
Despite the care taken in the preparation of this document, NUM cannot guarantee the accuracy of the information it contains and cannot be held responsible for any errors therein, nor for any damage which might result from the use or application of the document.

The physical, technical and functional characteristics of the hardware and software products and the services described in this document are subject to modification and cannot under any circumstances be regarded as contractual.

The programming examples described in this manual are intended for guidance only. They must be specially adapted before they can be used in programs with an industrial application, according to the automated system used and the safety levels required.

All rights reserved. No part of this manual may be copied or reproduced in any form or by any means whatsoever, including photographic or magnetic processes. The transcription on an electronic machine of all or part of the contents is forbidden.

© Copyright NUM 2023 software

This software is the property of NUM. Each memorized copy of this software sold confers upon the purchaser a non-exclusive licence strictly limited to the use of the said copy. No copy or other form of duplication of this product is authorized.

Summary

1	Foreword	9
1.1	Use of the manual	9
1.2	Document revisions	9
1.3	How to read the manual	10
1.4	NUM Agencies	11
2	NUM IloTgateway	15
2.1	System overview	15
2.2	General notes	15
2.3	Appearance in Windows	16
2.3.1	Windows Start Menu	16
2.3.2	Program States	17
2.4	NUM IloTgateway UI	20
2.4.1	Protocols	20
2.4.2	Control Parameters	23
2.4.3	Machine information	24
2.4.4	PLC Variables	25
2.4.5	E Parameters	27
2.4.6	Import/Export	28
2.5	OPC UA NUM's Nodeset	30
2.5.1	NUM IloTgateway Architecture	30
2.5.2	Server Endpoints	30
2.5.3	Sign of Life	31
2.5.4	Machine Information	31
2.5.5	Machine State	31
2.5.6	NrProdState	33
2.5.7	NrToolInfo	34
2.6	OPC UA umati	36
2.6.1	NUM IloTgateway Architecture	36
2.6.2	End points	36
2.6.3	Equipment	36
2.6.4	Identification	36
2.6.5	Monitoring	36
2.6.6	Notification	38
2.6.7	Production	38
2.7	MQTT	39
2.7.1	NUM IloTgateway Architecture	39
2.7.2	Topics and values	39
2.7.3	Sign of Life	40
2.7.4	Machine Information	40
2.7.5	Machine State	40
2.7.6	NUMroto	42
2.7.7	NrToolInfo	43
2.8	MTConnect	45
2.8.1	NUM IloTgateway Architecture	45
2.8.2	Introduction	45

2.8.3	General Data	47
2.8.4	Machine Information	47
2.8.5	Machine State	48
2.8.6	NrProdState	50
2.8.7	NrToolInfo	50
2.9	Troubleshooting / Diagnostics	52
2.10	NUM IloTgateway Installation	53
2.11	Windows Firewall Settings	59
2.11.1	Windows Firewall PC Settings	59
2.12	Test Client Settings	65
2.12.1	OPC UA Client Settings	65
2.12.2	MQTT Client Settings	68
2.12.3	MTConnect Client Settings	70
2.12.4	Monitoring of Communication	71
3	MQgateway	75
3.1	System overview	75
3.2	General notes	75
3.3	Program states	76
3.4	Settings	77
3.4.1	General settings	77
3.4.2	Machine information	78
3.4.3	PLC variables	78
3.4.4	E-Parameter variables	79
3.5	Topics and values	80
3.5.1	Lifecycle topics	80
3.6	Machine information	81
3.7	Machine State	82
3.8	NUMroto	84
3.9	Troubleshooting / Diagnostics	85
4	MTConnect	89
4.1	MTConnect introduction	89
4.2	NUM MDTO Software	90
4.3	NUM MDTO Adapter and Agent	91
4.4	NUM MDTO Standard Data Items	92
4.5	NUM MDTO Extended Data Items	93
4.6	NUM MDTO Installation	94
4.6.1	NUM MDTO Condition	94
4.6.1.1	Flexium and Flexium+ CNC series	94
4.6.1.2	Axiom Power CNC series	94
4.6.1.3	Microsoft .NET Framework installed	94
4.6.2	Installation NUM MDTO	95
4.6.3	NUM MDTO Installation Location	97
4.7	NUM MDTO Adapter Configuration	98
4.7.1	Configuration file NUMAdp.ini	98
4.7.2	NUM MDTO Activation	98
4.7.2.1	Flexium/Flexium+ CNC activation	98
4.7.2.2	Axiom Power/NUM Power CNC activation	98
4.7.3	Configuration NUMAdp.ini SYSTEM	99
4.7.3.1	Configuration NUMAdp.ini CNC	101
4.7.3.2	Configuration NUMAdp.ini Axes and Spindles	102
4.7.3.3	Configuration NUMAdp.ini Custom_PLC_Variables	103

4.7.3.4	Configuration NUMAdp.ini ProcessState	104
4.7.3.5	Configuration NUMAdp.ini NUMROTO_XML.....	106
4.7.3.6	Configuration NUMAdp.ini ProcessState version.....	107
4.7.3.7	Configuration NUMAdp.ini NUMROTO_XML version.....	109
4.7.3.8	Required NUMROTO version	110
4.7.3.9	Sample NUMAdp.ini configuration for Flexium CNC series	111
4.8	NUM MDTO Agent Configuration	114
4.8.1	Agent.cfg configuration	114
4.8.2	Devices.xml configuration.....	115
4.8.3	Structure NUM CNC Devices.xml	117
4.8.4	Customization NUM CNC Devices.xml	118
4.9	NUM MDTO Diagnostic	120
4.9.1	NUM MDTO Adapter Diagnostic.....	120
4.9.2	NUM MDTO Agent Diagnostic.....	122
4.10	Common NUM MDTO Network Configuration	125
4.10.1	Windows Firewall	126
5	OPC UA Server	131
5.1	Creating a project for OPC UA access	131
5.2	Creating a certificate for the Flexium OPC UA server.....	132
5.3	Setting up an encrypted connection with the “UaExpert” client	133
5.4	User management in OPC UA.....	134
5.5	Changing variables via the OPC UA client	135
5.6	Creating events in the Flexium project	136
5.7	Monitoring of events with the OPC UA client “UaExpert”	137
6	OPC DA.....	141
6.1	Overview on OPC DA	141
6.2	Overview on Flexium OPC DA Server.....	142
6.2.1	CODESYS Gateway V3	143
6.3	Installation of the Flexium OPC Server V3	145
6.3.1	Installation and Registration	145
6.3.2	Usage of the Flexium OPC DA Server	147
6.4	Configuration of the symbols in the programming system	149
6.4.1	V3 Interface	149
6.5	CODESYS OPC Server Configuration by OPCConfig.exe	151
6.5.1	Call OPCConfig.exe.....	152
6.5.1.1	Default call	152
6.5.1.2	Open a particular INI file.....	152
6.5.2	Menu Commands and Configuration Dialogs.....	153
6.5.2.1	File	153
6.5.2.2	Edit	153
6.5.2.3	Settings for OPC server	154
6.5.2.4	Settings for <PLC>	155
6.5.2.5	Settings for connection to <PLC>	157
6.5.3	Simulation Interface	159
6.5.3.1	SIMULATION3	159
6.5.4	Redundant Group	160
6.5.5	Configuration of OPCConfig with OPCConfig.ini	162
6.6	Configuration settings in OPCServer.ini	163
6.6.1	Configuration of the OPC server under section [Server]	163
6.6.2	Configuration of the communication to the controllers	164
6.6.3	Example of an INI file for the V3 interface.....	165
6.6.4	Configuration of ‘Redundant Group’	166
6.6.5	Information on the changed logfilter	168

6.6.6	Configuration of simulation	169
6.6.7	Changing directory and name of INI and LOG file.....	170
6.7	Connection to OPC servers on other computers with DCOMCNFG.EXE	171
6.8	Error Diagnosis	172
6.8.1	Means of error diagnosis	172
6.8.2	Brief checklist.....	172
6.8.3	Error cases	173

Summary

1	Foreword	9
1.1	Use of the manual	9
1.2	Document revisions	9
1.3	How to read the manual	10
1.4	NUM Agencies	11

1

2

3

4

5

6

Page deliberately empty

1 Foreword

1.1 Use of the manual

This manual is for use by qualified personnel only.

To reduce the risk due to an improper use of the product, do not perform any servicing other than that contained in this manual unless you are qualified to do so.

To ensure proper use of these products, please always read this User's manual carefully and retain for future reference. Should the unit require maintenance, please contact a NUM authorized service center only.

1.2 Document revisions

Date	Index	Reason for revision	See chapter
4 - 2019	00	Document creation	-
5 - 2023	01	New chapter "NUM IloTgateway" added	2
12 - 2024	02	NUM IloTgateway chapter update Updating images and tables	2

1

2

3

4

5

6

1.3 How to read the manual

This manual gives information about the NUM Industry 4.0 solution.

Chapter	Description
1	Foreword This chapter introduces: The use of the manual.
2	NUM IloTgateway This chapter introduces: - System overview; - NUM IloTgateway UI and appearance - OPC UA, MQTT, MTConnect - Troubleshooting / Diagnostics; - Installation; - Test client Settings;
3	MQgateway This chapter introduces: - System overview; - Program states; - Settings; - Topics and values and Machine State.
4	MT Connect This chapter introduces: - Installation; - System Configuration; - Testing NUM MDTO; - Diagnostics.
5	OPC UA Server This chapter introduces: - Creating a project for OPC UA access; - Creating a certificate for the CODESYS OPC UA server; - Setting up an encrypted connection with the "UaExpert" client; - User management in OPC UA; - Changing variables via the OPC UA client; - Creating events in the CODESYS project; - Monitoring of events with the OPC UA client "UaExpert".
6	OPC DA This chapter introduces: - Overview on OPC DA; - Overview on CODESYS OPC Server V3; - Installation of the CODESYS OPC Server V3; - Configuration of the symbols in the programming system; - CODESYS OPC Server Configuration by OPCConfig.exe; - Configuration settings in OPCServer.ini; - Connection to OPC servers on other computers with DCOMCNFG.EXE; - Error Diagnosis.

1.4 NUM Agencies

The list of NUM agencies is given click here <http://www.num.com>

1

2

3

4

5

6

Page deliberately empty

Summary

2	NUM IloTgateway	15
2.1	System overview	15
2.2	General notes.....	15
2.3	Appearance in Windows	16
2.3.1	Windows Start Menu.....	16
2.3.2	Program States	17
2.4	NUM IloTgateway UI	20
2.4.1	Protocols	20
2.4.2	Control Parameters	23
2.4.3	Machine information.....	24
2.4.4	PLC Variables	25
2.4.5	E Parameters	27
2.4.6	Import/Export.....	28
2.5	OPC UA NUM's Nodeset	30
2.5.1	NUM IloTgateway Architecture.....	30
2.5.2	Server Endpoints	30
2.5.3	Sign of Life	31
2.5.4	Machine Information	31
2.5.5	Machine State	31
2.5.6	NrProdState	33
2.5.7	NrToolInfo	34
2.6	OPC UA umati.....	36
2.6.1	NUM IloTgateway Architecture.....	36
2.6.2	End points	36
2.6.3	Equipment	36
2.6.4	Identification.....	36
2.6.5	Monitoring	36
2.6.6	Notification	38
2.6.7	Production.....	38
2.7	MQTT	39
2.7.1	NUM IloTgateway Architecture.....	39
2.7.2	Topics and values	39
2.7.3	Sign of Life	40
2.7.4	Machine Information	40
2.7.5	Machine State	40
2.7.6	NUMroto.....	42
2.7.7	NrToolInfo	43
2.8	MTConnect.....	45
2.8.1	NUM IloTgateway Architecture.....	45
2.8.2	Introduction	45
2.8.3	General Data.....	47
2.8.4	Machine Information	47
2.8.5	Machine State	48
2.8.6	NrProdState	50
2.8.7	NrToolInfo	50
2.9	Troubleshooting / Diagnostics.....	52

2.10	NUM IIoTgateway Installation.....	53
2.11	Windows Firewall Settings	59
2.11.1	Windows Firewall PC Settings	59
2.12	Test Client Settings	65
2.12.1	OPC UA Client Settings	65
2.12.2	MQTT Client Settings.....	68
2.12.3	MTConnect Client Settings	70
2.12.4	Monitoring of Communication	71

2 NUM IIoTgateway

2.1 System overview

The main goal of NUM IIoTgateway is to offer data of a NCK or from NUMroto. Receiver can get the information via OPC UA, MQTT or MTConnect.

Participating in this data chain are the members: NCK, IPC / Box PC, APServer / FXServer, and a message broker in case of MQTT.

The NCK and NUMroto are the source of most of the published data. Currently supported are only Flexium Systems above version 3.6.0.0 and Axiom Power above version R2 with Windows 7.

The industrial PC (IPC) or Box PC is the host of the NUM IIoTgateway. Since the APServer / FX-Server currently runs only on Windows, the NUM IIoTgateway runs only on the respective PC, too.

NUM IIoTgateway is a client, which collects all data from different sources. Most of the data are continuous updating while machining, but there are some static fields, which are updated only once.

NUM IIoTgateway is also available in a trial version. It works without a license and allows you to test the product for one hour.

2.2 General notes

APServer is a basic requirement for Axiom Power or Num Power systems and FXServer for Flexium or Flexium* systems. They have to be installed on the same PC as NUM IIoTgateway.



NUM IIoTgateway can be applied to the NUM CNC Series Axiom Power/ Num Power only with a TCP/IP Ethernet communication connection. Without this requirements met, NUM IIoTgateway is not able to offer any data.

The NUM IIoTgateway application runs as a Windows application and is minimized on the tray icon bar. A settings menu is available from context menu accessible through a right click on the icon. This menu should be available whether APServer / FXServer are installed.

If APServer / FXServer are installed and the license is available, NUM IIoTgateway starts sending data.



The data from NUMROTO is only available if the NUMROTO option 'Interface for machine state information (Industry 4.0) [CH-50052026]' is licensed for the corresponding machine.

1

2

3

4

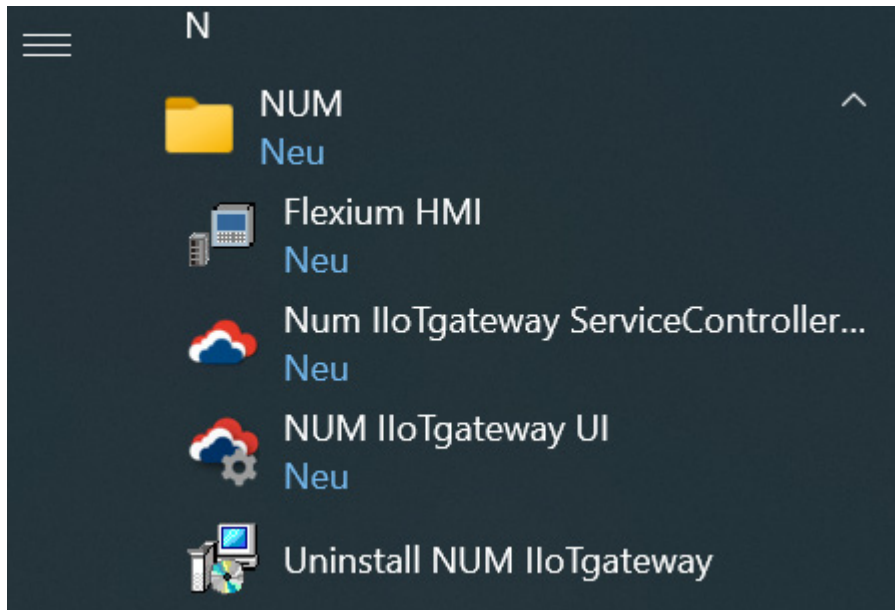
5

6

2.3 Appearance in Windows

2.3.1 Windows Start Menu

After installing NUM IloTgateway, the NUM IloTgateway tools are displayed in the Windows Start Menu NUM for selection.



NUM IloTgateway ServiceControllerTray

Clicking on this menu point starts NUM IloTgateway Services and displays the ServiceControllerTray icon.



NUM IloTgateway UI

Clicking on this menu point starts NUM IloTgateway UI Configuration Tool (see section [2.4](#)).

Uninstall NUM IloTgateway

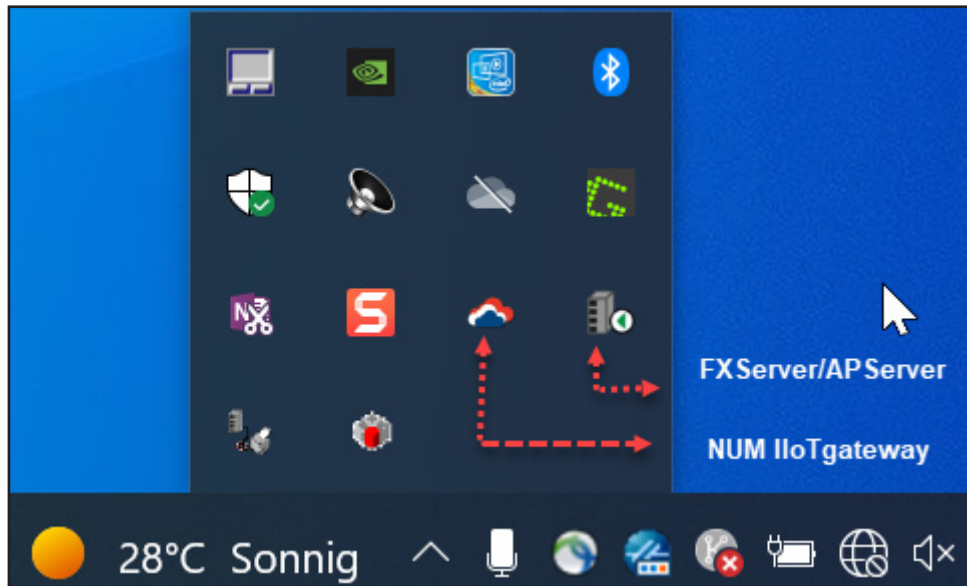
Clicking on this menu point uninstalls the NUM IloTgateway Software.

2.3.2 Program States

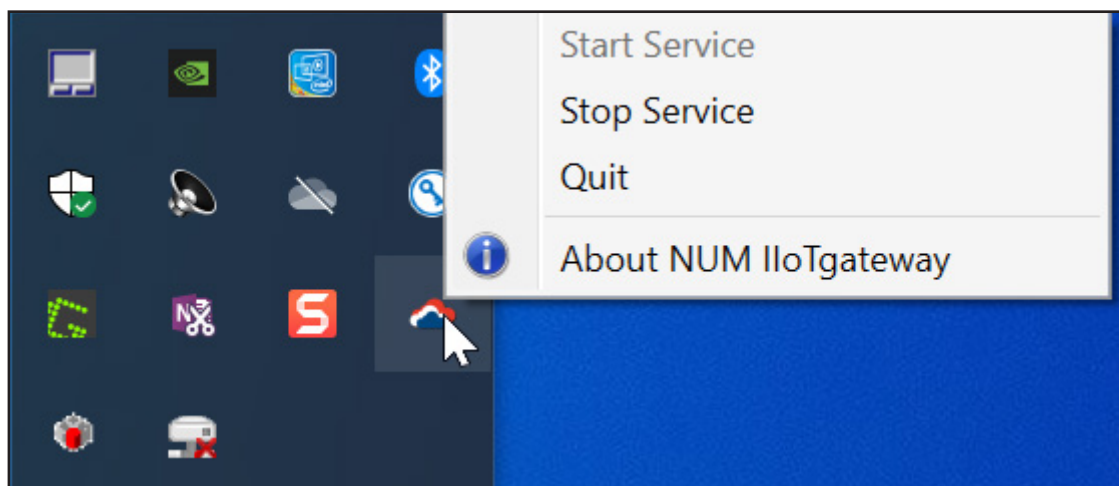
After the installation of NUM IloTgateway, the NUM IloTgateway SysTray icon for status monitoring is displayed in the Windows taskbar.

The NUM IloTgateway Services and the SysTray icon are started with every PC restart.

The NUM IloTgateway Service starts the communication Server FXServer (Flexium, Flexium+) or APServer (Axiom Power, Num Power) for the communication with the Flexium / Flexium+ or Axiom Power / Num Power for the acquisition of the data points of the CNC.



With a right click on the NUM IloTgateway SysTray icon, the following functions are available for the settings of the NUM IloTgateway ⇔ FXServer/APServer communication.



1

2

3

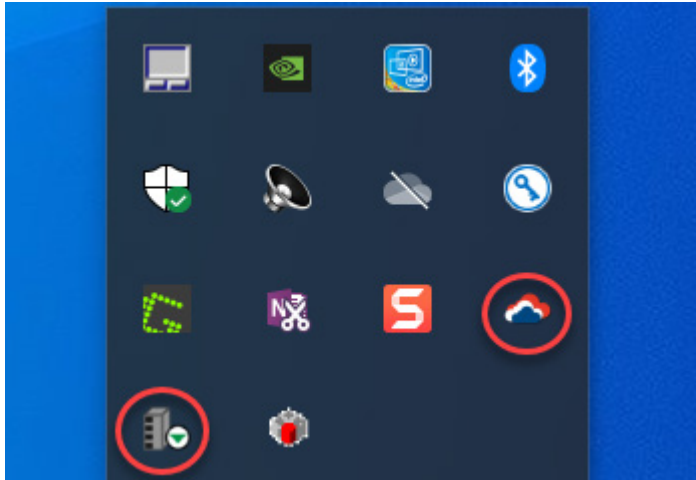
4

5

6

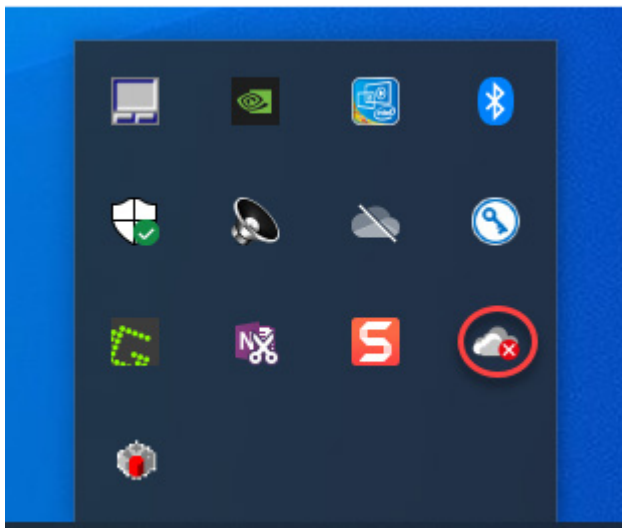
Start Service

Clicking on “Start Service” starts the NUM IloTgateway Services and the SysTray icon appears in the active state (blue-white-red cloud). The FXServer/APServer icon also appears in the active state (green triangle).



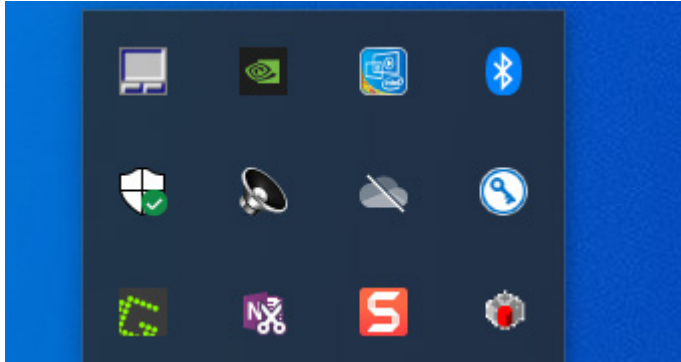
Stop Service

Clicking on “Stop Service” stops the NUM IloTgateway Services and the SysTray icon appears in the inactive state (grey-white-grey). FXServer/APServer is closed and the icon disappears.



Quit

Only the NUM IloTgateway SysTray icon disappears with Quit. However, the NUM IloTgateway Services remains started and connected to the FXServer/APServer.



The UI of the NUM IloTgateway Services can be restarted via the Windows Menu NUM IloTgateway ServiceControllerTray (see [2.3.1](#)).

1

2

3

4

5

6

2.4 NUM IloTgateway UI

This section explains the use and functions of the «NUM IloTgateway UI». It is the configuration tool for NUM IloTgateway.

Requirements

«NUM IloTgateway UI» is a software for Windows operating systems. Minimum Microsoft .NET Framework 4.0 must be installed for NUM IloTgateway on the PC system.

Installation

Run the NUM.IloT.Setup.msi to install NUM IloTgateway and UI. The NUM IloTgateway UI will be installed as a standalone application. It can be found within NUM folder in Start Menu.

2.4.1 Protocols

Connection technology settings can be accessed from the Protocols tab. One or more connection technologies can be selected from the three options in the context menu. As soon as connection type is selected, the relevant settings will appear below.

The NUM IloTgateway UI reads out the controller type with which connection is to be established from the configuration files supplied.

OPC UA

In the protocols section, the desired encryption settings and the port number for OPC UA can be set. Two different NodeSets are available: umati and a NUM specific one.

The screenshot shows the 'NUM IloTgateway Settings' application window with the 'Protocols' tab selected. The 'Connection' section has three radio buttons: 'OPC UA NUM NodeSet' (selected and highlighted with a red box), 'MTConnect', and 'MQTT'. Below this, the 'OPC UA Settings' section contains two radio buttons for security: 'Enable Security Unencrypted and Encrypted connection by Default Setting: False' (selected) and 'Enable Security Encrypted connection only by Setting: True'. A 'Port' field is set to '48030'. The bottom of the window shows the version '2.1.0.483' and a 'Save' button.

MQTT

The Id, Prefix, Startup Behavior, Host and Port number options are set in the MQTT section.

The screenshot shows the 'NUM IloTgateway Settings' window with the 'MQTT' section active. The 'MQTT' checkbox is checked and highlighted with a red box. The 'MQTT Broker' section contains the following fields and options:

- Id:** *MachineIdentifier* (with a 'Generate Guid' button)
- Prefix:** *TopicFilter*
- Startup Behavior:** Send all data (dropdown menu)
- Host:** *MQTTServerAddress* (with a 'Port' field set to 0)
- SSL/TLS:** No SSL /TLS (dropdown menu)
- SSL/TLS Version:** (dropdown menu)

The version 2.1.0.483 and a 'Save' button are visible at the bottom of the window.

Id must be a unique identifier of this publishing client. It could be a machine name or may be generated as a Guid by clicking "Generate Guid" button that is placed beside. To make sure the topics are not overlapping other applications, it is possible to set a prefix. The Prefix is a free text. However, it must comply with the MQTT requirements. It can also be empty, in this case the first level is the ID. Otherwise the first level is the prefix and the second level is this ID.

There are "Send nothing", "Send all data" and "Send changes" options within Start-up Behavior combo box.

"Send nothing" means that no information is published to the MQTT Broker in the moment when a connection is established. As soon as a topic value changes, it will be published.

If "Send all data" is selected, all current data of the topics will be sent to the MQTT Broker.

If the requester wants to receive only the changes since the last connection, "Send changes" must be selected.

In the Host entry field the IP-address or server name must be set.

The default value for the MQTT port is 0 (Automatic). If the security setting is set to "No SSL/TLS" and the port number entered is 0, the port number 1883 is used, in all other cases 8883.

The MQTT protocol can be secured. Possible security layer settings are: "No SSL/TLS", "SSL/TLS allow untrusted certifications" and "SSL/TLS". As soon as any SSL/TLS security layers are selected, the versions connected appear within the available "SSL/TLS Version" list.

1

2

3

4

5

6

MTConnect

The port number and the buffer size can be selected:

- Port number range: 1 ... 49'000 (Default: 5000).
- Buffer size range: 1 ... 10'000 (Default: 1024).

The screenshot shows the 'NUM IloTgateway Settings' window with the 'Control Parameters' tab selected. Under the 'Connection' section, the 'MTConnect' checkbox is checked and highlighted with a red box. Below this, the 'MTConnect Settings' section contains two input fields: 'Port' with the value '5000' and 'Buffer size' with the value '1024'. The bottom of the window displays the version '2.1.0.483' and a 'Save' button.

Connection	
☐ OPC UA NUM NodeSet	☒ MTConnect
☐ MQTT	

MTConnect Settings	
Port	5000
Buffer size	1024

2.1.0.483 Save

2.4.2 Control Parameters

In the Control Parameters tab, the user can make the settings for the number of NCKs as well as the number of channels and the spindle addresses for the selected channel.

The number of NCKs can only be selected for Flexium and Flexium+ controllers. It must be between one and five. After a valid NCK number is entered, channel and spindle menu is created as much as entered NCK number. As can be seen in the picture below, the NCK number was entered as two and accordingly, the channel menu was created 2 times just below. The newly created channel menu also changes dynamically depending on the number of channels entered.

As can be seen in the "Number of Channels of NCK 2" in the picture, three were entered for the number of channels, and accordingly, the spindle address for each channel appeared. Here, the spindle address can be selected from the options depending on the controller version. For Axiom Power, Num Power, and Flexium the addresses 24 to 28 are allowed, for Flexium+ 0 to 31.

The number of channels for the selected NCK must be between 1 and 8.

NUM IloTgateway Settings

Protocols Control Parameters Machine Information PLC Variables E-Parameters

Flexium+

Number of NCKs 2

Number of Channels of NCK 1: 1

Spindle Address@ Channel 1 None (0 to 31)

Number of Channels of NCK 2: 3

Spindle Address@ Channel 1 1 (0 to 31)

Spindle Address@ Channel 2 5 (0 to 31)

Spindle Address@ Channel 3 4 (0 to 31)

2.1.0.472 Save

1

2

3

4

5

6

2.4.3 Machine information

On the tab Machine Information, it is possible to set some information about the connected machine which are published on the respective data nodes.

In addition, free texts can be entered in three variables. These are published as soon as something has been entered in them.

The screenshot shows the 'NUM IloTgateway Settings' window with the 'Machine Information' tab selected. The window has a title bar with a help icon, the title 'NUM IloTgateway Settings', and standard window controls. Below the title bar is a toolbar with two icons. The main area contains a tabbed interface with five tabs: 'Protocols', 'Control Parameters', 'Machine Information' (selected), 'PLC Variables', and 'E-Parameters'. Under the 'Machine Information' tab, there is a list of fields on the left and corresponding input boxes on the right. The fields are: 'Production plant' with value '*ProductionPlant*', 'Production line' with value '*ProductionLine*', 'Machine serial' with value '*MachineSerial*', 'Machine type' with value '*MachineType*', 'Machine name' with value '*MachineName*', 'Manufacturer' with value '*Manufacturer*', and 'Product Instance Uri' with value '*ProductInstanceUri*'. Below these are three empty input boxes labeled 'Variable 1', 'Variable 2', and 'Variable 3'. At the bottom left of the window is the version number '2.1.0.472', and at the bottom right is a 'Save' button.

Field	Value
Production plant	*ProductionPlant*
Production line	*ProductionLine*
Machine serial	*MachineSerial*
Machine type	*MachineType*
Machine name	*MachineName*
Manufacturer	*Manufacturer*
Product Instance Uri	*ProductInstanceUri*
Variable 1	
Variable 2	
Variable 3	

2.4.4 PLC Variables

If the content of a PLC variable is to be published, the required information can be entered on the PLC variables tab. A maximum of 15 PLC variables can be monitored at the same time. The data type for captured PLC variables must be defined. It is also possible to enter a comment for a PLC variable. The data type options may vary depending on the control type.

Flexium⁺/Flexium

PLC objects consist of a prefix and a suffix. Together they form a Syntax Composition.

Syntax Composition: PLC objects according to the CNC-PLC exchange zone structure

[Application](#).[IoConfig_Globals](#).[Flexium_NCK](#).RCNC/WCNC.PLCCategory.[PLC VariableName](#)

Example: Application.IoConfig_Globals.Flexium_NCK.RCNC.General.Start

Syntax Composition: PLC objects according to the User structure

[Application](#).[User Structure Name](#).[PLC VariableName](#)

Example: Application.General_CNCFunction.PLCVarDIntMinus

Refer Commissioning Manual Flexium⁺ M00044EN Chapter 4 PLC Programming

Refer Commissioning Manual Flexium M00010EN Chapter 4 PLC Programming

Flexium/Flexium⁺ Example:

NUM IloTgateway Settings

Protocols Control Parameters Machine Information **PLC Variables** E-Parameters

NCK 1

	Parameter	Comment	Data Type
1	Application.IoConfig_Global	ProgramActive	Bool
2	Application.IoConfig_Global	Cycle Start	Bool
3	Application.IoConfig_Global	Feed Hold	Bool
4	Application.IoConfig_Global	Feed Enable Global	Bool
5	Application.IoConfig_Global	Feed Enable Channel	Bool
6	Application.IoConfig_Global	CNC Mode	INT
7	Application.IoConfig_Global	E30010	LReal
8	Application.General_CncFur	Real 32	Real
9	Application.General_CncFur	String	String
10	Application.General_CncFur	LINT	DINT
11	Application.General_CncFur	LowLimit	DWord

Add Delete

2.1.0.472 Save

Axium Power/Num Power

PLC objects are specified for reading/writing with the variable types and the following addresses
%I, %Q, %R, %W, %M, %V.

The PLC objects are entered in the NUM IloTgateway UI without the % character.

Refer Manual Automatic Control Function Programming Manual Ladder Language 0101938846

Example: R3.2 (E_CYCLE).

Axium Power/Num Power Example:

NUM IloTgateway Settings

Protocols Control Parameters Machine Information PLC Variables E-Parameters

NCK 1

	Parameter	Comment	Data Type
1	%R5.1	E_Prog Status	Bool
2	%R3.2	E_Cycle Status	Bool
3	%R3.1	E_Stop Status	Bool
4	%R5.0	E_CNC Ready	Bool
5	%R16.B	E_CNC Mode	Byte
6	%W4.0	Feed_Enable	Bool
7	%R9.0	Axis0 Move	Bool
8	%R9.1	Axis1 Move	Bool
9	%R9.2	Axis1 Move	Bool
10	%R1C.W	Spindle1 Value	INT

Add Delete

2.1.0.434 Save

2.4.5 E Parameters

E Parameters to be monitored can be added in the settings on the E Parameters tab.

Please note that it is only possible to monitor 15 parameters at the same time. For Flexium⁺/Flexium CNC there are 15 E parameters per NCK.



- It is not possible to add the same parameter twice.
- The data type of the E parameters is listed in the programming manual chapter External Parameters

Flexium/Flexium⁺ Example:

NUM IloTgateway Settings

Protocols Control Parameters Machine Information PLC Variables E-Parameters

NCK 1

	Parameter	Comment	Data Type
1	E81001	Get E81001	Double
2	E81050	Get 81050	Double
3	E83001	Get E83001	Double
4	E83050	Get 83050	Double
5	E30010	Get 30010	Double
6	E41004	NCK SerialNumber	SInt32
7	E41005	NCK CycleTime	SInt32
8	E70000	PositionX	Double
9	E41000	CNC Mode By Number	SInt8
10	E32016	Line Number	UInt16
11	E84100	Activity Nroto	Double
12	E56204	Schleifstücke Anzahl	Double

Add Delete

2.1.0.472 Save

1

2

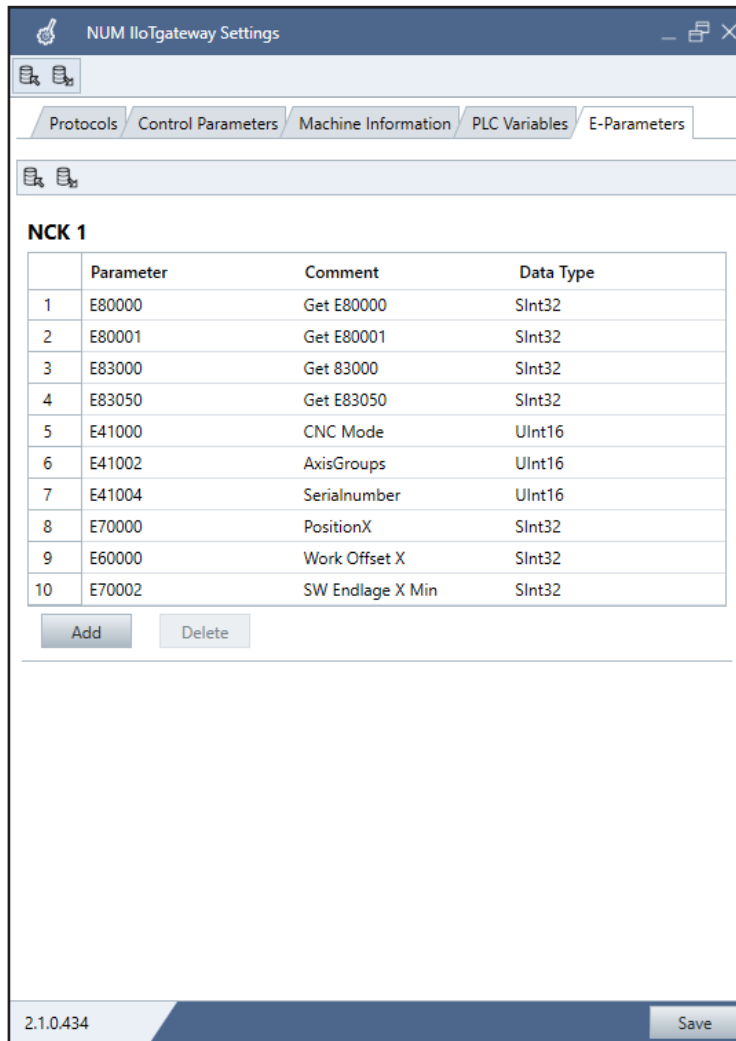
3

4

5

6

Axiom Power/Num Power Example:

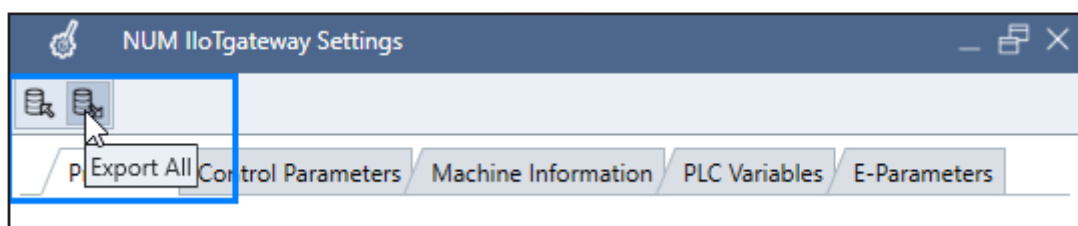


2.4.6 Import/Export

The NUM IloTgateway UI offers the possibility to export and import UI data. There are two different levels. The data is saved in a json file. The path to the storage location can be selected.

