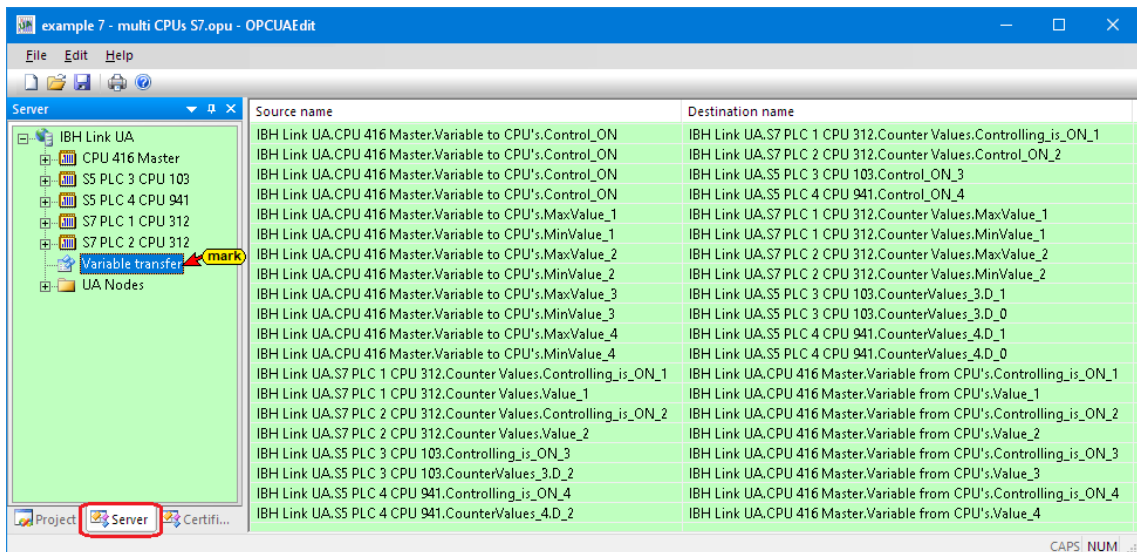
Information from the OPC UA server connected online with the CPUs are displayed.

## Show variable transfer

The individual OPC tags are displayed in the right server window with their status. The status of the OPC tags is updated continuously.

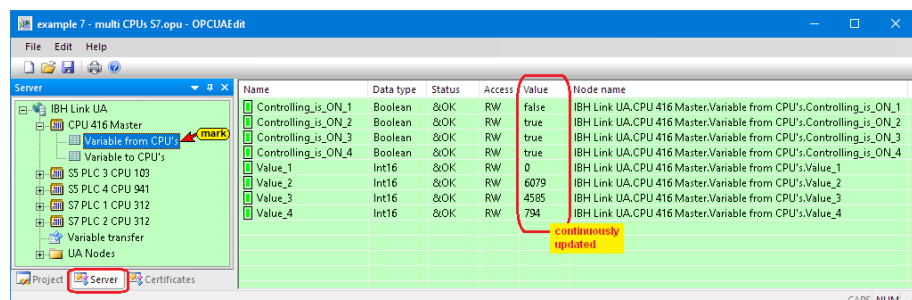


The source and destination names of the variable transfer are displayed.

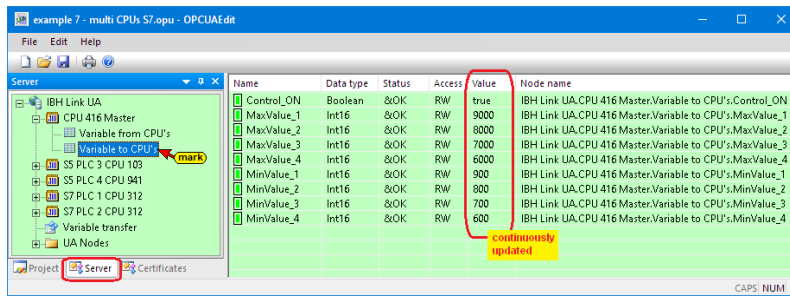


Marking the data block name, the individual variables (OPC tags) are displayed in the right server window with their status.

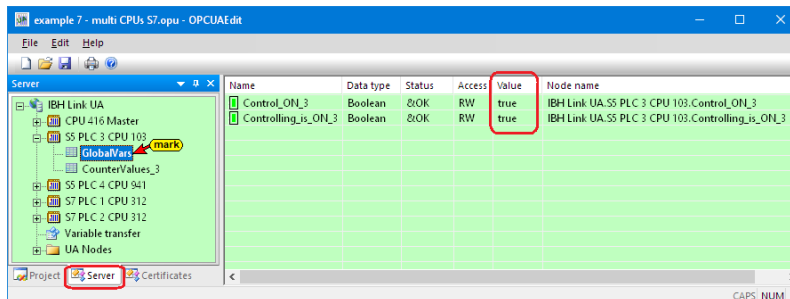
### CPU 416 Master; data block *Variable from CPU's [DB2]*



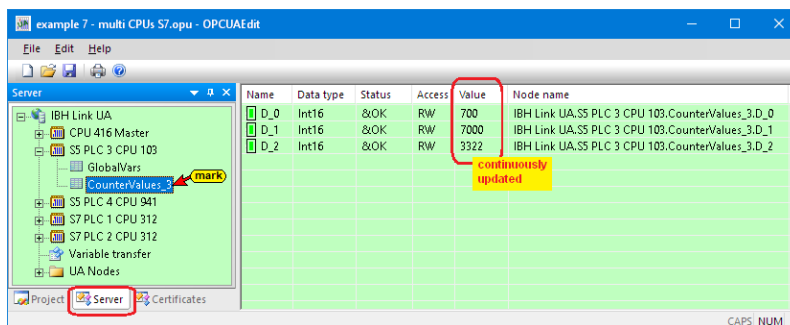
### CPU 416 Master; data block *Variable to CPU's [DB5]*



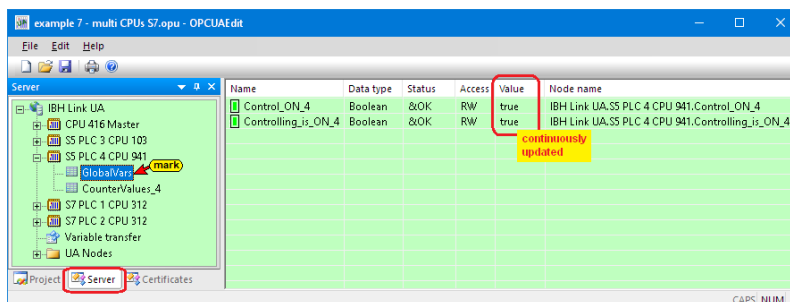
### S5 PLC 3 CPU 103; GlobalVars [F flag]



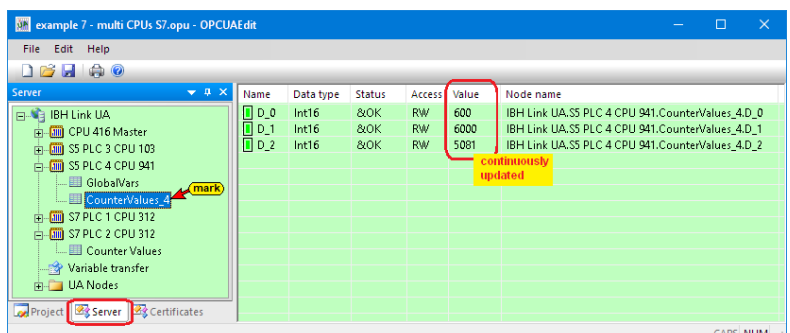
### S5 PLC 3 CPU 103; data block *CounterValues\_3 [DB2]*



### S5 PLC 4 CPU 941; GlobalVars [F flag]



### S5 PLC 4 CPU 941; data block *CounterValues\_4 [DB2]*



### S7 PLC 1 CPU 312; data block CounterValues [DB2]

Name	Data type	Status	Access	Value	Node name
Control_ON_1	Boolean	&OK	RW	false	IBH Link UA.S7 PLC 1 CPU 312.CounterValues.Control_ON_1
Controlling_is_ON_1	Boolean	&OK	RW	false	IBH Link UA.S7 PLC 1 CPU 312.CounterValues.Controlling_is_ON_1
MaxValue_1	Int16	&OK	RW	9000	IBH Link UA.S7 PLC 1 CPU 312.CounterValues.MaxValue_1
MinValue_1	Int16	&OK	RW	900	IBH Link UA.S7 PLC 1 CPU 312.CounterValues.MinValue_1
Value_1	Int16	&OK	RW	0	IBH Link UA.S7 PLC 1 CPU 312.CounterValues.Value_1

### S7 PLC 2 CPU 312; data block CounterValues [DB2]

Name	Data type	Status	Access	Value	Node name
Control_ON_2	Boolean	&OK	RW	true	IBH Link UA.S7 PLC 2 CPU 312.CounterValues.Control_ON_2
Controlling_is_ON_2	Boolean	&OK	RW	true	IBH Link UA.S7 PLC 2 CPU 312.CounterValues.Controlling_is_ON_2
MaxValue_2	Int16	&OK	RW	8000	IBH Link UA.S7 PLC 2 CPU 312.CounterValues.MaxValue_2
MinValue_2	Int16	&OK	RW	800	IBH Link UA.S7 PLC 2 CPU 312.CounterValues.MinValue_2
Value_2	Int16	&OK	RW	6164	IBH Link UA.S7 PLC 2 CPU 312.CounterValues.Value_2

## 2.7.13 Unified Automation UaExpert - The OPC Unified Architecture Client

The UaExpert program window lists the OPC tags transferred by the IBH OPC UA Editor and the associated UA nodes.

#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	IBHLinkUA@ib...	NS4[String]IBH...	Controlling_is_ON_1	false	Boolean	13:53:16.758	13:53:17.000	Good
2	IBHLinkUA@ib...	NS4[String]IBH...	Controlling_is_ON_2	false	Boolean	13:53:21.765	13:53:22.003	Good
3	IBHLinkUA@ib...	NS4[String]IBH...	Controlling_is_ON_3	true	Boolean	13:53:16.758	13:53:17.000	Good
4	IBHLinkUA@ib...	NS4[String]IBH...	Controlling_is_ON_4	true	Boolean	13:53:16.758	13:53:17.000	Good
5	IBHLinkUA@ib...	NS4[String]IBH...	Value_1	0	Int16	13:53:08.090	13:53:09.668	Good
6	IBHLinkUA@ib...	NS4[String]IBH...	Value_2	5941	Int16	13:53:21.765	13:53:22.003	Good
7	IBHLinkUA@ib...	NS4[String]IBH...	Value_3	4749	Int16	13:53:21.765	13:53:22.003	Good
8	IBHLinkUA@ib...	NS4[String]IBH...	Value_4	4025	Int16	13:53:21.765	13:53:22.003	Good
9	IBHLinkUA@ib...	NS4[String]IBH...	Control_ON	true	Boolean	13:53:21.594	13:53:21.753	Good
10	IBHLinkUA@ib...	NS4[String]IBH...	MaxValue_1	9000	Int16	13:53:21.594	13:53:21.753	Good
11	IBHLinkUA@ib...	NS4[String]IBH...	MaxValue_2	8000	Int16	13:53:21.594	13:53:21.753	Good
12	IBHLinkUA@ib...	NS4[String]IBH...	MaxValue_3	7000	Int16	13:53:21.594	13:53:21.753	Good
13	IBHLinkUA@ib...	NS4[String]IBH...	MaxValue_4	6000	Int16	13:53:21.594	13:53:21.753	Good
14	IBHLinkUA@ib...	NS4[String]IBH...	MinValue_1	900	Int16	13:53:21.594	13:53:21.753	Good
15	IBHLinkUA@ib...	NS4[String]IBH...	MinValue_2	800	Int16	13:53:21.594	13:53:21.753	Good
16	IBHLinkUA@ib...	NS4[String]IBH...	MinValue_3	700	Int16	13:53:21.594	13:53:21.753	Good
17	IBHLinkUA@ib...	NS4[String]IBH...	MinValue_4	600	Int16	13:53:21.594	13:53:21.753	Good

## 2.8 Example 8: Connecting two S7 CPU 300 via an IBH Link S7++

S7-300 series CPUs not having a free Ethernet port may be connected to the IBH Link UA via MPI bus via an IBH Link S7 ++ via Ethernet (protocol RFC 1006).

The example shows the creation of a project with the connection of two (2) CPU 312 to one (1) IBH Link UA via IBH Link S7 ++. Instead of the CPU 312, any other S7 CPU 300/400 that does not have a free Ethernet port could be used.

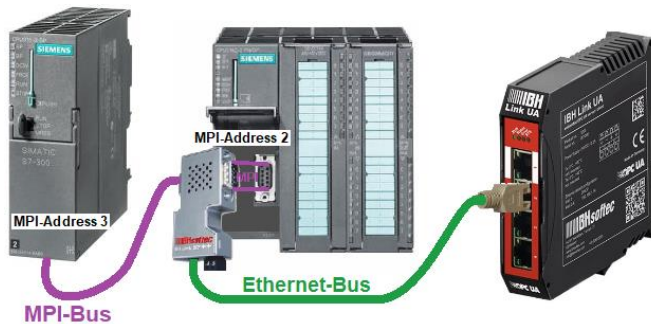
### IBH Link S7++

The IBH Link S7++ is an Ethernet converter. The standard TCP / IP is used. The user can benefit from all the advantages of Ethernet.

The use of **SimaticNet** and the use of a CP communication processor is not necessary on either the PC or the PLC side.