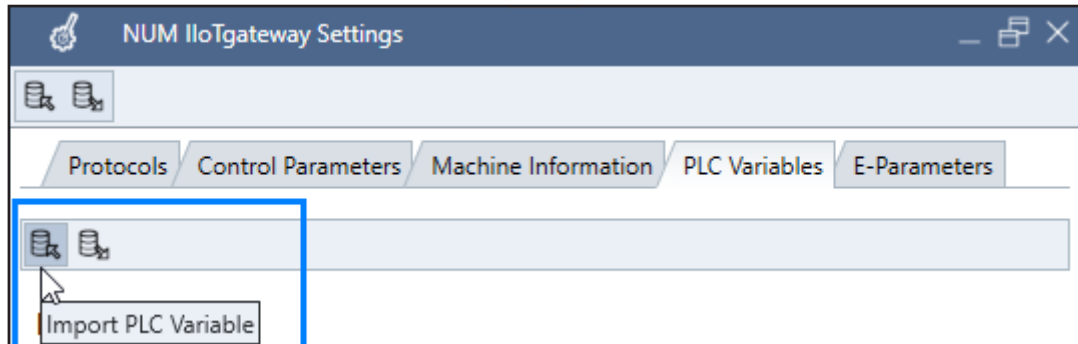
Complete data

All data that can be edited in the NUM IloTgateway UI can be exported or imported. The corresponding icons are located at the top left.



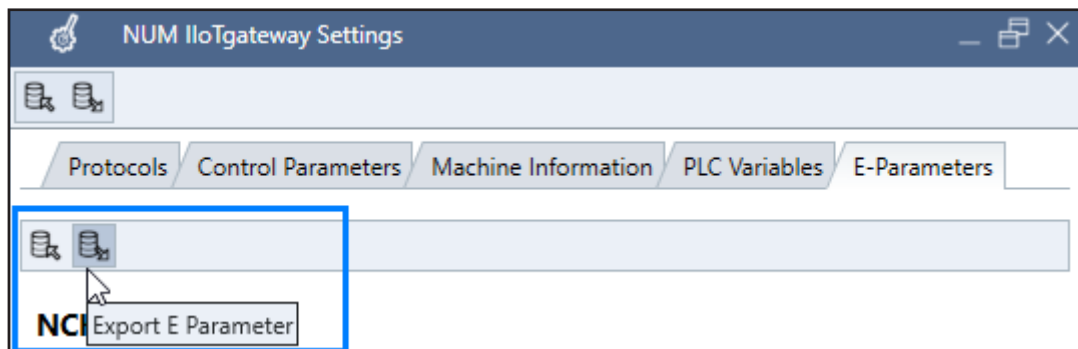
PLC Variables

Data that can be edited in the PLC variables folder of the NUM IloTgateway UI can be exported or imported. The corresponding icons are located under the folder bar on the left-hand side.



E-Parameters

Data that can be edited in the E-Parameters folder of the NUM IloTgateway UI can be exported or imported. The corresponding icons are located under the folder bar on the left-hand side.



1

2

3

4

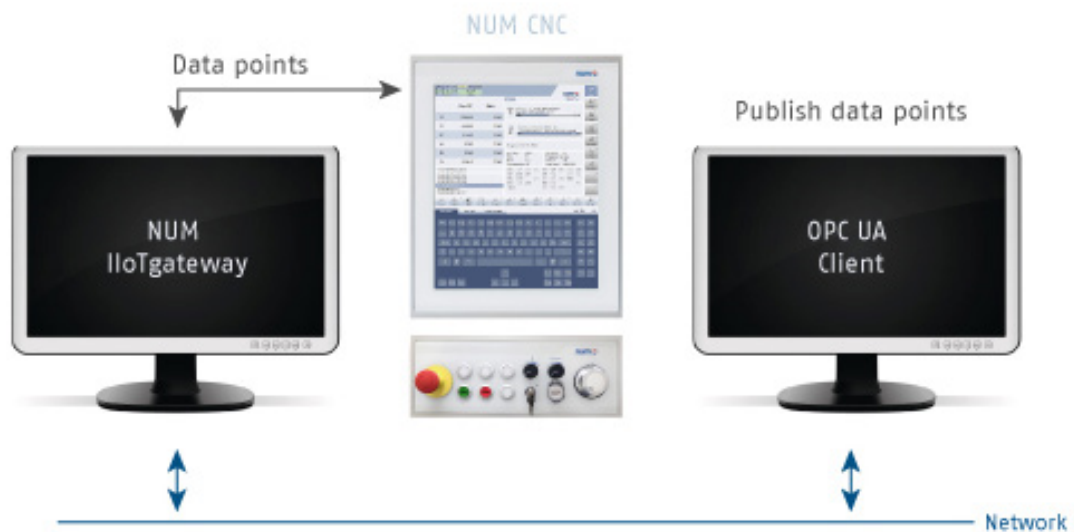
5

6

2.5 OPC UA NUM's Nodeset

2.5.1 NUM IloTgateway Architecture

Exemplary structure with OPC UA



2.5.2 Server Endpoints

The OPC UA Server of NUM IloTgateway provides access on NUM controls using the OPC UA protocol:

- Data Access (DA)
- History Access (HA)
- Alarms and Conditions (A&C)

Only one OPC UA server can be configured (see [2.4.1](#)).

The definition of the server endpoint is: **opc.tcp://<computername>:<port>**

<computername> is the name of the target computer or its IP address.

The number of the <port> used is set in the NUM IloTgateway (see [2.4.1](#)).

The server provides following endpoint types:

- None:None:Binary
- SignAndEncrypt:Basic256Sha256:Binary
- Sign:Basic256Sha256:Binary
- SignAndEncrypt:Aes128Sha256RsaOaep:Binary
- Sign:Aes128Sha256RsaOaep:Binary
- SignAndEncrypt:Aes256Sha256RsaPss:Binary
- Sign:Aes256Sha256RsaPss:Binary

The access on a node is defined over a path **NUM.*MachineName*. *Group*. *Node Id****.

MachineName is the Machine Name configured in the NUM IloTgateway UI.

Group is the name of a group of nodes. This is a grouping of nodes with similar information.

Node Id is the Node Id of an information.

2.5.3 Sign of Life

IloTgateway provides a "Sign of Life" that indicates whether the connection with the controller is still present. The signal alternates between 1 and 0 every second.

It can be reached via ***"NUM.*MachineName*.SignOfLife"***.

The display name is "Sign of Life".

2.5.4 Machine Information

The nodes of the machine information group contains mostly static information from the configuration file.

The nodes can be reached via ***"NUM.*MachineName*.MachineInformation.*Node Id*"***.

The Node Ids and the Display Names are as follows and the data type reflects the type of the value:

Node Id	Display Name	Data Type	Comment
ProductionPlant	Production Plant	String	From configuration
ProductionLine	Production Line	String	From configuration
MachineType	Machine Type	String	From configuration
MachineSerial	Machine Serial	String	From configuration
MachineName	Machine Name	String	From configuration
SystemType	System Type	String	CNC identifier from CNC. E.g. Flexium+ 68
Variable1	Variable 1	String	From configuration
Variable2	Variable 2	String	From configuration
Variable3	Variable 3	String	From configuration

2.5.5 Machine State

The machine state group contains actual data for all channels.

The nodes can be reached via ***"NUM.*MachineName*.MachineState.*Node Id*"***.

Node Id	Display Name	Data Type	Comment
ActStatus	Actual Status	String	Values: Unknown, Idle, Running, Paused, Error
ActFeedEnable	Actual Feed Enable	Boolean	
ActAlarms	Actual Alarms	Array [Alarm]	A JSON array from type Data Type Alarm
NumberOfPLCVars	Number of PLC Variables	Integer	0 to 15 0: No PLC Variables
NumberOfNCK	Number of NCKs	Integer	1 to 5
"PLCVarx"	"Path"	-	See below
"NCKx"	"NCKx"	-	See below

Data Type Alarm

Example: `{ "number": 2007, "text": "Flexium System was not shutdown correctly! Previous Zone 0 files are transferred to the NCK.", "type": "RTSMessage", "extension": 0 }`.

"PLCVarx"

Some PLC Variables values can be queried under the MachineState. The Node Id contains a number between 1 and the value of "Number of PLC Variables".

E.g. Node Id for first PLC Variable: ***"NUM.*MachineName*.MachineState.PLCVar1"***.

The Display Name is the path to the PLC variable entered via the NUM IloTgateway UI (see [2.4.4](#)).

“NCKx”

Is a node of MachineState whose elements are numbers between 1 and the value of “*Number of NCKs*” and contains several nodes.

The Display Name consists of the text “NCK” and the corresponding channel number (e.g. “**NCK 1**”).

E.g. Node Id: “**NUM.*MachineName*. MachineState.NCK1.*Node Id***”

Node Id	Display Name	Data Type	Comment
NumberOfChannels	Number of Channels	Integer	1 to 8
NumberOfEparams	Number of E Parameters	Integer	0 to 15 0: No E Parameters
*eParamx”	“User Comment”	Double	See below
“Channelx”	“Channel x”	-	See below
General	General	-	See below

“eParamx”

The E parameters are accessible over “**NUM.*MachineName*.MachineState.NCK1.eParam1**” where the number is between 0 and *Number of E Parameters*.

The Display Name is the E parameter number entered via the NUM IloTgateway UI (see [2.4.5](#)).

The value is from type Double.

“Channelx”

Are nodes of MachineState.NCKx whose elements are numbers between 1 and the value of “*Number of Channels*” and contains several nodes.

The Display Name consists of the text “Channel” and the corresponding channel number (e.g. “**Channel 1**”).

E.g. Node Id: “**NUM.*MachineName*.MachineState.NCK1.Ch1.*Node Id***”

A channel has the following nodes:

Node Id	Display Name	Data Type	Comment
ActFeedrate	Actual Feed Rate	Double	
ProgrammedFeedRate	Programmed Feed Rate	Double	
ActFeedrateOverride	Actual Feed Rate Override	Integer	
ActFeedEnable	Actual Feed Enable	Boolean	
ActSpindleSpeed	Actual Spindle Speed	Double	
ProgrammedSpindleSpeed	Programmed Spindle Speed	Double	
SpindleOverride	Spindle Override	Integer	
ActMainProgramNumber	Actual Main Program Number	String	
ActProgramFileName	Actual Program File Name	String	This information is only available for Flexium*
ActProgramComment	Actual Program Comment	String	
ActCallStack	Actual Call Stack	String	
ActOperatorMessage	Actual Operator Message	String	

General

General information can be reached via

"NUM.*MachineName*.MachineState.NCK1.General.*Node Id*".

Node Id	Display Name	Data Type	Comment
ActOperationMode	Actual Operation Mode	String	Values: Unknown, Auto, Single, MDI, Rapid, Search, Test, Manual, Home, Shifts, NoMode
PreSelectedOperationMode	Pre-selected Operation Mode	String	

2.5.6

NrProdState

The nodes of NUMROTO's NrProdState.xml are accessible under

"NUM.*MachineName*.NrProdState.*Node Id*".

Node Id	Display Name	Data Type	Comment
Content	Whole Content	json	Complete ProdState.xml in a JSON string
NumrotoVersion	Numroto Version	String	
MachineNumber	Machine Number	String	
MachineName	Machine Name	String	
Activity	Activity	String	Values: No, GrindProd, ReGrind, ToolProbe, WheelProbe, WheelDress, WheelCrush, WheelStick, ToolLoader, WheelChange
JobType	Job Type	String	
JobName	Job Name	String	
JobStarttime	Job Start Time	String	
JobEndtime	Job End Time	String	
ToolName	Tool Name	String	
ToolStartTime	Tool Start Time	String	
ToolEndTime	Tool End Time	String	
ToolNumber	Tool Number	Integer	
ToolNumberMax	Tool Number Max	Integer	
ToolNumberUnsuccessful	Tool Number Unsuccessful	Integer	
ProcessType	Process Type	String	

1

2

3

4

5

6

2.5.7 NrToolInfo

The nodes of the NUMroto's NrToolInfo.xml are accessible over
"NUM.*MachineName*.NrToolInfo.*Node Id*".

Node Id	Display Name	Data Type	Comment
Content	Whole Content	json	Complete NrToolInfo.xml in a JSON string
NumrotoVersion	Numroto Version	String	
ToolInfoVersion	Tool Info Version	String	
Name	Name	String	
Comment	Comment	String	
JobNumber	Job Number	String	
BlankLength	Blank Length	Double	
BlankDiameter	Blank Diameter	Double	
ShankLength	Shank Length	Double	
ShankDiameter	Shank Diameter	Double	
ShorteningAmount	Shortening Amount	Double	
Collet	Collet	-	See below
Configuration	Configuration	-	See below
Operations	Operations	-	See below

Collet

The nodes of the group Collet are accessible under
"NUM.*MachineName*.NrToolInfo.Collet.*Node Id*".

Node Id	Display Name	Data Type	Comment
Name	Name	String	

Configuration

The nodes of the group Configuration are accessible under
"NUM.*MachineName*.NrToolInfo.Configuration.*Node Id*".

Node Id	Display Name	Data Type	Comment
Package	Package	-	See below

Package

The Configuration nodes for Packages can be reached via the path
"NUM.*MachineName*.NrToolInfo.Configuration.Package.*Node Id*".

Node Id	Display Name	Data Type	Comment
Id	Id	Integer	
Name	Name	String	

Operation

The node information of the different Operations are available under **"NUM.*MachineName*.NrToolInfo.Operations.Operation[n].*Node Id*"**. Where "n" is the number of the operation ("n" is between 1 and 99).

Node Id	Display Name	Data Type	Comment
Id	Id	Integer	
Name	Name	String	
Machining	Machining	Integer	0: False / 1: True
Simulate3D	Simulate 3D	Integer	0: False / 1: True
MachiningTime	Machining Time	Double	
SimulatedCollision	Simulated Collision	Integer	
SimulatedPathError	Simulated Path Error	Double	
SimulatedRemovalRate	Simulated Removal Rate	Double	
Wheel	Wheel	-	See below

Wheel

The node information of the wheels of a package are accessible at **"NUM.*MachineName*.NrToolInfo.Operations.Operation[n].Wheel.*Node Id*"**.

Node Id	Display Name	Data Type	Comment
Name	Name	String	
Type	Type	String	
Package	Package	String	
Diameter	Diameter	Double	
RotationSpeed	Rotation Speed	Double	
CuttingSpeed	Cutting Speed	Double	
DirectionOfRotation	Direction of Rotation	String	

1

2

3

4

5

6

2.6 OPC UA umati

2.6.1 NUM IloTgateway Architecture

See section 2.5.1

2.6.2 End points

See section 2.5.2

2.6.3 Equipment

There is no information on this.

2.6.4 Identification

The variables of the identification object contains mostly static information from the configuration file.

The variables can be reached via "***Machines.*MachineName*.Identification.*DisplayName****".

The ****DisplayName**** is as follows and the data type reflects the type of the value:

Display Name	Node ID	Data Type	Comment
MachineName	NS9/6020	String	From configuration
MachineType	NS9/6019	String	From configuration
Manufacturer	NS9/6001	String	From configuration
ProductInstanceUri	NS9/6002	String	From configuration
ProductionLine	NS9/6022	String	From configuration
ProductionPlant	NS9/6035	String	From configuration
SerialNumber	NS9/6003	String	From configuration
SystemType	NS9/6038	String	CNC Type E.g. Flexium* 68

2.6.5 Monitoring

The Monitoring object contains following objects:

- Channelx
- MachineTool
- Spindex

Channelx

This object can exist in several instances. They are numbered between 1 and the value of "Number of Channels" and contain several variables.

The object name consists of the text "Channel" and the corresponding channel number (e.g. "Channel1").

A channel has the following variables and they can be reached via ***"Machines.*MachineName*.Monitoring.Channel1.*DisplayName*"*** (example for channel 1):

Display Name	Node ID	Data Type	Comment
ChannelMode	NS9/6023	Int32	0: Automatic 1: MdaMdi 2: JogManual 3: JogIncrement 4: TeachingHandle (5: Remote) (6: Reference) 7: Other Values in () not supported
ChannelState	NS9/6024	Int32	0: Active 1: Interrupted 2: Reset
FeedOverride	NS9/6025	Double	
Name	NS9/6028	String	

MachineTool

The variables of this object can be reached via ***"Machines.*MachineName*.Monitoring.MachineTool.*DisplayName*"***.

Display Name	Node ID	Data Type	Comment
OperationMode	NS9/6004	Int32	0: Manual 1: Automatic 2: Setup 3: AutoWithManualIntervention 4: Service 5: Other

Spindlex

This object can exist in several instances. They are numbered between 1 and the value of "Number of Spindles" and contain several variables.

The object name consists of the text "Spindle" and the corresponding spindle number (e.g. "Spindle1").

A spindle has the following variables and they can be reached via ***"Machines.*MachineName*.Monitoring.Spindle1.*DisplayName*"*** (example for spindle 1):

Display Name	Node ID	Data Type	Comment
IsRotating	NS9/6029	Boolean	
Override	NS9/6031	Double	
Name	NS9/6143	String	

1

2

3

4

5

6

2.6.6**Notification**

The Notification object contains the object messages.

Messages

A Message object has the following variables and they can be reached via ***"Machines.*MachineName*.Notification.*DisplayName*"***:

Display Name	Node ID	Data Type	Comment
Messages	NS6	Strings	Message: Message text ConditionName: Error number inclusive extension

2.6.7**Production**

The Production object contains the object ActiveProgram.

ActiveProgram

This object contains variables and an object and they can be reached via ***"Machines.*MachineName*.Production.ActiveProgram.*DisplayName*"***:

		Data Type	Comment
Name	NS9/6005	String	Name of the part program
State		-	See below
NumberInList	NS9/6006	UInt16	

State

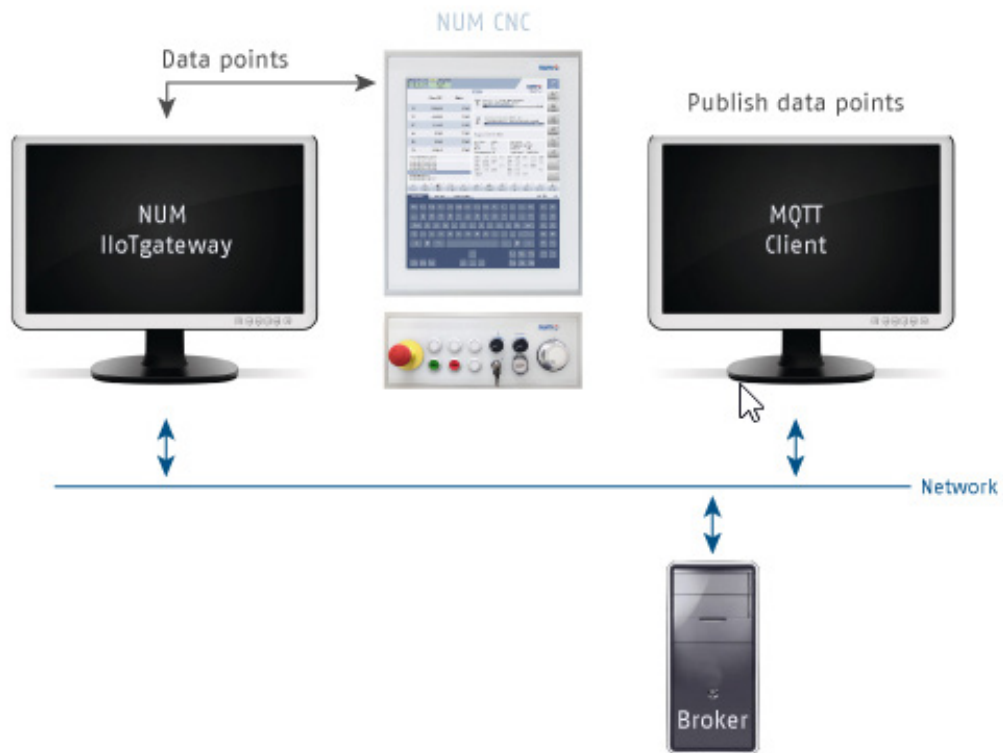
The variable can be reached via ***"Machines.*MachineName*.Production.ActiveProgram.State.*DisplayName*"***:

Display Name	Node ID	Data Type	Comment
CurrentState	NS9/6007	String	Initializing Running Ended Interrupted Ended / Aborted

2.7 MQTT

2.7.1 NUM IloTgateway Architecture

Exemplary structure with MQTT



2.7.2 Topics and values

Collected and already converted data is published via the MQTT protocol to a MQTT Broker.

A message broker is a service in between multiple clients. In this case multiple clients which understand the protocol MQTT. The broker itself isn't interested on which topics clients publish respective subscribe this only purpose is to relay messages from one publisher to multiple subscribers.

Every client can be a publisher, subscriber, or both. (The NUM IloTgateway has no subscription capabilities at the moment, i.e. it is not possible to receive commands from somewhere).

In some cases the broker is used to translate the payload from one or multiple topics to another format or topics and publishes them on himself or on another broker.

MQTT topics are in the form ***"prefix/Id/topic/subtopic"***.

The prefix, if present, comes on the first level "prefix". If no prefix is selected, the first level is the Id from the Protocols tab of the NUM IloTgateway UI.

All topics are prefixed with this combination: ***"prefix/Id"***. To be compatible with MQgateway the prefix name must have a leading / in e.g. ***/prefix***.

All topics have the retain flag set, that means if someone connects to the broker after a topic was published, he gets the latest information from the broker.

The payload of each topic is in ***JSON format***. Depending on the topic, it could be a list of payloads or just a single payload.

Every payload has a timestamp in the format ***"2018-08-08T07:53:11.218Z"***. The timestamp is in UTC.

There is also a property value, which contains the actual data.

1

2

3

4

5

6

A simple payload with a string looks like this:

```
{“timestamp”:“2018-08-08T07:52:00.541Z”,“value”:“Some strings”,“state”:“Good”}
```

Moreover, a payload with a number like this:

```
{“timestamp”:“2018-08-08T07:52:00.556Z”,“value”:100.4553,“state”:“BadNotActive”}
```

It is also possible to have complex data values like this:

```
{“timestamp”:“2018-08-08T07:52:57.606Z”,“value”:{“parameter”:“75000”,“value”:47373.9324},  
“state”:“Good”}
```

The data is always sent, regardless of whether a value is valid or not. To avoid invalid data, the payload contains the state information (Good or BadNotActive).

The topics are grouped in three logical units; they are described in the following sections.

2.7.3 Sign of Life

IIoTgateway provides a “Sign of Life” that indicates whether the connection with the controller is still present. The signal alternates between 1 and 0 every second.

The payload is published over the topic “*prefix/Id/SignOfLife*”.

2.7.4 Machine Information

The machine information topic contains mostly static information from the configuration file.

The topics can be reached via the path “*prefix/Id/MachineInformation/topic*”.

The topics are as follows and the data type reflects the type of the value (without timestamp).

Topic	Data Type	Comment
ProductionPlant	string	From configuration
ProductionLine	string	From configuration
MachineType	string	From configuration
MachineSerial	string	From configuration
MachineName	string	From configuration
SystemType	string	CNC identifier from CNC. E.g. Flexium+ 68
Variable1	string	From configuration
Variable2	string	From configuration
Variable3	string	From configuration

2.7.5 Machine State

The machine state topic contains actual data for all channels.

The topics can be reached via the path “*prefix/Id/MachineState/topic*”.

Topic	Data Type	Comment
ActStatus	string	Values: Unknown, Idle, Running, Paused, Error
ActFeedEnable	bool	
ActAlarms	Array[Alarm]	A JSON array from type Data Type Alarm
NumberOfPLCVars	int	0 to 15 / 0: No PLC Variables
NumberOfNCK	int	1 to 5
PLCVar	-	See below
NCK	-	See below

Data Type Alarm

Example: `{"number":2007,"text":"Flexium System was not shutdown correctly! Previous Zone 0 files are transferred to the NCK.", "type":"RTSMessage", "extension":0}`

PLCVar

Is a subtopic of MachineState whose elements are numbers between 1 and the value of "NumberOfPLCVars". E.g. **`prefix/Id/MachineState/PLCVar/1`**.

The value of such an element has the format as follows, where the parameter is the subscribed variable: `{"timestamp":"2018-08-08T07:18:30.377Z", "value":{"parameter":"Application.IoConfig_Globals.Flexium_NCK.RCNC.Axis[0].Status", "value":1}}`

NCK

Is a subtopic of MachineState whose elements are numbers between 1 and the value of "NumberOfNCK". E.g. **`prefix/Id/MachineState/NCK/1/topic`**.

Topic	Data Type	Comment
NumberOfChannels	int	1 to 8
NumberOfEparams	int	0 to 15 / 0: No E Parameters
eParam	-	See below
Channel	-	See below
General	-	See below

eParam

The E parameters are accessible over **`prefix/Id/MachineState/NCK/1/eParam/number`** where the number is between 0 and *NumberOfEparams*.

The value is a complex type as follows; the parameter is the subscribed *eParam number*:

`{"timestamp":"2018-08-08T07:53:06.429Z", "value":{"parameter":"70000", "value":1044.1040308}}`

Channel

The topics of the corresponding channel are accessible through **`prefix/Id/MachineState/NCK/1/Channel/number/topic`** where the number is between 1 and *NumberOfChannels*.

A channel has the following topics:

Topic	Data Type	Comment
ActFeedrate	double	
ProgrammedFeedRate	double	
ActFeedrateOverride	int	
ActFeedEnable	bool	
ActSpindleSpeed	double	
ProgrammedSpindleSpeed	double	
SpindleOverride	int	
ActMainProgramName	string	Actual program number of the main program (e.g. %0100)
ActProgramFileName	string	This information is only available for Flexium+
ActProgramComment	string	
ActCallStack	string	
ActOperatorMessage	string	

General

General information can be reached via “*prefix/Id/MachineState/NCK/1/General/topic*”.

Topic	Data Type	Comment
ActOperationMode	string	Values: Unknown, Auto, Single, MDI, Rapid, Search, Test, Manual, Home, Shifts, NoMode
PreSelectedOperationMode	string	

2.7.6

NUMroto

The topics of NUMROTO's NrProdState.xml are accessible under “*prefix/Id/NUMroto/topic*”.

Topic	Data Type	Comment
NrProdState	json	Complete ProdState.xml in a JSON string
Info	-	See below
JobState	-	See below
WorkpieceState	-	See below
ProcessState	-	See below

Info

The topics for Info can be reached via the path “*prefix/Id/NUMroto/Info/topic*”.

Topic	Data Type	Comment
NumrotoVersion	string	
MachineNumber	string	
MachineName	string	

JobState

The topics for JobState are accessible over “*prefix/Id/NUMroto/JobState/topic*”.

Topic	Data Type	Comment
Activity	string	Values: No, GrindProd, ReGrind, Tool-Probe, WheelProbe, WheelDress, WheelCrush, WheelStick, ToolLoader, WheelChange
JobMode	string	ProdState.xml: /NrProdData/JobType
JobName	string	ProdState.xml: /NrProdData/JobName
JobStarttime	DateTime	ProdState.xml: /NrProdData/JobStart-Time
JobEndtime	DateTime	ProdState.xml: /NrProdData/JobEnd-Time
JobTotalNoOfWorkpcs	int	ProdState.xml: /NrProdData/ToolNum-berMax
JobNoOfCurrentWorkpcs	int	ProdState.xml: /NrProdData/ToolNum-ber
JobNoOfUnsuccessful Workpcs	int	ProdState.xml: /NrProdData/ToolNum-berUnsuccessful

WorkpieceState

The topics for WorkpieceState are accessible over “*prefix/Id/NUMroto/WorkpieceState/topic*”.

Topic	Data Type	Comment
WorkpcsName	string	ProdState.xml: /NrProdData/ToolName
WorkpcsStartTime	DateTime	ProdState.xml: /NrProdData/ToolStart-Time
WorkpcsEndTime	DateTime	ProdState.xml: /NrProdData/ToolEnd-Time

ProcessState

The topics for ProcessState are accessible over “*prefix/Id/NUMroto/ProcessState/topic*”.

Topic	Data Type	Comment
ProcessType	string	

2.7.7**NrToolInfo**

The topics of the NUMroto's NrToolInfo.xml are accessible over “*prefix/Id/NrToolInfo/topic*”.

Topic	Data Type	Comment
NrToolInfo	json	Complete NrToolInfo.xml in a JSON string
NumrotoVersion	string	
ToolInfoVersion	string	
Name	string	
Comment	string	
JobNumber	string	
BlankLength	double	
BlankDiameter	double	
ShankLength	double	
ShankDiameter	double	
ShorteningAmount	double	
Collet	-	See below
Configuration	-	See below
Operations	-	See below
NrToolInfo	json	Complete NrToolInfo.xml in a JSON string

Collet

The Collet topics are accessible under “*prefix/Id/NrToolInfo/Collet/topic*”.

Topic	Data Type	Comment
Name	string	

Configuration

The Configuration topics are available under “*prefix/Id/NrToolInfo/Configuration/topic*”.

Topic	Data Type	Comment
Package	-	See below

Package

The Configuration topics for Packages can be reached via the path
“**prefix/Id/NrToolInfo/ Configuration/Package/topic**”.

Topic	Data Type	Comment
Id	int	
Name	string	

Operation

The topics of the different Operations are available under
“**prefix/Id/NrToolInfo/Operations/ Operation[n]/topic**”. Where “n” is the number of the operation
 (“n” is between 1 and 99).

Topic	Data Type	Comment
Id	int	
Name	string	
Machining	int	0: False / 1: True
Simulation3D	int	0: False / 1: True
MachiningTime	double	
SimulatedCollision	int	
SimulatedPathError	double	
SimulatedRemovalRate	double	
Wheel	-	See below

Wheel

The topics of the wheels of a package are accessible at
“**prefix/Id/NrToolInfo/Operations/ Operation[n]/Wheel/topic**”.

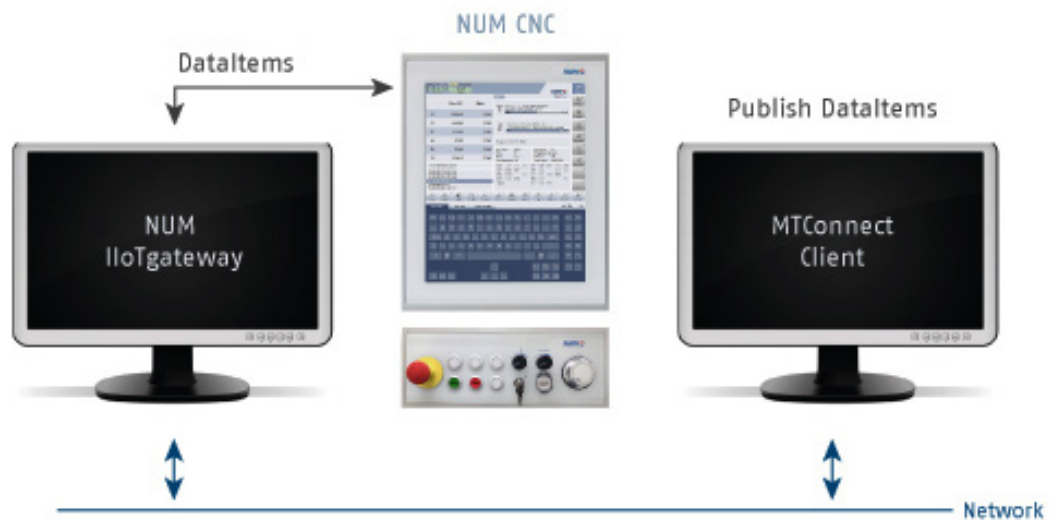
Where “n” is the number of the operation (“n” is between 1 and 99).

Topic	Data Type	Comment
Name	string	
Type	string	
Package	string	
Diameter	double	
RotationSpeed	double	
CuttingSpeed	double	
DirectionOfRotation	string	

2.8 MTConnect

2.8.1 NUM IloTgateway Architecture

Exemplary Structure with MTConnect



2.8.2 Introduction

MTConnect is a universal communications protocol for the factory floor. It is designed specifically for the shop floor environment.

While there are numerous communication solutions available for the shop floor, MTConnect offers one very distinct difference.

MTConnect is the first standard to define a "dictionary" for manufacturing data. This means that data from multiples machines will have a common definition - name, units, values and context.

With MTConnect, the data is defined only once at the MTConnect compliant interface to the device or machine tool.

Once the data is defined based on the MTConnect standard protocol, it can be easily be used by all MTConnect compliant software applications. This eliminates the need to redefine the data within each application.

MTConnect significantly reduces startup time, overall project costs, and long term maintenance of software system interfaces. MTConnect compliant devices process information locally and then provide that data in a consistent format to any MTConnect compliant application - ERP, MES, and Production Management Systems.

(MTConnect.org: Getting Started with MTConnect:Architecture)

1

2

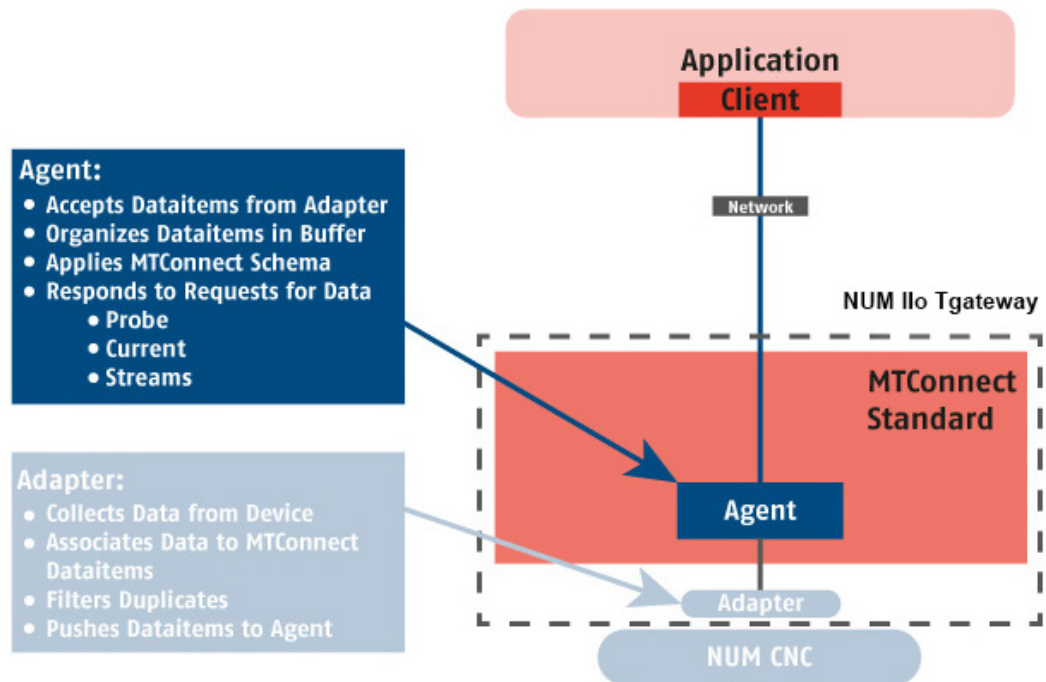
3

4

5

6

MTConnect Most Basic Architecture



MTConnect of the NUM IloTgateway acts as server for the CNC and NUMROTO data via http requests.