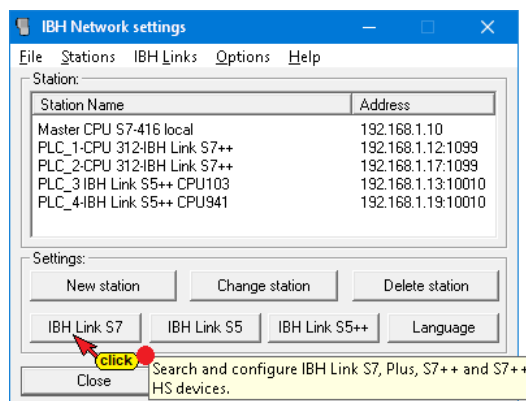


### Connecting the IBH Link UA with 2x S7 PLC CPU 312

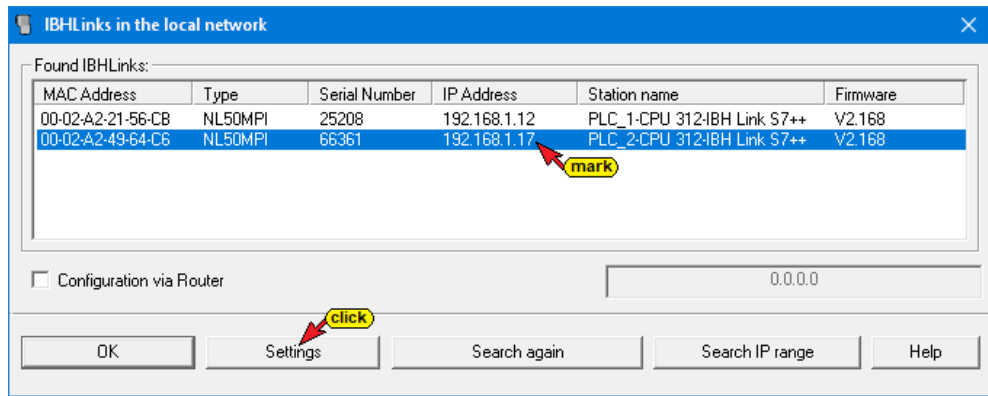


### Configuration of the IBH Link S7 ++

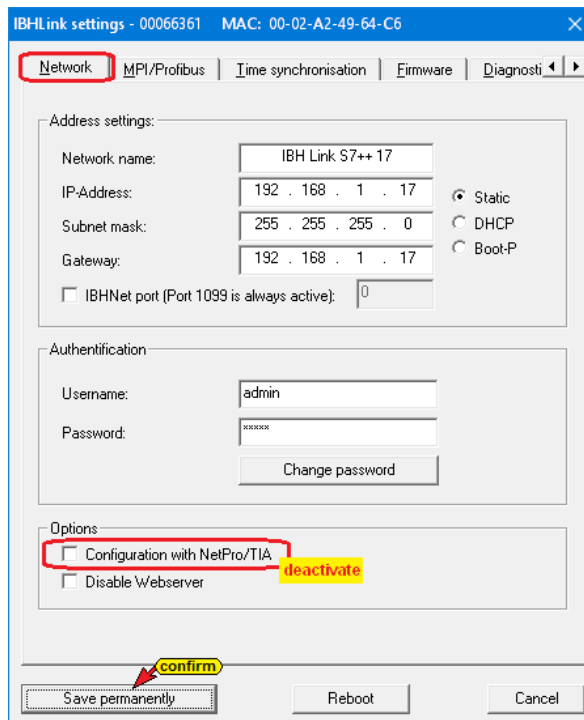
Open the IBH Network settings dialog box to deactivate the option Configuration with NetPro/TIA.





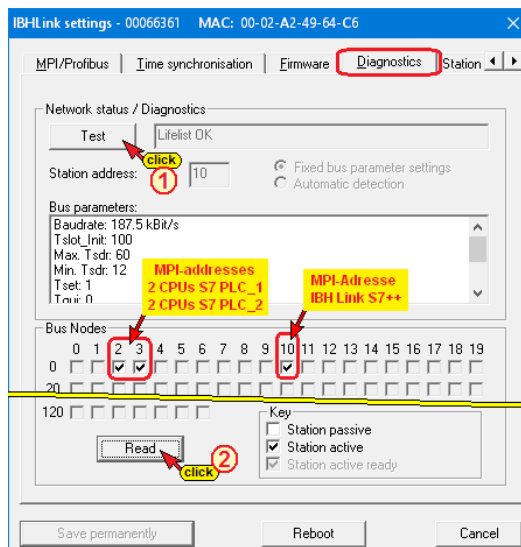


### Deactivate the option Configuration with NetPro / TIA



### IBH Link S7++ MPI addresses

Diagnostics are available to show the MPI bus parameters and the MPI addresses in use.



## IP / MPI addresses / PLC programs of the devices

Device	IP-Address	MPI-Address
IBH Link UA	192.168.1.14	not applicable
IBH Link S7++	192.168.1.12	10
<b>S7-CPU 1 – [CPU 312]</b> <b>PLC-Project / Program:</b> 2 CPU's S7 PLC_1 / S7 CPU 1	IBH Link S7++ together for: 2 CPU's S7 PLC_1 2 CPU's S7 PLC_2 <b>192.168.1.12</b>	2
<b>S7-CPU 2 – [CPU 312]</b> <b>PLC-Project / Program:</b> 2 CPU's S7 PLC_2 / S7 CPU 2		3

### 2.8.1 PLC programs

The S5 / S7 CPUs programs count value until **MaxValue** is reached. Then the **Value** is counted down until **MinValue** is reached. This up and down counting is repeated continuously.

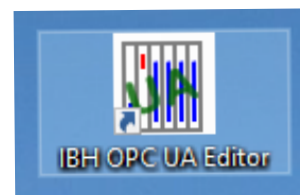
The data blocks **CounterValue\_1 [DB2]** and **CounterData\_from\_PLC\_2 [DB5]** are in the **S7-CPU 1**. The data blocks **CounterValue\_2 [DB2]** and **CounterData\_from\_PLC\_1 [DB5]** are in the **S7-CPU 2**.

These data blocks are available for data exchange with one another. The variables of these blocks are defined as (OPC tags).

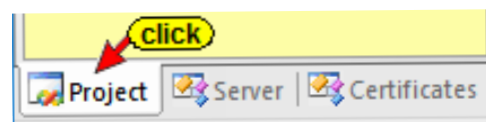
The programs for the S7 CPUs are available in the **STEP® 7** - SIMATIC Manager format.

### Calling the IBH OPC UA Editor

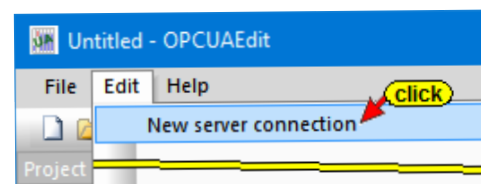
Double-click the **IBH OPC UA Editor** icon to open the program window.



Open the **Project window** by clicking on the **Project** tab.



Open the **New Server Connection** dialog box with the New Server Connection command from the **Edit** menu or by clicking the icon.



The new server connection setup was explained in example 1 (see chapter 2, page 2-3).

## Server Connection dialog box

Server connection properties

Name of the server connection: IBH Link UA

Server address:

Host name or IP address: 192.168.1.14

Port: 48010

URL: opc.tcp://192.168.1.14:48010

Select endpoint...

Security settings:

None

Basic128Rsa15

Basic256

BasicSha256

Aes128Sha256RsaOaep

Aes256Sha256RsaPss

Message mode:

Signatur

Signature and Encryption

Inverse connection:

Connect invers

Properties...

Login:

Anonymous

User name and password

User name: \_\_\_\_\_

Password: \_\_\_\_\_  Store

Session Name: example 8: 2x S7 CPU 312

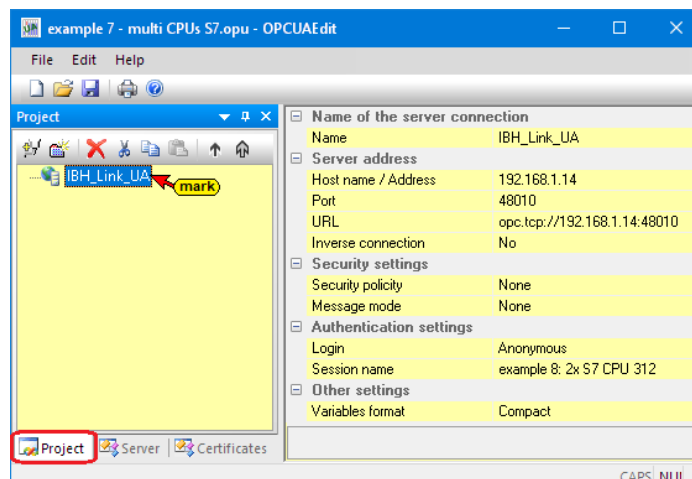
Variables format: Compact

confirm

OK Cancel Help

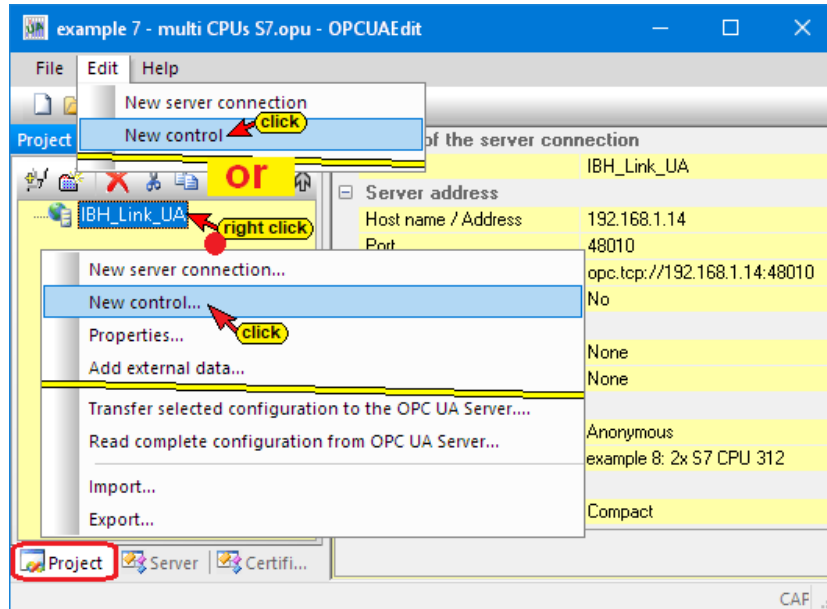
All necessary settings are shown. The possible encryptions of the data to be transmitted are displayed. The example is using the security method **None**. By clicking **OK** the content of the dialog box is saved and closed.

The settings for the connection to the **IBH Link UA** OPC UA server are displayed in the right part of the **project window**.

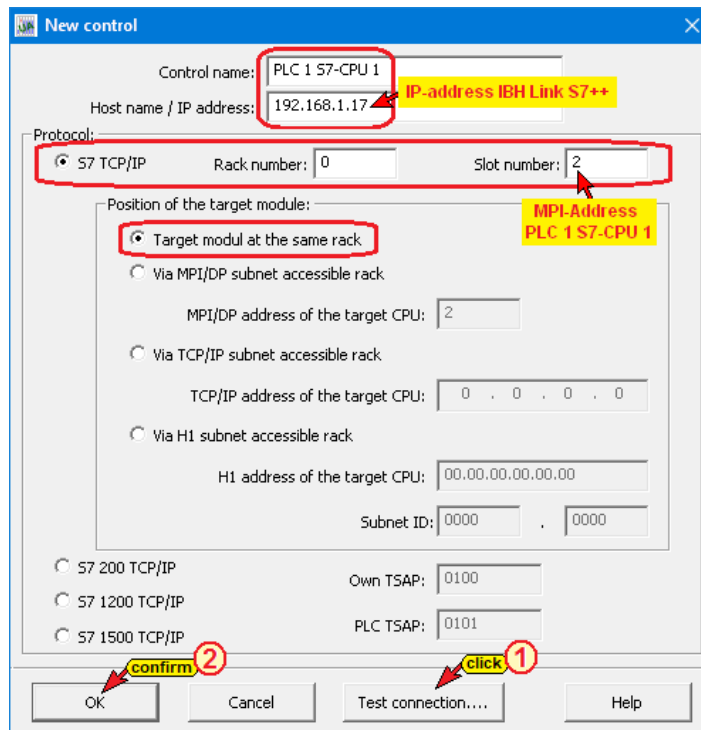


## 2.8.2 Inserting a New controls (PLC)

The **New control** command from the context menu (or menu Edit / New control) opens the dialog box **New control** to specify the access to the control (CPU).