NUM IloTgateway offers many Data Items. That they can be separated from the others, a Data Item Name consists of ***Group*/*Sub-Name***.

Group is the name of a group of data items or a component with similar information.

Sub-Name is the name of an information.

In the following tables, the group name is not mentioned but listed before. E.g., Data Item Name is "Machine Information/Production Plant". The text "Machine Information" is above the table and "Production Plant" is in the table.

2.8.3 General Data

General Data of the NUM device.

The Data Item Name is **"*Sub-Name*"**.

Sub-Name	Category	Comment
Availability	Event	Values: UNAVAILABLE / AVAILABLE
Sign of Life	Event	IloTgateway provides a „Sign of Life“ that indicates whether the connection with the controller is still present. The signal alternates between 1 and 0 every second.

2.8.4 Machine Information

The “component” machine information contains mainly static information from the configuration file.

The Data Item Name is **"Machine Information/*Sub-Name*"**.

Sub-Name	Category	Comment
Production Plant	Event	From configuration
Production Line	Event	From configuration
Machine Type	Event	From configuration
Machine Serial	Event	From configuration
Machine Name	Event	From configuration
System Type	Event	CNC identifier from CNC. E.g. Flexium ⁺ 68
Variable1	Event	From configuration
Variable2	Event	From configuration
Variable3	Event	From configuration

2.8.5 Machine State

The “component” Machine State contains actual data for all channels.

The Data Item Name is “**Machine State/*Sub-Name***”.

Sub-Name	Category	Comment
Actual Status	Event	Values: Unknown, Idle, Running, Paused, Error
Actual Feed Enable	Event	
Number of Active Alarms	Event	0 to 5
Active Alarm Error Type 0	Event	CNC, Drive etc.
Active Alarm Error Number 0	Event	Error number
Active Alarm Error Index 0	Event	Extension of the error number
Active Alarm Error Line 0	Event	N number of the CNC program where the problem was detected
Active Alarm Error Message 0	Event	Message text
Active Alarm Error Additional 0	Event	Line number of the CNC program where the problem was detected
...		Repetition of the Data Items of Active Alarms 1 to 4
Number of PLC Variables	Event	0 to 15 / 0: No PLC Variables
Number of NCKs	Event	1 to 5
“PLC Variable x”	-	See below
“NCKx”	-	See below

“PLC Variable x”

Some PLC Variables values can be queried under the “component” Machine State. The Data Item Name contains a number between 1 and the value of “*Number of PLC Variables*”.

Data Item Name for the first PLC Variable is e.g. “**Machine State/PLC Variable 1**”.

This Data Item is from the Event category.

“NCKx”

The “component” Machine State contains the component NCK. Its ID is a number between 1 and the value of “*Number of NCKs*”.

The Data Item Name for the first NCK is e.g. “**Machine State/NCK 1/*Sub-Name***”.

An NCK has the following Data Items:

Sub-Name	Category	Comment
Number of Channels	Event	1 to 8
Number of E Parameters	Event	0 to 15 / 0: No E Parameter
“E Parameter x”	-	See below
“Channel x”	-	See below
General	-	See below

“E Parameter x”

Some E Parameters values can be queried under the component NCK. The Data Item Name contains a number between 1 and the value of “*Number of E Parameters*”.

Data Item Name for the first E Parameter of the NCK1 is e.g.

“Machine State/NCK 1/E Parameter 1”.

This Data Item is from the Event category.

“Channel x”

The “component” NCK contains the “component” Channel. Its ID is a number between 1 and the value of “*Number of Channels*”.

The Data Item Name for the first Channel of NCK1 is e.g.

“Machine State/NCK 1/Channel 1/ *Sub-Name*”.

A channel has the following Data Items:

Sub-Name	Category	Comment
Actual Feed Rate	Event	
Programmed Feed Rate	Event	
Actual Feed Rate Override	Event	
Actual Feed Enable	Event	
Actual Spindle Speed	Event	
Programmed Spindle Speed	Event	
Spindle Override	Event	
Actual Main Program Number	Event	
ActProgramFileName	Event	This information is only available for Flexium*
Actual Program Comment	Event	
Actual Call Stack	Event	
Actual Operator Message	Event	ASCII string

General

The component NCK contains the “component” General.

The Data Item Names of it is **“Machine State/NCK 1/General/*Sub-Name*”**.

Sub-Name	Category	Comment
Actual Operation Mode	Event	Values: Unknown, Auto, Single, MDI, Rapid, Search, Test, Manual, Home, Shifts, NoMode
Pre-selected Operation Mode	Event	

1

2

3

4

5

6

2.8.6 NrProdState

The device NrProdState offers information of NUMROTO's NrProdState.xml. They are accessible under ***“/NrProdState/*Sub-Name*”***.

Sub-Name	Category	Comment
NrProdState Content JSON	Event	Complete ProdState.xml in a JSON string
Numroto Version	Event	
Machine Number		
Machine Name	Event	
Activity	Event	Values: No, GrindProd, ReGrind, Tool-Probe, WheelProbe, WheelDress, WheelCrush, WheelStick, ToolLoader, WheelChange
Job Type	Event	
Job Name	Event	
Job Start Time	Event	
Job End Time	Event	
Tool Name	Event	
Tool Start Time	Event	
Tool End Time	Event	
Tool Number	Event	
Tool Number Max	Event	
Tool Number Unsuccessful	Event	
Process Type	Event	

2.8.7 NrToolInfo

The device NrToolInfo offers information of NUMROTO's NrToolInfo.xml. They are accessible under ***“NrToolInfo/*Sub-Name*”***.

Sub-Name	Category	Comment
NrToolInfo Content XML	Event	Complete NrToolInfo.xml in a JSON string
Numroto Version	Event	
Tool Info Version	Event	
Name	Event	
Comment	Event	
JobNumber	Event	
Blank Length	Event	
Blank Diameter	Event	
Shank Length	Event	
Shank Diameter	Event	
Shortening Amount	Event	
Collet	-	See below
Configuration	-	See below
Operations	-	See below

Collet

The Data Items of the group Collet are accessible under **"NrToolInfo/Collet/*Sub-Name*"**.

Sub-Name	Category	Comment
Name	Event	

Configuration

The Data Items of the group Configuration are accessible under **"NrToolInfo/Configuration/*Sub-Name*"**.

Sub-Name	Category	Comment
Package	-	See below

Package

The Data Items of the group Package are accessible under **"NrToolInfo/Package/*Sub-Name*"**.

Sub-Name	Category	Comment
Id	Event	
Name	Event	

Operation

The Data Items of the group Operations are accessible under **"NrToolInfo/Operations/ Operation n/*Sub-Name*"**. Where "n" is the number of the operation ("n" is between 1 and 99).

Sub-Name	Category	Comment
Id	Event	
Name	Event	
Machining	Event	0: False / 1: True
Simulate 3D	Event	0: False / 1: True
Machining Time	Event	
Simulated Collision	Event	
Simulated Path Error	Event	
Simulated Removal Rate	Event	
Wheel	-	See below

Wheel

The Data Items of the group Wheel are accessible under **"NrToolInfo/Operations/ Operation n/ Wheel/*Sub-Name*"**. Where "n" is the number of the operation ("n" is between 1 and 99).

Sub-Name	Category	Comment
Name	Event	
Type	Event	
Package	Event	
Diameter	Event	
Rotation Speed	Event	
Cutting Speed	Event	
Direction of Rotation	Event	

1

2

3

4

5

6

2.9 Troubleshooting / Diagnostics

If there is no data available, please check the following:

- Is there a license? Has it the correct serial number? The license file is stored under C:\Program Files (x86)\NUM\NUM IIoTgateway\Gateway\Licenses
- Is FXServer / APServer connected to the correct CNC?
- Runs NUM IIoTgateway with elevated permissions? (It should not, when FXServer does not.)
- In case of MQTT: Is the broker reachable?

If all of the above questions can be answered with yes and it is still not running, please check the logfiles.

The log file are stored under "C:\User\Public\Documents\NUM\NUM IIoTgateway\Logs". A file is created daily if NUM IIoTgateway is started. A maximum of 20 files are stored.

Make sure there is no exception displayed.

2.10 NUM IloTgateway Installation



Administration rights are required for the installation.

NUM delivers the NUM IloTgateway installation software digitally via your NUM branch. Two variants are available: The standard version and a trial version.

When installing a standard version for the first time, a license (*.lic) and a configuration file (controller.json) are supplied in a zip file. This zip file must be unzipped and saved together with installation program on the target system.

The directory should look like this:

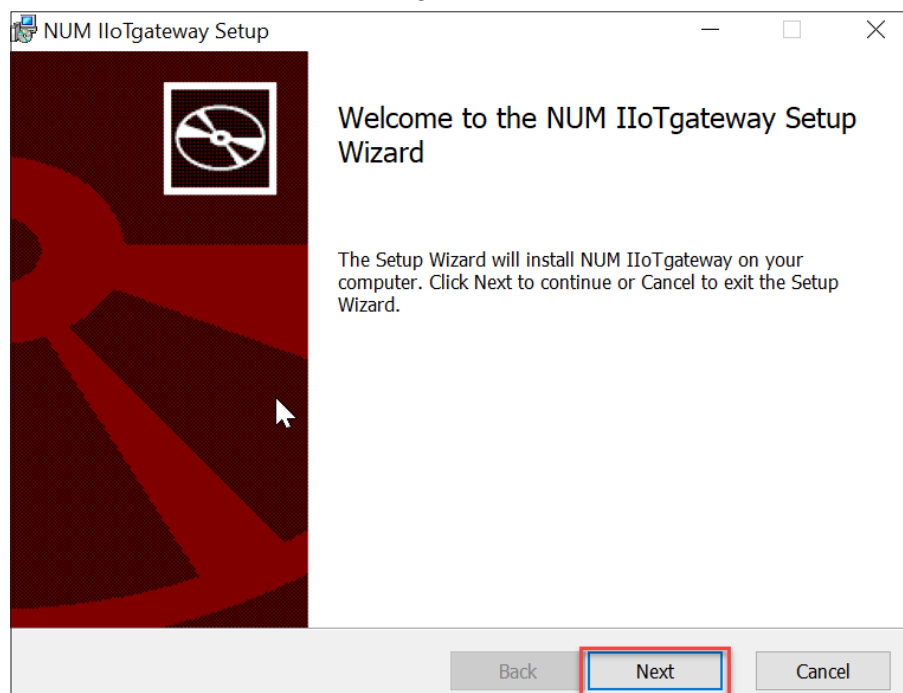
 Config	18.03.2024 08:34	Dateiordner	
 Licenses	18.03.2024 08:34	Dateiordner	
 NUM.IloT.Setup_2.1.0.480.msi	28.02.2024 16:16	Windows Installer...	188'837 KB

In the case of a software upgrade or a test version, only the installation program needs to be saved on the target system.

Double-click the Microsoft installation program NUM.IloT.Setup_x.x.x.xxx.msi or NUM.IloT.Setup_x.x.x.xxx-Trial.msi to install the NUM IloTgateway.

NUM IloTgateway Setup Wizard Page appears.

Press Button: Next ⇒ Installation begins



1

2

3

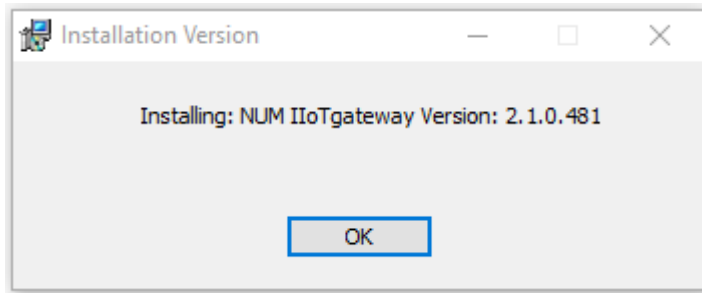
4

5

6

A window appears with the version number to be installed. This is a hint so that it is clear which software version is being installed.

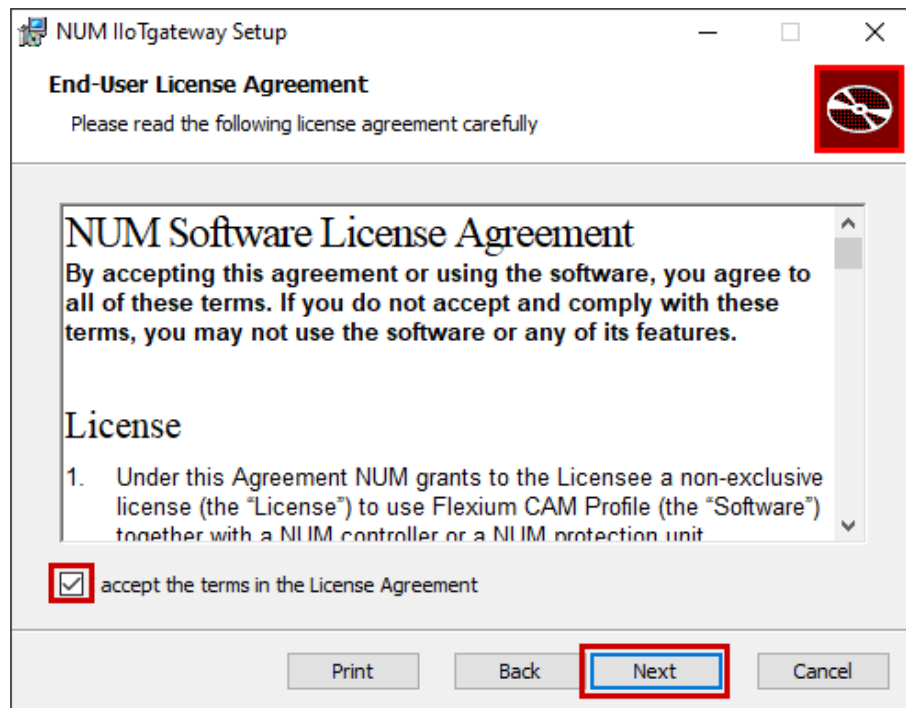
Press the OK button.



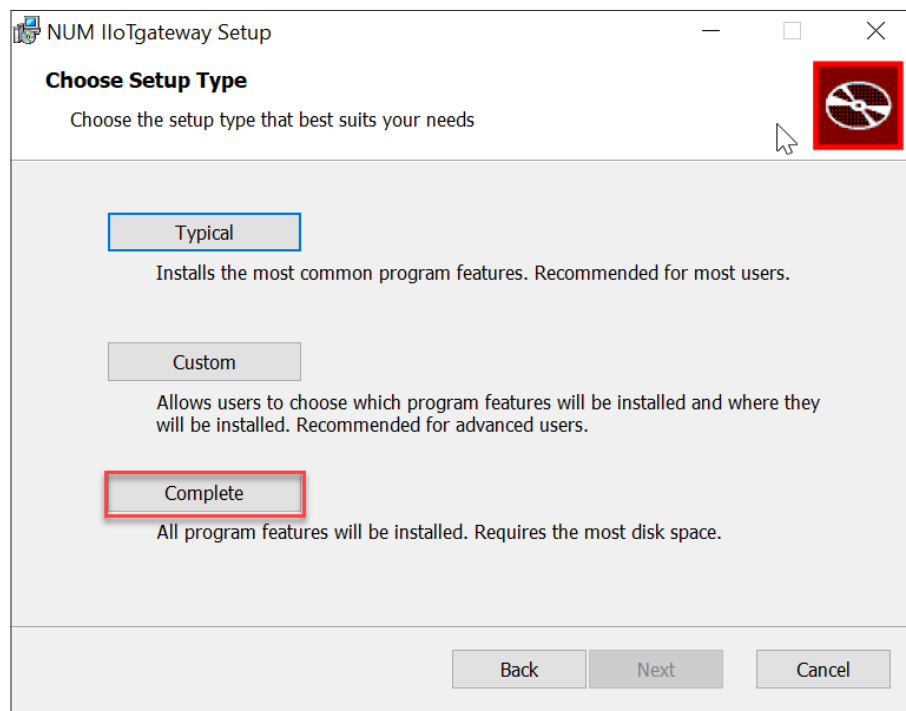
NUM IloTgateway Software License Agreement window appears.

Activate the checkbox to agree to the license agreement.

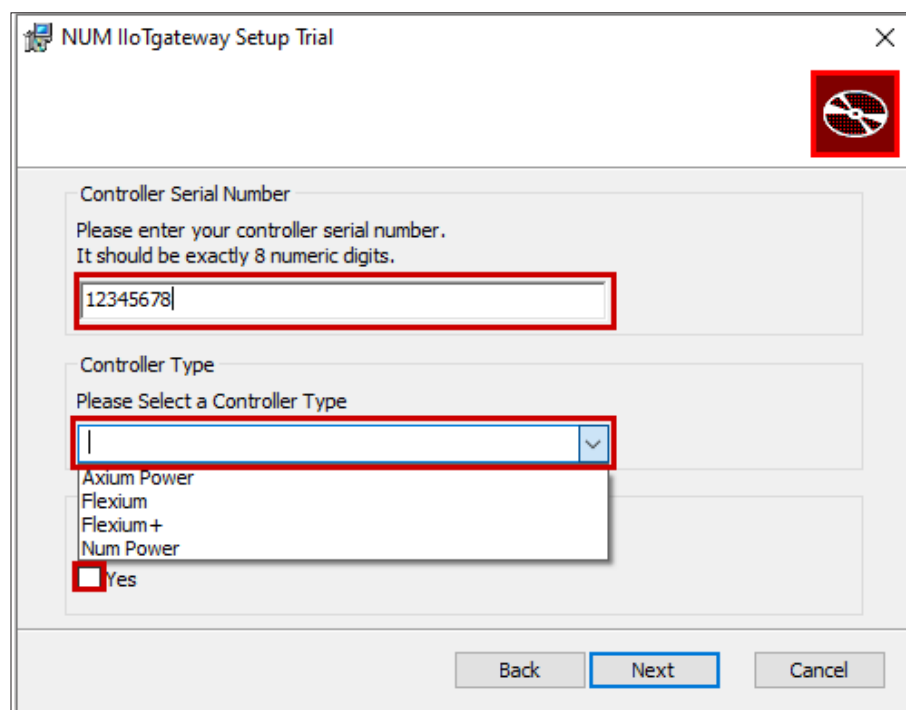
Press Button: Next ⇒ Continue the installation



In case of a standard installation, NUM IloTgateway Setup window appears.
Choose a Setup Type, for example Complete.



In case of installing a trial version, NUM IloTgateway Setup Trial window appears.
Enter the serial number of the CNC to be used for the test.
Chose the according CNC type.
Activate the Yes checkbox if the umati nodeset is to be tested.



Press Button: Next

1

2

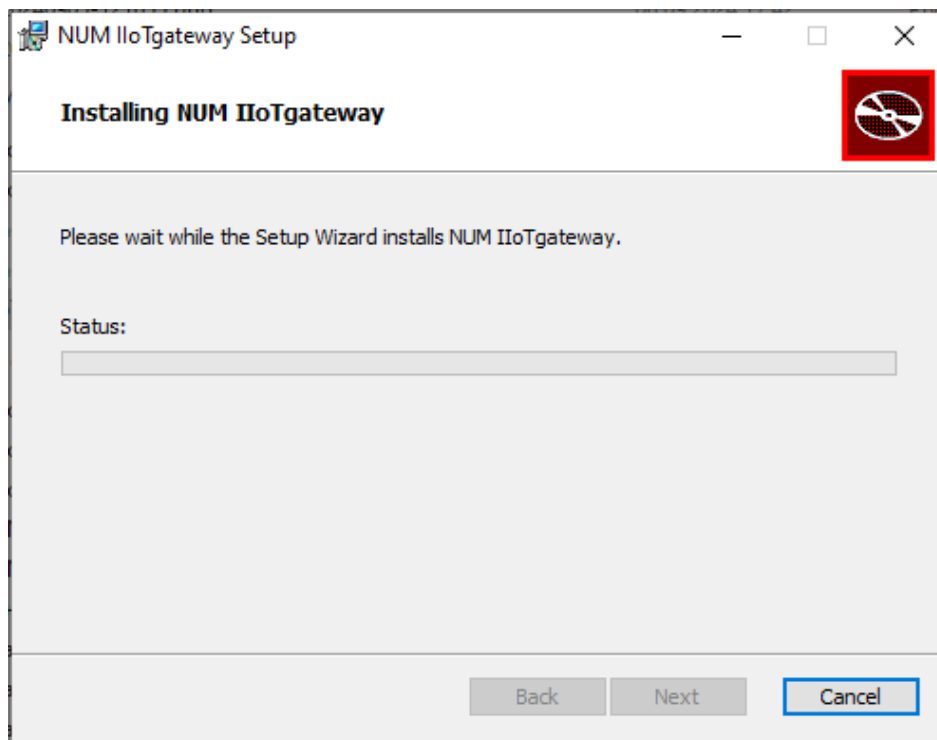
3

4

5

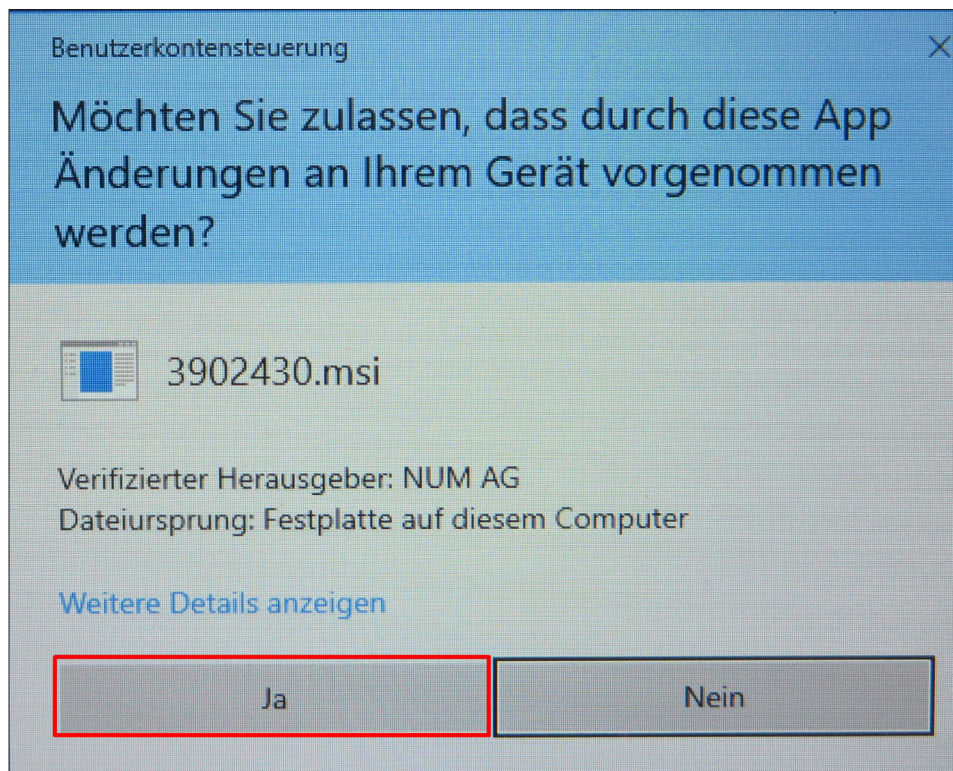
6

The NUM IIoTgateway Installation page appears.



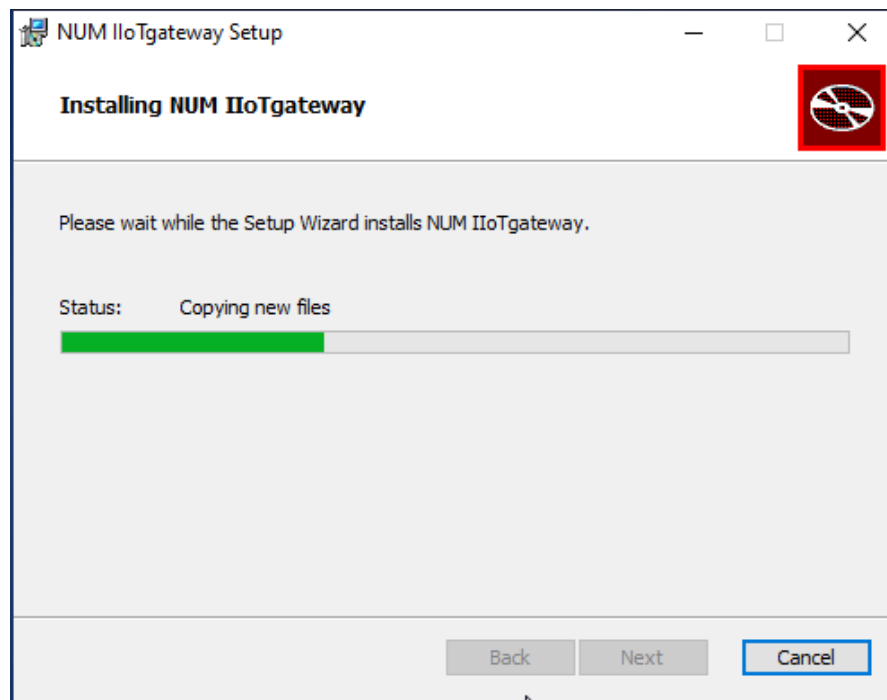
The following window appears during installation.

Press Button: Yes



The NUM IIoTgateway Installation page appears again.

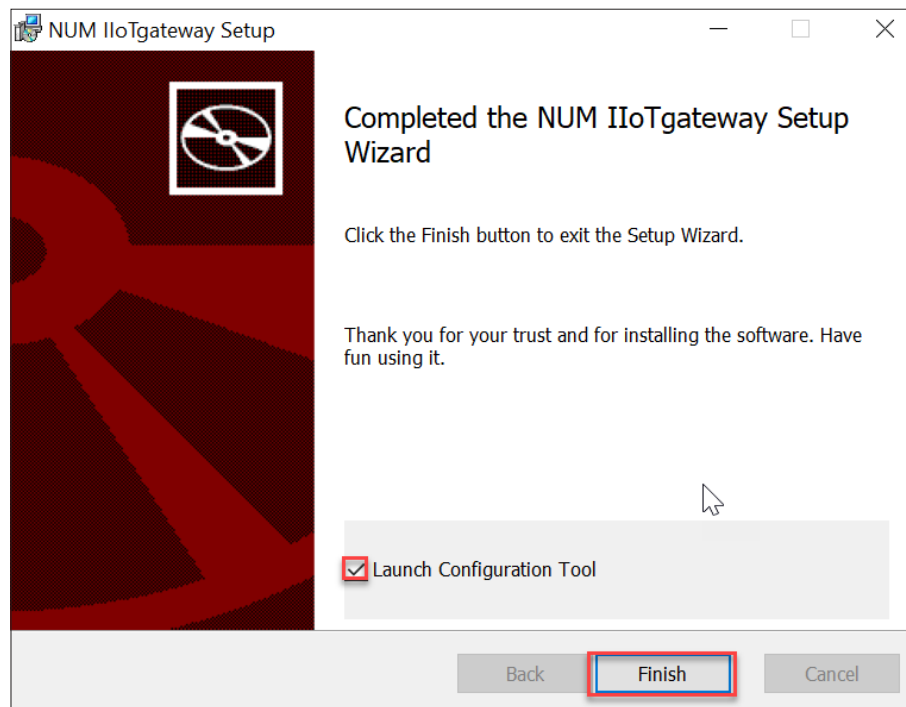
The installation status is displayed.



After some time the NUM IIoTgateway Installing End Page appears.

You may click the checkbox ⇒ Launch the NUM IIoTgateway UI Configuration Tool after installation.

Press Button: Finish ⇒ Complete the installation.



1

2

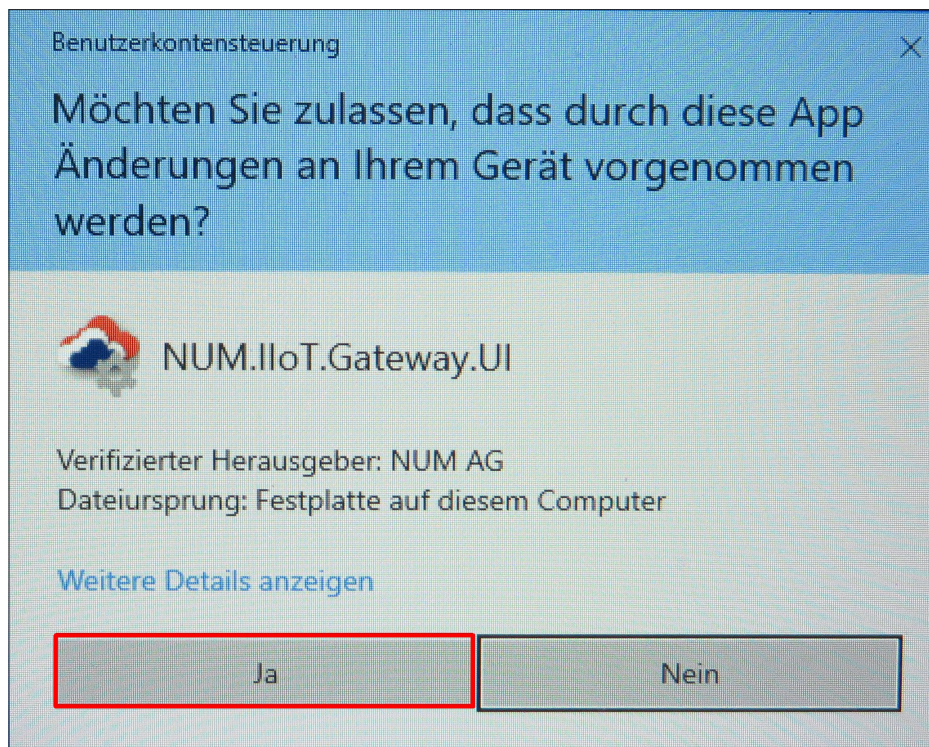
3

4

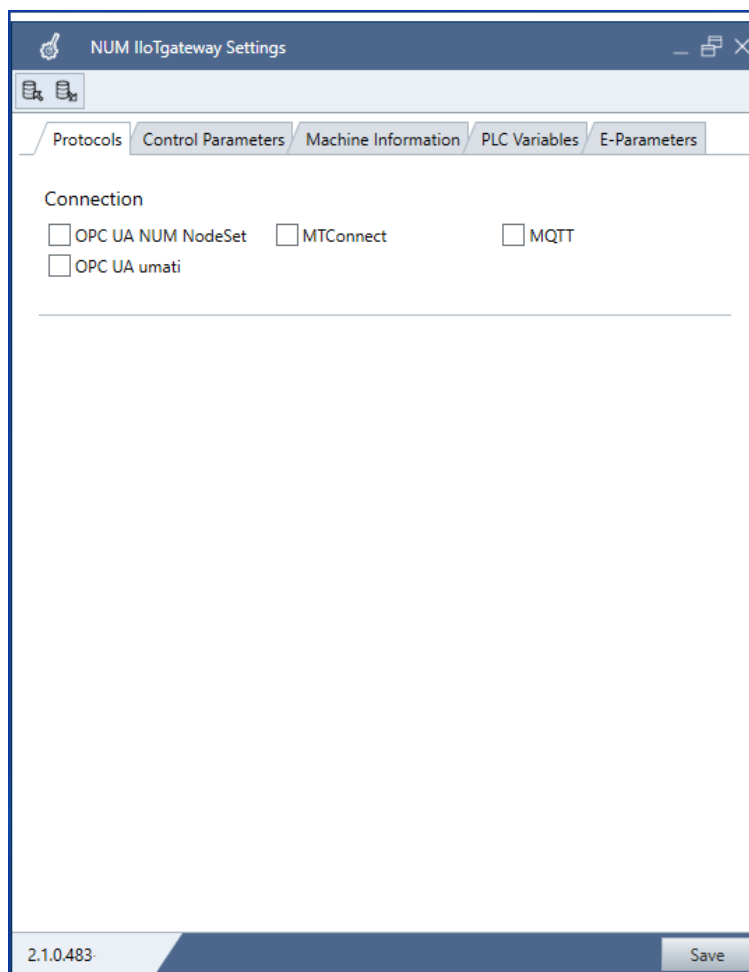
5

6

When the NUM IloTgateway UI is started for the first time after successful installation, the following window appears:



The first folder of the NUM IloTgateway UI then appears.

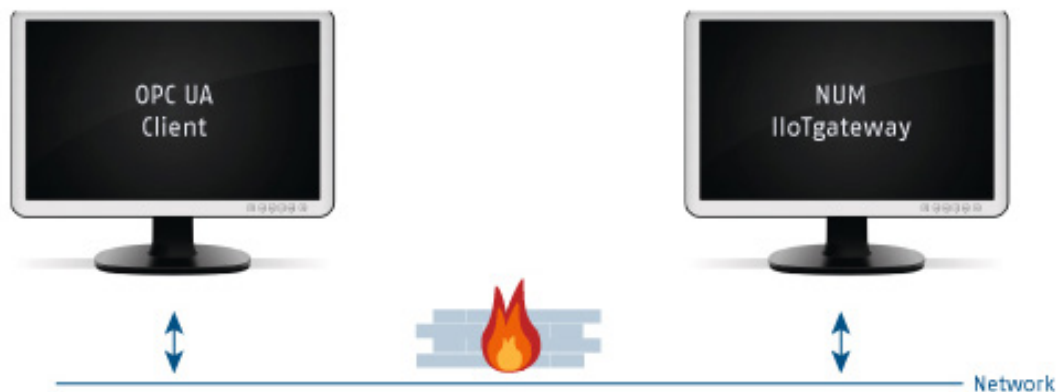


2.11 Windows Firewall Settings

The following Windows Firewall settings are exemplary settings used during development and testing of the NUM IloTgateway in the NUM network environment using OPC UA as an example.

Therefore, the Windows Firewall settings are described as guiding paths and may differ from the customer's network structure.

Settings of the NUM IloTgateway PC, so that the OPC UA Client PC has access to the NUM IloTgateway PC via the network over the UA OPC Port 48030.

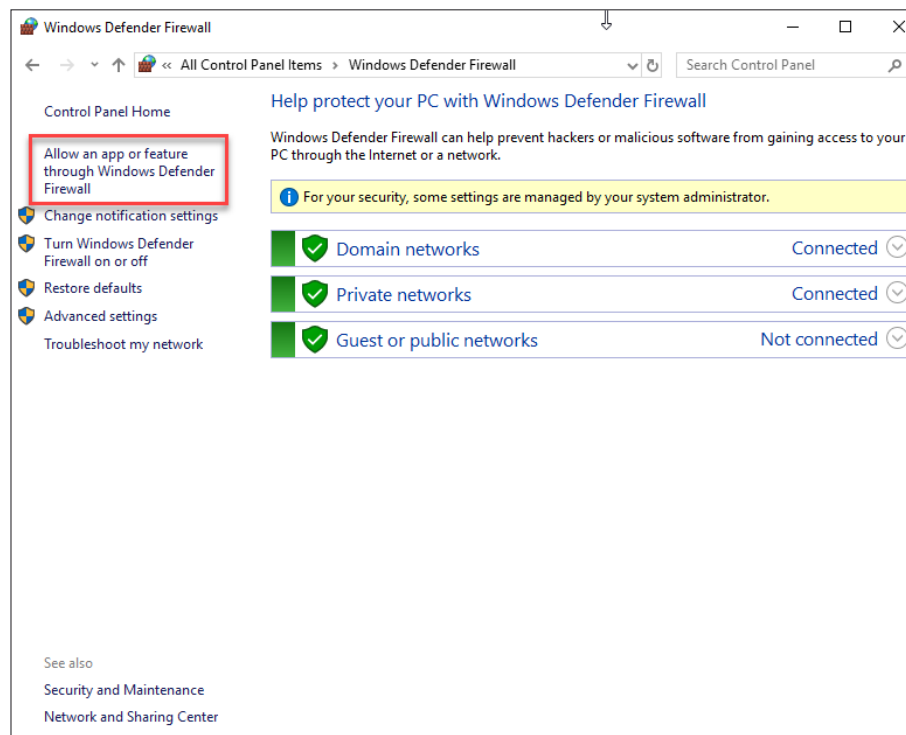


2.11.1 Windows Firewall PC Settings

Windows Firewall Settings for NUM IloTgateway Applications

Select Windows Defender Firewall

Select menu: "Allow an app or feature through Windows Defender Firewall"



1

2

3

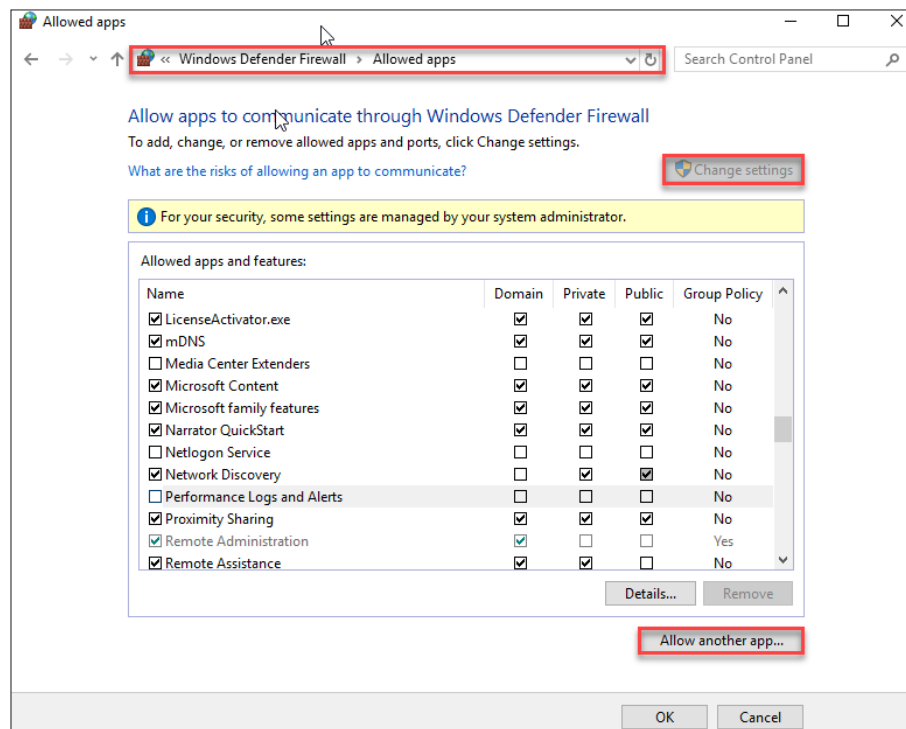
4

5

6

Press Button: Change settings

Press Button: Allow another app...

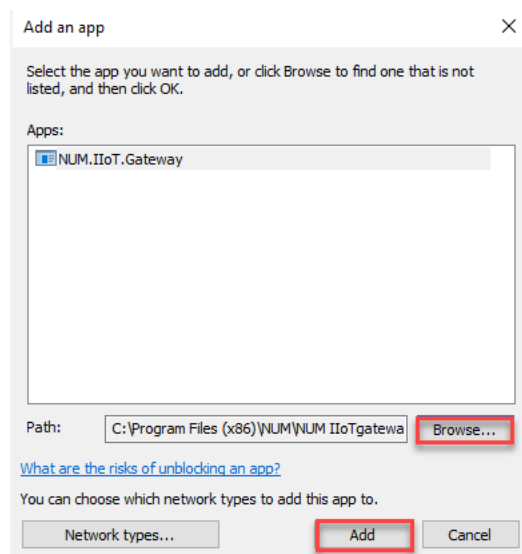
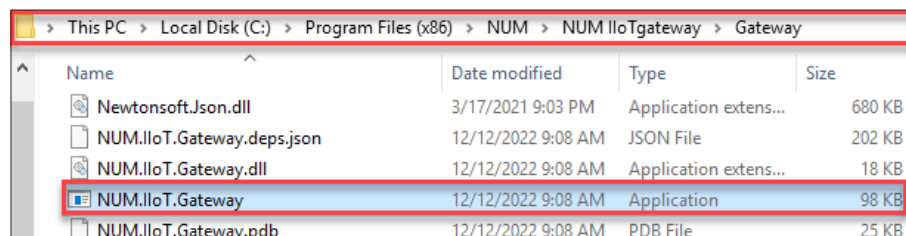


Press Button:

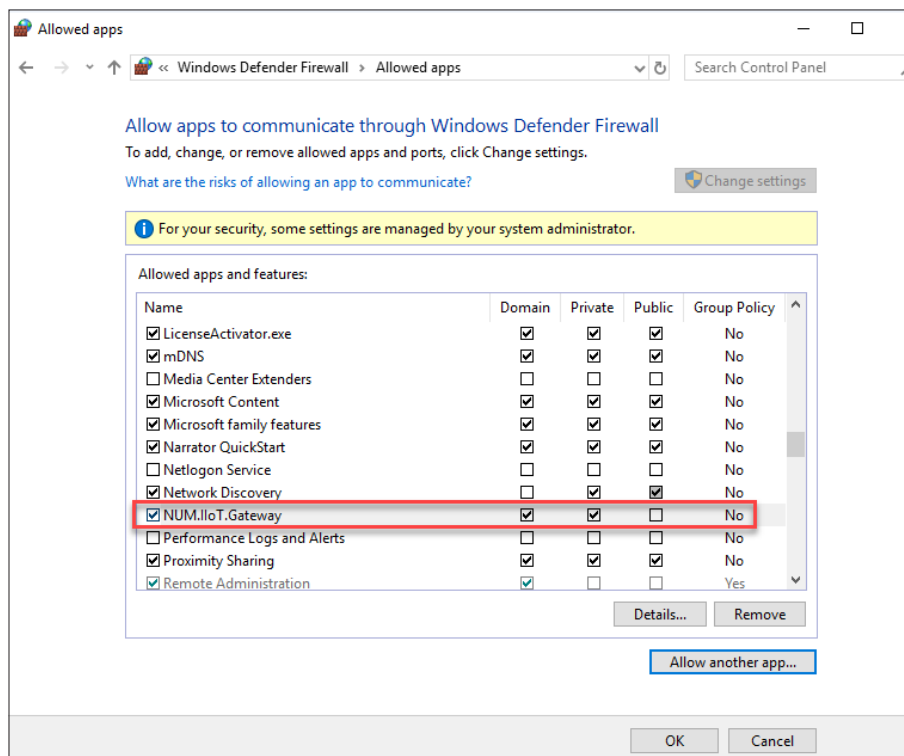
Browse ⇒ C:\Program files (x86)\NUM\NUM IloTgateway\Gateway\NUM.IloTgateway.exe

Press Button: Add the NUM.IloTgateway app to it

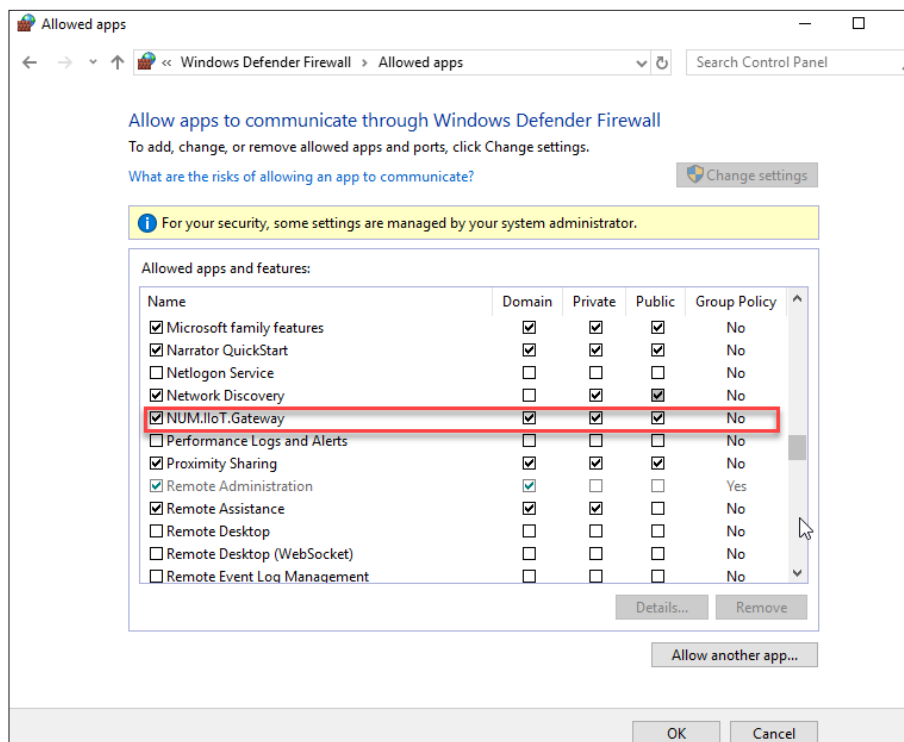
See directory structure for file NUM.IloTgateway.exe



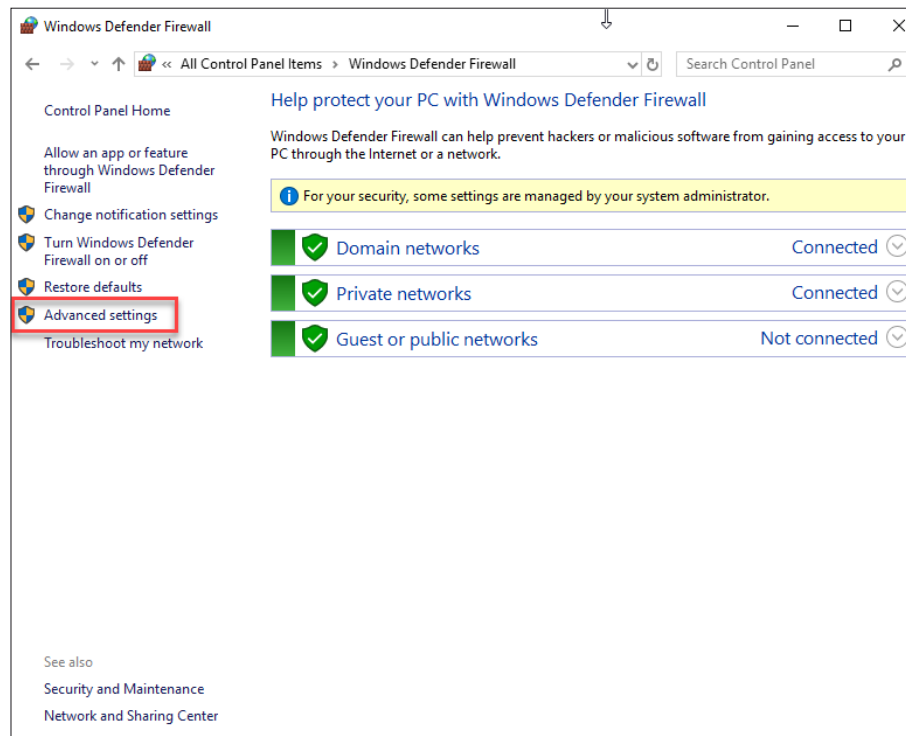
App NUM.IloTgateway has been added
Network app connections are displayed



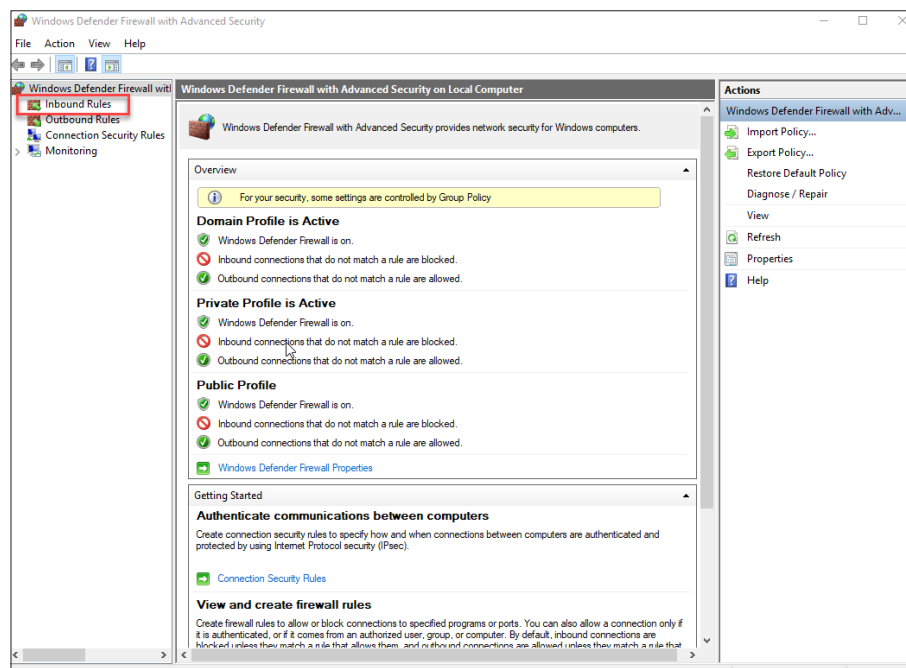
The networks Domain, Private and Public can be allowed for the NUM IloTgateway per checkbox.



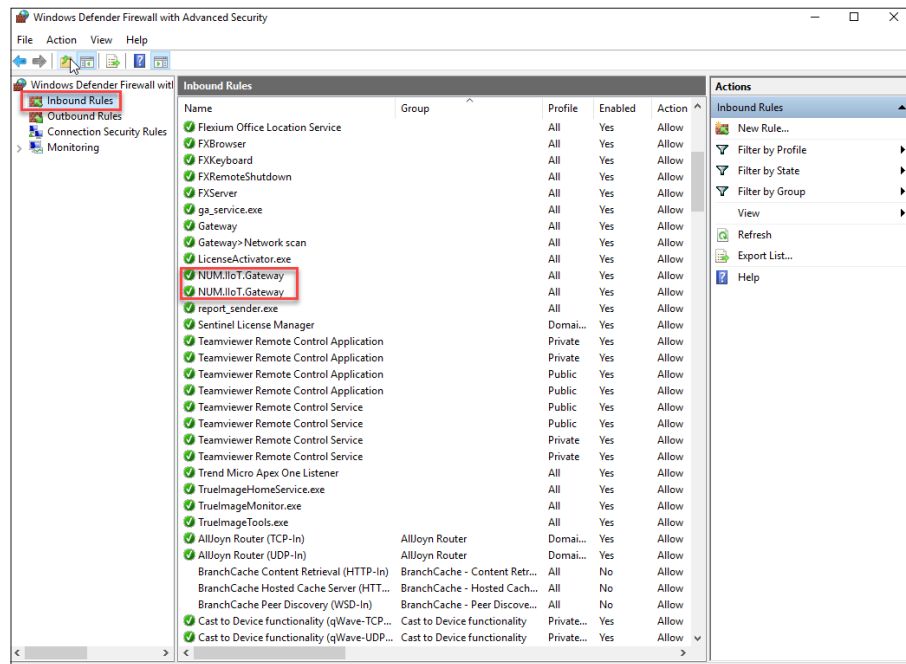
Select menu "Advanced settings"



Select menu: Inbound Rules



Select Name: NUM.IloTgateway and open with Double Click

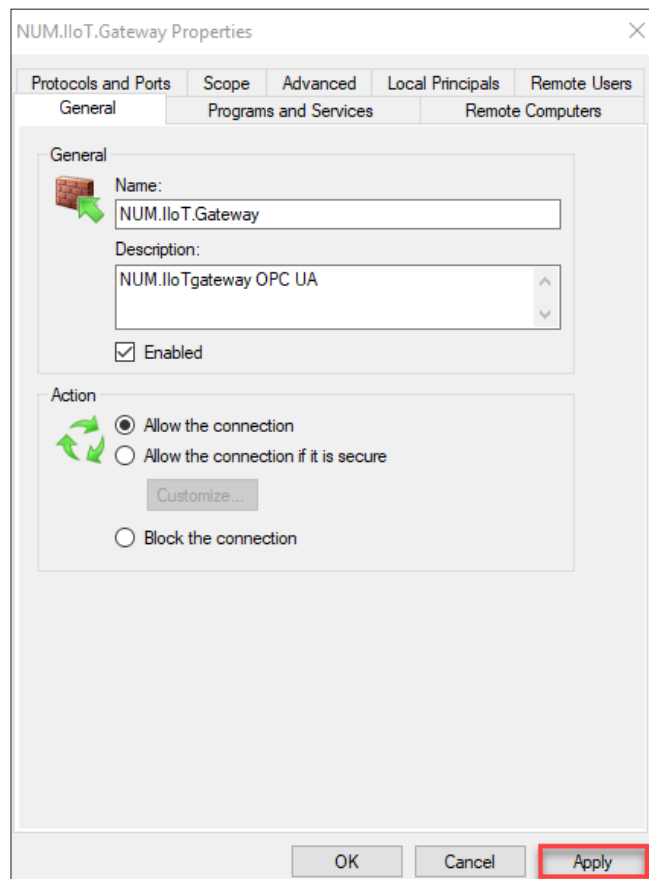


Select Tab General:

Name text box appears: NUM.IloT.Gateway

Description text box: A description can be entered.

Press Button: Apply



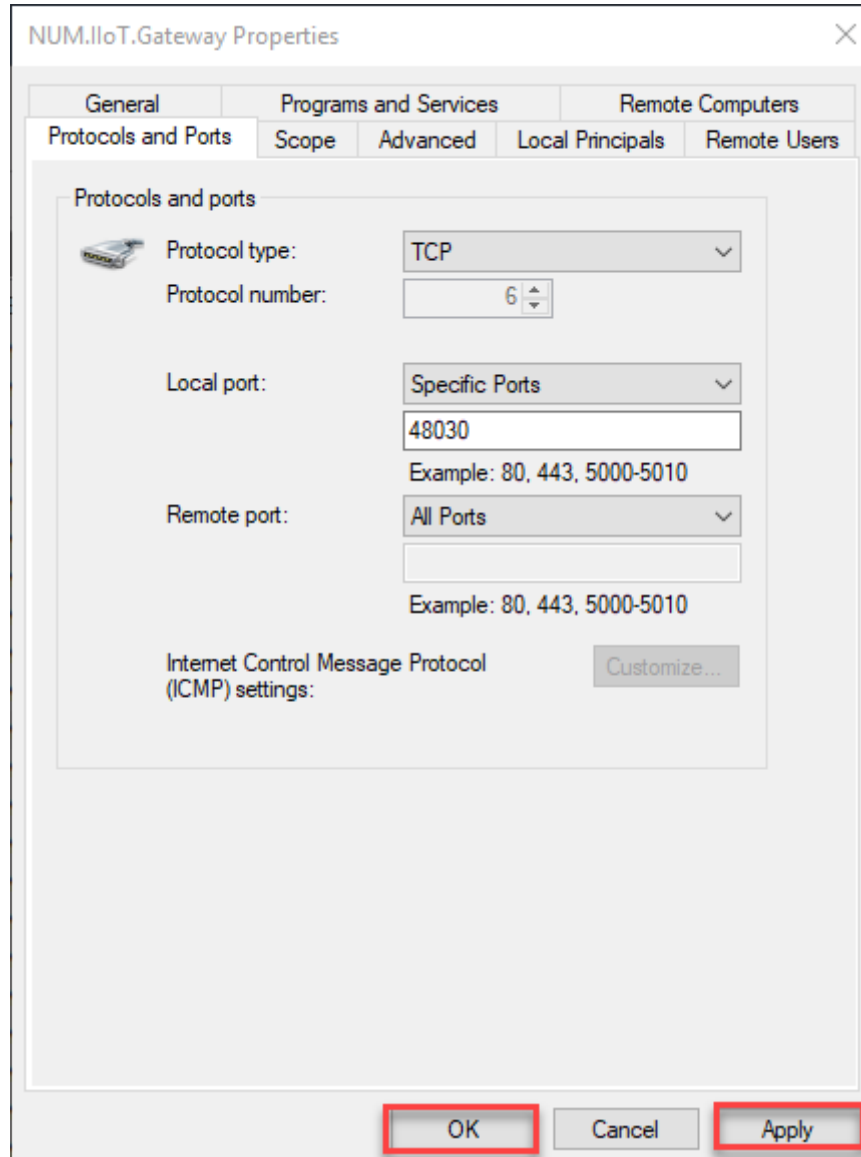
Select Tab: Protocols and Ports

Select Protocol type: Choose "TCP"

Select Local Port: Choose "Specific Ports"

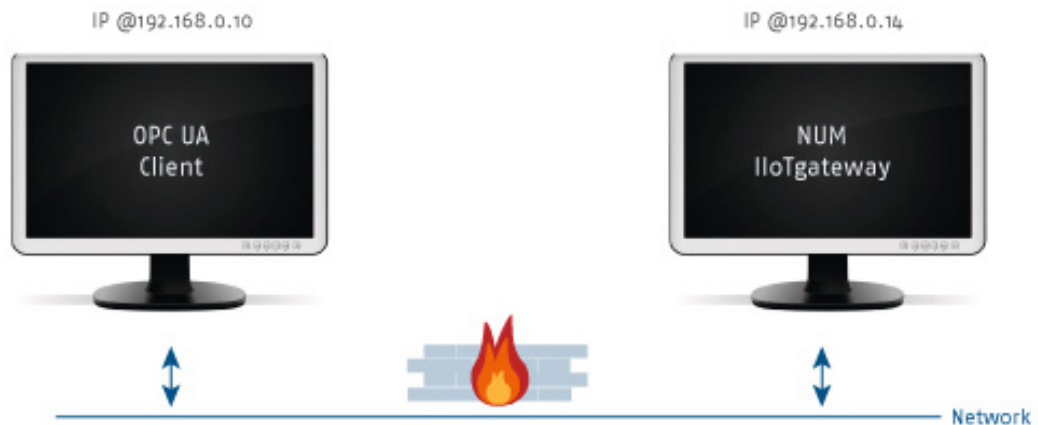
Enter in the text box: 48030 (Port for OPC UA)

Press Button: Apply - OK



2.12 Test Client Settings

2.12.1 OPC UA Client Settings



Unified Automation's UaExpert may be used as a test client. It shows the essential and exemplary settings for a connection with the NUM IloTgateway over OPC UA. OPC UA clients from other providers may therefore show deviations when applying the settings.

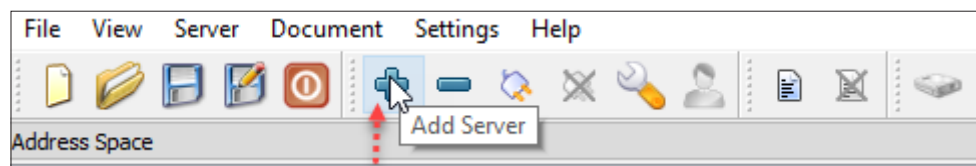
NUM IloTgateway UI

Activate the OPC UA connection in the NUM IloTgateway UI tab "Protocols" by clicking the checkbox OPC UA. Thus, the OPC UA Server is validated in the NUM IloTgateway and according settings appears.

The OPC UA settings are described in see section [2.4.1](#).

UaExpert Settings

Add NUM Server and configure the IP address and OPC UA port number for accessing the data points with the OPC UA client.



Set the Endpoint Url with the OPC UA format `opc.tcp://192.168.0.14:48030/` for IP@ + Port in the OPC UA Client.

1

2

3

4

5

6

Add Server ? X

Configuration Name

Discovery Advanced

Server Information

Endpoint Url

Security Settings

Security Policy

Message Security Mode

Authentication Settings

☒ Anonymous

☐ Username

☐ Password ☐ Store

☐ Certificate ...

☐ Private Key ...

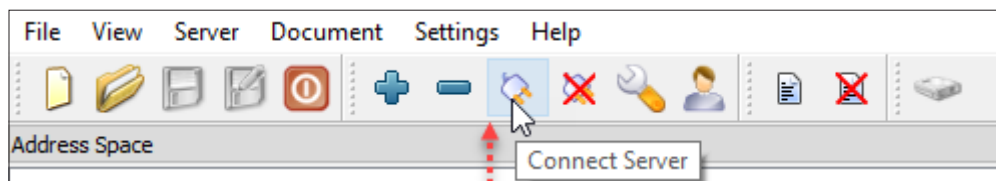
Session Settings

Session Name

☒ Connect Automatically

OK Cancel

Establish connection to the NUM IloTgateway OPC UA Server.

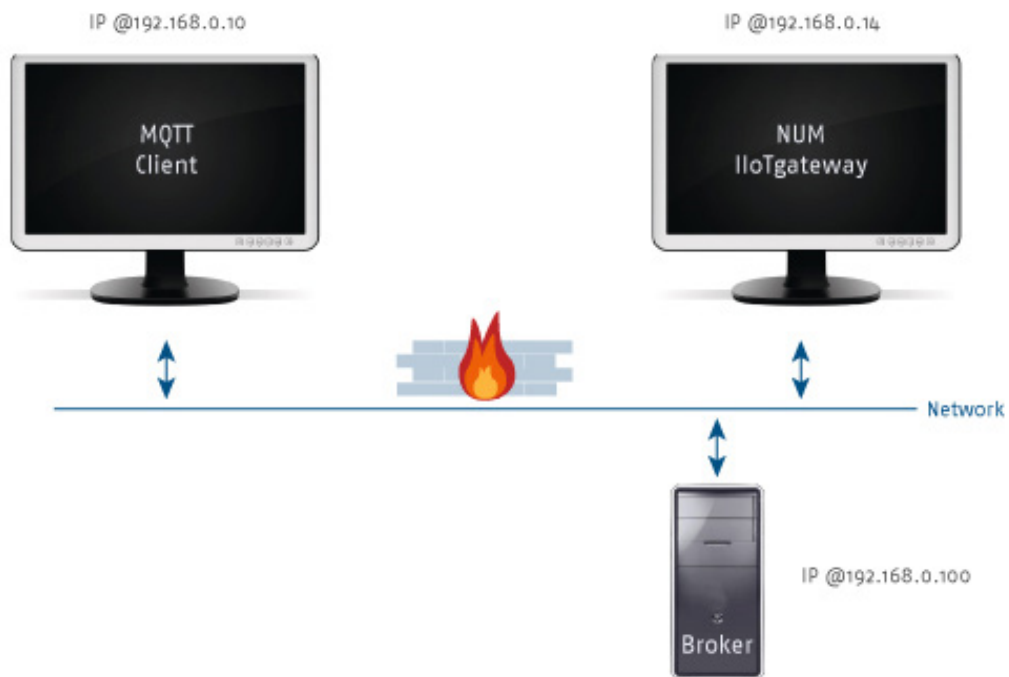


In online mode, the UaExpert should show the address space with nodes configuration and the data access view with the Node Ids, Values etc.

The screenshot displays the UaExpert interface in online mode. On the left, the 'Address Space' tree shows the hierarchy of nodes, including 'FXTBench01_MachineName', 'Machine Information', 'Machine State', 'NCK1', and 'E Parameters'. On the right, the 'Data Access View' table lists the nodes with their Node Ids, Display Names, Values, Datatypes, and Source Timestamps.

#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp
1		NS2String(MachineInformation.MachineName)	Machine Information/MachineName	FXTBench01_MachineName	String	143638.615
2		NS2String(MachineInformation.MachineSerial)	Machine Information/MachineSerial	FXTBench01_MachineSerial	String	143638.615
3		NS2String(MachineInformation.MachineType)	Machine Information/MachineType	FXTBench01_MachineType	String	143638.615
4		NS2String(MachineInformation.ProductionLine)	Machine Information/ProductionLine	FXTBench01_ProductionLine	String	143638.615
5		NS2String(MachineInformation.ProductionPlant)	Machine Information/ProductionPlant	FXTBench01_ProductionPlant	String	143638.615
6		NS2String(MachineInformation.SystemType)	Machine Information/SystemType	Flexium+ 60	String	143638.615
7		NS2String(MachineState.ActualFeedEnable)	Machine State/Actual Feed Enable	true	Boolean	143638.615
8		NS2String(MachineState.ActualStatus)	Machine State/Actual Status	Running	String	144214.597
9		NS2String(MachineState.NumberOfNCKs)	Machine State/Number of NCKs	1	Int32	143638.615
10		NS2String(MachineState.NumberOfPLCVars)	Machine State/Number of PLC Vars	14	Int32	143638.615
11		NS2String(MachineState.NCK1.ActualFeedBack)	Machine State/NCK1/ActualFeedBack	163002 L18 N40	String	144340.547
12		NS2String(MachineState.NCK1.Ch1.ActFeedEnable)	Machine State/NCK1/Ch1/ActualFeedEnable	true	Boolean	143638.615
13		NS2String(MachineState.NCK1.Ch1.ActFeedOverride)	Machine State/NCK1/Ch1/ActualFeedOverride	0	Int32	144341.608
14		NS2String(MachineState.NCK1.Ch1.ActMainProgramNumber)	Machine State/NCK1/Ch1/ActualMainProgramNumber	163002	String	144341.608
15		NS2String(MachineState.NCK1.Ch1.ActOperatorMessage)	Machine State/NCK1/Ch1/ActualOperatorMessage	Exxxxx Init	String	144111.360
16		NS2String(MachineState.NCK1.Ch1.ActProgramComment)	Machine State/NCK1/Ch1/ActualProgramComment	M001T Read Eparameter	String	143638.615
17		NS2String(MachineState.NCK1.Ch1.ActProgramFileName)	Machine State/NCK1/Ch1/ActualProgramFileName	m00t_PPMainParameter.xpi	String	143953.104
18		NS2String(MachineState.NCK1.Ch1.ActSpindleSpeed)	Machine State/NCK1/Ch1/ActualSpindleSpeed	0	Double	143638.615
19		NS2String(MachineState.NCK1.Ch1.ProgrammedFeedrate)	Machine State/NCK1/Ch1/ProgrammedFeedrate	300	Double	144030.850
20		NS2String(MachineState.NCK1.Ch1.ProgrammedSpindleSpeed)	Machine State/NCK1/Ch1/ProgrammedSpindleSpeed	0	Double	143638.615
21		NS2String(MachineState.NCK1.Ch1.SpindleOverride)	Machine State/NCK1/Ch1/SpindleOverride	70	Int32	144015.098
22		NS2String(MachineState.NCK1.aParam3)	Machine State/NCK1/ActChannel	0	Int16	143638.615
23		NS2String(MachineState.NCK1.aParam3)	Machine State/NCK1/ActChannel	18	Int16	144340.610
24		NS2String(MachineState.NCK1.aParam3)	Machine State/NCK1/CNC Mode	0	Int16	143935.598
25		NS2String(MachineState.NCK1.aParam3)	Machine State/NCK1/Feedrate C...	0	Int32	144341.608
26		NS2String(MachineState.NCK1.aParam3)	Machine State/NCK1/Get E00001	123.456	Double	144245.609
27		NS2String(MachineState.NCK1.aParam3)	Machine State/NCK1/Get E00020	111.222	Double	143638.610
28		NS2String(MachineState.NCK1.aParam3)	Machine State/NCK1/Get E00001	456.789	Double	144245.609
29		NS2String(MachineState.NCK1.aParam3)	Machine State/NCK1/Get E00020	222.333	Double	144341.610
30		NS2String(MachineState.NCK1.aParam3)	Machine State/NCK1/NCK Serial...	2550000	Int32	143638.615
31		NS2String(MachineState.NCK1.aParam3)	Machine State/NCK1/PositionX	206768.75	Double	144341.608
32		NS2String(MachineState.NCK1.General.ActOperationMode)	Machine State/NCK1/General/Act...	Auto	String	143935.598
33		NS2String(MachineState.NCK1.General.PreSelectedOperationMo...	Machine State/NCK1/General/Pr...	NoMode	String	143638.615
34		NS2String(MachineState.NCK1.NumberOfEParams)	Machine State/NCK1/Number of...	10	Int32	143638.615
35		NS2String(MachineState.NCK1.NumberOfChannels)	Machine State/NCK1/Number of...	3	Int32	143638.615
36		NS2String(MachineState.NCK1.aParam7)	Application.General.CncFunction...	0	Int32	143638.615
37		NS2String(MachineState.NCK1.aParam7)	Application.General.CncFunction...	0	Int32	143638.615
38		NS2String(MachineState.NCK1.aParam7)	Application.General.CncFunction...	0	Int32	143638.615
39		NS2String(MachineState.NCK1.aParam10)	Application.General.CncFunction...	0	Int32	143638.615
40		NS2String(MachineState.NCK1.aParam11)	Application.General.CncFunction...	0	Int32	143638.615
41		NS2String(MachineState.NCK1.aParam12)	Application.General.CncFunction...	0	Int32	143638.615
42		NS2String(MachineState.NCK1.aParam13)	Application.General.CncFunction...	0	Int32	143638.615
43		NS2String(MachineState.NCK1.aParam14)	Application.General.CncFunction...	0	Int32	143638.615
44		NS2String(MachineState.NCK1.aParam14)	Application.General.CncFunction...	0	Int32	143638.615
45		NS2String(MachineState.NCK1.aParam14)	Application.General.CncFunction...	0	Int32	143638.615
46		NS2String(MachineState.NCK1.aParam14)	Application.General.CncFunction...	0	Int32	143638.615
47		NS2String(MachineState.NCK1.aParam14)	Application.General.CncFunction...	0	Int32	143638.615
48		NS2String(MachineState.NCK1.aParam14)	Application.General.CncFunction...	0	Int32	143638.615
49		NS2String(MachineState.NCK1.aParam14)	Application.General.CncFunction...	0	Int32	143638.615
50		NS2String(MachineState.NCK1.aParam14)	Application.General.CncFunction...	0	Int32	143638.615
51		NS2String(MachineState.NCK1.aParam14)	Application.General.CncFunction...	0	Int32	143638.615
52		NS2String(MachineState.NCK1.aParam14)	Application.General.CncFunction...	0	Int32	143638.615
53		NS2String(MachineState.NCK1.aParam14)	Application.General.CncFunction...	0	Int32	143638.615
54		NS2String(MachineState.NCK1.aParam14)	Application.General.CncFunction...	0	Int32	143638.615
55		NS2String(MachineState.NCK1.aParam14)	Application.General.CncFunction...	0	Int32	143638.615
56		NS2String(MachineState.NCK1.aParam14)	Application.General.CncFunction...	0	Int32	143638.610

2.12.2 MQTT Client Settings



The MQTT Explorer may be used as a test client. It shows the essential and exemplary settings for a connection with the NUM IloTgateway over MQTT. MQTT clients from other providers may therefore show deviations when applying the settings.

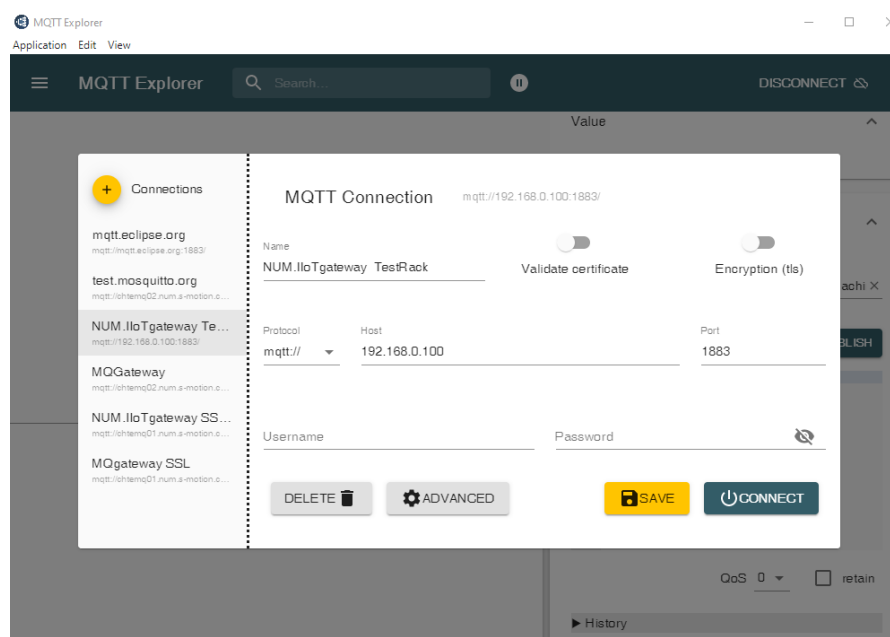
NUM IloTgateway UI

Activate the MQTT connection in the NUM IloTgateway UI tab “Protocols” by clicking the checkbox MQTT. Thus, the MQTT Server is validated in the NUM IloTgateway and according settings appears.

The MQTT settings are described in see section [2.4.1](#).

MQTT Explorer Settings

Configuration of the connection to the MQTT Broker (Server address and Port number):



In online mode, the MQTT Explorer should show the topics tree etc.