### New control dialog box – PLC 1 S7-CPU 1

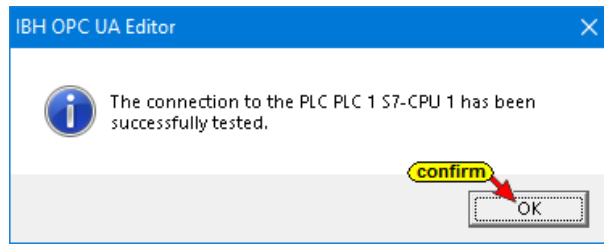


#### Test connection

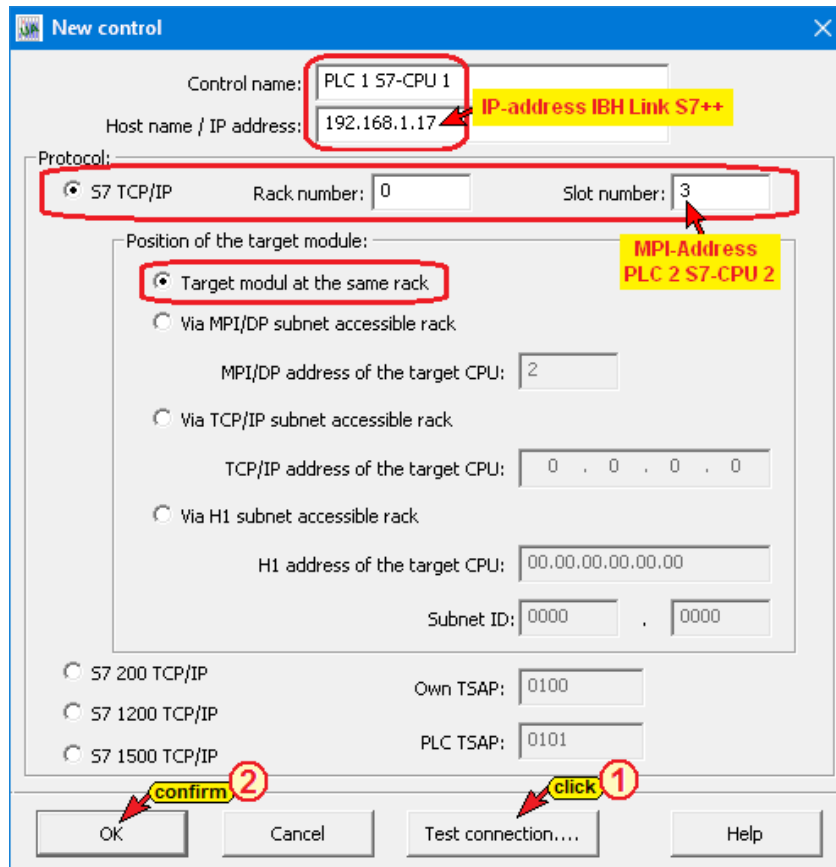
After completing the New Control dialog box, the connection to the online connected CPU can be tested.

Test connection....

Information about the successful connection is displayed.



## New control dialog box – PLC 2 S7-CPU 2

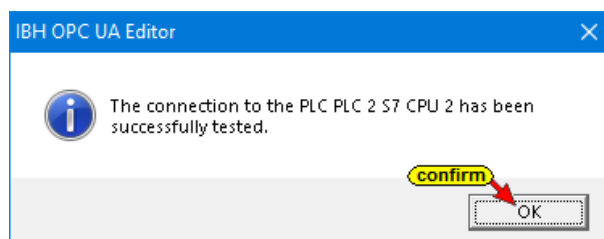


### Test connection

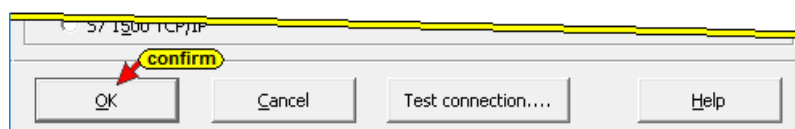
After completing the New Control dialog box, the connection to the online connected CPU can be tested.



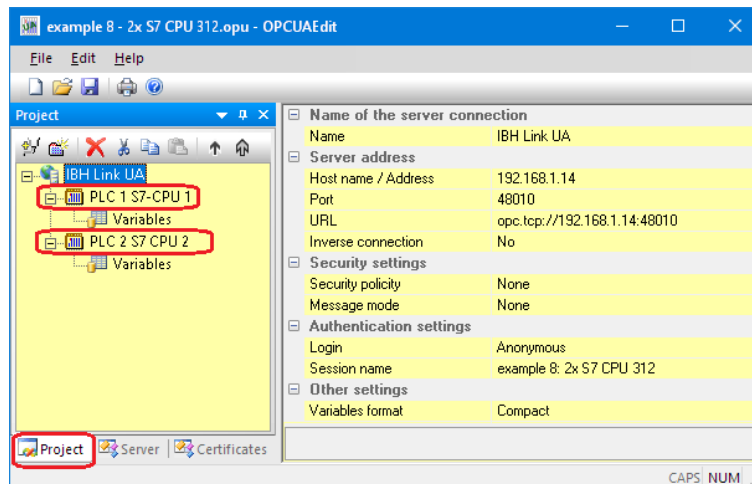
Information about the successful connection is displayed.



To accept and close the **New control** dialog box settings click on **OK**.



The specified PLC controls are displayed in the left project window.



### 2.8.3 IBH Link S7 ++ setting

If no connection is established from the PC via **IBH Link S7 ++**, the settings must be checked (see chapter 1 page 1-13 – IBH Link UA - S7 CPU 300 / 400 connection via IBH Link S7++).

#### Note:

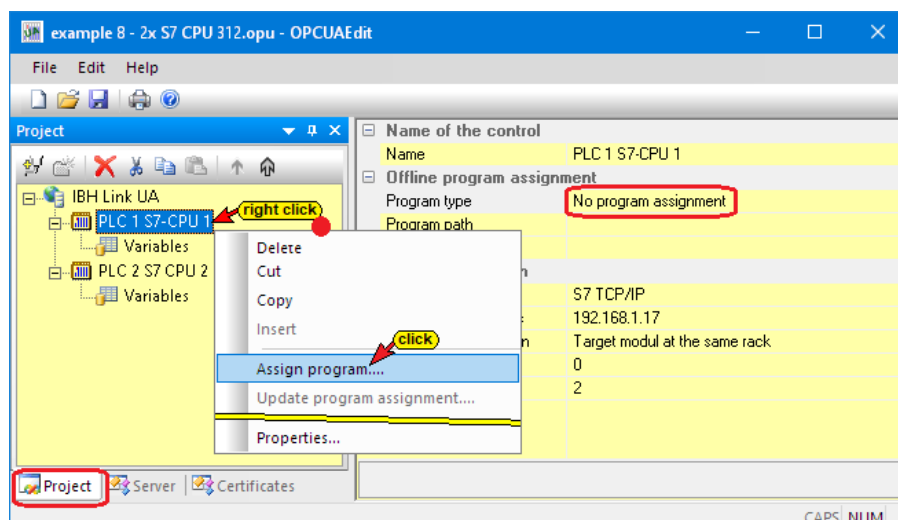


To address an CPU 300/400 via the **IBH Link S7++**, the routing option (dialog box **IBHLink settings / Network tab**) **Configuration with NetPro** must be deactivated (Apply permanently).

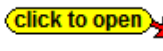


**This applies to all S7 300/400 CPUs with IBH Link S7++ connection.**

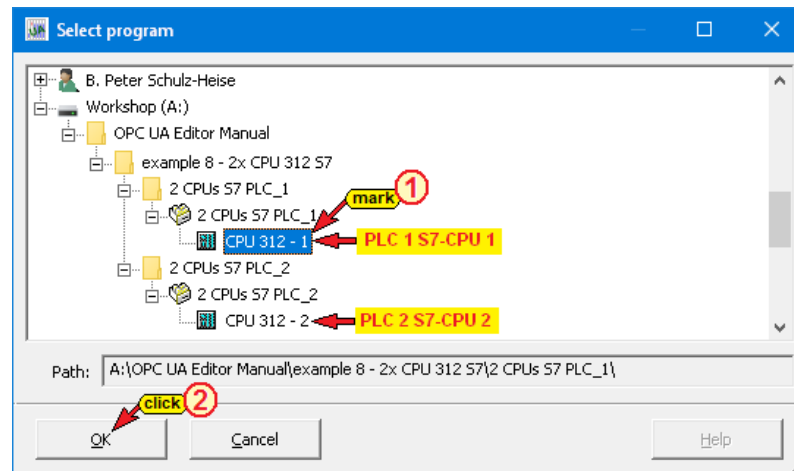
### 2.8.4 Program assignment

The Assign Program command opens the Program Selection dialog box.



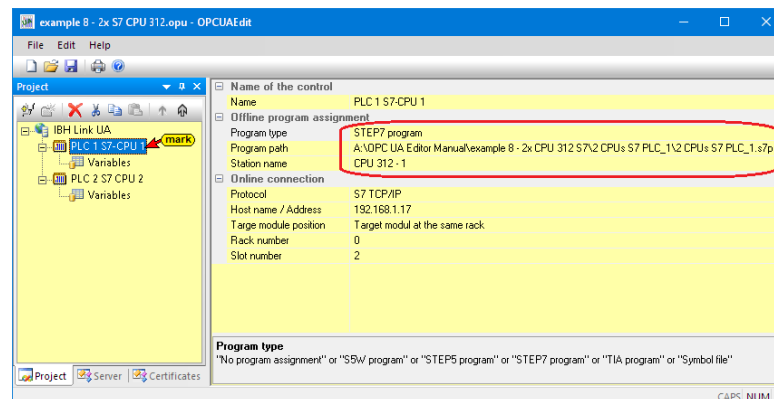
## Select Program dialog box

Select the PLC program in the **Select program**    dialog box. Clicking the **Plus** symbol in front of the PLC project name. Mark the PLC program (CPU) and click OK to assign the PLC program.



## Assigned PLC program

In the right part of the project window information about the **program assignment** are displayed.



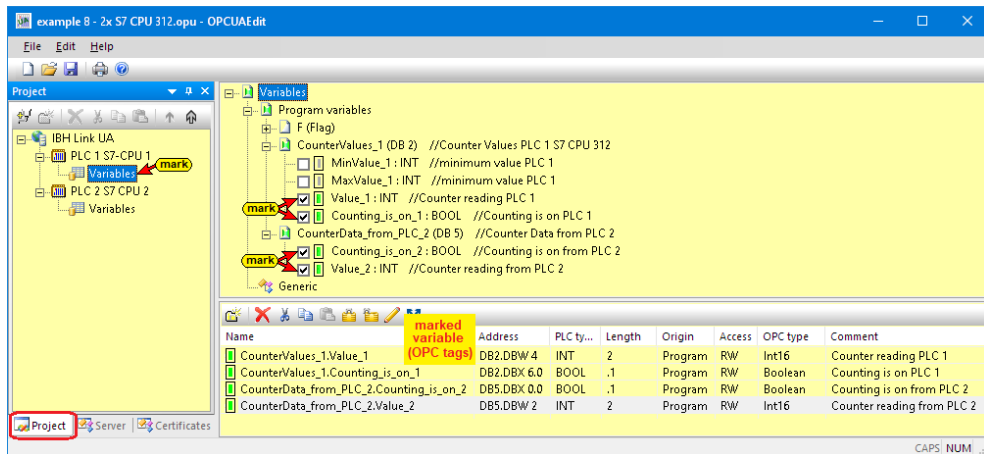
The corresponding PLC program must be assigned to the two (2) PLC controls.

### 2.8.5 Define variables as OPC tags

Clicking **Variables** lists the variables / data (data blocks) from the PLC in the right part of the project window. The selected **OPC tags** are listed in the lower part of the right window.

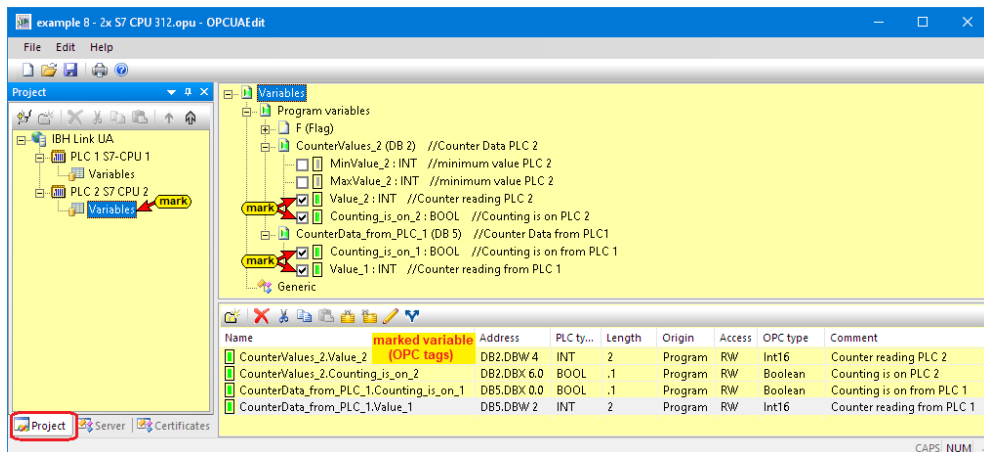
#### OPC tags of the controller –PLC 1 S7-CPU 1

Two (2) variables of the data block **CounterValues\_1 [DB2]** and two (2) variables of data block **CounterData\_from\_PLC\_2** are defined as OPC tags.



## OPC tags of the controller –PLC 2 S7-CPU 2

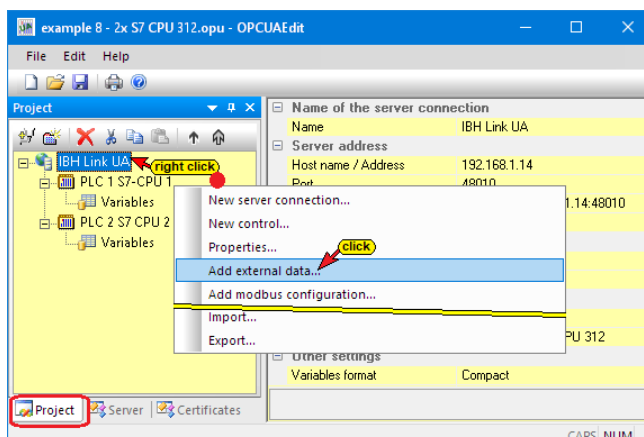
Two (2) variables of the data block **CounterValues\_2 [DB2]** and two (2) variables of data block **CounterData\_from\_PLC\_1** are defined as OPC tags.



### 2.8.6 Add external data

To exchange data between two OPC UA servers, the second server and its variables are defined using **Add external data**.