The screenshot displays the MQTT Explorer application interface. The main window is titled "MQTT Explorer" and includes a menu bar (Application, Edit, View), a search bar, and a status bar (DISCONNECT, a green user icon, and a refresh icon). The interface is divided into several sections:

- Topics Tree (Left):** A hierarchical view of MQTT topics. The selected topic is `FXTBench01_295090`, which contains a sub-topic `MachineInformation` (9 topics, 9 messages). Under `MachineInformation`, there is a `MachineState` sub-topic, which further contains `ActAlarms`, `ActFeedEnable`, `ActFeedRate`, `ActMainProgramName`, `ActOperatorMessage`, `ActProgramComment`, `ActProgramFileName`, `ActSpindleSpeed`, and `ProgrammedFeedRate`. The `ActCallStack` sub-topic is also visible under `MachineState`.
- Message Details (Right):** A panel showing the details of the selected message. It includes a "Topic" field with a trash icon, a "Value" field with a trash icon, and a "Publish" button. The message content is a JSON object: `{ "timestamp": "2023-03-27T13:08:42.4907626Z", "value": "%3002 L18 N40" }`. The QoS is 0, and the message is marked as "RETAINED". The timestamp is 27.03.2023 15:31:41.
- Data Plots (Bottom):** Two plots are shown. The left plot is titled "value" and the right plot is titled "value.NrProdData.ToolNum...". Both plots show a "No Data" message on a y-axis ranging from -1 to 1.

1

2

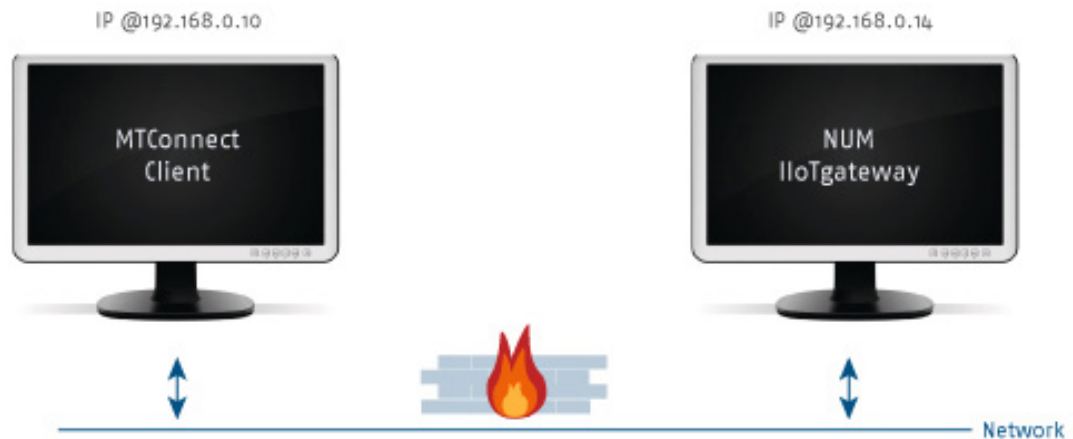
3

4

5

6

2.12.3 MTConnect Client Settings



The MTExplorer may be used as a test client. It shows the essential and exemplary settings for a connection with the NUM IIoTgateway over MTConnect. MTConnect clients from other providers may therefore show deviations when applying the settings.

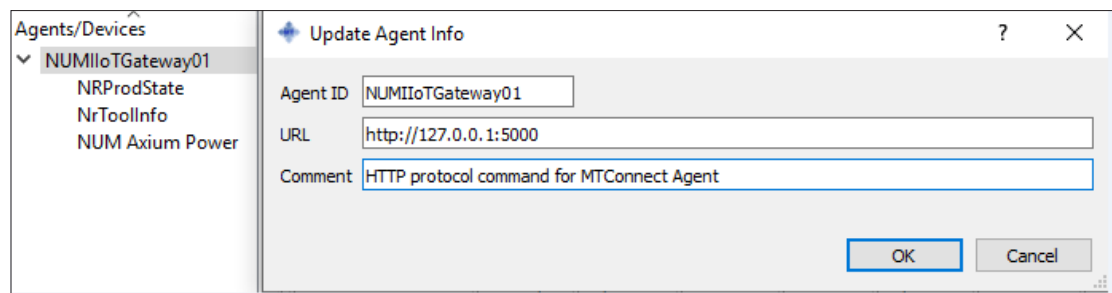
NUM IIoTgateway UI

Activate the MTConnect connection in the NUM IIoTgateway UI tab "Protocols" by clicking the checkbox MTConnect. Thus, the MTConnect Server is validated in the NUM IIoTgateway and according settings appears.

The MTConnect settings are described in see section [2.4.1](#).

MTExplorer Settings

HTTP Protocol Setting with localhost IP address and Portnumber



In online mode, the MTExplorer should show the components etc.

Device: NUM Axiom Power; UUID: 49cb009a-47b5-433c-9299-8b333763a797

Device : NUM Axiom Power

Events

Timestamp	Key	Name	Sub Type	Comp Id	Id	Sequence	Value
2023-03-27 13:56:06	Availability	Availability			Availability	1	AVAILABLE
2023-03-27 13:56:54	Message	Sign of Life			SignOfLife	1707	1

Condition

Timestamp	Key	Name	Type	Sub Type	Native Code	Id	Sequence	Value
2023-03-27 13:56:05	Fault	AlarmEventPool	SYSTEM		2[0]	099bc29d-6548-461c-aa06-d64713e21778	1648	Die batterien müssen erneuert werden

Component : Machine Information

Events

Timestamp	Key	Name	Sub Type	Comp Id	Id	Sequence	Value
2023-03-27 13:56:06	SystemType	Machine Information/System Type			MachineInformation.SystemType	5	Axiom Power Advanced
2023-03-27 13:56:06	ProductionPlant	Machine Information/Production Plant			MachineInformation.ProductionPlant	6	AxiomProductionPlant
2023-03-27 13:56:06	ProductionLine	Machine Information/Production Line			MachineInformation.ProductionLine	7	AxiomProductionLine
2023-03-27 13:56:06	MachineType	Machine Information/Machine Type			MachineInformation.MachineType	8	AxiomMachineType
2023-03-27 13:56:06	MachineSerial	Machine Information/Machine Serial			MachineInformation.MachineSerial	9	AxiomMachineSerial
2023-03-27 13:56:06	MachineName	Machine Information/Machine Name			MachineInformation.MachineName	10	Axiom Power TRack2
2023-03-27 13:56:06	Variable1	Machine Information/Variable 1			MachineInformation.Variable1	11	AxiomVariable1

2.12.4 Monitoring of Communication

The NUM IloTgateway contains the signal SignOfLife. This monitors the communication chain NUM CNC ⇌ NUM IloTgateway ⇌ OPC UA (2.5.3), MQTT (2.7.3) and MTConnect clients (2.8.3).

The signal SignOfLife changes its state (0-1-0-1 etc.) in a cycle of approx. 1s.

When the signal changes in this temporal pattern, the communication is stably present. If the signal does not change in this temporal pattern, communication is interrupted.

Address Space

No Highlight

- Root
 - Objects
 - NUM
 - FXTBench01_MachineName
 - HistoricalEventFilter
 - Machine Information
 - Machine State
 - Machine State/Actual Alarms
 - Sign of Life** (highlighted with a red arrow)
 - HistoricalEventFilter
 - NrProdState
 - NrToolInfo
 - getConfigData
 - Server
 - Types
 - Views

NS2 String SignOfLife	Sign of Life	1
NS2 String SignOfLife	Sign of Life	0

Page deliberately empty

Summary

3	MQgateway	75
3.1	System overview	75
3.2	General notes.....	75
3.3	Program states.....	76
3.4	Settings	77
3.4.1	General settings.....	77
3.4.2	Machine information.....	78
3.4.3	PLC variables.....	78
3.4.4	E-Parameter variables	79
3.5	Topics and values	80
3.5.1	Lifecycle topics.....	80
3.6	Machine information.....	81
3.7	Machine State	82
3.8	NUMroto.....	84
3.9	Troubleshooting / Diagnostics.....	85

1

2

3

4

5

6

Page deliberately empty

3 MQgateway

3.1 System overview

The main goal of this MQgateway is to publish data of a *NCK* to a *MQTT Broker*.

Participating in this data chain are the members: *NCK*, *IPC / Box PC*, *APServer / FXServer*, and *MQTT Broker*.

The *NCK* is the source of most of the published data. Currently supported are only Flexium Systems above version 3.6.0.0 and Axiom Power above version R2 with Windows 7.

The *industrial PC (iPC)* or *Box PC* is the host of the *MQgateway*. Since the *APServer / FXServer* currently runs only on Windows, the *MQgateway* runs only on the respective PC also.

MQgateway is a special *MQTT* client which collects all data from different sources. Most of the data are continuous updating while machining, but there are some static fields, which are updated only once.

There is also an additional source where data is collected from an *xml*-file. After collection, the data is packed with a timestamp to a *JSON object*.

This collected and already converted data is published via the *MQTT* protocol to a *MQTT Broker*.

A message broker is a service in between multiple clients. In this case multiple clients which understand the protocol *MQTT*. The broker itself isn't interested on which topics clients publish respective subscribe, his only purpose is to relay messages from one publisher to multiple subscribers.

Every client can be a publisher, subscriber, or both. (The *MQgateway* has no subscription capabilities at the moment, i.e. it is not possible to receive commands from somewhere).

In some cases the broker is used to translate the payload from one or multiple topics to another format or topics and publishes them on himself or on another broker.

3.2 General notes

As a basic requirement for Axiom Power systems *APServer* and for Flexium systems *FXServer* has to be installed on the same machine as *MQgateway*.

Without this requirements met, *MQgateway* is not able to publish any data to *MQTT* topics.

The *MQgateway* application runs as a Windows application and is minimized on the tray icon bar.

A settings menu is available from context menu accessible through a right click on the icon. This menu should be available whether *APServer / FXServer* are installed.

If *APServer / FXServer* are installed and the licence is available, *MQgateway* starts publishing data through *MQTT* topics to the broker.

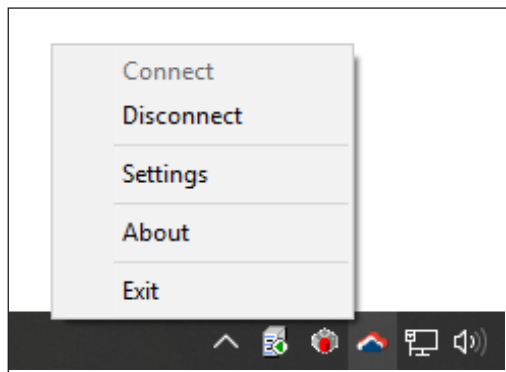
MQTT topics are in the form *"/prefix/Id/topic/subtopic"*.

3.3 Program states

The application can have seven different states.

	Unknown: Transition while starting up.
	Connected: Connection established between MQgateway and broker.
	Disconnected: No connection between MQgateway and broker.
	Unlicenced: Missing MQgateway licence.
	Error: Only possible if the error is not recoverable.
  Flashing	Connecting / Disconnecting: Flashing icon signals the establishment or termination of the connection.

3.4 Settings



3.4.1 General settings

The settings are accessible through a context menu.

On the first tab General settings, Group MQTT broker, one can adjust the IP-address or server name and port.

If the port is empty or 0, it uses internally the default port for *MQTT* which is 1883.

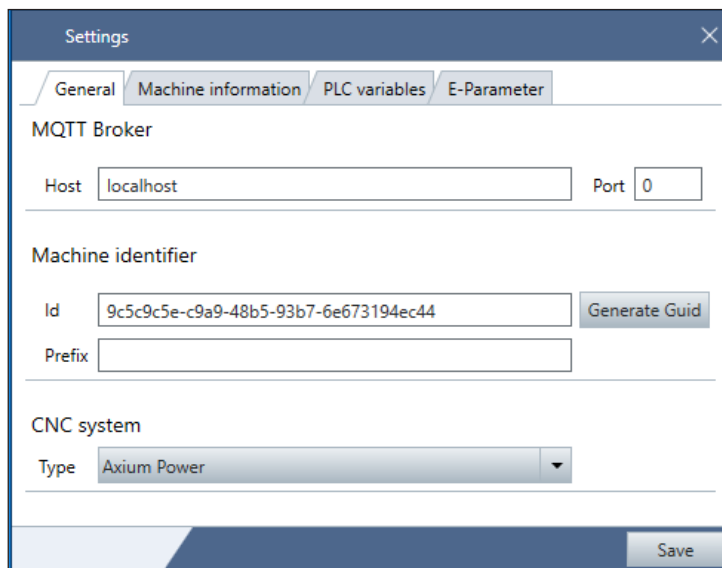
The General settings, Group Machine identifier *ID* is the unique identifier of this publishing client. To make sure the topics are not overlapping other applications, it is possible to set a prefix. This prefix can also be empty, in which case the first level is the ID.

Otherwise the first level is the prefix and the second level is this ID.

It's possible to select between Axiom Power and Flexium / Flexium*.

The default selection is Automatic.

In this mode MQgateway tries to connect to a Flexium system first, and if not present to an Axiom system.



1

2

3

4

5

6

3.4.2 Machine information

On the second tab *Machine information*, it is possible to set some information about this machine which get published on the respective topics.

The screenshot shows a 'Settings' dialog box with a close button (X) in the top right corner. It has four tabs: 'General', 'Machine information' (which is selected and highlighted with a dashed border), 'PLC variables', and 'E-Parameter'. The 'Machine information' tab contains several text input fields for the following labels: 'Production plant', 'Production line', 'Machine serial', 'Machine type', 'Machine name', 'Variable 1', 'Variable 2', and 'Variable 3'. A 'Save' button is located at the bottom right of the dialog box.

3.4.3 PLC variables

On the third tab *PLC variables*, it is possible to add some PLC watch variables.

Please note: It is not possible to add the same variable twice.

Also note it is only possible to watch 10 variables simultaneously.

The screenshot shows the 'Settings' dialog box with the 'PLC variables' tab selected. It features a table with two columns: 'Parameter' and 'Comment'. The first row of the table contains the text 'RCNC.Axis[0].Status' in the 'Parameter' column. Below the table, there are two buttons: 'Add' and 'Delete'. A 'Save' button is located at the bottom right of the dialog box.

Parameter	Comment
RCNC.Axis[0].Status	

3.4.4 E-Parameter variables

On the fourth tab *E-parameter variables*, one can add E-Parameter which should be watched.

Please note: it is also not possible to add the same parameter twice.

Also note it is only possible to watch 15 parameters simultaneously.



Enter the E-Parameter without E into the Parameter field.

Parameter	Comment
70000	X-Axis Position

These settings are saved in a config file “C:\ProgramData\NUM\MQgateway\Config\MQgateway.config”.

The config file is in an *xml* like format and the constraints are the same as in the settings editor:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <sectionGroup name="mqGatewaySettings" type="Num.MqGateway.Models.AppSettings, MQgateway" >
      <section name="cncSystem" type="Num.MqGateway.Models.Settings.CncSystemSettings, MQgateway" />
      <section name="client" type="Num.MqGateway.Models.Settings.ClientSettings, MQgateway" />
      <section name="machineInformation" type="Num.MqGateway.Models.Settings.MachineInformation, MQgateway" />
      <section name="mqttServer" type="Num.MqGateway.Models.Settings.MqttServerSettings, MQgateway" />
      <section name="sourceSettings" type="Num.MqGateway.Models.Settings.SourceSettings, MQgateway" />
    </sectionGroup>
  </configSections>
  <mqttPublisherSettings>
    <cncSystem type="Axium"/>
    <mqttServer serverName="localhost" port="0"/>
    <client guid="8fd7aa94-9298-4d1f-9fd7-257158941500" prefix="num"/>
    <machineInformation productionPlant="NUM AG - Test Publisher" productionLine="" machineType="" machineSerial=""
machineName="" Var1="" Var2="" Var3=""/>
  </mqttPublisherSettings>
  <sourceSettings>
    <plcVariables>
      <plcVariable parameter="RCNC.Axis[0].Status" comment=""/>
      <plcVariable parameter="RCNC.Axis[1].Status" comment=""/>
    </plcVariables>
    <eParameters>
      <eParameter parameter="70000" comment="X-Axis Position"/>
      <eParameter parameter="75000" comment=""/>
      <eParameter parameter="78000" comment=""/>
    </eParams>
  </sourceSettings>
</configuration>
```

3.5 Topics and values

As stated before, topics are starting always with a slash “/”.

The prefix if present comes on the first level “/prefix”. If no prefix is chosen, the first level is the Id from General Settings, Machine Identifier.

All topics are prefixed with this combination, i.e. “/prefix/Id”.

All topics have the retain flag set, that means if someone connects to the broker after a topic was published, he gets the latest information from the broker.

The payload of each topic is in *JSON format*. Depending on the topic it could be a list of payloads or just a single payload.

Every payload has a timestamp in the format “2018-08-08T07:53:11.218Z”. The timestamp is in UTC.

There is also a property value which contains the actual data.

A simple payload with a string looks like this:

```
{"timestamp":"2018-08-08T07:52:00.541Z","value":"Some string"}
```

And a payload with a number like this:

```
{"timestamp":"2018-08-08T07:52:00.556Z","value":100.4553}
```

It is also possible to have complex data values like this:

```
{"timestamp":"2018-08-08T07:52:57.606Z","value":{"parameter":"75000","value":47373.9324}}
```

The topics are grouped in three logical units; they are described in the following sections.

3.5.1 Lifecycle topics

To make it easier to debug or other use cases there is a lifecycle topic.

The payload “Online” or “Offline” is published to the topic “/prefix/Id/status”.

It is not guaranteed, that a message is only sent once. But it does not matter if a disconnection is intended or not.

“Offline” is sent if the connection is disrupted either by a hardware or software error.

3.6 Machine information

The machine information topic contains mostly static information from the configuration file.

The topics are accessible through the path *“/prefix/Id/MachineInformation/topic”*.

The topics are as follows and the datatype reflects the type of the value (without timestamp).

Topic	Datatype	Comment
ProductionPlant	String	From configuration
ProductionLine	String	From configuration
MachineType	String	From configuration
MachineSerial	String	From configuration
MachineName	String	From configuration
SystemType	String	CNC identifier from CNC. E.g. Flexium+ 68
Variable1	String	From configuration
Variable2	String	From configuration
Variable3	String	From configuration

3.7 Machine State

The machine state topic contains actual data for multiple channels.

The topics are accessible through the path *“/prefix/Id/MachineState/topic”*.

Topic	Datatype	Comment
ActStatus	String	Values: Unknown, Idle, Running, Paused, Error
ActFeedEnable	bool	
ActAlarms	Array[Alarm]	A JSON Array with the type Alarm
NumberOfPLCVars	int	0 to 10
NumberOfNCK	int	At the moment the value is 1 always.
NCK		
PLCVar		

Type Alarm

```
{"extension":0,"number":2007,"text":"Flexium System was not shutdown correctly!
```

```
Previous Zone 0 files are transferred to the NCK.", "type":"RTSMessage"}
```

SPSVar

Is also a subtopic of MachineState whose elements are numbers between 1 and the value of *“NumberOfSPSVars”*. E.g. *“/prefix/Id/MachineState/PLCVar/1”*.

The value of such an element has the format as follows, where the parameter is the subscribed variable.

```
{"timestamp":"2018-08-
```

```
08T07:18:30.377Z","value":{"parameter":"Application.IoConfig_Globals.Flexium_NCK.RCNC.Axis[0].Status","value":1}}
```

NCK

Is a subtopic of MachineState whose elements are numbers between 1 and the value of *“NumberOfNCK”*. E.g. *“/prefix/Id/MachineState/NCK/1/topic”*.

Topic	Datatype	Comment
NumberOfChannels	int	
NumberOfEparams	int	
eParam		
Channel		
General		

eParam

The E-parameters are accessible through `"/prefix/Id/MachineState/NCK/1/eParam/number"` where the number is between 1 and *NumberOfEparams*.

The value is the same complex type as for SPSVar, the parameter is the subscribed *eParam number*.
`{"timestamp":"2018-08-08T07:53:06.429Z","value":{"parameter":"70000","value":1044.1040308}}`

Channel

The topics of the corresponding channel are accessible through `"/prefix/Id/MachineState/NCK/1/Channel/number/topic"` where the number is between 1 and *NumberOfChannels*.

A channel has the following topics:

Topic	Datatype	Comment
ActFeedrate	double	
ProgrammedFeedRate	double	
ActFeedrateOverride	int	
ActFeedEnable	bool	
ActSpindleSpeed	double	
ProgrammedSpindleSpeed	double	
SpindleOverride	int	
ActMainProgramName	string	
ActProgramComment	string	
ActCallStack	string	

General

General information is accessible through `"/prefix/Id/MachineState/NCK/1/General/topic"`.

Topic	Datatype	Comment
ActOperationMode	string	Values: Unknown, Auto, Single, MDI, Rapid, Search, Test, Manual, Home, Shifts, NoMode
PreSelectedOperationMode	string	

1

2

3

4

5

6

3.8 NUMroto

The topics for NUMroto are accessible under *“/prefix/Id/NUMroto/topic”*.

Topic	Datatype	Comment
Info		
JobState		
WorkpieceState		

Info

The topics for Info are under *“/prefix/Id/NUMroto/Info/topic”*.

Topic	Datatype	Comment
NumrotoVersion	string	ProdState.xml -> /NrProdData/NumrotoVersion
MachineNumber	string	ProdState.xml -> /NrProdData/MachineNumber
MachineName	string	ProdState.xml -> /NrProdData/MachineName

JobState

The topics for JobState are under *“/prefix/Id/NUMroto/JobState/topic”*.

Topic	Datatype	Comment
Activity	string	E-Parameter
JobMode	string	ProdState.xml -> /NrProdData/JobType
JobStarttime	DateTime	ProdState.xml -> /NrProdData/JobStartTime
JobEndtime	DateTime	ProdState.xml -> /NrProdData/JobEndTime
JobTotalNoOfWorkpcs	int	ProdState.xml -> /NrProdData/ToolNumberMax
JobNoOfCurrentWorkpcs	int	ProdState.xml -> /NrProdData/ToolNumber
JobNoOfUnsuccessfulWorkpcs	int	ProdState.xml -> /NrProdData/ToolNumberUnsuccessful

WorkpieceState

The topics for WorkpieceState are under *“/prefix/Id/NUMroto/WorkpieceState/topic”*.

Topic	Datatype	Comment
WorkpcsName	string	ProdState.xml -> /NrProdData/ToolName
WorkpcsEndTime	DateTime	ProdState.xml -> /NrProdData/ToolEndTime

3.9 Troubleshooting / Diagnostics

If there is no data published on the broker, please check the following:

- Is FXServer / APServer connected to the correct CNC?
- Are all the dependencies met?
- Runs MQgateway with elevated permissions? (It should not, when FXServer does not.)
- Is there a licence?
- Is the broker reachable? If it is, what is the state of MQgateway?

If all of the above questions can be answered with yes and it's still not running, please check the logfile.

The logfile located at "*C:\Users\Public\Documents\NUM\MQgateway\Logs*", which is rotated every day or when MQgateway is started.

Make sure there is no exception displayed.

If needed, one can adjust the logging level in the file "*nlog.config*" alongside the "*MQgateway.exe*".

Page deliberately empty

Summary

4	MTConnect.....	89
4.1	MTConnect introduction.....	89
4.2	NUM MDTO Software.....	90
4.3	NUM MDTO Adapter and Agent	91
4.4	NUM MDTO Standard Data Items	92
4.5	NUM MDTO Extended Data Items	93
4.6	NUM MDTO Installation	94
4.6.1	NUM MDTO Condition	94
4.6.1.1	Flexium and Flexium* CNC series	94
4.6.1.2	Axium Power CNC series	94
4.6.1.3	Microsoft .NET Framework installed.....	94
4.6.2	Installation NUM MDTO	95
4.6.3	NUM MDTO Installation Location	97
4.7	NUM MDTO Adapter Configuration	98
4.7.1	Configuration file NUMAdp.ini	98
4.7.2	NUM MDTO Activation.....	98
4.7.2.1	Flexium/Flexium* CNC activation	98
4.7.2.2	Axium Power/NUM Power CNC activation	98
4.7.3	Configuration NUMAdp.ini SYSTEM	99
4.7.3.1	Configuration NUMAdp.ini CNC	101
4.7.3.2	Configuration NUMAdp.ini Axes and Spindles	102
4.7.3.3	Configuration NUMAdp.ini Custom_PLC_Variables.....	103
4.7.3.4	Configuration NUMAdp.ini ProcessState	104
4.7.3.5	Configuration NUMAdp.ini NUMROTO_XML.....	106
4.7.3.6	Configuration NUMAdp.ini ProcessState version.....	107
4.7.3.7	Configuration NUMAdp.ini NUMROTO_XML version	109
4.7.3.8	Required NUMROTO version	110
4.7.3.9	Sample NUMAdp.ini configuration for Flexium CNC series	111
4.8	NUM MDTO Agent Configuration	114
4.8.1	Agent.cfg configuration	114
4.8.2	Devices.xml configuration.....	115
4.8.3	Structure NUM CNC Devices.xml.....	117
4.8.4	Customization NUM CNC Devices.xml	118
4.9	NUM MDTO Diagnostic	120
4.9.1	NUM MDTO Adapter Diagnostic.....	120
4.9.2	NUM MDTO Agent Diagnostic.....	122
4.10	Common NUM MDTO Network Configuration	125
4.10.1	Windows Firewall	126

1

2

3

4

5

6

Page deliberately empty

4 MTConnect

4.1 MTConnect introduction

MTConnect is a universal factory floor communications protocol. It is designed specifically for the shop floor environment.

While there are numerous communication solutions available for the shop floor, *MTConnect* offers one very distinct difference.

MTConnect is the first standard to define a "dictionary" for manufacturing data. This means that data from multiples machines will have a common definition - name, units, values and context.

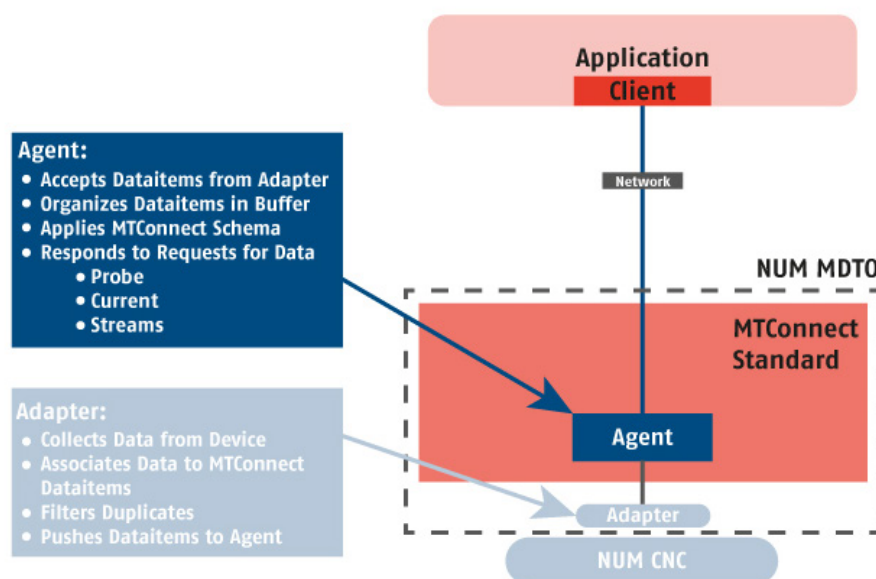
With *MTConnect*, the data is defined only once at the *MTConnect* compliant interface to the device or machine tool.

Once the data is defined based on the *MTConnect* standard protocol, it can be easily be used by all *MTConnect* compliant software applications. This eliminates the need to redefine the data within each application.

MTConnect significantly reduces startup time, overall project costs, and long term maintenance of software system interfaces. *MTConnect* compliant devices process information locally and then provide that data in a consistent format to any *MTConnect* compliant application - ERP, MES, and Production Management Systems.

(*MTConnect.org: Getting Started with MTConnect:Architecture*)

MTConnect Most Basic Architecture

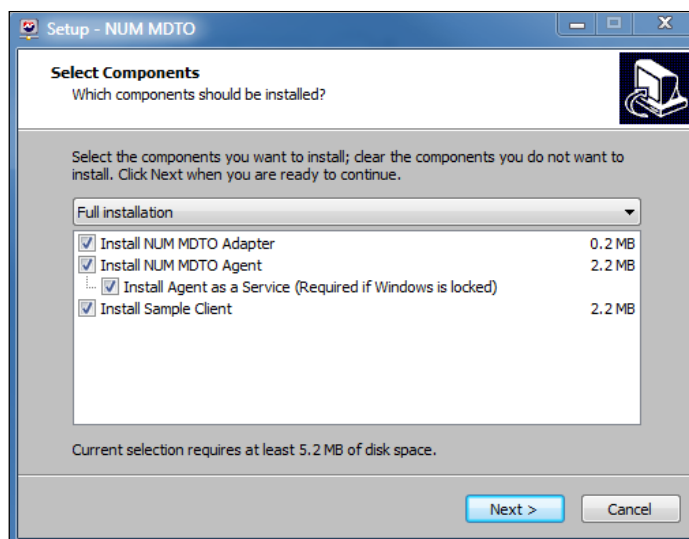
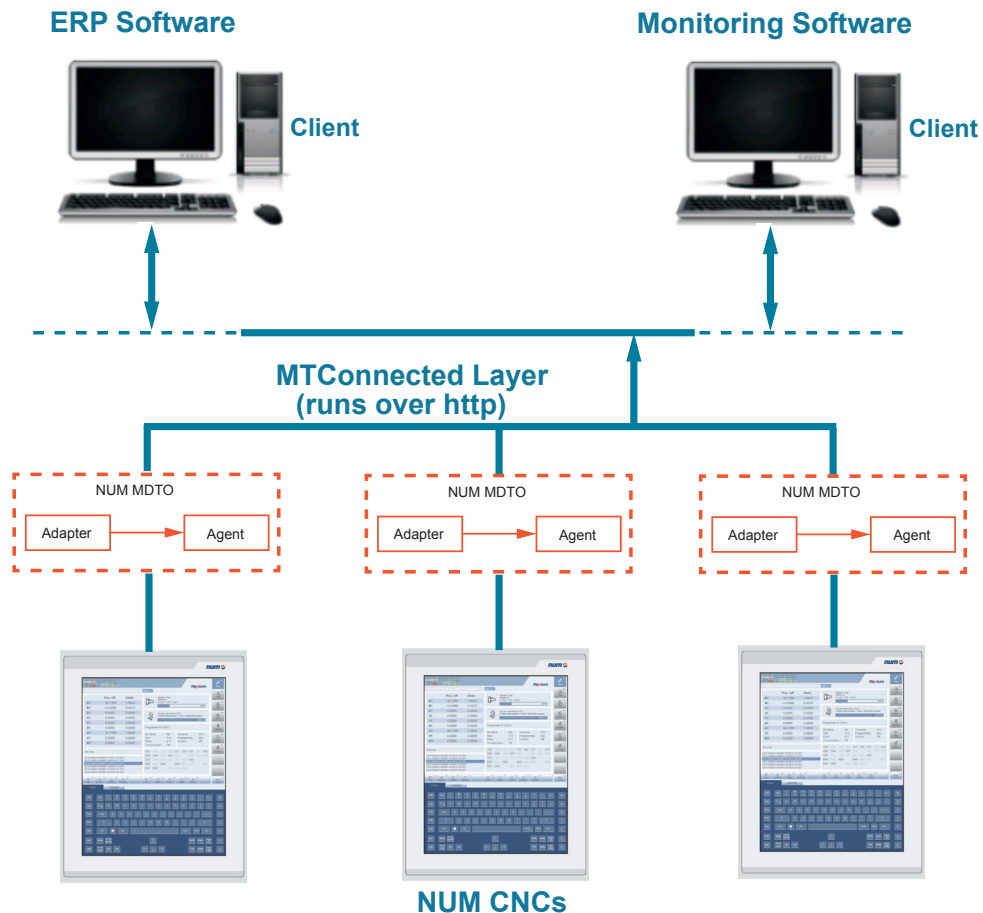


4.2 NUM MDTO Software

NUM MDTO (*MTConnect Data Transfer Object*) is the software component which connects NUM CNC's into a network over MTConnect

NUM MDTO is composed of two different components that act as one piece of *software* → *Adapter and Agent*. The *Adapter* gathers the data directly from the NUM CNC.

The *Agent* collect the data from the Adapter packages and implements it into the MTConnect protocol to serve it to MTConnect compliant client software over the network.



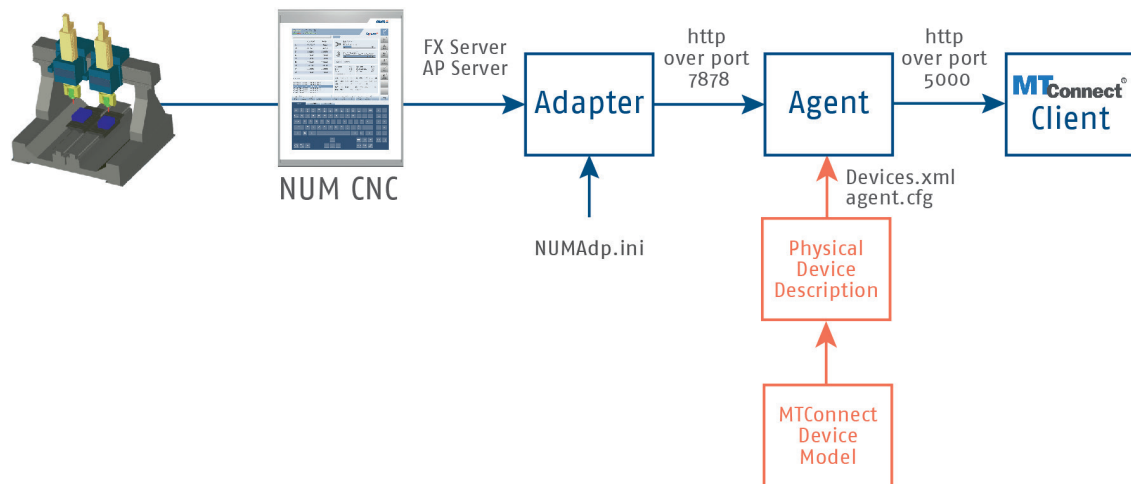
4.3 NUM MDTO Adapter and Agent

NUM MDTO Adapter use the NUM communication server *FXServer* and *APServer* to gather data from the NUM CNC.

The Adapter acts as a server for those CNC data via *http requests*.

NUM MDTO Agent gathers the CNC data from the *NUM MDTO Adapter*, stores and packages the CNC data into MTConnect compliant *XML format*.

The Agent acts as server for the packaged CNC data via *http requests*.



4.4 NUM MDTO Standard Data Items

NUM MDTO provides the *MTConnect Client software* with following standard data items.
Valid from NUM MDTO version 1.4.17.

General Data Items		
Data Item	Data Type	Description
PLC State	Condition	Flexium RTS/ Axiom PLC state
Communications	Condition	Adapter-NC communications state
Availability	Event	Adapter running state

CNC Data Items		
Data Item	Data Type	Description
Alarm	Condition	General CNC alarms
Controller Mode	Event	CNC mode
Execution	Event	CNC program state
Tool ID	Event	Tool Corrector Number (D value)
Tool Asset ID	Event	Tool Number (T value)
Sub Program	Event	Full Active Program List
Program	Event	Main Active Program
Path Feed Rate	Sample	CNC interpolation speed
Requested Feed Rate	Sample	Programmed Feed Rate
Feed Rate Override	Sample	Feed rate override value

Axis/Spindle Data Items		
Data Item	Data Type	Description
Command Position	Sample	Axes commanded positions
Command Rotary Velocity	Sample	Commanded Spindle RPM
Measured Rotary Velocity	Sample	Measured Spindle RPM
Rotary Velocity Override	Sample	Spindle speed override value
Load	Sample	Spindle load
Rotary Mode	Event	Spindle mode

4.5 NUM MDTO Extended Data Items

NUM MDTO version $\geq 1.4.17$.

Provides new extended data items concerned to system information & messages and NUMRoto process data.

System Information & Messages		
Data Item	Data Type	Description
Project Information	Event	Flexium PLC project information.
CNC Serialnumber	Event	Serialnumber of the NUM CNC device.
CNC Software Version	Event	Software version of the NUM CNC device.
Global System Messages	Event	All System Messages from the NUM CNC device.

NUMRoto Process Data		
Data Item	Data Type	Description
NUMRoto Process State value	Event	NUMRoto Tool Grinding process state value by <i>*xml</i> .
NUMRoto Process Label	Event	NUMRoto Tool Grinding process text label by <i>NUMAdp.ini</i> .

1
2
3
4
5
6

4.6 NUM MDTO Installation

4.6.1 NUM MDTO Condition

NUM MDTO is a software for Windows operating systems: Windows XP, 7 , 8.1, 10.

4.6.1.1 Flexium and Flexium+ CNC series

NUM MDTO requires the **licence FXSW283100** "*NUM MDTO MTConnect Data Transfer Objects*" bound to the CNC serial number (Affaire number).
Reference number: **FXSW283100**.

NUM iPC Panel or BoxPC must provide for MDTO the NUM communication interface FXServer.

4.6.1.2 Axiom Power CNC series

NUM MDTO requires an **Activation key** bound to the CNC serial number (Affaire number).
NUM supplies the corresponding Activation key to the customer to register the Activation key to the configuration file *NUMAdp.ini* section *[System] Activation*.

Industrial/Office PC must provide for MDTO the NUM communication interface APServer.
APServer requires the Axiom Power version $\geq$ 204482 R2.

4.6.1.3 Microsoft .NET Framework installed

Microsoft .NET Framework 4.0 must be installed for MDTO on the PC system.

1

2

3

4

5

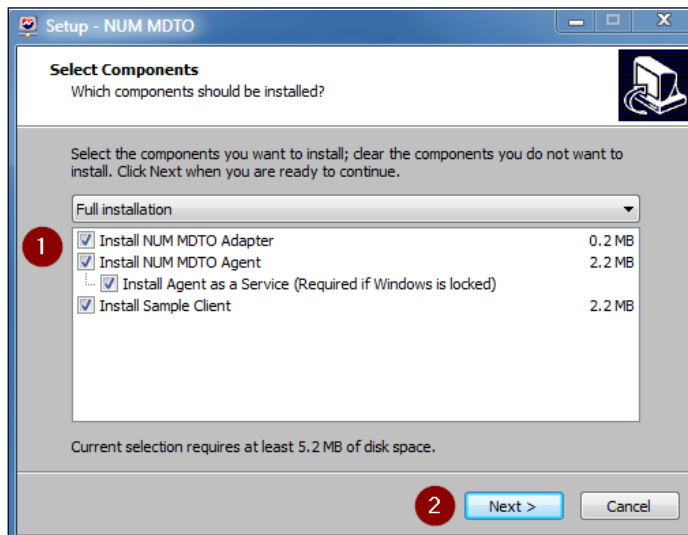
6

4.6.2 Installation NUM MDTO

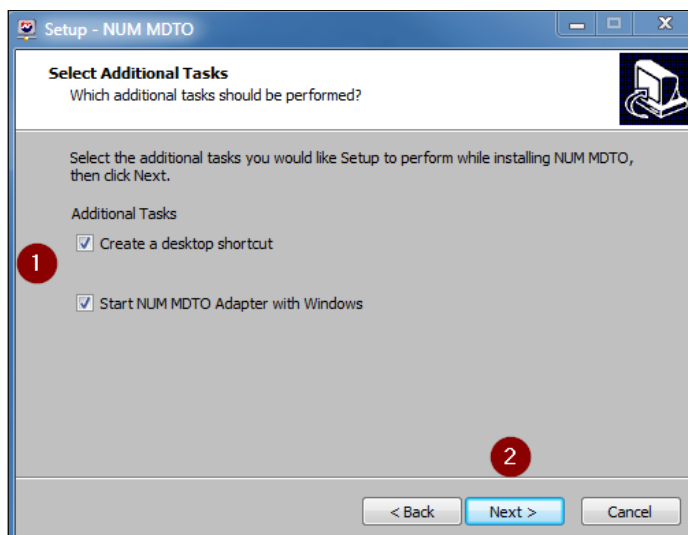
Run the *NUM MDTO Setup.exe* to install the NUM MDTO Adapter and Agent. The Adapter will be installed as *SysTray* application in the Windows taskbar.



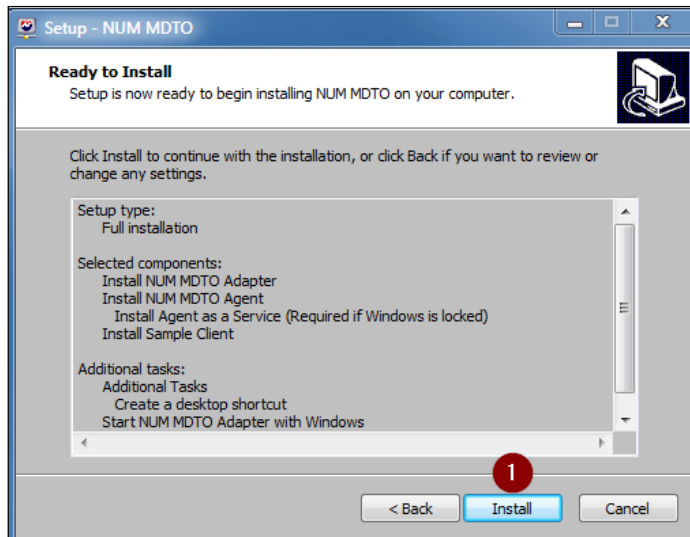
The Agent will be installed as Windows Service NUM MDTO Agent.



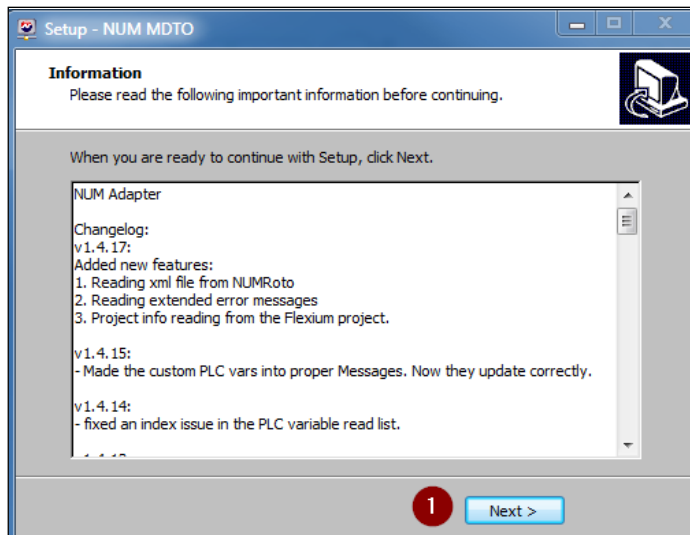
1. Check on all checkboxes.
2. Press button *Next*.



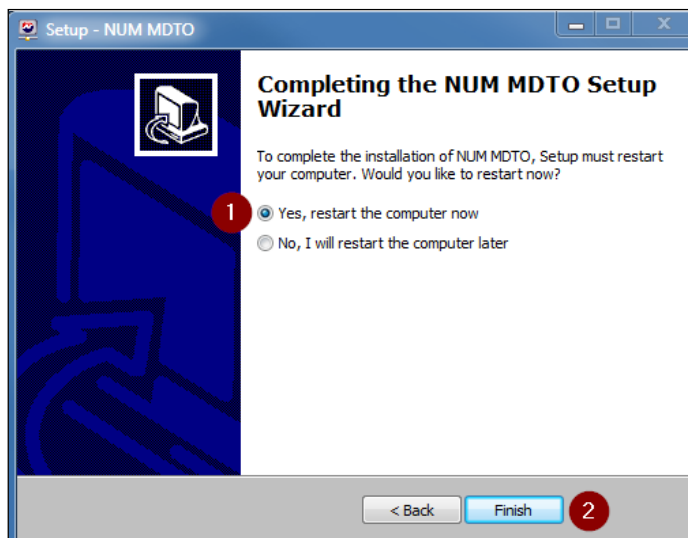
1. Check on all checkboxes.
2. Press button *Next*.



1. Press button *Install*.



1. Press button *Next*.



1. Select Radio Button "Yes, restart computer now".
2. Press button *Finish*.



The MDTO installation is completed and the computer will be restarted!

4.6.3 NUM MDTO Installation Location

Adapter location

C:\Programm Files(86)\NUM\NUM MDTO\Adapter
Includes the Configurationfile: *NUMAdp.ini*.

Agent location

C:\Programm Files(86)\NUM\NUM MDTO\Agent
Includes the Configurationfiles: *Agent.cfg* and *Devices.xml*.

1
2
3
4
5
6

4.7 NUM MDTO Adapter Configuration

4.7.1 Configuration file NUMAdp.ini

The configuration file NUMAdp.ini gives the details of some main features from the NUM CNC series and machine. Therefore the Adapter recognizes how the machine is configured to gather the machine data.

The following configuration table section shows the main features to consider of the NUMAdp.ini configuration file for individual modification.

The configuration file NUMAdp.ini is segmented in section as below:

Note: The character ";;" is in the NUMAdp.ini file a comment! .

+	[System]
+	[CNC]
+	[Axes]
+	[Spindles]
+	[Custom PLC Variables]
+	[ProcessState]
+	[NUMRoto XML]

4.7.2 NUM MDTO Activation

4.7.2.1 Flexium/Flexium+ CNC activation

NUM MDTO requires the licence NUM MDTO MTConnect Data Transfer Object bound to the CNC serial number (Affaire number) : *Reference Number FXSW283100*.

4.7.2.2 Axiom Power/NUM Power CNC activation

NUM MDTO requires an Activation code bound to the CNC serial number (Affaire number).

NUM supplies the corresponding Activation code to the customer.

The Activation code has to register to the configuration file *NUMAdp.ini* section *[System] Data Item: Activation*.

4.7.3 Configuration NUMAdp.ini SYSTEM

NUM Adp.ini section SYSTEM

NUM Adp.ini section SYSTEM		
MDTO Item	Default Value	Description
Activation	-	Axium Power CNC: Entry of the Activation code bound to the serial number (Affaire) supplied by NUM.
Port	7878	TCP/IP port that the Adapter exchange data over http protocol. Must match the agent.cfg configurationfile.
Affaire	-	Serial number (Affaire) of the NUM CNC.
deviceName	NUMCNC	A unique device name without special characters. Must match the agent.cfg configurationfile.
generation	Flexium	Definition of the NUM CNC series type: Axium or Flexium.
debug	false	Debug mode for MDTO debugging: Debug On=true Debug Off=false ⇒ default setting.
dataRefresh	15000	Cyclical update time in [ms] of the data from the device (CNC). Smaller values raise up the data refreshement ⇒ e.g. 2000[ms] .
connectionRefresh	5000	Time in [ms] between attempts to connect to the device (CNC).
demoMode	false	Demo mode acts without connection to the device (CNC),with fake data from the Adapter. demoMode On=true
applicationName	Application	The PLC application name used by the Flexium Tool project.

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;NAME: NUMAdp.ini                                                    ;;
;;Adapter Version: 1.4.17                                           ;;
;;MTConnect Version: 1.3                                           ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
[System]
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Activation code. Contact NUM USA for                               ;;
;; details.                                                         ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Activation=xxxxxxxxxx
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;Port must match the agent.cfg port                                ;;
;;configuration. Usually stays at 7878.                             ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
port=7878
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Affaire number must match the CNC.                                ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Affaire=6xxxxxx
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; DeviceName must match the agent.cfg                               ;;
;; Adapters Name.                                                  ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
deviceName=NUMCNC
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Generation must be Flexium or Axiom.                             ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
generation=Flexium
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Debug mode. true == ON, takes up extra CPU ;;
;; but gives more activity feedback.                               ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
debug=false
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; dataRefresh: Time, in milliseconds, between;;
;; each data update cycle.                                         ;;
;; Smaller numbers make more CPU load, but ;;
;; gather more data. Default = 15000.                               ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
dataRefresh=6000
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; ConnectionRefresh: Time, in milliseconds, ;;
;; between attempts to connect to the CNC. ;;
;; Default value is 5000.                                           ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
connectionRefresh=10000
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; demoMode: boolean                                               ;;
;; True makes the adapter fake data.                               ;;
;; Will not connect to CNC.                                         ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
demoMode=false
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Application name: string                                         ;;
;; Enter the PLC Application name as it ;;
;; appears in the FXTools project.                                  ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
applicationName=Application

```

4.7.3.1 Configuration NUMAdp.ini CNC

NUMAdp.ini section CNC

NUM Adp.ini section CNC		
MDTO Item	Default Value	Description
axisgroup	0	Selection of CNC axis channel from 8 axis channel to read the device (CNC) data. Axis channel range number = 0-7.
cncNumber	0	Selection of one CNC number from 32 CNC number to read the device (CNC) data. Axis channel range number = 0-31.
channel	1	Selection of the quantity of CNC axis channels to read simultaneous the device (CNC) data. Actual only 1 channel is supported.

```

[CNC]
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; axisGroup, value 0-7                                ;;
;; Define the CNC group to read.                        ;;
;; Default value is 0.                                  ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
axisGroup=0
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; cncNumber: value, 0-32 ;;
;; Define the cnc Number to read.                      ;;
;; Default value is 0.                                  ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
cncNumber=0
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Choose the channel for the adapter to               ;;
;; listen to. At the moment, only one channel ;;
;; at a time is supported. Default is 1.              ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
channel=1

```

4.7.3.2

NUMAdp.ini section Axes and Spindles

NUM Adp.ini section Axes and Spindles		
MDTO Item Axes	Default Value	Description
xAxis	true	Setting = true indicates the axis X is valid on this machine. Setting = false indicates the axis X is not valid on this machine.
yAxis	true	Setting = true indicates the axis Y is valid on this machine. Setting = false indicates the axis Y is not valid on this machine.
zAxis	true	Setting = true indicates the axis Z is valid on this machine. Setting = false indicates the axis Z is not valid on this machine.
uAxis	true	Setting = true indicates the axis U is valid on this machine. Setting = false indicates the axis U is not valid on this machine.
vAxis	true	Setting = true indicates the axis V is valid on this machine. Setting = false indicates the axis V is not valid on this machine.
wAxis	true	Setting = true indicates the axis W is valid on this machine. Setting = false indicates the axis W is not valid on this machine.
aAxis	true	Setting = true indicates the axis A is valid on this machine. Setting = false indicates the axis A is not valid on this machine.
bAxis	true	Setting = true indicates the axis B is valid on this machine. Setting = false indicates the axis B is not valid on this machine.
cAxis	true	Setting = true indicates the axis C is valid on this machine. Setting = false indicates the axis C is not valid on this machine.
MDTO Item Spindles	Default Value	Description
adr24	true	Setting = true indicates the spindle address=24 is valid on this machine. Setting = false indicates the spindle address=24 is not valid on this machine.
adr25	true	Setting = true indicates the spindle address=25 is valid on this machine. Setting = false indicates the spindle address=25 is not valid on this machine.
adr26	true	Setting = true indicates the spindle address=26 is valid on this machine. Setting = false indicates the spindle address=26 is not valid on this machine.
adr27	true	Setting = true indicates the spindle address=27 is valid on this machine. Setting = false indicates the spindle address=27 is not valid on this machine.

[illegible]

NUMAdp.ini section ProcessState

NUM Adp.ini section ProcessState		
MDTO Item	Default Value	Description
ProcessEParam	-	NUMROTO process state can be gather by a predefined External E-Parameter type E80xxx. Based on the E80xxx value MDTO provides the NUMROTO process state as text.
ProcessValue0=0	0	Process state value = 0
ProcessTextEN0	-	No NUMROTO program active ⇒ process text language 1
ProcessTextDE0	-	Kein NUMROTO Programm aktiv ⇒ process text language 2
ProcessValue1=1	1	Process state value = 1
ProcessTextEN1	-	Grinding NUMROTO manufacturing ⇒ process text language 1
ProcessTextDE1	-	Schleifen NUMROTO Herstellen ⇒ process text language 2
ProcessValue2=2	2	Process state value = 2
ProcessTextEN2	-	Grinding NUMROTO resharpending ⇒ process text language 1
ProcessTextDE2	-	Schleifen NUMROTO Nachschärfen ⇒ process text language 2
ProcessValue3=10	10	Process state value = 10
ProcessTextEN3	-	Tool probing ⇒ process text language 1
ProcessTextDE3	-	Werkzeug tasten ⇒ process text language 2
ProcessValue4=11	11	Process state value = 11
ProcessTextEN4	-	Wheel probing ⇒ process text language 1
ProcessTextDE4	-	Scheibe tasten ⇒ process text language 2
ProcessValue5=20	20	Process state value = 20
ProcessTextEN5	-	Wheel dressing ⇒ process text language 1
ProcessTextDE5	-	Scheibe abrichten ⇒ process text language 2
ProcessValue6=21	21	Process state value = 21
ProcessTextEN6	-	Wheel crushing ⇒ process text language 1
ProcessTextDE6	-	Scheibe crushieren ⇒ process text language 2
ProcessValue7=22	22	Process state value = 22
ProcessTextEN7	-	Wheel sticking ⇒ process text language 1
ProcessTextDE7	-	Scheibe aufräumen ⇒ process text language 2
ProcessValue8=30	30	Process state value = 30
ProcessTextEN8	-	Tool loader ⇒ process text language 1
ProcessTextDE8	-	Werkzeug laden ⇒ process text language 2

```

[ProcessState]
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; These items are for NUMROTO integration.      ;;
;; Process state values must be followed by      ;;
;; text translations. EN and DE are allowed.      ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
ProcessEParam=E80047
ProcessValue0=0
ProcessTextEN0=No NUMROTO program active
ProcessTextDE0=Kein NUMROTO Programm aktiv
ProcessValue1=1
ProcessTextEN1=Grinding NUMROTO manufacturing
ProcessTextDE1=Schleifen NUMROTO Herstellen
ProcessValue2=2
ProcessTextEN2=Grinding NUMROTO resharpener
ProcessTextDE2=Schleifen NUMROTO Nachschärfen
ProcessValue3=10
ProcessTextEN3=Tool probing
ProcessTextDE3=Werkzeug tasten
ProcessValue4=11
ProcessTextEN4=Wheel probing
ProcessTextDE4=Scheibe tasten
ProcessValue5=20
ProcessTextEN5=Wheel dressing
ProcessTextDE5=Scheibe abrichten
ProcessValue6=21
ProcessTextEN6=Wheel crushing
ProcessTextDE6=Scheibe crushieren
ProcessValue7=22
ProcessTextEN7=Wheel sticking
ProcessTextDE7=Scheibe aufräumen
ProcessValue8=30
ProcessTextEN1=Tool loader
ProcessTextDE1=Werkzeug laden

```

1

2

3

4

5

6

4.7.3.5 Configuration NUMAdp.ini NUMROTO_XML

NUMAdp.ini section NUMRoto_XML

NUM Adp.ini section NUMROTO_XML		
MDTO Item	Default Value	Description
XMLFilePath	-	NUMROTO Control creates *.xml files , which contains various production step information. These *.xml files are located by a NUMROTO configurable path definition. The entry <i>XMLFilePath</i> must match with the NUMROTO setting.
DeleteAfterRead	true	The entry <i>DeleteAfterRead = true</i> setting defines to delete the *.xml file after gathering the production step information. The entry <i>DeleteAfterRead = false</i> setting defines to remain the *.xml file after gathering the production step information

```
[NUMRoto_XML]
;;;;;;;;;;;;;
;; For NUMRoto XML integration.           ;;
;; Indicate the path to the XML.           ;;
;;;;;;;;;;;;;
XMLFilePath=C:\\Users\\Public\\Desktop\\NumrotoXML\\
DeleteAfterRead=TRUE
```

4.7.3.6 Configuration NUMAdp.ini ProcessState version

- Configuration NUMAdp.ini ProcessState from NUM MDTO version 1.5.1
- Extension of the Process state value 40. Requires NUMROTO version 4.1.0

NUM Adp.ini section ProcessState		
MDTO Item	Default Value	Description
ProcessEParam	-	NUMROTO process state can be gather by a predefined External E-Parameter type E80xxx. Based on the E80xxx value MDTO provides the NUMROTO process state as text.
ProcessValue0=0	0	Process state value = 0
ProcessTextEN0	-	No NUMROTO program active ⇒ process text language 1
ProcessTextDE0	-	Kein NUMROTO Programm aktiv ⇒ process text language 2
ProcessValue1=1	1	Process state value = 1
ProcessTextEN1	-	Grinding NUMROTO manufacturing ⇒ process text language 1
ProcessTextDE1	-	Schleifen NUMROTO Herstellen ⇒ process text language 2
ProcessValue2=2	2	Process state value = 2
ProcessTextEN2	-	Grinding NUMROTO resharpening ⇒ process text language 1
ProcessTextDE2	-	Schleifen NUMROTO Nachschärfen ⇒ process text language 2
ProcessValue3=10	10	Process state value = 10
ProcessTextEN3	-	Tool probing ⇒ process text language 1
ProcessTextDE3	-	Werkzeug tasten ⇒ process text language 2
ProcessValue4=11	11	Process state value = 11
ProcessTextEN4	-	Wheel probing ⇒ process text language 1
ProcessTextDE4	-	Scheibe tasten ⇒ process text language 2
ProcessValue5=20	20	Process state value = 20
ProcessTextEN5	-	Wheel dressing ⇒ process text language 1
ProcessTextDE5	-	Scheibe abrichten ⇒ process text language 2
ProcessValue6=21	21	Process state value = 21
ProcessTextEN6	-	Wheel crushing ⇒ process text language 1
ProcessTextDE6	-	Scheibe crushieren ⇒ process text language 2
ProcessValue7=22	22	Process state value = 22
ProcessTextEN7	-	Wheel sticking ⇒ process text language 1
ProcessTextDE7	-	Scheibe aufräuen ⇒ process text language 2
ProcessValue8=30	30	Process state value = 30
ProcessTextEN8	-	Tool loader ⇒ process text language 1
ProcessTextDE8	-	Werkzeug laden ⇒ process text language 2
ProcessValue8=40	40	Process state value = 40
ProcessTextEN9	-	Wheel change ⇒ process text language 1
ProcessTextDE9	-	Schleifscheibe wechseln ⇒ process text language 2

```

[ProcessState]
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; These items are for NUMROTO integration.      ;;
;; Process state values must be followed by      ;;
;; text translations. EN and DE are allowed.      ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
ProcessEParam=E80047
ProcessValue0=0
ProcessTextEN0=No NUMROTO program active
ProcessTextDE0=Kein NUMROTO Programm aktiv
ProcessValue1=1
ProcessTextEN1=Grinding NUMROTO manufacturing
ProcessTextDE1=Schleifen NUMROTO Herstellen
ProcessValue2=2
ProcessTextEN2=Grinding NUMROTO resharpending
ProcessTextDE2=Schleifen NUMROTO Nachschärfen
ProcessValue3=10
ProcessTextEN3=Tool probing
ProcessTextDE3=Werkzeug tasten
ProcessValue4=11
ProcessTextEN4=Wheel probing
ProcessTextDE4=Scheibe tasten
ProcessValue5=20
ProcessTextEN5=Wheel dressing
ProcessTextDE5=Scheibe abrichten
ProcessValue6=21
ProcessTextEN6=Wheel crushing
ProcessTextDE6=Scheibe crushieren
ProcessValue7=22
ProcessTextEN7=Wheel sticking
ProcessTextDE7=Scheibe aufrauen
ProcessValue8=30
ProcessTextEN8=Tool loader
ProcessTextDE8=Werkzeug laden
ProcessValue9=40
ProcessTextEN9=Wheel change
ProcessTextDE9=Schleifscheibe wechseln

```

4.7.3.7

- Configuration NUMAdp.ini NUMRoto_XML from NUM MDTO version 1.5.1
- Extension of the Process state value 40. Requires NUMROTO version 4.1.0

NUM Adp.ini section NUMROTO_XML		
MDTO Item	Default Value	Description
XMLFilePath	-	NUMROTO Control creates *.xml files , which contains various production step information. These *.xml files are located by a NUMROTO configurable path definition. The entry <i>XMLFilePath</i> must match with the NUMROTO setting.
DeleteAfterRead	true	The entry <i>DeleteAfterRead = true</i> setting defines to delete the *.xml file after gathering the production step information. The entry <i>DeleteAfterRead = false</i> setting defines to remain the *.xml file after gathering the production step information
ActRegKey	-	Entry to get the machine activity E-Parameter E80xxx by Windows Registry. <i>HKEY_CURRENT_USER\Software\Numrotol\NUMROTOplus\EparamMachineActivity</i>
ToolRegKey	-	Entry to get the tool counter E-Parameter E80xxx by Windows Registry. <i>HKEY_CURRENT_USER\Software\Numrotol\NUMROTOplus\EparamToolCount</i>

```
[NUMRoto_XML]
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; For NUMRoto XML integration.                      ;;
;; Indicate the path to the XML. Replace username.  ;;
XMLFilePath=C:\Users\Public\AppData\Local\NumrotoXML\
DeleteAfterRead=false
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; For NUMRoto integration. Indicate the            ;;
;; registry path to the machine activity and        ;;
;; tool count E-Parameter.                          ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
ActRegKey=HKEY_CURRENT_USER\Software\Numroto\NUMROTOplus\EparamMachineActivity
ToolRegKey=HKEY_CURRENT_USER\Software\Numroto\NUMROTOplus\EparamToolCount
```



NUM MDTO version $\geq 1.5.1$: Configuration file NUMAdp.ini extension for the section [ProcessState] to provide additional machine activity data value: 40= Wheel change Requires NUMROTO version $\geq 4.1.0$.



NUM MDTO version $\geq 1.5.1$: Configuration file NUMAdp.ini extension for the section [NUMRoto_XML] to get machine activity E-Parameter E80xxx by Windows registry setting:
ActRegKey=HKEY_CURRENT_USER\\Software\\Numroto\\NUMROTOplus\\EparamMachineActivity Requires NUMROTO version $\geq 4.1.0$.



NUM MDTO version $\geq 1.5.1$: Configuration file NUMAdp.ini extension for the section [NUMRoto_XML] to get tool counter E-Parameter E80xxx by Windows registry setting:
ToolRegKey=HKEY_CURRENT_USER\\Software\\Numroto\\NUMROTOplus\\EparamToolCount Requires NUMROTO version $\geq 4.1.0$.

4.7.3.9 Sample NUMAdp.ini configuration for Flexium CNC series

- Sample configuration NUMAdp.ini
- Flexium CNC
- Serial number/Affaire: 612123
- Linear axes: X-Y-Z
- Rotary axes: B-C
- Spindle: 1 Spindle with address @24
- Read the CNC Mode state
- Read NUMROTO process state
- Read NUMROTO production step by *.xml
- Delete *.xml

```
[System]
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Activation code. Contact NUM USA for      ;;
;; details.                                ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Activation=
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;Port must match the agent.cfg port        ;;
;;configuration. Usually stays at 7878.     ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
port=7878
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Affaire number must match the CNC.       ;;
;;                               j:         ;;
Affaire=612123
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; DeviceName must match the agent.cfg      ;;
;; Adapters Name.                          ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
deviceName=NUMCNC
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Generation must be Flexium or Axium.     ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
generation=Flexium
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Debug mode. true == ON, takes up extra CPU ;;
;; but gives more activity feedback.        ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
debug=false
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; dataRefresh: Time, in milliseconds, between;;
;; each data update cycle.                 ;;
;; Smaller numbers make more CPU load, but  ;;
;; gather more data. Default = 15000.       ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
dataRefresh=3000
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; ConnectionRefresh: Time, in milliseconds, ;;
;; between attempts to connect to the CNC.  ;;
;; Default value is 5000.                   ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
connectionRefresh=10000
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; demoMode: boolean                       ;;
;; True makes the adapter fake data.        ;;
;; Will not connect to CNC.                 ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
demoMode=false
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Application name: string                 ;;
;; Enter the PLC Application name as it     ;;
;; appears in the FXTTools project.         ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
applicationName=Application
```

```

[CNC]
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; axisGroup, value 0-7                ;;
;; Define the CNC group to read.        ;;
;; Default value is 0.                  ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
axisGroup=0
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; cncNumber: value, 0-32 ;;
;; Define the cnc Number to read.        ;;
;; Default value is 0.                  ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
cncNumber=0
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Choose the channel for the adapter to ;;
;; listen to. At the moment, only one channel ;;
;; at a time is supported. Default is 1.    ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
channel=1

[Axes]
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; A true indicates that this axis exists on ;;
;; the machine.                             ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
xAxis=true
yAxis=true
zAxis=true
uAxis=false
vAxis=false
wAxis=false
aAxis=false
bAxis=true
cAxis=true

[Spindles]
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; A true indicates that this spindle address ;;
;; exists on the machine.                     ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
adr24=true
adr25=false
adr26=false
adr27=false

[Custom PLC Variables]
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Up to 5 custom PLC variables can be declared;;
;; Variables must be declared using full name ;;
;; syntax, E.g.                               ;;
;; Application.IOCONFIG_GLOBALS.Flexium_NCK.W_ALLCNC.Mode;
;; or Application.MYPOU.MySymbolicVariable    ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
customVar1=Application.IoConfig_Globals.RCNC.General.Mode
customVar2=
customVar3=
customVar4=
customVar5=

```

```

[ProcessState]
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; These items are for NUMROTO integration.      ;;
;; Process state values must be followed by      ;;
;; text translations. EN and DE are allowed.      ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
ProcessEParam=E80047
ProcessValue0=0
ProcessTextEN0=No NUMROTO program active
ProcessTextDE0=Kein NUMROTO Programm aktiv
ProcessValue1=1
ProcessTextEN1=Grinding NUMROTO manufacturing
ProcessTextDE1=Schleifen NUMROTO Herstellen
ProcessValue2=2
ProcessTextEN2=Grinding NUMROTO resharpending
ProcessTextDE2=Schleifen NUMROTO Nachschärfen
ProcessValue3=10
ProcessTextEN3=Tool probing
ProcessTextDE3=Werkzeug tasten
ProcessValue4=11
ProcessTextEN4=Wheel probing
ProcessTextDE4=Scheibe tasten
ProcessValue5=20
ProcessTextEN5=Wheel dressing
ProcessTextDE5=Scheibe abrichten
ProcessValue6=21
ProcessTextEN6=Wheel crushing
ProcessTextDE6=Scheibe crushieren
ProcessValue7=22
ProcessTextEN7=Wheel sticking
ProcessTextDE7=Scheibe aufräumen
ProcessValue8=30
ProcessTextEN8=Tool loader
ProcessTextDE8=Werkzeug laden
[NUMROTO_XML]
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; For NUMROTO XML integration.                  ;;
;; Indicate the path to the XML.                  ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
XMLFilePath=C:\\Users\\Public\\Desktop\\XML\\
DeleteAfterRead=TRUE

```

1

2

3

4

5

6

4.8 NUM MDTO Agent Configuration

4.8.1 Agent.cfg configuration

- Location: *C:\Program Files\NUM MDTO\Agent\agent.cfg*
The configuration file Agent.cfg defines a network environment and informs the Agent where the NUM MDTO Adapter connection is located by the definition of the PC hardware IP address.

- Agent.cfg parameter settings:

```
Devices = Devices.xml
Port = 5000
Adapters {
    NUM MDTO {
        Device = NUMCNC
        Host = 127.0.0.1
        Port = 7878
    }
}
```

- Device = NUMCNC

Device must match with the with entry *<Device name="NUMCNC"....>* of the XML file *Devices.xml*

```
<Devices>
  <Device name="NUMCNC" uuid="cncid" id="num001">
```

- Host:127.0.0.1

IP address from the PC hardware where the Adapter is executed in the network structure.
Common setting: localhost 127.0.0.1 Adapter and Agent on the same PC.

- Port: 7878

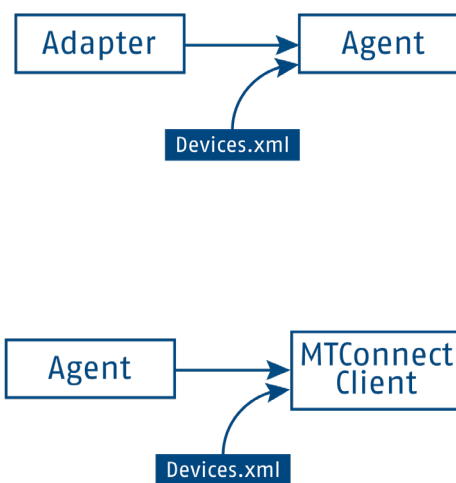
Common and predefined port number 7878 for the http protocol over TCP transport layer between Adapter <=> Agent .

1	<MTConnectDevices>
2	<Header></Header>
3	<Devices>
4	<Device>
5	<Axes>
6	<Rotary>...</Rotary>
7	</Axes>
8	</Device>
9	</Devices>
10	</MTConnectDevices>

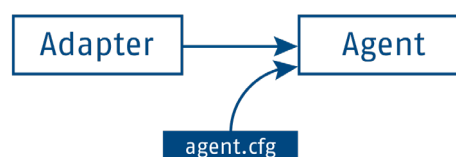
4.8.2 Devices.xml configuration

- Location: *C:\Program Files\NUM MDT\Agent\Devices.xml*
The content of the *MTConnect XML* document *Devices.xml* specifies in the MTConnect standard, the encoding of *Dataltems* in the XML schema.
The machine - CNC data served by the Adapter should match the Adapter data output to a Agent *Dataltem* tag in the *Devices.xml*.
- Common data flow:
The Adapter gather the machine - CNC data and serves the Adapter output data over http to the Agent. The Agent reads the XML document *Devices.xml* to understand what data are transferred from the Adapter.
The Agent provides the *Devices.xml* to the MTConnect compliant client application (ERP, OOE, Dashboards etc.) to understand what *Dataltems* are provided from the Agent.

The MTConnect compliant Client requests data from the MTConnect Agent, which acts as Server. The Agent responds with the requested data and parses the data.



Devices.xml contains description of a complete device:



Header	Protocol related information.
Components	The building blocks of the device.
DataItems	The description of the data available from the device.
Samples	A point-in-time measurement of a data item that is continuously changing.
Events	Discrete changes in state that can have no intermediate value. They indicate the state of a specific attribute of a component.
Condition	A piece of information the device provides as an indicator of its health and ability to function. A condition can be one of Normal, Warning, Fault, or Unavailable. A single condition type can have multiple Faults or Warnings at any given time. This behavior is different from Events and Samples where a data item MUST only have a single value at a given time.