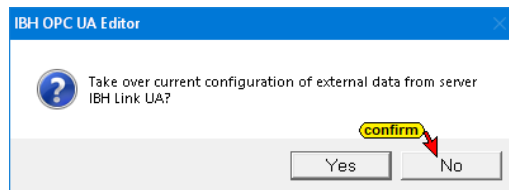
Right-click on **IBH Link UA** and execute the **Add External Data ...** command.





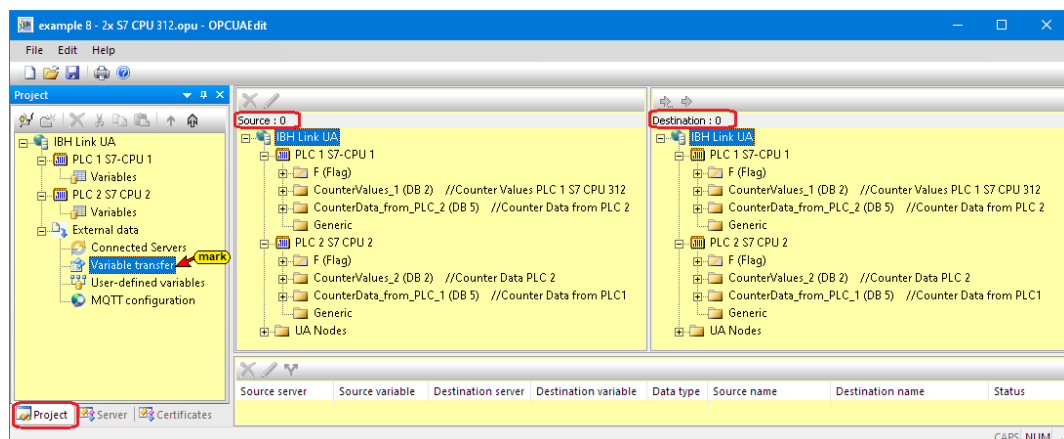
The OPC tags defined in the IBH OPC UA Editor are to be used. Confirm the dialog box with **No**.

No



## 2.8.7 Variable transfer – define source and destination OPC tags

**External data** with additional commands was inserted in the left part of the project window.



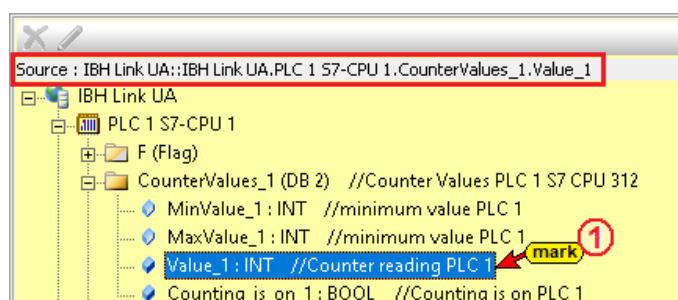
Mark **Variable transfer** to define the source and destination OPC tags. The right project window is divided into two parts. The **Source** window is on the left and the **Destination** window on the right. The Source window and the Destination window lists the OPC tags of the OPC server and the PLC Controllers.

The **OPC tags** to be read (read variables) are specified in the **Source** window. The **OPC tag** to be linked with the variable read is specified in the **Destination** window.

### Variable connection

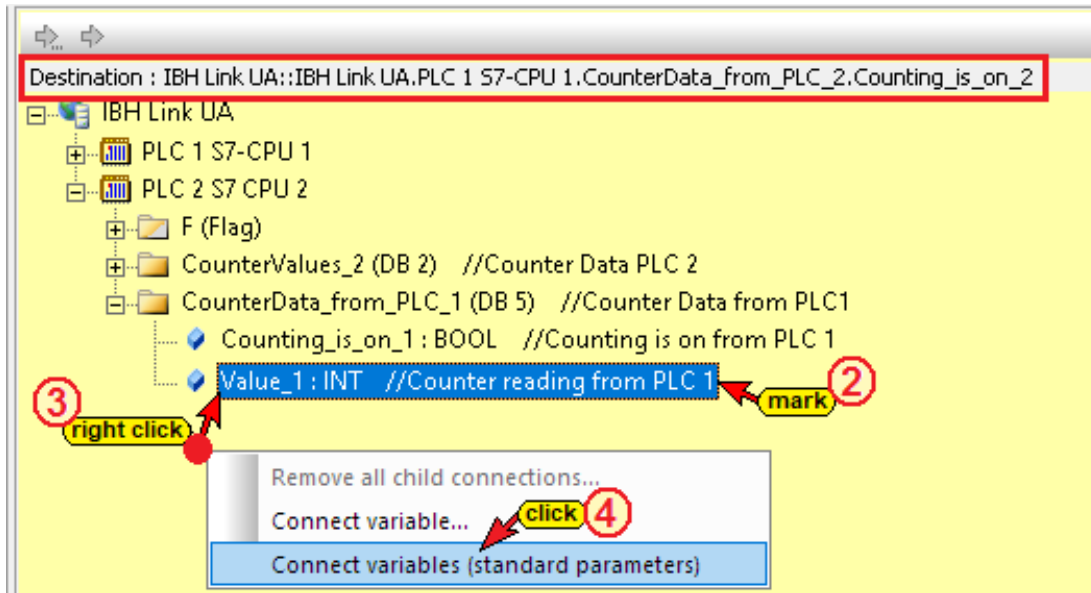
Mark the source OPC tag and right-click the destination OPC tag.

#### Source window



### Destination window

The command **Connect variable (standard parameters)** finalizes the definition. The connected OPC tags are listed in the lower part of the right project window.

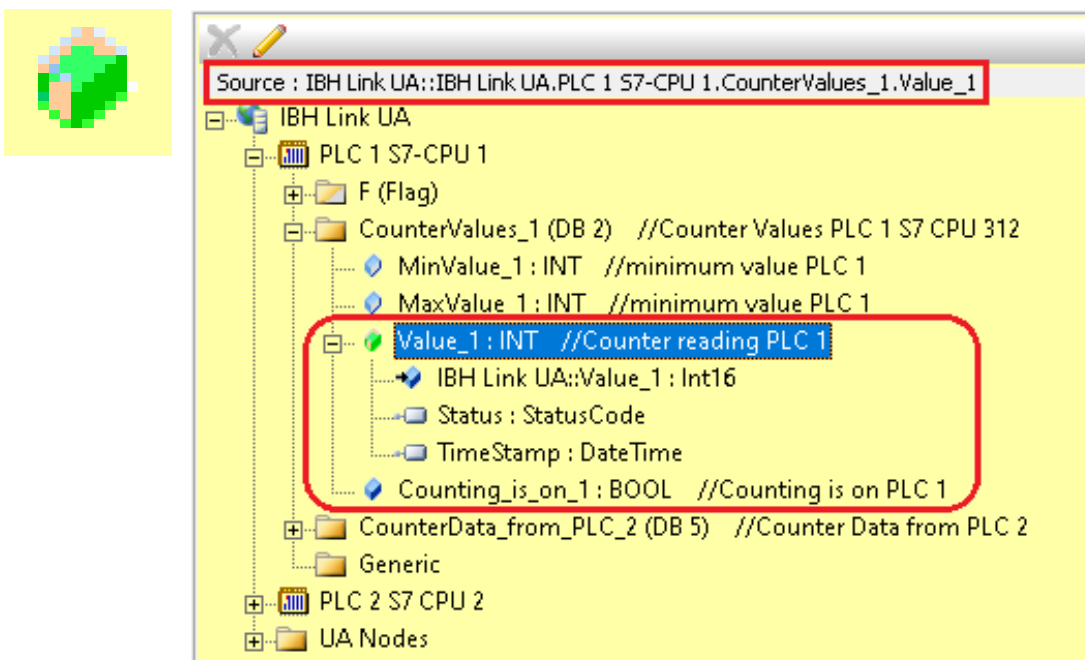


### Established connection

Source server	Source variable	Destination server	Destination variable	Data type	Source name	Destination name	Status
IBH Link UA	Value_1	IBH Link UA	Value_1	Int16	IBH Link UA.PLC 1 S7-CPU 1.CounterValues_1.Value_1	IBH Link UA.PLC 2 S7 CPU 2.CounterData_from_PLC_1.Value_1	

Once a connection has been established, the symbols in front of the OPC tags changes. In addition to the **value**, the source OPC tag also offers the **time stamp** and the **status** of the OPC tag. To use these OPC tags, the corresponding destination variables must be available.

### Source variables connection established



### Destination variable connection established



### Defined Connections

Source	Destination
<b>PLC 1 S7-CPU 1;</b> <b>CounterValues_1 [DB2]</b>	<b>PLC 2 S7-CPU 2;</b> <b>CountingData_from_PLC_1 [DB5]</b>
Value_1	Value_1
Counting_is_on_1	Counting_is_on_1
<b>PLC 2 S7-CPU 2;</b> <b>CounterValues_2 [DB2]</b>	<b>PLC 1 S7-CPU 1;</b> <b>CountingData_from_PLC_1 [DB5]</b>
Value_2	Value_2
Counting_is_on_2	Counting_is_on_2

The connections are displayed in the lower part of the right project window.

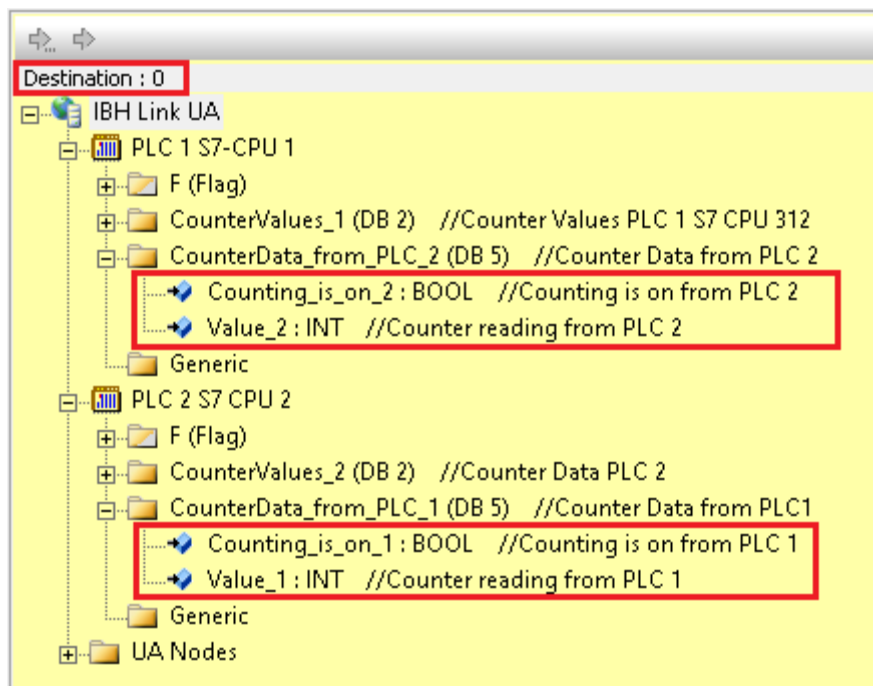
Source server	Source variable	Destination server	Destination variable	Data type	Source name	Destination name
IBH Link UA	Value_1	IBH Link UA	Value_1	Int16	IBH Link UA.PLC 1 S7-CPU 1.CounterValues_1.Value_1	IBH Link UA.PLC 2 S7 CPU 2.CounterData_from_PLC_1.Value_1
IBH Link UA	Counting_is_on_1	IBH Link UA	Counting_is_on_1	Boolean	IBH Link UA.PLC 1 S7-CPU 1.CounterValues_1.Counting_is_on_1	IBH Link UA.PLC 2 S7 CPU 2.CounterData_from_PLC_1.Counting_is_on_1
IBH Link UA	Value_2	IBH Link UA	Value_2	Int16	IBH Link UA.PLC 2 S7 CPU 2.CounterValues_2.Value_2	IBH Link UA.PLC 1 S7-CPU 1.CounterData_from_PLC_2.Value_2
IBH Link UA	Counting_is_on_2	IBH Link UA	Counting_is_on_2	Boolean	IBH Link UA.PLC 2 S7 CPU 2.CounterValues_2.Counting_is_on_2	IBH Link UA.PLC 1 S7-CPU 1.CounterData_from_PLC_2.Counting_is_on_2



### Linked read variables have this green symbol.

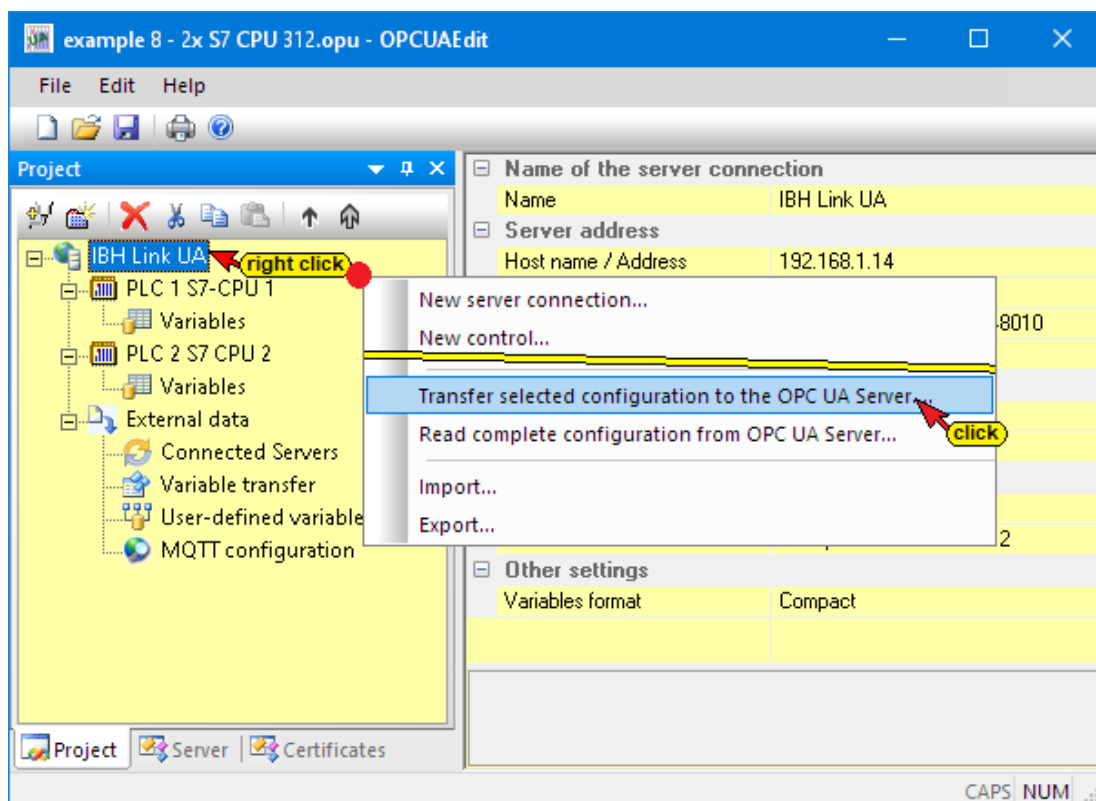


Linked target variables have this blue symbol.



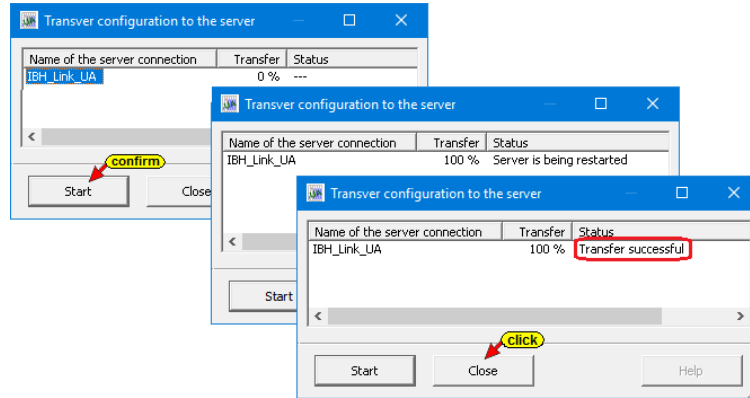
### 2.8.8 Transfer configuration to the OPC UA server (IBH Link UA).

A right-click on the Server icon (IBH Link UA) opens the context menu.



The command ***Transfer Selected Configuration to OPC UA Server*** command opens the ***Transfer Configuration to Server*** dialog box.

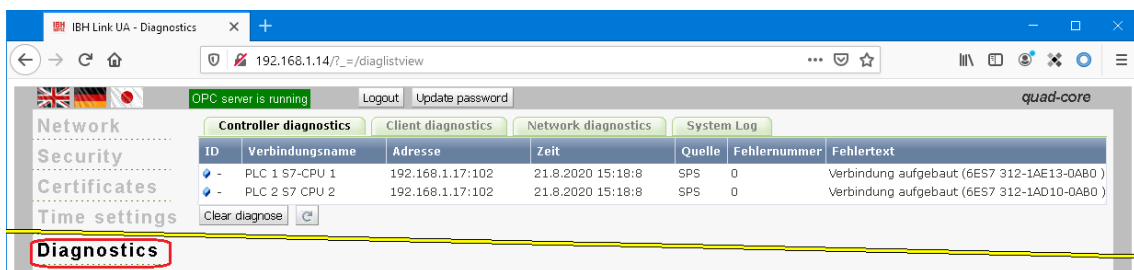
Select the server **IBH Link UA** and then click Start. The configuration is transferred to the **IBH Link UA**.



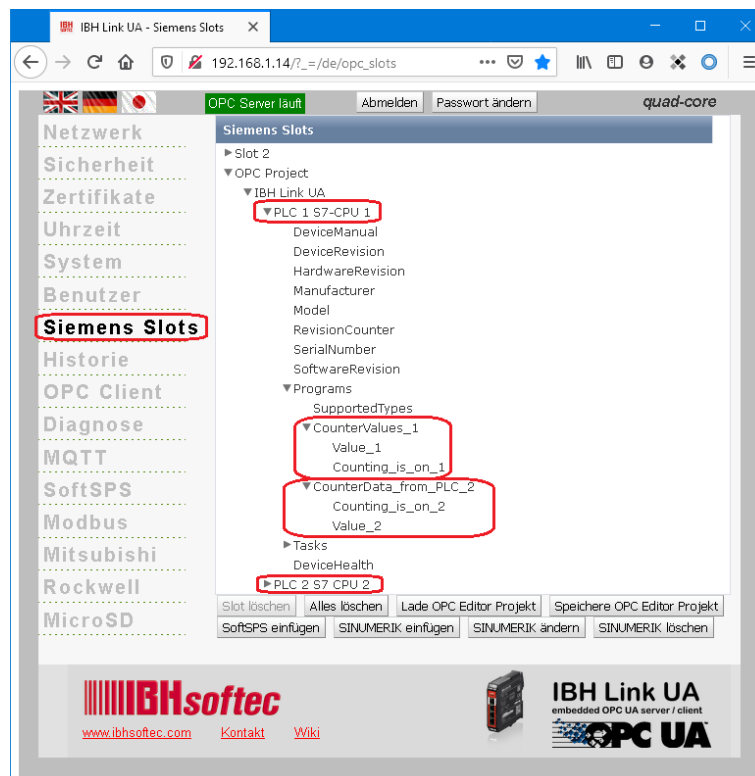
The successful transfer is displayed.

## 2.8.9 IBH Link UA browser windows

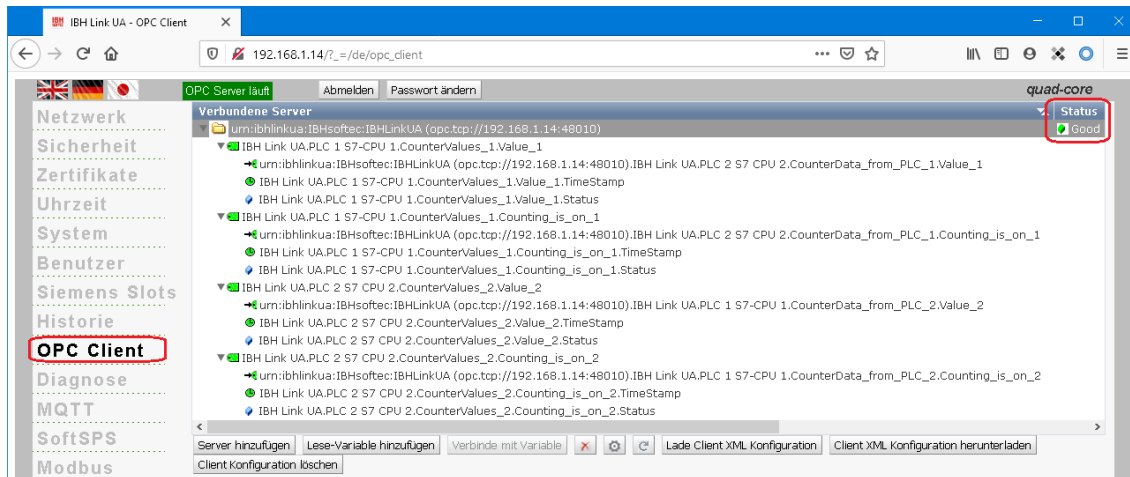
The browser window **Diagnostics** displays the status of the connection **IBH Link UA – PLCs**.



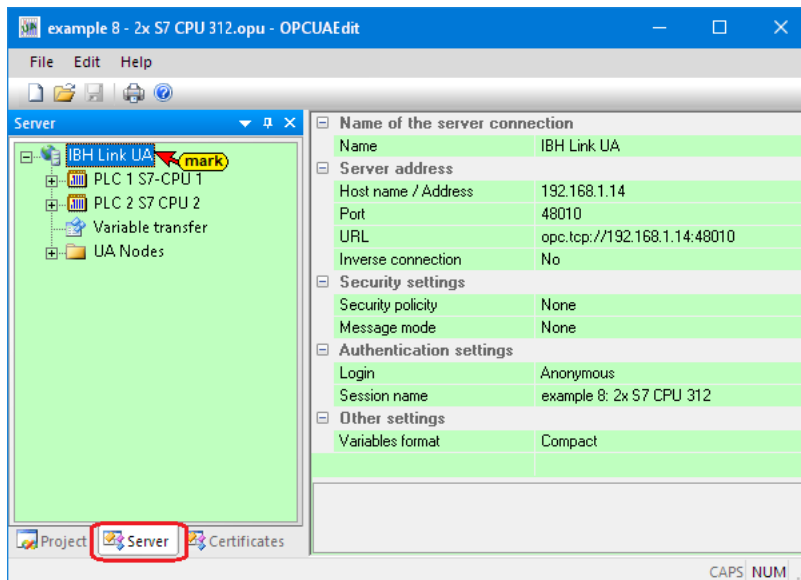
## IBH Link UA – Siemens Slots – OPC Project



## IBH Link UA – OPC Client



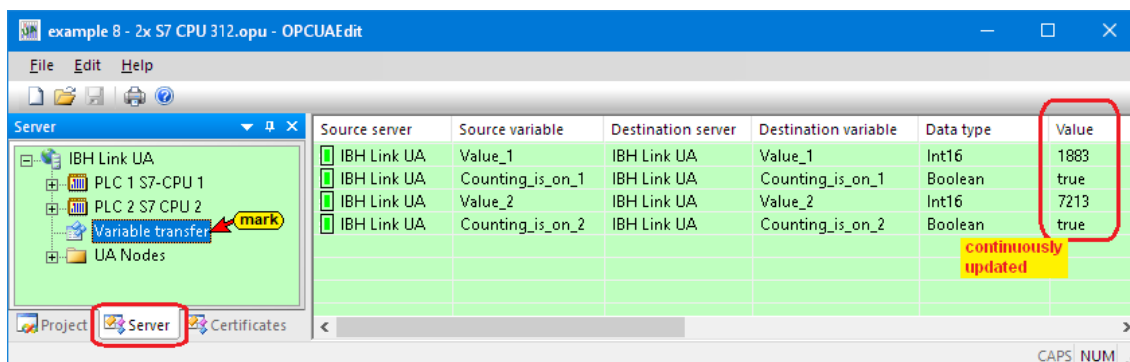
### 2.8.10 Online OPC UA Server Information



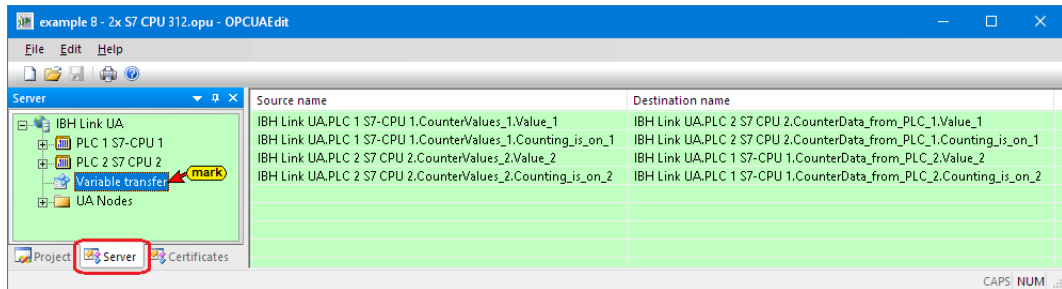
Information from the OPC UA server connected online with the CPUs are displayed.

### Show variable transfer

The individual OPC tags are displayed in the right server window with their status. The status of the OPC tags is updated continuously.

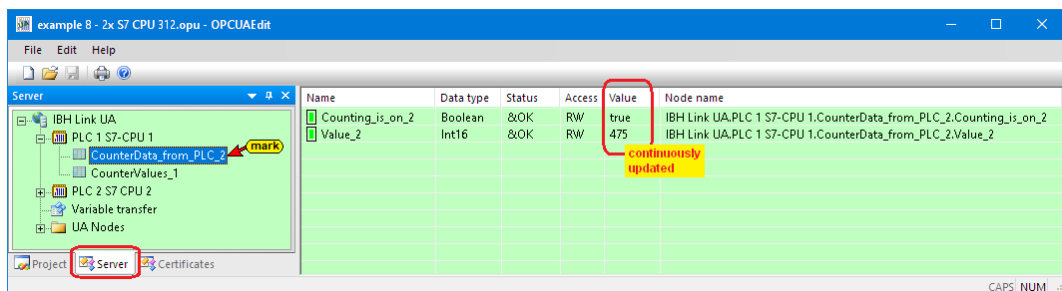


The source and destination names of the variables are displayed.

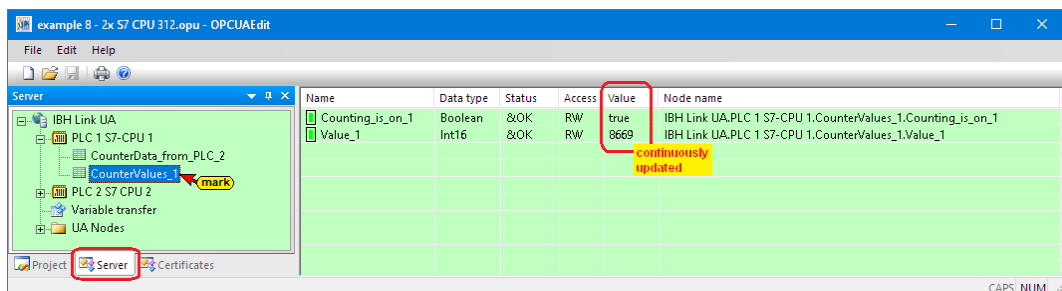


Marking the data block name, the individual variables (OPC tags) are displayed in the right server window with their status.