4.8.3 Structure NUM CNC Devices.xml

To give an overview, the Devices.xml is structured in the following parts:

- Messages**

```
<Devices>
  <Device name="NUMCNC" uuid="gncid" id="num001">
    <Description model="Model" manufacturer="OEM">Machine Description</Description>
    <DataItems>
      <DataItem type="AVAILABILITY" category="EVENT" id="id002" name="avail" />
      <DataItem category="EVENT" id="adapterState" type="AVAILABILITY"/>
      <DataItem category="EVENT" id="dev_asset_chg" type="ASSET_CHANGED"/>
      <DataItem category="EVENT" id="dev_asset_rem" type="ASSET_REMOVED"/>
      <DataItem type="MESSAGE" category="EVENT" id="projectVersion" name="prjVersion" />
      <DataItem type="MESSAGE" category="EVENT" id="projectTitle" name="prjTitle" />
      <DataItem type="MESSAGE" category="EVENT" id="projectDescript" name="prjDesc" />
      <DataItem type="MESSAGE" category="EVENT" id="projectAuthor" name="prjAuthor" />
      <DataItem type="MESSAGE" category="EVENT" id="projectCompany" name="prjCompany" />
      <DataItem type="MESSAGE" category="EVENT" id="project" name="prj" />
      <DataItem type="MESSAGE" category="EVENT" id="affairs" name="affaireNo" />
      <DataItem type="MESSAGE" category="EVENT" id="software" name="swVersion" />
      <DataItem type="MESSAGE" category="EVENT" id="CncNumber" name="ActiveAlarmCncNumber" />
      <DataItem type="MESSAGE" category="EVENT" id="numberOfError" name="ActiveAlarmNumberOfError" />
      <DataItem type="MESSAGE" category="EVENT" id="ErrorTyp0" name="ActiveAlarmErrorTyp0" />
      <DataItem type="MESSAGE" category="EVENT" id="ErrorIndex0" name="ActiveAlarmErrorIndex0" />
      <DataItem type="MESSAGE" category="EVENT" id="ErrorNumber0" name="ActiveAlarmErrorNumber0" />
      <DataItem type="MESSAGE" category="EVENT" id="ErrorLine0" name="ActiveAlarmErrorLine0" />
      <DataItem type="MESSAGE" category="EVENT" id="ErrorMessage0" name="ActiveAlarmErrorMessage0" />
      <DataItem type="MESSAGE" category="EVENT" id="ErrorAdditional0" name="ActiveAlarmErrorAdditional0" />
      <DataItem type="MESSAGE" category="EVENT" id="ErrorTyp1" name="ActiveAlarmErrorTyp1" />
    </DataItems>
  </Device>
</Devices>
```

- Axes**

```
<Components>
  <Axes name="axes" id="id005">
    <Components>
      <Linear name="X" id="axX">
        <DataItems>
          <DataItem type="POSITION" category="SAMPLE" id="posX" name="xComd" subType="COMMANDED" units="MILLIMETER" />
        </DataItems>
      </Linear>
      <Linear name="Y" id="axY">
        <DataItems>
          <DataItem type="POSITION" category="SAMPLE" id="posY" name="yComd" subType="COMMANDED" units="MILLIMETER" />
        </DataItems>
      </Linear>
      <Linear name="Z" id="axZ">
        <DataItems>
          <DataItem type="POSITION" category="SAMPLE" id="posZ" name="zComd" subType="COMMANDED" units="MILLIMETER" />
        </DataItems>
      </Linear>
      <Linear name="U" id="axU">
        <DataItems>
        </DataItems>
      </Linear>
    </Components>
  </Axes>
</Components>
```

- Spindle**

```
<Rotary name="S1" nativeName="824" id="sp24">
  <DataItems>
    <DataItem type="ROTARY_VELOCITY" category="SAMPLE" id="sp24RPM" name="sp24RPM" units="REVOLUTION/MINUTE" s />
    <DataItem type="ROTARY_VELOCITY" category="SAMPLE" id="sp24Req" name="sp24Req" units="REVOLUTION/MINUTE" s />
    <DataItem type="ROTARY_VELOCITY_OVERRIDE" category="EVENT" id="sp24OVR" name="sp24OVR" units="PERCENT" />
    <DataItem type="ROTARY_MODE" category="EVENT" id="sp24Mode" name="sp24Mode">
      <Constraints>
        <Value>SPINDLE</Value>
      </Constraints>
    </DataItem>
    <DataItem type="LOAD" category="SAMPLE" id="sp24Load" name="sp24Load" units="PERCENT" />
  </DataItems>
</Rotary>
<Rotary name="S2" nativeName="825" id="sp25">
  <DataItems>
    <DataItem type="ROTARY_VELOCITY" category="SAMPLE" id="sp25RPM" name="sp25RPM" units="REVOLUTION/MINUTE" s />
    <DataItem type="ROTARY_VELOCITY" category="SAMPLE" id="sp25Req" name="sp25Req" units="REVOLUTION/MINUTE" s />
    <DataItem type="ROTARY_VELOCITY_OVERRIDE" category="EVENT" id="sp25OVR" name="sp25OVR" units="PERCENT" />
    <DataItem type="ROTARY_MODE" category="EVENT" id="sp25Mode" name="sp25Mode">
      <Constraints>
      </Constraints>
    </DataItem>
  </DataItems>
</Rotary>
```

- NCK**

```
<Controller name="NCK" id="id050">
  <DataItems>
    <DataItem id="systemAlarm" name="sysAlarm" type="SYSTEM" category="CONDITION" />
    <DataItem type="COMMUNICATIONS" category="CONDITION" id="com01" name="commState" />
    <DataItem category="CONDITION" id="plc01" name="plcState" type="LOGIC_PROGRAM" />
  </DataItems>
  <Components>
    <Path id="pth" name="path">
      <DataItems>
        <DataItem type="CONTROLLER_MODE" category="EVENT" id="id051" name="mode" />
        <DataItem type="EXECUTION" category="EVENT" id="id052" name="exec" />
        <DataItem type="PROGRAM" category="EVENT" id="id053" name="program" />
        <DataItem type="PATH_FEEDRATE" category="SAMPLE" id="id003" name="pathFeed" uni />
        <DataItem type="FEEDRATE_OVERRIDE" category="SAMPLE" id="id069" name="feedOVR" />
      </DataItems>
    </Path>
  </Components>
</Controller>
```

4.8.4 Customization NUM CNC Devices.xml

To provide the possibility to modify some parts from the standard Devices.xml file for customized settings.

- **Line part:** `<Device name="NUMCNC" uuid="cncid" id="num001">`

Modify value: name=

Consider name=: Must match with the NUMAdp.ini entry deviceName=

Modify value: uuid=

Consider uuid: According to choice.

- **Line part:** `<Description model="Model" manufacturer="OEM">Machine Description</Description>`

Modify value: model=

Consider model=: According to choice.

Modify value: manufacturer=

Consider manufacturer=: According to choice.

Modify value: description=

Consider description=: According to choice.



This data is for display purposes.

- **Line part: Line part: <Axes name="axes" id="id005">
<Linear name=".." id=".."> and <Rotary name=".." id="..">**

The Devices.xml part Linear name and Rotary name contains by default the dataitems for all linear and rotary axes.

This part must be modified corresponding the physical available axes defined in the configuration file NUMAdp.ini.

Not used axis should delete the entire section <Linear name=".." id=".."> to </Linear> and <Rotary name=".." id=".."> to </Rotary>.



See here under the note.

```
<Components>
  <Axes name="axes" id="id005">
    <Components>
<Linear name=".." id="..">
  <DataItems>
    <DataItem type="POSITION" category="SAMPLE" id="posX" name="xComd" subType="COMMANDED"
units="MILLIMETER" nativeUnits="MILLIMETER" coordinateSystem="MACHINE" />
  </DataItems>
</Linear>
.....
    <Rotary name=".." id="..">
<DataItems>
    <DataItem type="POSITION" category="SAMPLE" id="posB" name="bComd" subType="COMMANDED"
units="DEGREE" nativeUnits="DEGREE" coordinateSystem="MACHINE" />
  </DataItems>
    </Rotary>
```

1

2

3

4

5

6

4.9 NUM MDTO Diagnostic

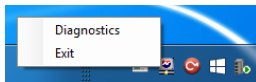
4.9.1 NUM MDTO Adapter Diagnostic

The NUM MDTO Adapter provide a diagnostic tool by the Adapter SysTray application. It monitors the communication between the device NUM CNC <=> Adapter.

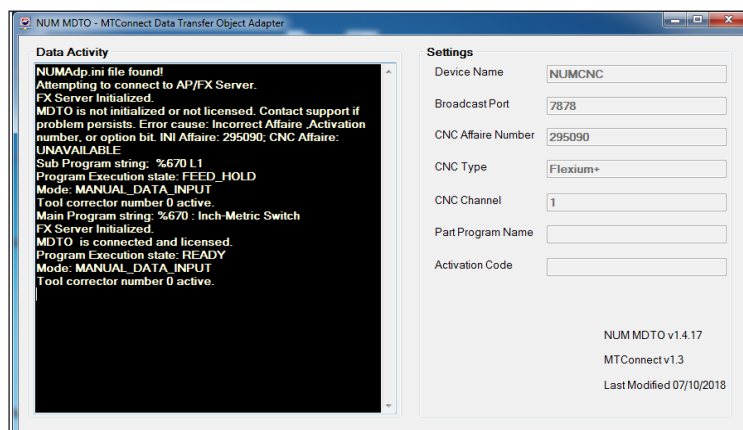
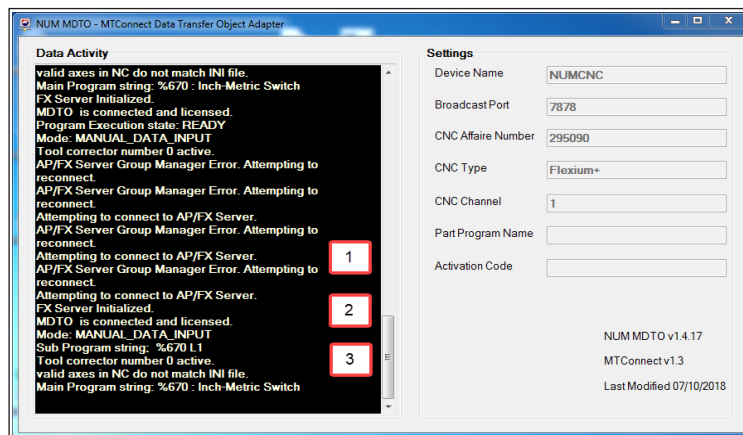
- Right mouse click on the MDTO Adapter SysTray icon in the Windows taskbar.



- Select in the context menu Diagnostics to open the diagnostic tool.



- The NUM MDTO Adapter Diagnostic tool is divided in the Data Activity window and Setting window.



- Data Activity Window:

This window area shows the communication state between Machine/NUM CNC <=> Adapter.

Item1: Shows the communication build-up state by the NUM FXServer (Flexium) or APServer (Axiom) to the NUM MDTO Adapter.

Item2: Shows the NUM MDTO Adapter connection and licence state.

Item3: Shows from the NUM CNC gathered data e.g. CNC mode, tool correction number, etc.



The valid axis configuration in the CNC doesn't corresponds with the NUMAdp.ini settings => "Valid axes in NC do not not match INI file".

- Settings window:

This window area shows static information between the NUM CNC <=> NUM MDTO Adapter.

1

2

3

4

5

6

4.9.2 NUM MDTO Agent Diagnostic

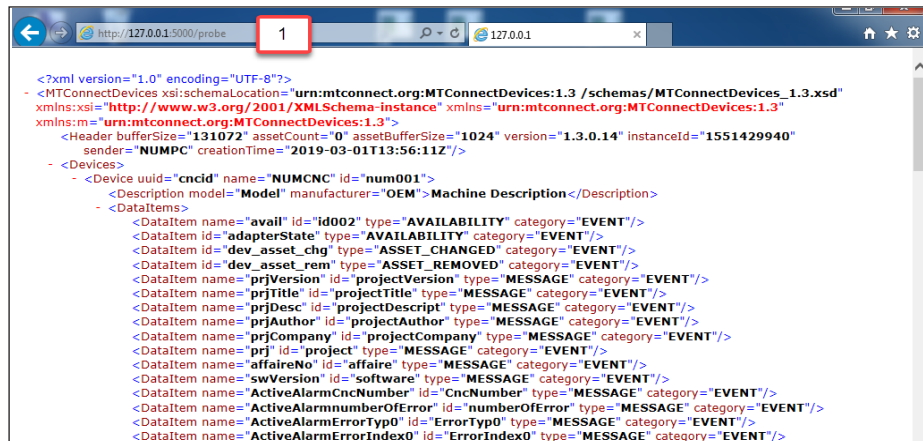
The Agent diagnostic can be performed by a Webbrowser. The Webbrowser applies as MTConnect compliant client and the Agent acts as HTTP Server.
The Webbrowser as MTConnect client can request from the Agent (Server) by HTTP commands and Agent response the requested data.

- **HTTP command: `http://127.0.0.1:5000/probe`**

MTConnect client "Webbrowser" request for device's descriptive information from the Agent by the http command and the probe command will return the Devices.xml contents to parse which DataItems are available from the device NUM CNC.

Sample:

Open a Webbrowser and type in the Agent URL `http://127.0.0.1:5000/probe` => MDTO Adapter and Agent are at the same PC.



```
<?xml version="1.0" encoding="UTF-8"?>
- <MTConnectDevices xsi:schemaLocation="urn:mtconnect.org:MTConnectDevices:1.3 /schemas/MTConnectDevices_1.3.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:mtconnect.org:MTConnectDevices:1.3"
  xmlns:m="urn:mtconnect.org:MTConnectDevices:1.3">
  <Header bufferSize="131072" assetCount="0" assetBufferSize="1024" version="1.3.0.14" instanceId="1551429940"
    sender="NUMPC" creationTime="2019-03-01T13:56:11Z"/>
  - <Devices>
    - <Device uuid="cncid" name="NUMCNC" id="num001">
      <Description model="Model" manufacturer="OEM">Machine Description</Description>
      - <DataItems>
        <DataItem name="avail" id="id002" type="AVAILABILITY" category="EVENT"/>
        <DataItem id="adapterState" type="AVAILABILITY" category="EVENT"/>
        <DataItem id="dev_asset_chg" type="ASSET_CHANGED" category="EVENT"/>
        <DataItem id="dev_asset_rem" type="ASSET_REMOVED" category="EVENT"/>
        <DataItem name="prjVersion" id="projectVersion" type="MESSAGE" category="EVENT"/>
        <DataItem name="prjTitle" id="projectTitle" type="MESSAGE" category="EVENT"/>
        <DataItem name="prjDesc" id="projectDescription" type="MESSAGE" category="EVENT"/>
        <DataItem name="prjAuthor" id="projectAuthor" type="MESSAGE" category="EVENT"/>
        <DataItem name="prjCompany" id="projectCompany" type="MESSAGE" category="EVENT"/>
        <DataItem name="prj" id="project" type="MESSAGE" category="EVENT"/>
        <DataItem name="affaireNo" id="affaire" type="MESSAGE" category="EVENT"/>
        <DataItem name="swVersion" id="software" type="MESSAGE" category="EVENT"/>
        <DataItem name="ActiveAlarmCncNumber" id="CncNumber" type="MESSAGE" category="EVENT"/>
        <DataItem name="ActiveAlarmnumberOfError" id="numberOfError" type="MESSAGE" category="EVENT"/>
        <DataItem name="ActiveAlarmErrorTyp0" id="ErrorTyp0" type="MESSAGE" category="EVENT"/>
        <DataItem name="ActiveAlarmErrorIndex0" id="ErrorIndex0" type="MESSAGE" category="EVENT"/>
      </DataItems>
    </Device>
  </Devices>
</MTConnectDevices>
```

- **HTTP command: `http://127.0.0.1:5000/current`**

MTConnect client "Webbrowser" request for the actual DataItems values from the machine/NUM CNC.

Parameter for the current request: Refer the table below.

Timestamp	Based on the Universal Time Coordinated UTC format.
Type	The type of DataItem being measured => Example: Position, Rotary_Velocity.
SubType	A sub-categorization of the DataItem type => Example: subType of Position is Actual.
name	Name of the DataItem.
id	Unique identifier for this DataItem.
Sequence	Sequence number that was assigned at the time the data was collected.
Value	DataItem.



If all Value fields are marked "UNAVAILABLE" then the Agent is not communicating with NUM MDTO Adapter or the Adapter is not communicating with the NUM CNC over FXServer/APServer.

- If some Value fields are marked "UNAVAILABLE" then this DataItem is not available for e.g. rotary axis not available.
- The DataItems AssetChanged and AssetRemoved Value field are always marked "UNAVAILABLE".
- If all DataItems are not populated, try pressing key F5 to refresh the data on the Webbrowser page.

Example

Open a Webbrowser and type in the Agent URL <http://127.0.0.1:5000/current> => MDTO Adapter and Agent are at the same PC.

creationTime: 2019-03-01T14:01:00Z
 sender: NUMPC
 instanceId: 1551429940
 version: 1.3.0.14
 bufferSize: 131072
 nextSequence: 964
 firstSequence: 1
 lastSequence: 963

Device: NUMCNC; UUID: cncid

Rotary : A

Samples

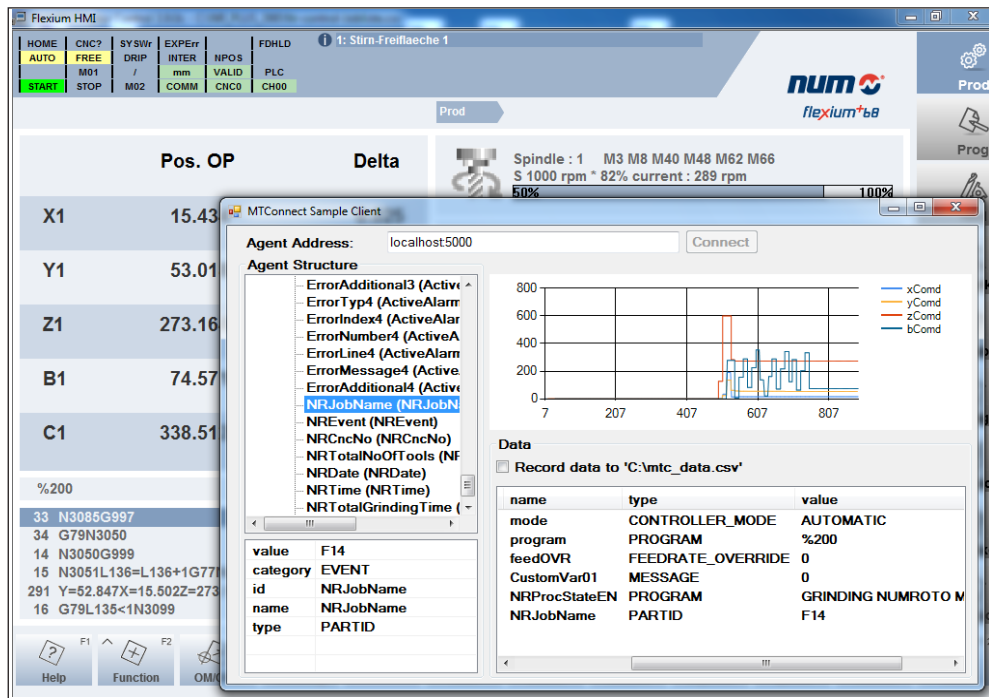
Timestamp	Type	Sub Type	Name	Id	Sequence	Value
2019-03-01T08:45:40.610441Z	Position	COMMANDED	aComd	posA	65	UNAVAILABLE

Rotary : B

Events

Timestamp	Type	Sub Type	Name	Id	Sequence	Value
2019-03-01T13:48:38.5751619Z	Message		CustomVar01	CustVar01	471	0
2019-03-01T13:45:42.6360988Z	Message		CustomVar02	CustVar02	355	UNAVAILABLE
2019-03-01T13:45:42.6360988Z	Message		CustomVar03	CustVar03	356	UNAVAILABLE
2019-03-01T13:45:42.6360988Z	Message		CustomVar04	CustVar04	357	UNAVAILABLE
2019-03-01T13:45:42.6360988Z	Message		CustomVar05	CustVar05	358	UNAVAILABLE
2019-03-01T13:59:27.2222624Z	Program	x:SUB	NRProcStateDE	NRProcessStateDE	874	SCHLEIFEN NUMROTO HERSTELLEN
2019-03-01T13:59:27.2222624Z	Program	x:SUB	NRProcStateEN	NRProcessStateEN	873	GRINDING NUMROTO MANUFACTURING
2019-03-01T13:45:44.3981996Z	Program	x:SUB	programComment	comment	410	
2019-03-01T13:48:38.5751619Z	ControllerMode		mode	id051	470	AUTOMATIC
2019-03-01T13:50:56.7090627Z	Execution		exec	id052	481	ACTIVE
2019-03-01T13:45:44.3981996Z	Program		program	id053	409	%200

Generic view by a MTConnect sample client with connection to the Agent URL <http://127.0.0.1:5000/>
current => MDTO Adapter and Agent are at the same PC = localhost.

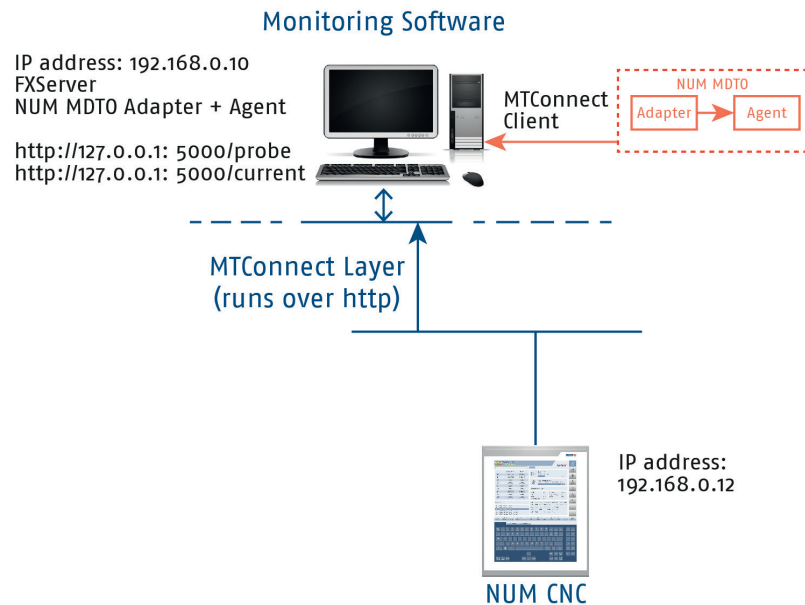


4.10 Common NUM MDTO Network Configuration

Generic common NUM MDTO configuration examples with Flexium CNC series.

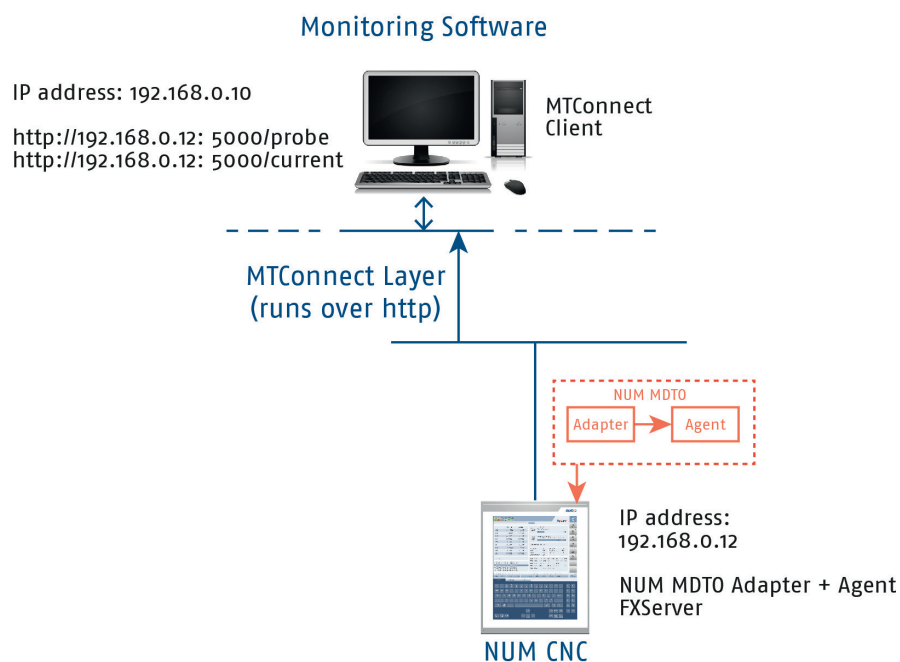
- Example 1**

NUM MDTO component Adapter and Agent are installed at the same PC hardware where the MTConnect client is also installed:



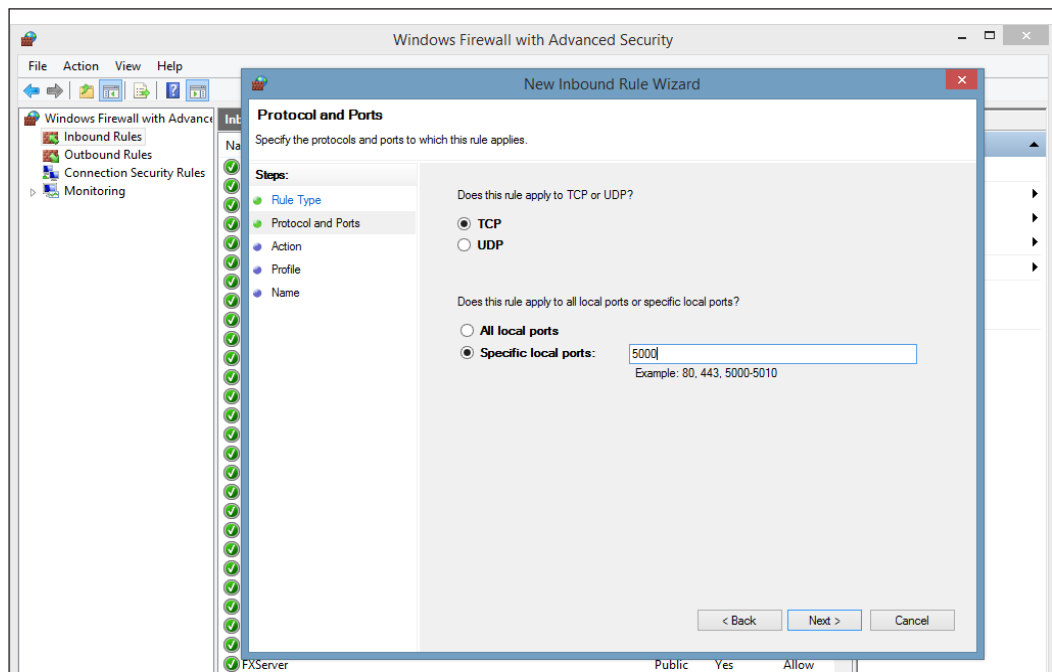
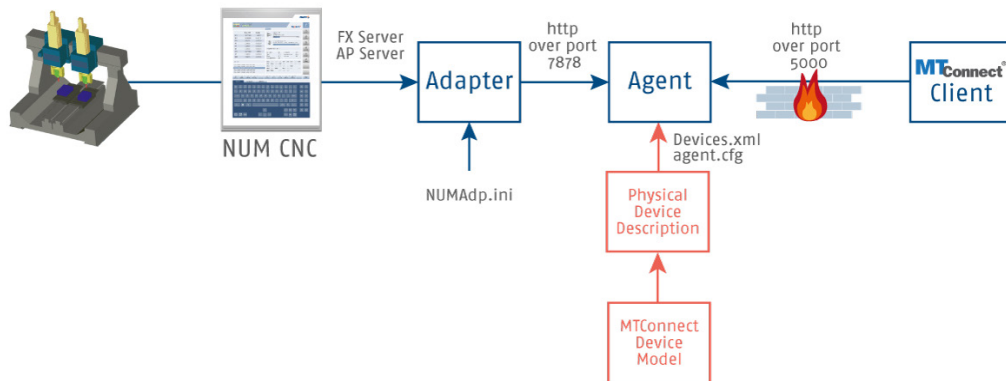
- Example 2**

NUM MDTO component Adapter and Agent are installed at the NUM iPC Panel. The MTConnect client is installed at a different PC hardware:



4.10.1 Windows Firewall

In case, the MTConnect client can't access to the NUM MDTO component Agent, which is located at a different PC hardware (e.g. NUM iPC Panel), by http protocol over TCP transport layer the Windows Firewall could prevent the access over the port 5000. A possible reason could be to enable the port 5000 in the Inbound Rules of the Windows Firewall.



Page deliberately empty

1

2

3

4

5

6

Page deliberately empty

Summary

5	OPC UA Server	131
5.1	Creating a project for OPC UA access	131
5.2	Creating a certificate for the Flexium OPC UA server	132
5.3	Setting up an encrypted connection with the “UaExpert” client	133
5.4	User management in OPC UA	134
5.5	Changing variables via the OPC UA client	135
5.6	Creating events in the Flexium project	136
5.7	Monitoring of events with the OPC UA client “UaExpert”	137

1

2

3

4

5

6

Page deliberately empty

5 OPC UA Server

The standard installation of Flexium RTS / Flexium RTS x64 includes an OPC UA server. Then you can access the variable interface of the controller via a client..

At the same time, communication can be protected by means of encryption.

The CODESYS OPC UA server supports the following features:

- Browsing of data types and variables
- Standard read/write services
- Notification for value changes: subscription and monitored item services
- Encrypted communication according to "OPC UA standard (profile: Basic256SHA256)"
- Imaging of the IEC application according to "OPC UA Information Model for IEC 61131-3"
- Supported profile: Micro Embedded Device Server Profile
- By default, there is no restriction in the number of sessions, monitored items, and subscriptions. The number depends on the performance of the respective platform.
- Events are sent according to the OPC UA standard.

5.1 Creating a project for OPC UA access

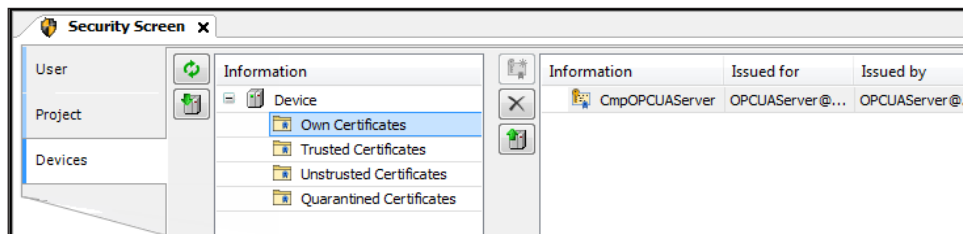
1. Create a new project with a Flexium RTS / Flexium RTS x64 controller.
2. Declare some variables of different types in the *PLC_PRG* program.
3. Add a *Symbol configuration* object below the application.
4. Activate the option *Support OPC UA Features* in the dialog *Add symbol configuration*.
5. Open the symbol configuration in the editor.
6. Click *Build* : The variables are displayed in a tree structure.
7. Activate the variables that you want to change with an OPC UA client. Specify the access rights.
8. Download the project to the controller.

5.2 Creating a certificate for the Flexium OPC UA server

In order to encrypt data and exchange it with the client safely, the server needs a certificate that the client must classify as trusted when a connection is established for the first time.

Requirement: The active path to the controller is set.

1. Install the Security Agent add-on.
2. Click *View Security-Screen*.
3. Select the *Devices* tab.
4. Select the controller in the left view.
All services of the controller that require a certificate are displayed in the right view.
5. Select the service *CmpOPCUAServer*.
6. Create a new certificate for the device. Click the icon 
The *Certificate Settings* dialog opens
7. Define the certificate parameters and *click OK* to close the dialog.
The certificate is created on the controller.



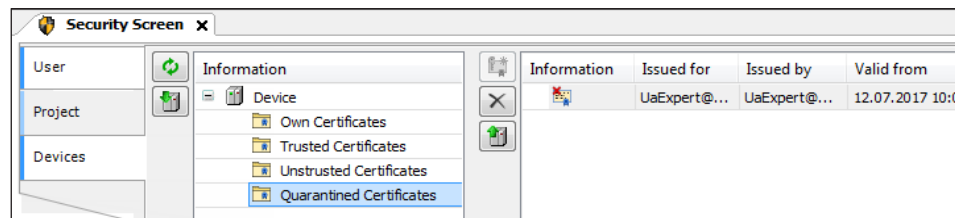
8. Restart the runtime system.

5.3 Setting up an encrypted connection with the “UaExpert” client

The OPC UA client “UaExpert” is freely accessible software that you can download from the Internet. Using this client, you can connect to the CODESYS OPC UA server.

The following description refers to this program. Other OPC UA clients work in the same way.

1. Start the “UaExpert” program.
2. Click *Server ► Add*. The *Add server* dialog opens.
3. Expand *Local ► OPCUAServer@...* in the tree view.
4. Select the connection type *Basic256Sha256 - Sign & Encrypt (uatcp-uasc-uabinary)* and click *OK* to close the dialog. .. **figure:: PNG72_cds_img_uaexpert_1.png alt:**
5. Click *Server ► Connect*. The *Certificate Validation* dialog opens with an error message.
6. Activate the option *Accept the server certificate temporarily for this session* and click *Continue*.
7. In Flexium Tools, click the  symbol. The view is refreshed.
8. Select the certificate folder *Quarantined Certificates*.
The client certificate *UaExpert@...* is displayed in the right view.



9. Drag the certificate to the certificate folder *Trusted Certificates*.
Now the client certificate is classified by the server as trusted.
10. Click *Server ► Connect* in the UaExpert client.
The *Certificate Validation* dialog opens with an error message.
11. Activate the option *Accept the server certificate temporarily for this session* and click *Continue*.
The connection is established and objects are displayed in the *Address Space* view.

5.4 User management in OPC UA

The Flexium OPC UA server supports the Flexium Tools user management. You set the access rights on the server from the *Access rights* tab on the controller.

To do this, select the object *RuntimeSystemObjects* ► *RemoteConnections* ► *OPCUAServer*.

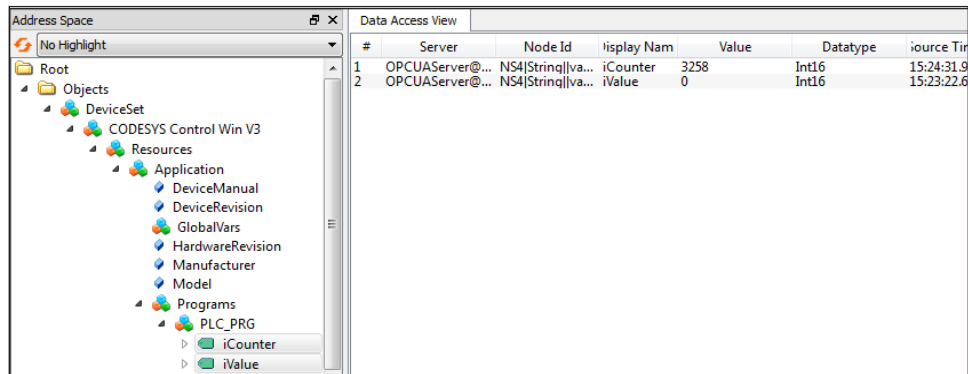
Access rights can be checked at both the service and objects levels. This means that a variable cannot be written by a user, although this user is generally allowed to write to the OPC UA server.

OPC UA service	Access rights
AttributeRead	View
AttributeRead	Change
CreateMonitoredItem	View
ModifyMonitoredItem	View
SetMonitoringMode	View
DeleteMonitoredItem	View
CloseSession	View
CreateSubscription	View
ModifySubscription	View
SetPublishingMode	View
DeleteSubscriptions	View
Publish	View
Republish	View
Browse	View
BrowseNext	View
TranslateBrowsePathsToNodeIds	View
RegisterNodes	View
UnregisterNodes	View

See also *Handling User Management* on the Controller and Tab 'Access Rights'.

5.5 Changing variables via the OPC UA client

1. Expand the object *Objects* ► *DeviceSet* ► *CODESYS OPC UA* ► *Application* ► *Global Vars* ► *GVL* in the “UaExpert” client of the view *Address Space*.
The variables of the global variable list are visible.
2. Select the variables and drag them to the *Data Access View*.
The variables and their current values are shown.



3. Change the variable values by double-clicking the *Value* field.

5.6 Creating events in the Flexium project

The OPC UA server provides the capability of sending standard OPC UA events.

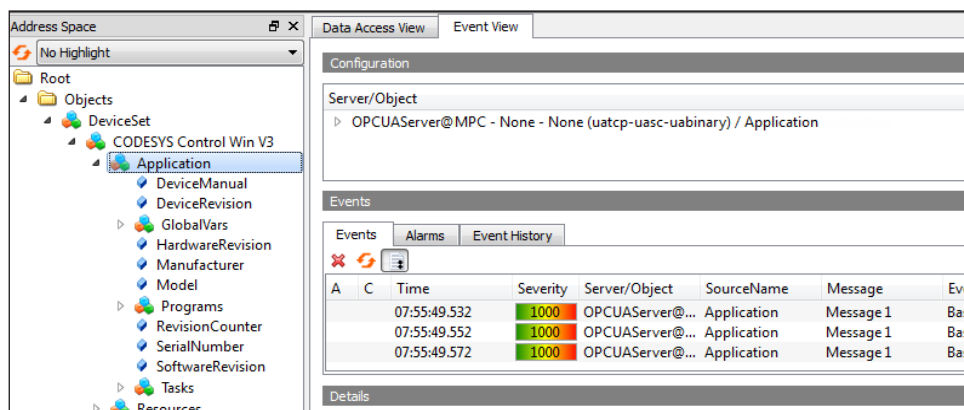
1. Create a new project with a Flexium RTS / Flexium RTS x64 Control Win V3 controller.
2. Add an *Alarm configuration* object below the application.
3. Add an *Alarm class object* below the *Alarm configuration*. Specify a name, for example `Event`. The new alarm group opens in the editor.
4. Select the acknowledgement method *REP*.
5. Add an Alarm group object below the *Alarm configuration*. Specify a name, for example `ApplicationEvent`.
6. Change the following parameters.
 - Monitoring type: `Event`
 - Class: `Event`
 - Message: `"Message 1"`.
7. Add an Visualization object below the *Application*.
8. Add a *Symbol configuration* object below the application.
9. In the Program (for example, POU `PLC_PRG`), add a program call for triggering the event alarm.

```
AlarmManager.AlarmGlobals.g_AlarmHandler.RaiseEvent(Alm_AlarmConfiguration_Alarmgroup_IDs.ID_ApplicationEvent,
Alm_ApplicationEvent_Alarm_IDs.ID_0);
```
10. Add the library `CmpOPCUAProviderAlarmConfiguration` to the library manager.
When the library is added, it connects automatically as a client to the alarm configuration and sends the events to the OPC UA server.
11. Send the project to the controller and start it.

See also Object '*Alarm group*' and Calling event alarms in the program.

5.7 Monitoring of events with the OPC UA client “UaExpert”

1. Start the “UaExpert” program.
2. Click *Server* ► *Add*. The Add server dialog opens.
3. Expand *Local* ► *OPCUAServer@...* in the tree view.
4. Select the connection type *None* and click *OK* to close the dialog.
5. Click *Server* ► *Connect*. An object tree is shown in the *Address Space* view.
6. Click *Documents* ► *Add*. The Add documents dialog opens.
7. Select the “Document Type” Event View. The *Event View* tab opens.
8. Expand the object *Objects* ► *DeviceSet* ► Flexium RTS in the *Address Space* view.
9. Select the object Flexium RTS Control Win V3 in *Address Space* and drag it to the *Event view*. The events are displayed.