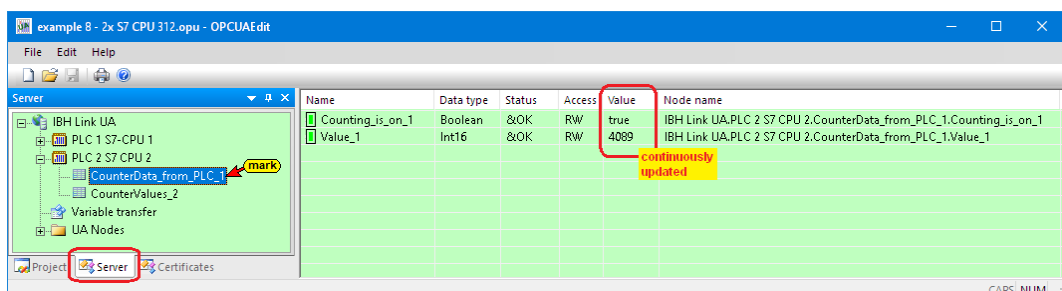
**PLC1 S7-CPU1; data block CounterData\_from\_PLC\_2 [DB5]**



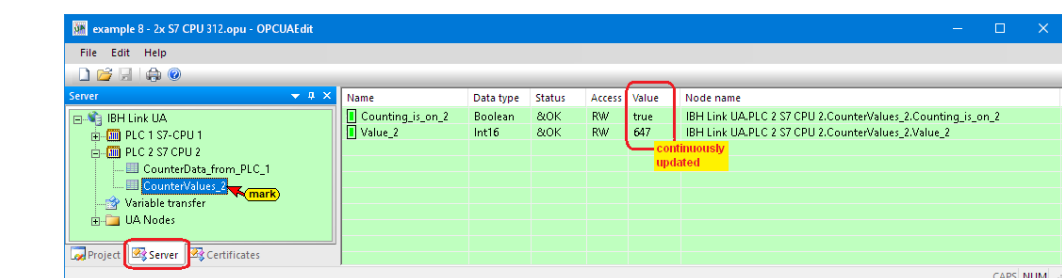
**PLC1 S7-CPU1; data block CounterValues\_1 [DB2]**



**PLC2 S7-CPU2; data block CounterData\_from\_PLC\_1 [DB5]**



**PLC1 S7-CPU1; data block CounterValues\_1 [DB2]**



### 2.8.11 Unified Automation UaExpert - The OPC Unified Architecture Client

The UaExpert program window lists the OPC tags transferred by the IBH OPC UA Editor and the associated UA nodes.

Use Drag & Drop to pull the OPC tags into the Data Access Viewer window.

The screenshot shows the Unified Automation UaExpert interface. The 'Data Access View' window is open, displaying a list of OPC tags. A red box highlights a specific tag with a value of 4643 and a 'continuously updated' status. Another red box highlights a table of tag values and timestamps. A third red box highlights a table of tag values, datatypes, and timestamps. A fourth red box highlights a table of tag values, datatypes, source timestamps, server timestamps, and status codes. A fifth red box highlights a table of tag values, servers, and node IDs.

#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	IBHLinkUA@ib...	NS4 String IBH ...	Counting_is_on_2	true	Boolean	18:04:17.096	18:04:18.111	Good
2	IBHLinkUA@ib...	NS4 String IBH ...	Value_2	4128	Int16	18:05:21.630	18:05:21.877	Good
3	IBHLinkUA@ib...	NS4 String IBH ...	Counting_is_on_1	true	Boolean	18:04:20.774	18:04:21.360	Good
4	IBHLinkUA@ib...	NS4 String IBH ...	Value_1	3743	Int16	18:05:21.627	18:05:21.877	Good
5	IBHLinkUA@ib...	NS4 String IBH ...	Counting_is_on_1	true	Boolean	18:04:32.007	18:04:32.867	Good
7	IBHLinkUA@ib...	NS4 String IBH ...	Value_1	3244	Int16	18:05:21.630	18:05:21.877	Good
8	IBHLinkUA@ib...	NS4 String IBH ...	Counting_is_on_2	true	Boolean	18:04:35.939	18:04:36.117	Good
				4643	Int16	18:05:21.702	18:05:21.877	Good

Display Name	Value	Datatype	Source Timestamp
Counting_is_on_2	true	Boolean	18:06:47.910
Value_2	6790	Int16	18:16:41.972
Counting_is_on_1	true	Boolean	18:06:47.910
Value_1	4785	Int16	18:16:42.154
Counting_is_on_1	true	Boolean	18:06:47.910
Value_1	6316	Int16	18:16:41.972
Counting_is_on_2	true	Boolean	18:06:47.910
Value_2	8668	Int16	18:16:42.154

Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
true	Boolean	18:06:47.910	18:06:48.160	Good
1232	Int16	18:21:35.843	18:21:36.028	Good
true	Boolean	18:06:47.910	18:06:48.160	Good
3386	Int16	18:21:36.028	18:21:36.279	Good
true	Boolean	18:06:47.910	18:06:48.160	Good
2380	Int16	18:21:35.843	18:21:36.028	Good
true	Boolean	18:06:47.910	18:06:48.160	Good
2479	Int16	18:21:36.028	18:21:36.279	Good

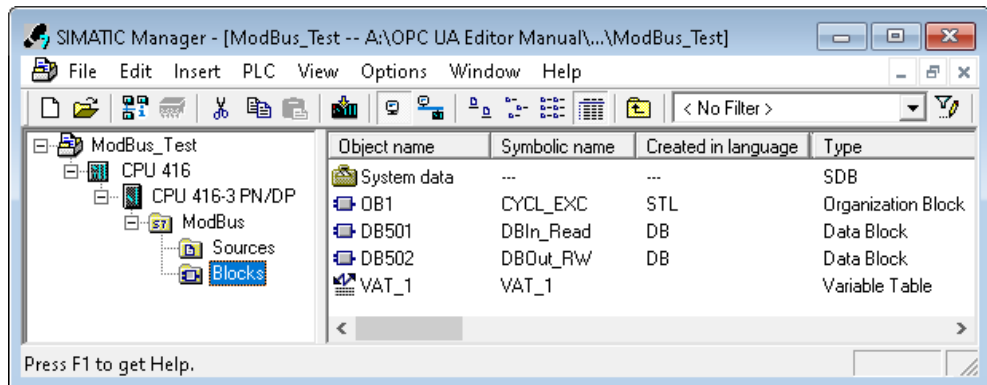
#	Server	Node Id
1	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.PLC 1 S7-CPU 1.CounterData_from_PLC_2.Counting_is_on_2
2	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.PLC 1 S7-CPU 1.CounterData_from_PLC_2.Value_2
3	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.PLC 1 S7-CPU 1.CounterValues_1.Counting_is_on_1
4	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.PLC 1 S7-CPU 1.CounterValues_1.Value_1
5	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.PLC 2 S7 CPU 2.CounterData_from_PLC_1.Counting_is_on_1
6	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.PLC 2 S7 CPU 2.CounterData_from_PLC_1.Value_1
7	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.PLC 2 S7 CPU 2.CounterValues_2.Counting_is_on_2
8	IBHLinkUA@ibhlinkua-SC-14	NS4 String IBH Link UA.PLC 2 S7 CPU 2.CounterValues_2.Value_2

## 2.9 Modbus connection - examples

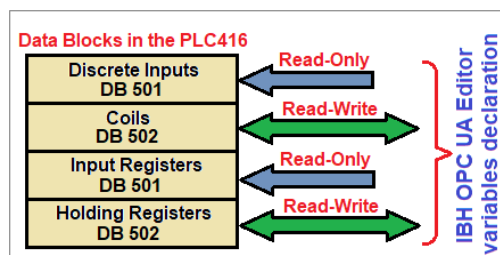
The IBH SoftPLC PLC416 has the option of a Modbus connection. In the example, variables are defined as OPC tags. This Modbus configuration is transmitted to the IBH Link UA and the variables are displayed in the *UaExpert client program*.



## SIMATIC Manager PLC project ModBus\_Test



The data blocks DB501 (**DBIn - Read**) and DB502 (**DBOut - Read / Write**) are created in the PLC416.



### Organization block OB1

In OB1, values are written to the data blocks DB501 (**DBIn\_Read**) and DB502 (**DBOut\_RW**) and reset to zero after a few seconds' delay.

**Example:** Modbus list used: - Register start addresses, access options to variables (fictitious device manufacturer information).

1 Bit	Register	Starting address					Address used				
		Hex	Dez	Bit			Hex	Dez	Bit	DB 501	DB 502
X <sup>(1)</sup>		0x000	0	0	Read /Write	Process data interface. Physical inputs Process image	0x00A	10	80		DBX 20.0
	X <sup>(2)</sup>	0x014	20	160	Read /Write		0x016	22	176		DBW44
	X <sup>(3)</sup>	0x034	52	416	Read only		0x034	59	464	DBX 118.0	
	X <sup>(4)</sup>	0x048	72	576	Read only	Process data interface. Physical inputs Process image	0x048	76	608	DBW 152	
	X <sup>(5)</sup>	0x068	104	832	Read /Write		0x068	112	896		DBD 224
	X <sup>(6)</sup>	0x07C	124	992	Read /Write		0x07C	126	1008		DBX 252.0
	X <sup>(7)</sup>	0x09C	156	1248	Read only	Status register	0x09C	160	1280	DBD 320	
	X <sup>(8)</sup>	0x0BC	188	1504	Read only	Process image length in bits, analog outputs	0x0BC	190	1520	DBW 380	
	X <sup>(9)</sup>	0x0FC	252	2016	Read/Write	Watchdog register	0x0FC	254	2032		DBD 508
	X <sup>(10)</sup>	0x10C	268	2144	Read/Write	Error register	0x10C	272	2176		DBD 544

(nn) Available as an example for the definition of a variable.

#### Note:

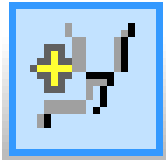
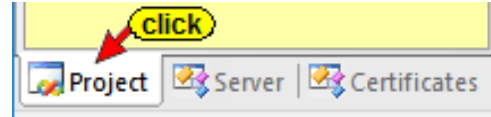
The addresses in the Modbus device manufacturer information are often in hexadecimal form. These addresses are to be converted into a decimal address for input in the IBH OPC UA editor.

### 2.9.1 Calling the IBH OPC UA Editor

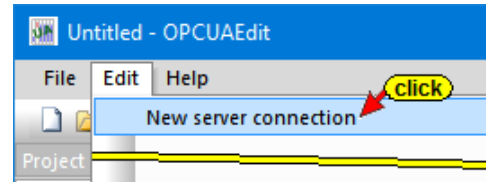
Double-click the *IBH OPC UA Editor* icon to open the program window.



Open the *Project window* by clicking on the *Project* tab.

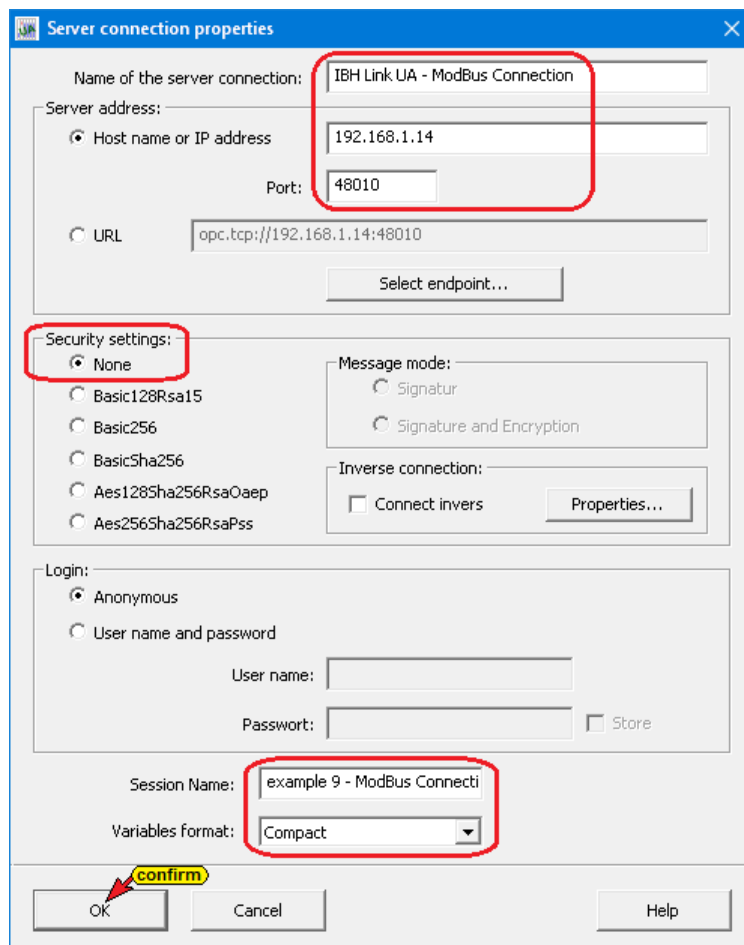


Open the *New Server Connection* dialog box with the New Server Connection command from the *Edit* menu or by clicking the icon.



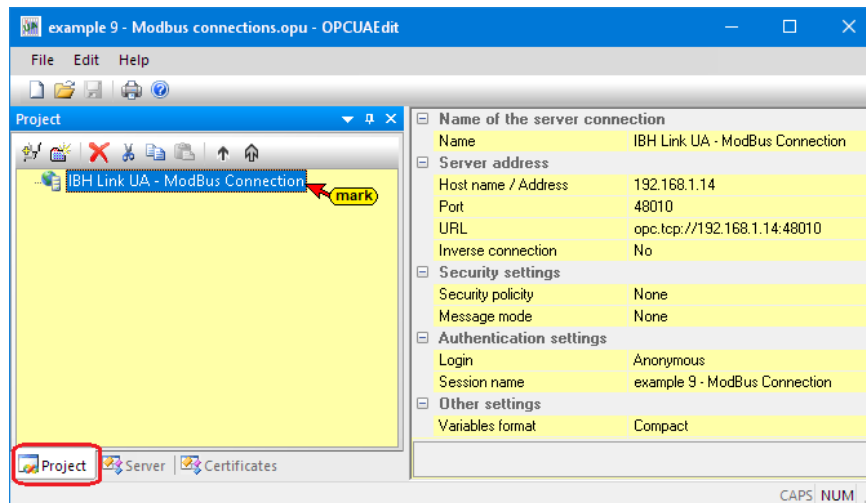
The new server connection setup was explained in example 1 (see chapter 2, page 2-3).

### Server Connection dialog box



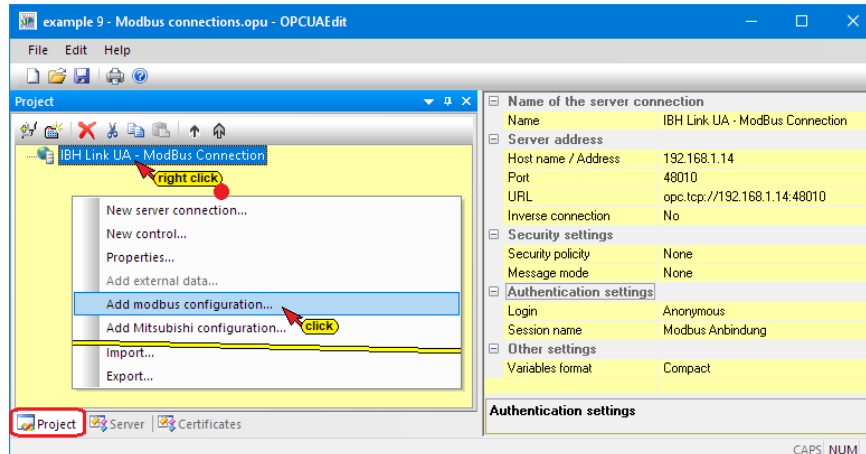
All necessary settings are shown. The possible encryptions of the data to be transmitted are displayed. The example is using the security method *None*. By clicking **OK** the content of the dialog box is saved and closed.

The settings for the connection to the **IBH Link UA** OPC UA server are displayed in the right part of the **project window**.

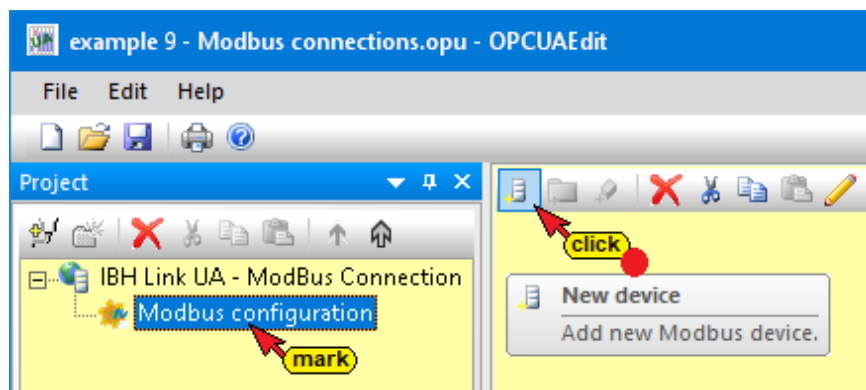


## 2.9.2 Modbus configuration

The OPC tags from the IBH UA editor program ModBus connection.opu can be used. Open the file with the IBH UA Editor and transfer the Modbus configuration to the IBH Link UA. The Modbus configuration can be created with the following steps.



## 2.9.3 Add new Modbus device



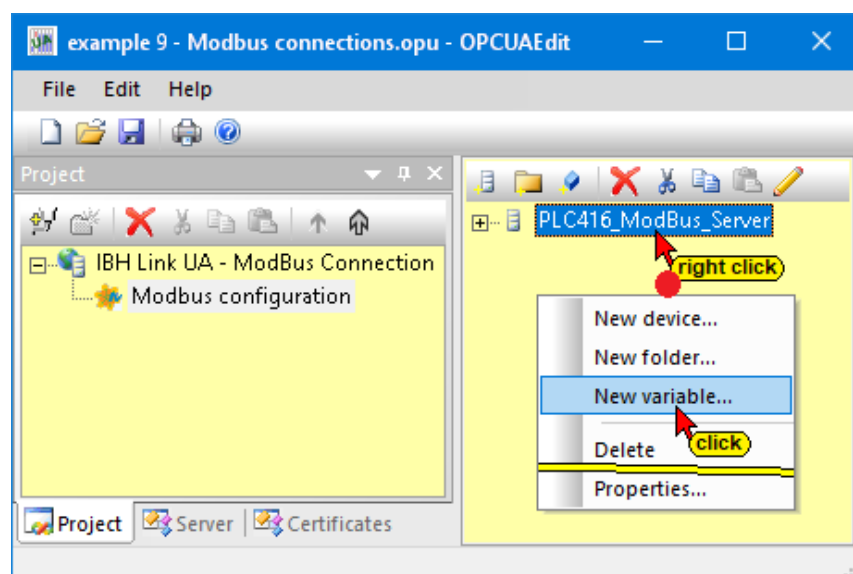
Mark Modbus configuration in the left part of the project window. Click the icon New device to open the Modbus device properties dialog box.



Specify device name and interface.

## 2.9.4 Define Modbus variable in the IBH OPC UA editor

Right-click on the Modbus device name (PLC416\_ModBus\_Server) to open the context menu.



The **New Variable** command opens the **Modbus Variable Properties** dialog box. Ten (10) OPC tags are defined for the connection with the OPC server.

## 1. Read\_Write\_Multiple\_Coils\_Boolean

Properties modbus variable

Name: 1\_Read\_Write\_Multiple\_Coils\_Boolean Unit ID: 255

Type:

- Boolean
- UInt16
- Int16
- UInt32
- Int32
- Float

Access:

Read:

- Read
- Address: 160
- ReadCoils
- ReadDiscreteInputs
- Number: 5

Write:

- Write
- Address: 160
- WriteSingleCoil

Analog limit values:

Check limit values Minimum: 0 Maximum: 0

History:

Save in history Sampling interval (sec): 0.5 Buffer size: 1000

UA node recognition: ns=9;s=CPU416\_Modbus\_Server.1\_Read\_Write\_Multiple\_Coils\_Boolean

OK Cancel Help

Read-Write, bit access, data type Boolean. Start word address 0014<sub>hex</sub> = 20<sub>dec</sub>; (DBX 20.0) Bit address 160<sub>dec</sub> = A0<sub>hex</sub>. Five (5) bits defined as an OPC tag.

## 2. Read\_Write\_Multiple\_Holding\_Register\_Int

Properties modbus variable

Name: 2\_Read\_Write\_Multiple\_Holding\_Register\_Int Unit ID: 255

Type:

- Boolean
- UInt16
- Int16
- UInt32
- Int32
- Float

Access:

Read:

- Read
- Address: 22
- ReadHoldingRegisters
- ReadInputRegisters
- Number: 3

Write:

- Write
- Address: 22
- WriteMultipleRegisters

Analog limit values:

Check limit values Minimum: 0 Maximum: 0

History:

Save in history Sampling interval (sec): 0.5 Buffer size: 1000

UA node recognition: ns=9;s=CPU416\_Modbus\_Server.2\_Read\_Write\_Multiple\_Holding\_Register\_Int

OK Cancel Help

Read-Write, data type INT16 fixed point number. Start word address, 22<sub>dec</sub> = 0016<sub>hex</sub>, word access. Three (3) fixed point numbers defined as an OPC tag.

## 3. Read\_Discrete\_Inputs\_Boolean

Properties modbus variable

Name: 3\_Read\_Discrete\_Inputs\_Boolean Unit ID: 255

Type:

- Boolean
- UInt16
- Int16
- UInt32
- Int32
- Float

Access:

Read:

- Read
- Address: 944
- ReadDiscreteInputs
- ReadCoils
- Number: 7

Write:

- Write
- Address:

Analog limit values:

Check limit values Minimum: 0 Maximum: 0

History:

Save in history Sampling interval (sec): 0.5 Buffer size: 1000

UA node recognition: ns=9;s=CPU416\_Modbus\_Server.3\_Read\_Discrete\_Inputs\_Boolean

OK Cancel Help

Read only, bit access, data type Boolean. Start word address 59<sub>dec</sub>, = 003B<sub>hex</sub>, Bit address 944<sub>dec</sub> = 03B0<sub>hex</sub> - bit access. Seven (7) bits defined as OPC tags.

#### 4. Read\_Input\_Registers\_Int16

Properties modbus variable

Name: 4\_Read\_Input\_Registers\_Int16 Unit ID: 255

Type:

- Boolean
- UInt16
- Int16
- UInt32
- Int32
- Float

Access:

Read:

- Read
- Address: 76
- ReadInputRegisters
- ReadHoldingRegisters
- Number: 4
- 32-bit swap word order

Write:

- Write
- Address:

Analog limit values:

Check limit values Minimum: 0 Maximum: 0

History:

Save in history Sampling interval (sec): 0.5 Buffer size: 1000

UA node recognition: ns=9;s=CPU416\_Modbus\_Server.4\_Read\_Input\_Registers\_Int16

confirm

OK Cancel Help

Read only, all data types except Boolean (Int16).

Start word address  $76_{\text{dez}} = 004C_{\text{hex}}$ , word access. Four (4) fixed point numbers defined as OPC tags.

#### 5. RW\_Holding\_Register\_Multiple\_Reg\_Real

Properties modbus variable

Name: 5\_RW\_Holding\_Register\_Multiple\_Reg\_Real Unit ID: 255

Type:

- Boolean
- UInt16
- Int16
- UInt32
- Int32
- Float

Access:

Read:

- Read
- Address: 111
- ReadInputRegisters
- ReadHoldingRegisters
- Number: 3
- 32-bit swap word order

Write:

- Write
- Address: 111
- WriteMultipleRegisters

Analog limit values:

Check limit values Minimum: 0 Maximum: 0

History:

Save in history Sampling interval (sec): 0.5 Buffer size: 1000

UA node recognition: ns=9;s=CPU416\_Modbus\_Server.5\_RW\_Holding\_Register\_Multiple\_Reg\_Real

confirm

OK Cancel Help

Read-Write, data type INT16 fixed point number.

Start word address  $111_{\text{dez}} = 006F_{\text{hex}}$ , word access. Three (3) floating point numbers (float) defined as OPC tag.

#### 6. Read\_Coils\_Boolean

Properties modbus variable

Name: 6\_Read\_Coils\_Boolean Unit ID: 255

Type:

- Boolean
- UInt16
- Int16
- UInt32
- Int32
- Float

Access:

Read:

- Read
- Address: 2016
- ReadCoils
- ReadDiscreteInputs
- Number: 8
- 32-bit swap word order

Write:

- Write
- Address:

Analog limit values:

Check limit values Minimum: 0 Maximum: 0

History:

Save in history Sampling interval (sec): 0.5 Buffer size: 1000

UA node recognition: ns=9;s=CPU416\_Modbus\_Server.6\_Read\_Coils\_Boolean

confirm

OK Cancel Help

Read only, bit access, data type Boolean.

Start bit address  $2016_{\text{dec}} = 07E0_{\text{hex}}$ . Eight (8) bits defined as OPC tags.

## 7. Read\_Input\_Registers\_Int32

Properties modbus variable

Name: 7\_Read\_Input\_Registers\_Int32 Unit ID: 255

Type:

- Boolean
- UInt16
- Int16
- UInt32
- Int32
- Float

Access:

Read:

- Read
- Address: 160
- ReadInputRegisters
- ReadHoldingRegisters
- Number: 6

Write:

- Write
- Address:

Analog limit values:

Check limit values Minimum: 0 Maximum: 0

History:

Save in history Sampling interval (sec): 0.5 Buffer size: 1000

UA node recognition: ns=9;s=CPU416\_Modbus\_Server.7\_Read\_Input\_Registers\_Int32

**confirm**

OK Cancel Help

Read only, all data types except Boolean (Int32 - fixed point number).

Start word address 160<sub>dez</sub> = 00A0<sub>hex</sub>, word access. Six (6) fixed point numbers defined as OPC tags.

## 8. Read\_Input\_Registers\_UInt

Properties modbus variable

Name: 8\_Read\_Input\_Registers\_UInt Unit ID: 255

Type:

- Boolean
- UInt16
- Int16
- UInt32
- Int32
- Float

Access:

Read:

- Read
- Address: 190
- ReadInputRegisters
- ReadHoldingRegisters
- Number: 6

Write:

- Write
- Address:

Analog limit values:

Check limit values Minimum: 0 Maximum: 0

History:

Save in history Sampling interval (sec): 0.5 Buffer size: 1000

UA node recognition: ns=9;s=CPU416\_Modbus\_Server.8\_Read\_Input\_Registers\_UInt

**confirm**

OK Cancel Help

Read only, all data types except Boolean (UInt16 - unsigned fixed point number), Start word address 190<sub>dez</sub> = 00BE<sub>hex</sub>, word access. Six (6) unsigned fixed point numbers defined as OPC tags.

## 9. RW\_Holding\_Register\_Multiple\_Reg\_Int32

Properties modbus variable

Name: 9\_RW\_Holding\_Register\_Multiple\_Reg\_Int32 Unit ID: 255

Type:

- Boolean
- UInt16
- Int16
- UInt32
- Int32
- Float

Access:

Read:

- Read
- Address: 254
- ReadInputRegisters
- ReadHoldingRegisters
- Number: 3

Write:

- Write
- Address: 254
- WriteMultipleRegisters

Analog limit values:

Check limit values Minimum: 0 Maximum: 0

History:

Save in history Sampling interval (sec): 0.5 Buffer size: 1000

UA node recognition: ns=9;s=CPU416\_Modbus\_Server.9\_RW\_Holding\_Register\_Multiple\_Reg\_Int32

**confirm**

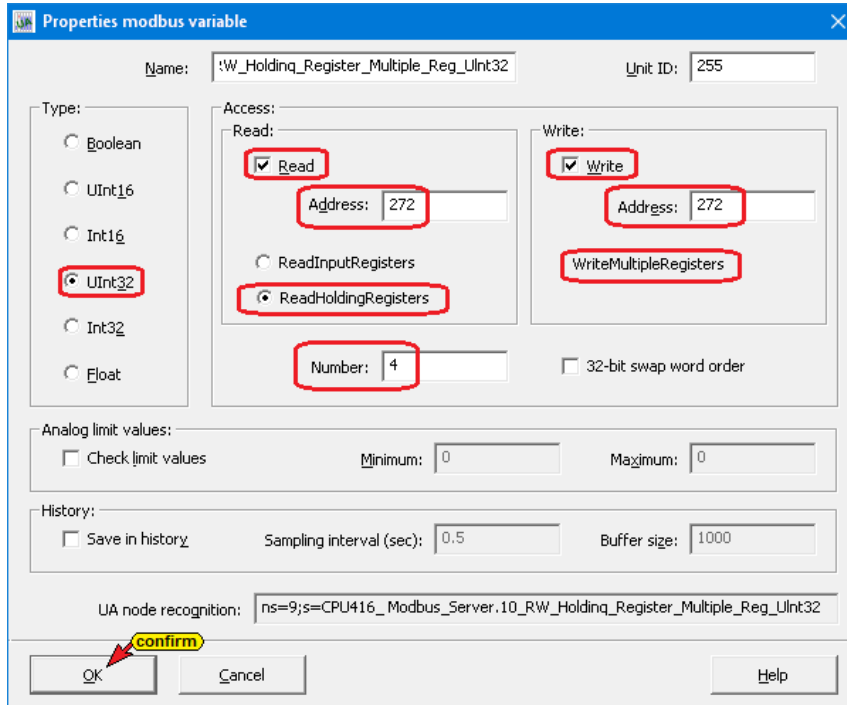
OK Cancel Help

Read-Write, all data types except Boolean (Int32 - fixed point number).

Start word address 254<sub>dez</sub> = 001FC<sub>hex</sub>, word access. Six (3) fixed point numbers defined as OPC tags.

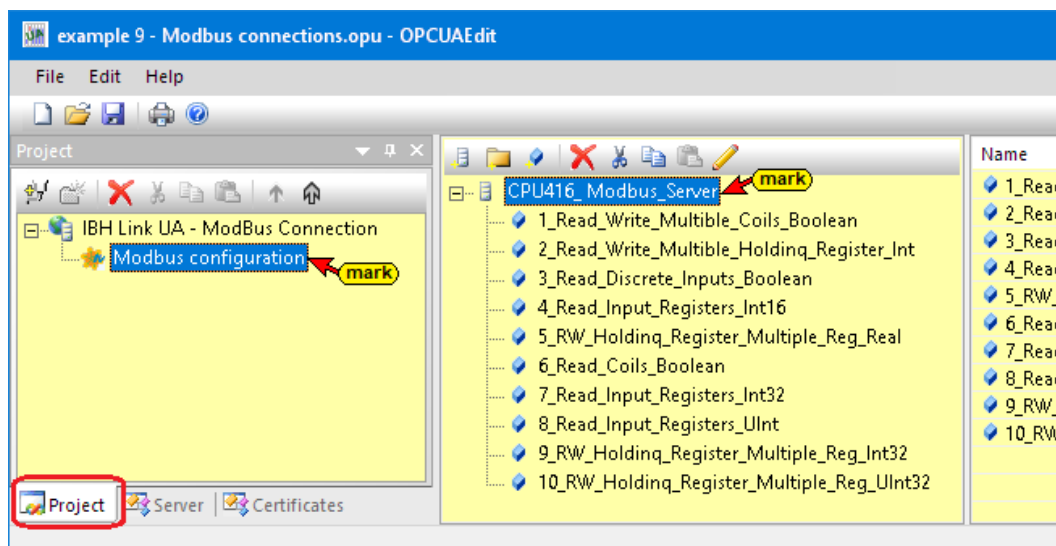
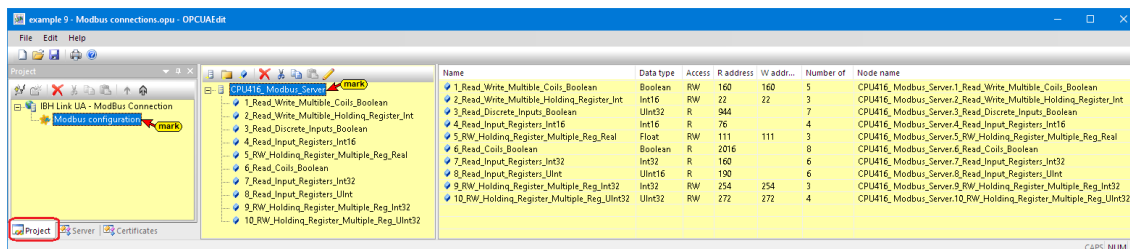
### 10.RW\_Holding\_Register\_Multiple\_Reg\_UInt32

Read-Write, all data types except Boolean (UInt32 - unsigned fixed point number). Start word address 272dec = 00110hex, word access. Four (4) unsigned fixed point numbers defined as OPC tags.



### Defines OPC variables (OPC tags)

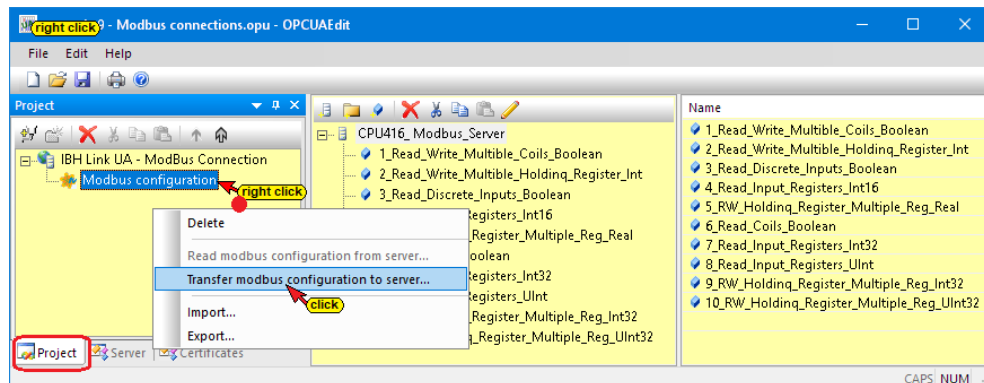
Ten (10) OPC variables (OPC tags) are defined with the Modbus *Variable Properties* dialog box.



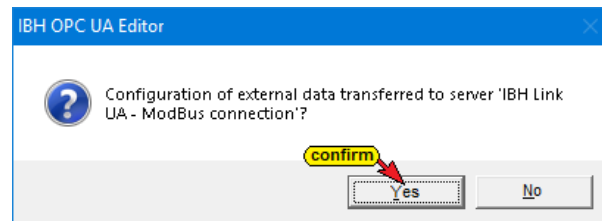


## 2.9.5 Transfer Modbus configuration to the IBH Link UA.

A right-click on *Modbus configuration* opens the context menu.



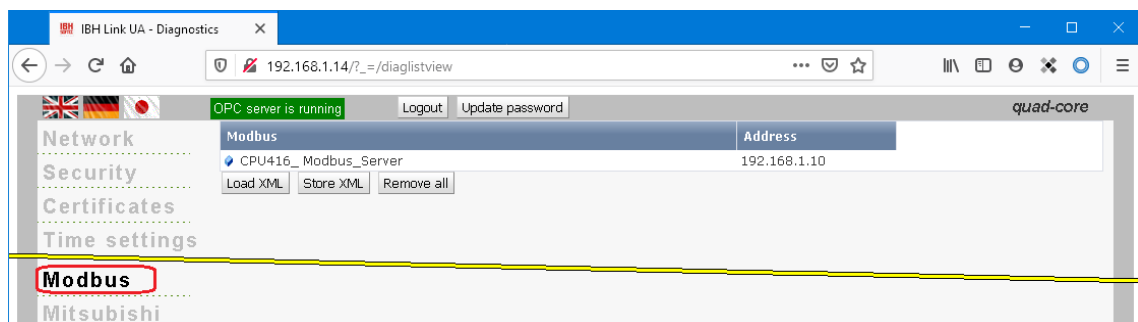
The command to transfer the Modbus configuration must be confirmed.



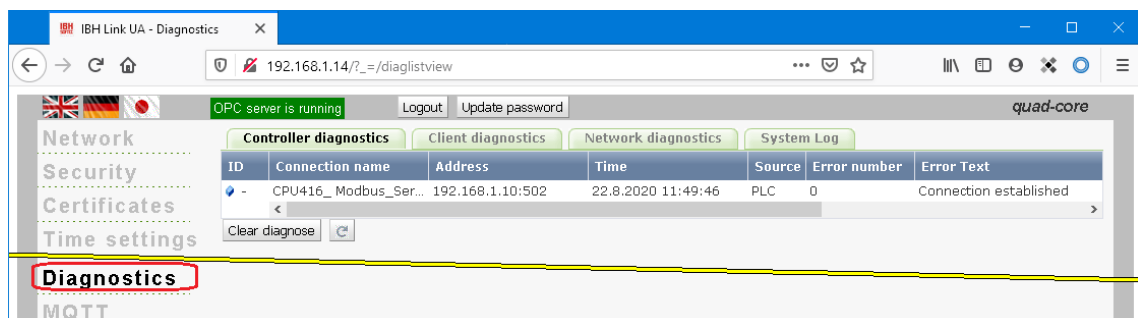
## 2.9.6 IBH Link UA browser window

The transmitted Modbus configuration is displayed under *Modbus* in the IBH Link UA Browser window.

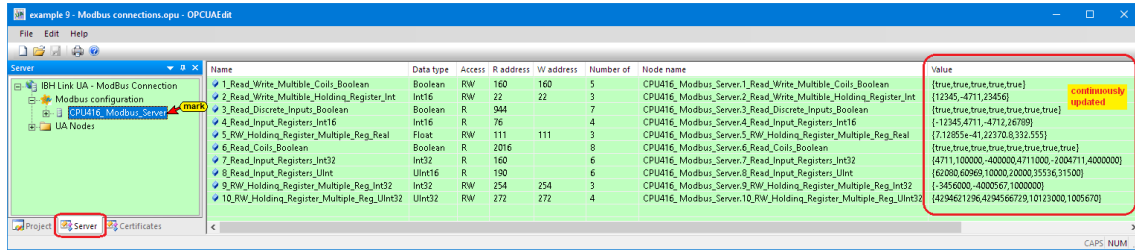
### Browser-Fenster-ModBus



### Browser window diagnostics



## 2.9.7 IBH OPC UA Editor Server Information - Online



Name	Data type	Access	R address	W address	Number of
1_Read_Write_Multiple_Coils_Boolean	Boolean	RW	160	160	5
2_Read_Write_Multiple_Holding_Register_Int	Int16	RW	22	22	3
3_Read_Discrete_Inputs_Boolean	Boolean	R	944		7
4_Read_Input_Registers_Int16	Int16	R	76		4
5_RW_Holding_Register_Multiple_Reg_Real	Float	RW	111	111	3
6_Read_Coils_Boolean	Boolean	R	2016		8
7_Read_Input_Registers_Int32	Int32	R	160		6
8_Read_Input_Registers_UInt	UInt16	R	190		6
9_RW_Holding_Register_Multiple_Reg_Int32	Int32	RW	254	254	3
10_RW_Holding_Register_Multiple_Reg_UInt32	UInt32	RW	272	272	4

Node name	Value
CPU416_Modbus_Server.1_Read_Write_Multiple_Coils_Boolean	{true,true,true,true,true}
CPU416_Modbus_Server.2_Read_Write_Multiple_Holding_Register_Int	{12345,-4711,23456}
CPU416_Modbus_Server.3_Read_Discrete_Inputs_Boolean	{true,true,true,true,true,true}
CPU416_Modbus_Server.4_Read_Input_Registers_Int16	{-12345,4711,-4712,26789}
CPU416_Modbus_Server.5_RW_Holding_Register_Multiple_Reg_Real	{7.12855e-41,22370.8,332.555}
CPU416_Modbus_Server.6_Read_Coils_Boolean	{true,true,true,true,true,true,true}
CPU416_Modbus_Server.7_Read_Input_Registers_Int32	{4711,100000,-400000,4711000,-2004711,4000000}
CPU416_Modbus_Server.8_Read_Input_Registers_UInt	{62080,60969,10000,20000,35536,31500}
CPU416_Modbus_Server.9_RW_Holding_Register_Multiple_Reg_Int32	{-3456000,-4000567,1000000}
CPU416_Modbus_Server.10_RW_Holding_Register_Multiple_Reg_UInt32	{4294621296,4294566729,10123000,1005670}

## 2.9.8 UaExpert – The OPC Unified Architecture Client