Page deliberately empty

Summary

6	OPC DA	141
6.1	Overview on OPC DA	141
6.2	Overview on Flexium OPC DA Server.....	142
6.2.1	CODESYS Gateway V3	143
6.3	Installation of the Flexium OPC Server V3	145
6.3.1	Installation and Registration	145
6.3.2	Usage of the Flexium OPC DA Server	147
6.4	Configuration of the symbols in the programming system.....	149
6.4.1	V3 Interface	149
6.5	CODESYS OPC Server Configuration by OPCConfig.exe	151
6.5.1	Call OPCConfig.exe.....	152
6.5.1.1	Default call	152
6.5.1.2	Open a particular INI file	152
6.5.2	Menu Commands and Configuration Dialogs.....	153
6.5.2.1	File	153
6.5.2.2	Edit	153
6.5.2.3	Settings for OPC server	154
6.5.2.4	Settings for <PLC>	155
6.5.2.5	Settings for connection to <PLC>	157
6.5.3	Simulation Interface	159
6.5.3.1	SIMULATION3	159
6.5.4	Redundant Group	160
6.5.5	Configuration of OPCConfig with OPCConfig.ini	162
6.6	Configuration settings in OPCServer.ini	163
6.6.1	Configuration of the OPC server under section [Server]	163
6.6.2	Configuration of the communication to the controllers	164
6.6.3	Example of an INI file for the V3 interface	165
6.6.4	Configuration of 'Redundant Group'	166
6.6.5	Information on the changed logfilter	168
6.6.6	Configuration of simulation	169
6.6.7	Changing directory and name of INI and LOG file.....	170
6.7	Connection to OPC servers on other computers with DCOMCNFG.EXE	171
6.8	Error Diagnosis	172
6.8.1	Means of error diagnosis	172
6.8.2	Brief checklist.....	172
6.8.3	Error cases.....	173

1

2

3

4

5

6

Page deliberately empty

6 OPC DA

6.1 Overview on OPC DA

OPC¹ is a standardized interface for access on process data. It is based on the Microsoft standard **COM/DCOM**² and enlarged due to the requirements of data access in automation, where the interface is mainly used to read data from or write data to the controller.

Typical OPC clients are visualizations, programs in purpose to register operating data, etc. Typical providers of OPC Servers are PLC systems and field bus interface cards.

The OPC Server is an executable program that is started automatically during the establishment of a connection between client and PLC. Hence, the OPC Server is able to inform the client about changed variable values or states.

The OPC Server provides all variables (referred to as **Items**) that are available on the PLC (**Item Pool** or **Address Space**). These items are managed within a **Data Cache** ensuring a fast access to their values. Also possible is a direct, not cached access on the items of the PLC.

Additionally alarm and event information of items can be considered to be another type of data. For this purpose items can be linked with digital- and limit alarms or events. For doing this it is highly recommended to use the **AECon figuration** tool. The tool can be found in the program directory of CODESYS. However, in the OPC Server the items can be organized in so-called **Groups (Private** and – only for V2.3 - **Public)**.

The private groups can be composed in the client arbitrarily from particular items. Initially they do not effect the groupings in the OPC Server, but if necessary can in case of V2.3 be transformed to Public Groups. Working with Private Groups for example is useful in order to be able to activate or deactivate certain groups of variables with just one single command, depending on whether they should be accessible or not.

Grouped data should be read from the OPC Server consistently, i.e. all variables should be read at the same time. Regard however that this is not always possible in case of target systems with restricted communication buffers!

Due to the characteristics of DCOM it is possible to access the OPC data on another computer. It is also allowed that more than one client accesses the data source at the same time. The applicability of different languages (C++, Visual Basic, Delphi, Java) is another benefit of employing COM.

¹ OPC = OLE for Process Control; OLE = Object Linking and Embedding.

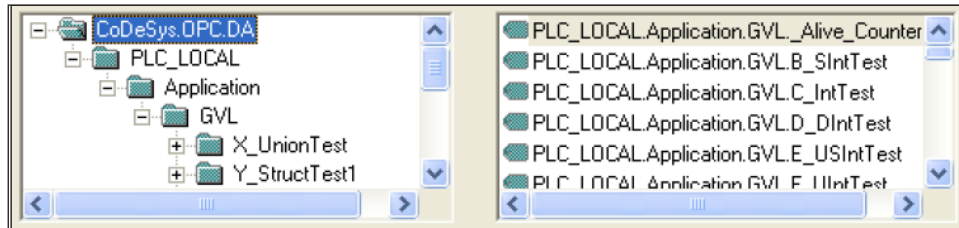
² COM = Component Object Model (base for OLE); DCOM = Distributed Component Object Model.

6.2 Overview on Flexium OPC DA Server

The CODESYS OPC Server V3 is based on the PLCHandler of 3S-Smart Software Solutions GmbH. This communication module permits a direct communication to **all** PLCs programmable with CODESYS: Thereby, the OPC Server V3.x supports the following OPC specifications:

- OPC Common Definitions and Interfaces Version 1.0
- Data Access Custom Interface Standard Version 1.0
- Data Access Custom Interface Standard Version 2.05A
- Data Access Custom Interface Standard Version 3.0
- Data Access Automation Interface Standard Version 2.0
- Alarms and Events Custom Interface Version 1.10.

Furthermore hierarchical browsing (read of variables available on the PLC) for items is supported.



Communication between OPC Server and PLC can be carried out via one of the following interfaces:

- Gateway V3
- ARTI
- ARTI3
- SIMULATION3.

Which interface has to be chosen for the communication depends on the particular PLC and has to be configured in the INI file of the OPC Server (parameter: interface type).

With help of the tool OPCConfig the INI file can be comfortably configured. See chap. 5.5 “CODESYS OPC Server Configuration by OPCConfig.exe”.

Furthermore a missing PLC can be simulated.

A symbol file can be created to provide the OPC server with all the symbols of the missing PLC file (XML file or SDB file).

6.2.1 CODESYS Gateway V3

A symbol/item list has to be generated in the programming system. Within the *Symbol Configuration* page, variables can be chosen for the symbol list. Depending on the settings in the target description file, the symbol list is transferred either implicitly within the application or as a separate child application. After this, the symbols can be browsed by the OPC server independent of the transfer method. Additionally a symbol file with extension XML is stored in the project path. If the developer of the visualization (OPC client) has not a PLC, this XML file can be used for simulation purposes.

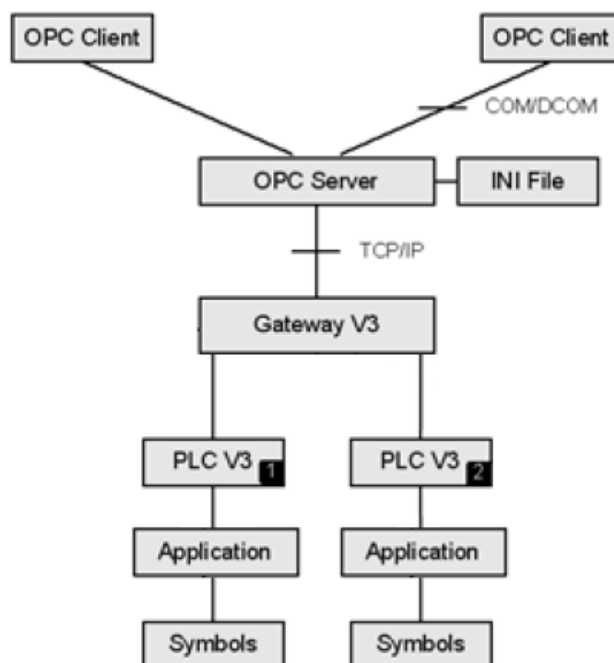
The XML file is named according to the following syntax:

```
<Project name>.<Device name>.<Application name>.xml
```

The communication between the OPC server and a CODESYS V3 controller is always done via CODESYS Gateway V3.

In contrast to Gateway Server V2.3, this server is not bound any longer to the Windows operating system and can be positioned at any desired position within the PLC network except embedded within PLCs.

See in the following picture a possible structure of a CODESYS V3 PLC network:

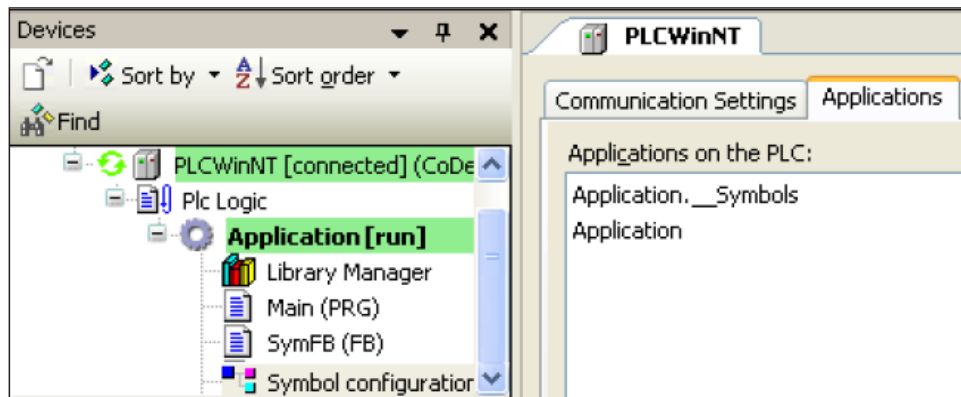


The application as well as the symbol information are filed on the PLC. Requests concerning the symbol information, the values and the status information are handled by the PLC itself. A symbol list is generated based off of the variables defined in the Symbol Configuration. Usually the symbol list is implicitly transferred to the PLC with the download of the PLC application. However, this can also be done in an explicit transfer with downloading a separate child application on the plc, which then can be handled separately.

In this case the syntax of the symbol application is:

```
<Application name>.__Symbols
```

If the *Symbol configuration* e.g. has been defined for application *Application* during downloading the application, also the symbol information gets loaded to the PLC as *Application.__Symbols* :



The symbol application must be regarded as „*normal*“ application, also concerning the maximum number of applications on a PLC.

Contemporaneously with the application download the symbol list will be exported in a symbol file (XML format) and stored in the project directory.

The XML file is named according to the following syntax:

```
<Project name>.<Device name>.<Application name>.xml
```



If the developer of the visualization (OPC client) has not available a PLC, this symbol file can be used for simulation purposes.

6.3 Installation of the Flexium OPC Server V3

OPC Server V3 is delivered in setup form. All files needed for the OPC communication get installed and the OPC Server will be registered automatically as COM server.



For using the OPC Server V3 on a Windows Vista system, it is strongly recommended to run the OPC client in Windows XP SP2 compatibility mode and with extended administrator rights in order to get a continuous updating of the values.

6.3.1 Installation and Registration

Further on, there is the possibility to register resp. to uninstall the OPC Server manually either as COM server or as service.

1..Parallel installation and registration

With command:

```
WinCODESYSOPC /RegServer
```

the server is registered as COM server.

Thereby as location path always the current position of WinCODESYSOPC.exe will be used. Thus the call only may be done from a local path.

2..Registration as service

With command:

```
WinCODESYSOPC /Service
```

WinCODESYSOPC.exe gets installed as system service.

Started once, the service will stay "started" until the system gets terminated. The communication to the configured PLCs survives.

Also here the service gets installed in the current position of WinCODESYSOPC.exe.

3..Uninstalling the

With command:

```
WinCODESYSOPC /UnRegServer
```

all entries of the OPC Server will be removed from the registry. The installed files will not be removed. Settings in the registry are to be found under path "*Computer\HKEY_LOCAL_MACHINE\SOFTWARE\3S-Smart Software Solutions GmbH\CODESYSOPC*".

Entries are "*AEConfigPath*", "*INIFileName*" and "*LOGFileName*".

The values *INIFileName* and *LOGFileName* can contain name and path information together.

The name of the INI file is "*OPCServer.ini*". See also chapter 5.6.7 "*Changing directory and name of INI and LOG file*".

1

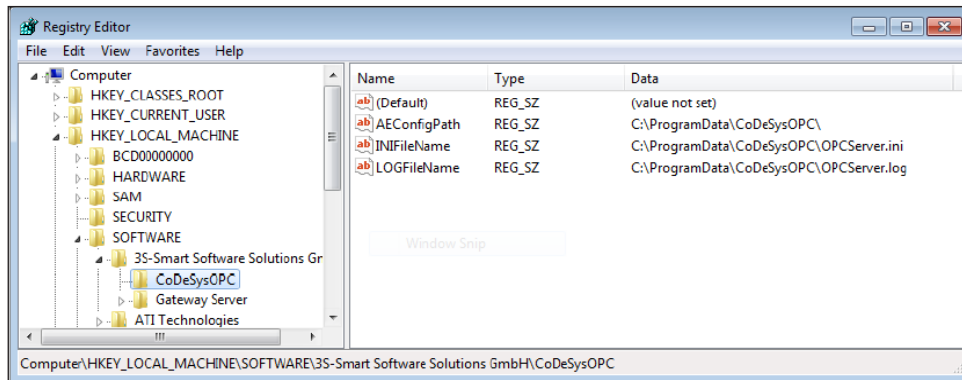
2

3

4

5

6



If a labeled version of CODESYS is used, consider that not only the product name (in the figure above “CoDeDysOPC”) is modified but also the vendor name (in the figure above “3S-Smart Software Solutions GmbH”) and their subordinate entries could also be changed.

6.3.2 Usage of the Flexium OPC DA Server

To receive a communication from the PLC via OPC server to the OPC client, the following steps are necessary:

- 1 The application on the control contains symbols (Items) configured with symbol configuration. See chapter 5.4 *“Configuration of the symbols in the programming system”*.
- 2 The OPC server receives all necessary settings to establish the connections in *“OPCServer.ini”*. The name and path of the OPC server INI file is by default *“OPCServer.ini”* in the installation directory. These default settings can be easily modified, see chapter see chapter 5.6.7 *“Changing directory and name of INI and LOG file”*. How to create the *OPCServer.ini* see chapter 5.5 *“CODESYS OPC Server Configuration by OPCConfig.exe”*.
- 3 After selection of server *CODESYS.OPC.DA* the desired OPC-Client can browse all configured symbols.

After the installation, the OPC Server should be offered for selection by the OPC client (e.g. visualization).

The name of the OPC Server is **CODESYS.OPC.DA**.

The OPC Server will be started automatically by the operating system as soon as a client establishes a connection. The OPC Server then will terminate automatically as soon as all clients have closed their connections to the server.

There will be no OPC Server icon in the task bar, but it will be only handled in the Windows Task Manager as a process.

For each configured PLC the OPC Server V3 generates the status variables: **_CommState** and **_CommStateOK**. **_CommState** can be interpreted from an OPC Client and the following states can be requested:

```
STATE_TERMINATE      = -1
STATE_PLC_NOT_CONNECTED = 0
STATE_PLC_CONNECTED  = 1
STATE_NO_SYMBOLS     = 2
STATE_SYMBOLS_LOADED = 3
STATE_RUNNING        = 4
STATE_DISCONNECT      = 5
STATE_NO_CONFIGURATION = 6
```

_CommStateOK is a Boolean variable indicating in state TRUE a communication with the OPC server. The name of the PLC can be taken from the prefix of the Boolean variable. If the variable is FALSE, currently no communication is possible. For example, this case can happen during a download.

In redundant groups the status variables are generated for each PLC in the group.

Then for differentiation the name of the status variables follows the syntax:

```
__CommStateOK<PLCName>
__CommState<PLCName>
```

1

2

3

4

5

6

Example

```
OPCServer.ini:  
[RedGroup:0]  
Name=MyGroup  
NumPLCs=2  
DefaultPLC=0  
PLCName0=SPWinV3  
PLCName1=SPWinV3R
```

Thus, there are the following status variables:

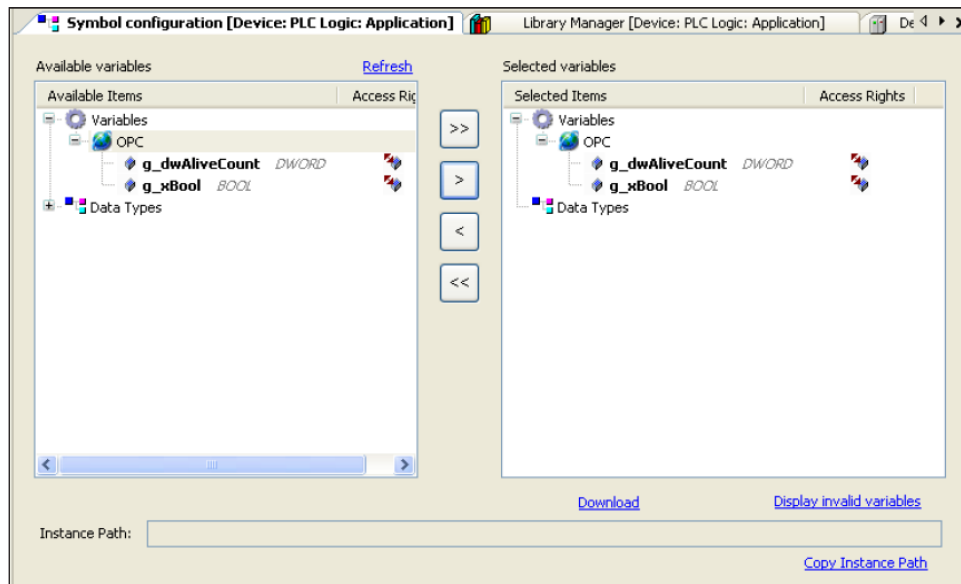
```
___CommStateOKSPWinV3  
___CommStateSPWinV3  
___CommStateOKSPWinV3R  
___CommStateSPWinV3R
```

6.4 Configuration of the symbols in the programming system

6.4.1 V3 Interface

Step 1: Add Symbol Configuration

Select the *Application* object in the device tree and add object *Symbol configuration* (*Project-> Add Object -> Symbol configuration*). The symbol configuration editor appears:



Step 2: Configure symbols

The dialog shown above will be opened on a double-click on the symbol configuration object.

In order to get the currently available item pool, first click on the *Refresh* button.

Thereupon, in the left part of the editor window all variables available in the application will be displayed. These are only variables with read or write access. In order to add a variable or node currently selected in the left tree to the right window use the arrow button *< >* between the two windows.

The access right for a selected item can be modified in the right window by clicking on the symbol in column *Access Rights*.

Each mouse-click switches to the next one of the possible symbols and thus to the respective access right:

-  read+write (read and write access),
-  write-only (only write access),
-  read-only (only read access).

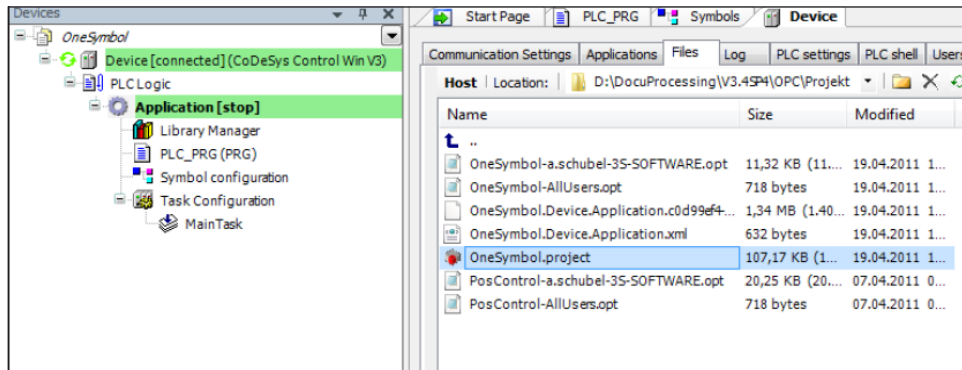


In case of arrays and structures the aggregate variable as well as the particular components will be generated implicitly.

If the symbol configuration will be modified in online mode, the modified application can be downloaded via button *Download*.

Step 3: Login

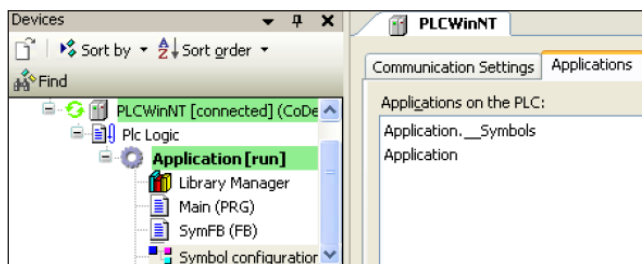
With commanding *Login* the application will be compiled and downloaded. If the controller is configured in a way that it downloads the symbols within the task application (default way), the following happens: The application with the implicit symbol information will be created and downloaded to the plc. The symbol file is stored next to the project file as well.



If the controller is configured that it downloads the symbols in its own application, the following happens: The syntax of the symbol application is:

`<Application name>.__Symbols`

The application and the child application will be created and downloaded to the PLC. The symbol file is stored next to the project file as well.



The symbol application has to be regarded as „normal“ application, also concerning the maximum number of applications on a PLC.

6.5 CODESYS OPC Server Configuration by OPCConfig.exe

For the connection between OPC client and the controller(s)/PLC(s) there are parameters defined in the file *"OPCServer.ini"*.

Regard that the default path of the INI file depend on the used operating system. The file can be found in versions of Windows prior to Vista in the in the installation directory of the OPC server (default „C:\Program Files\3S CODESYS\CODESYS OPC Server V3“).

In all subsequent windows systems the file is put in „C:\ProgramData\CODESYSOPC“ by default. The OPC server expects a file as it is defined in the Registry. That means by default the INI file name is *"C:\ProgramData\CODESYSOPC\OPCServer.ini"*.

This setting can be be changed by visiting chapter 5.6.7 *"Changing directory and name of INI and LOG file"* . Only this file is used from the CODESYS OPC server for establishing a connection. Files with other names are ignored.

Certainly *"OPCServer.ini"* can directly be edited. But it is much more comfortable to do the settings with the configuration tool *OPCConfig* be found in the installation directory (by default „C:\Program Files\3S CODESYS\CODESYS OPC Server V3“).

1

2

3

4

5

6

6.5.1 Call OPCConfig.exe

6.5.1.1 Default call

With calling *OPCConfig.exe*, the tool '*OPCConfig*' opens with the registered INI file.

6.5.1.2 Open a particular INI file

OPCConfig can be called with a command line parameter to open a particular INI file as well. If the *INI file* isn't stored in the same directory as *OPCConfig.exe*, call the tool with a full path in argument *<file name>*.

Syntax:

```
"<path of OPCConfig>\OPCConfig.exe" /open <file name>
```

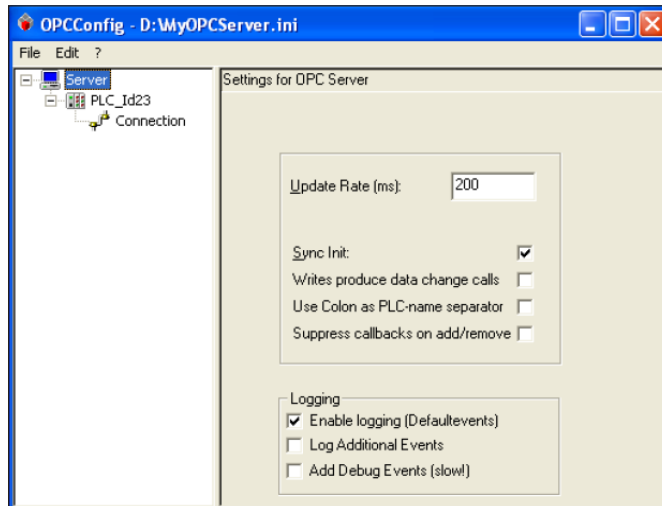
Example

Following command opens the file "*MySpecialConf.ini*"

```
C:\Program Files\3S CODESYS\CODESYS OPC Server V3\OPCConfig.exe"  
/open MyOPCServer.ini
```



The OPC server doesn't use a particular file if it is not registered, meaning that is not the active INI file.



6.5.2 Menu Commands and Configuration Dialogs

6.5.2.1 File

This menu provides commands for loading and saving the configuration files to / from the configuration tool:

Command	Description
Open <Ctrl>+<O>	For editing an existing configuration. The default dialog for opening a file will appear where you can select an already existing INI file. The filter is automatically set to <i>OPCConfig Files *.ini</i> . The configuration described in the chosen INI file will be loaded to the configuration tool.
New <Ctrl>+<N>	For creating a new configuration. If a configuration is currently opened you will be asked whether it should be saved before getting closed. Then the configuration tool will show the default settings like after having started the tool.
Save <Ctrl>+<S>	Saves the current configuration to the currently loaded INI file. Regard that modifications of a configuration do not get valid if the INI file is not the active one.
Save as	Saves the current configuration to a file, which you can specify in the default dialog for storing a file. Regard that modifications of a configuration do not get valid if the INI file is not the active one.
<n> recently opened ini-files	List of the INI files which had been edited since having started the tool last time. You can select any of them to get it reloaded to the configuration tool.
Exit	The tool will be terminated. If any unsaved changes are detected, the user will be prompted to save before exiting.

6.5.2.2 Edit

This menu provides the commands for editing the configuration tree in the left part of the configuration.

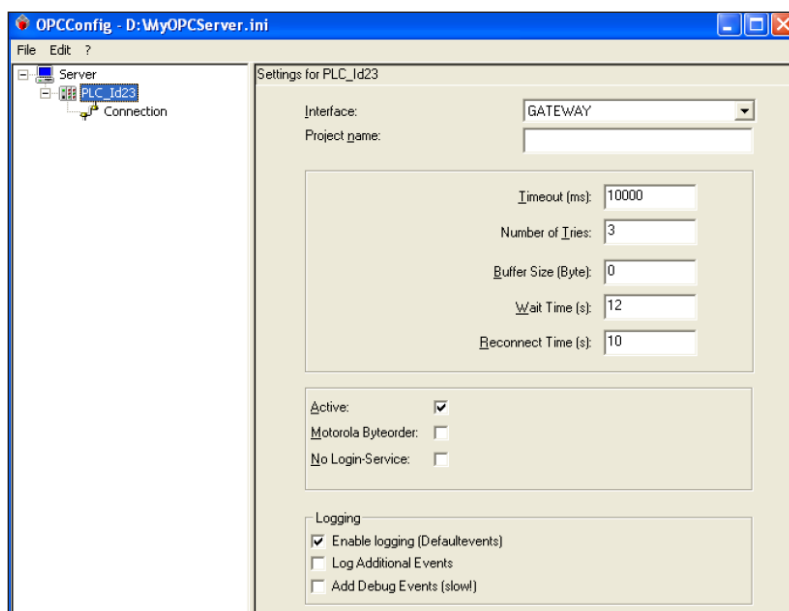
Command	Description
New Redundancygroup <Ctrl>+<G>	A Redundant Group entry will be added below <i>Server</i> . If there are already <i>PLC or Redundant Groups</i> listed in the tree, the new Redundant Group will be appended at the end. Per default a new entry is named <i>Redundant<n></i> , whereby n is a running number starting with 1. If you want to rename the entry, select it in the tree and either use command Edit/Rename PLC or perform a mouse click on the entry to get the string editable.
Append PLC <Ctrl>+<O>	A PLC entry will added below <i>Server</i> . A new PLC will always be appended at the end of the existing tree. Per default a new entry is named <i>PLC<n></i> , whereby n is a running number starting with 1. If you want to rename the entry, select it in the tree and either use command Edit / Rename PLC or perform a mouse click on the entry to get the string editable.
Delete PLC <Ctrl>+<D>	The currently selected PLC entry will be removed from the configuration tree.
Rename PLC <Ctrl>+<R>	The currently selected PLC entry can be renamed.
Reset PLC <Ctrl>+<Z>	Not available yet! The settings of the currently selected PLC entry will be reset to the default values defined in the PLC Default Settings.

6.5.2.3 Settings for OPC server

This dialog is available when the root entry *Server* is selected in the configuration tree. It provides the following settings.

Update Rate (ms)	Default: 200. Basic update rate of the OPC-Server in milliseconds. This is the cycle time according to which all item data are read from the PLC. The data get written into the cash with which the client communicates with a separately defined update rate. Note: With the status information the server also provides a parameter Bandwidth Usage giving the ratio of the actual update rate to the configured update rate in percents.
Sync Init	Default: not active. If this option is activated, a synchronous connection is initialized.
Writes produce data change calls	See chap. 5.6.1, Entry " <i>ReadCyclesAfterWrite</i> ".
Use Colon as PLC-name separator	See chap. 5.6.1, Entry " <i>UseColonAsPlcDivider</i> ".
Suppress callbacks on add/remove	See chap. 5.6.1, Entry " <i>GroupUpdateBehaviour</i> ".
Enable Logging (Defaultevents)	Default: not active. If this option is activated, any actions as well as errors on the OPC server will be recorded in a LOG file. This file will be stored in the installation directory and will be named OPCServer.log. After a shutdown of the OPC Server the LOG file can be evaluated. The messages of several OPC sessions are lined up in one LOG file until it reaches the size of 1 MB. Then the current date will be added in the file name (OPCServer<date>.log, e.g. OPCServer12.10.2008.log) and the file will be saved. After that a new LOG file <i>OPCServer.log</i> will be created
Log Additional Events	Default: not active. If this option is active, then additional events are written into the LOG file.
Add Debug Events (slow)	Default: not active. If this option is active, then debug events are written into the LOG file.

6.5.2.4 Settings for <PLC>



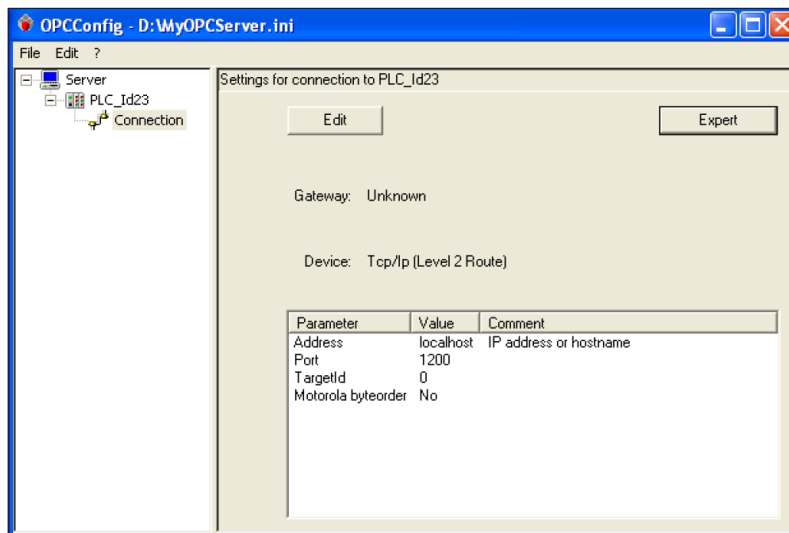
This dialog is available, if the <PLC> entry is selected in the configuration tree.

Interface	Name of the interface used for the communication between programming system and PLC. The selection list per default contains the following interface types: ARTI, GATEWAY, SIMULATION for projects created with CODESYSV2.3; GATEWAY3, SIMULATION3 for V3 projects. Concerning the SIMULATION or SIMULATION3 interface please see chap. 5.5.3, "Simulation Interface".
Project name	Name of the currently concerned symbol file. It has to be used in case of simulation. If there is no path the file has to be found in the OPC server directory. In Flexium V3 the symbol file name follows the syntax: <code><project name>.<device>.<application>.xml</code>. In Flexium V2.3 : <code><project name>.sdb</code>. Note. This entry is only used if interface SIMULATION or SIMULATION3 is contemporary used too.
Timeout (ms)	Default: 10000. If within this period of time the OPC server does not receive any feedback on a sent service from the PLC, it will close down automatically.
Number of Tries	Default: 3. Number of attempts the gateway communication driver will make to transfer a block correctly. As soon as the last attempt has failed, a communication error message will be created. (this entry is only relevant for drivers which support a communication via blocks, Level 2).
Buffer Size (Byte)	Default: 0. Size of the communication buffer on the target device; if the entry is "0" the device driver will be searched for this size information. If it does not provide any, it will be assumed that the buffer is unlimited.
Wait Time (s)	Default: 10. Time in seconds the OPC server will wait until the controller is ready (important during the automatic start of the controller).

Reconnect Time (s)	Default: 15. Time interval, according to which the OPC server repeatedly tries to reconnect with the controller via the gateway.
Active	Default: activated. Only if this option is activated, the PLC will be regarded by the OPC Server.
Motorola Byteorder	Default: not activated. The target system of the project does not use the Motorola byte order (this aspect is to be given special regard when working with 68K, 8051, Power PC !). This Entry has to coincide with the entry in the settings for connection of the PLCr.
No Login Service	Default: not activated. This option must be activated for specific target systems which request the sending of a login service.
Logging	Options for the logging of events in a LOG file, see chap. 5.6.7. Per default the logging of the default events is activated (<i>Enable logging (Default events)</i> : Information on setup and shutdown of the connection). For error analysis you can explicitly activate the logging of additional events (<i>Log Additional Events</i>) and debugging events (<i>Add Debug Events (slow!)</i>). Note that the latter two options will drastically increase the system load!.

6.5.2.5 Settings for connection to <PLC>

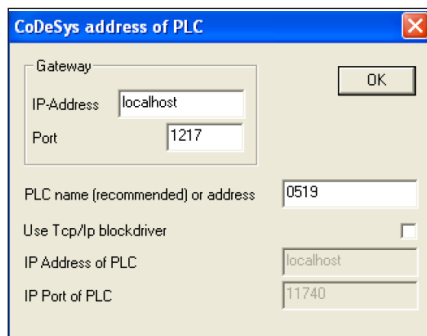
This dialog will be available if entry *Connection* below a <PLC> is selected in the configuration tree..



Here the communication settings as defined for the application in the respective CODESYS V3 project must be entered.

If no communication setting is defined the OPC-Server ends with STATE_TERMINATE. So have a closer look at the appropriate LOG file, where all sequences the OPC-Server had done are listed. There you can find the logged and unsuccessful attempts for starting communication.

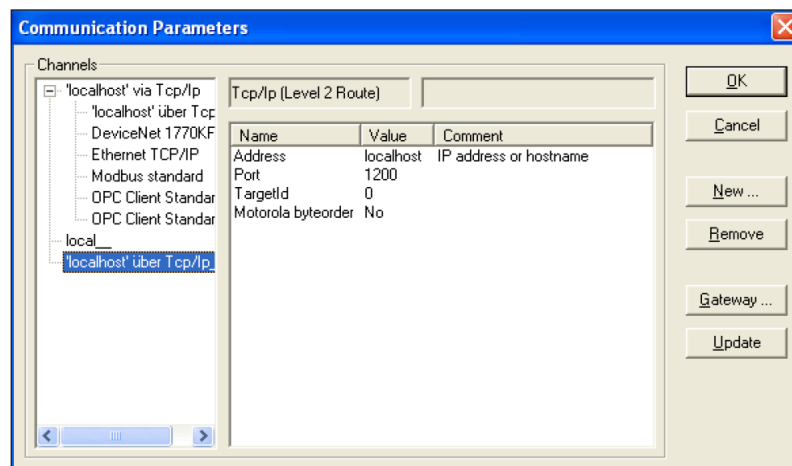
For creating a new communication resp. for modifying an existing one, use button Edit to open the dialog *CODESYS address of PLC*:



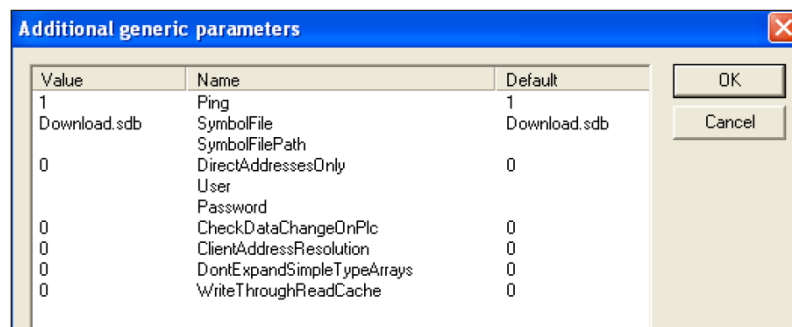
Enter the *IP address* and port for the gateway and for identifying the PLC specify either the name or address (*PLC name (recommended) or address*), or address and port of the Tcp/Ip blockdriver (*Use Tcp/blockdriver*); each the settings must correspond to those defined in *Communication Settings* tab of the Device Editor for the concerned CODESYS project.

After closing the dialog with *OK* the settings will be displayed in the main dialog.

In a CODESYS V2.3 project the valid settings are performed in the dialog *Communication Parameters*. From all the channels shown in the tree structure the valid one can be selected for the plc.



Extended configuration is possible via button *Expert*, in dialog *Additional generic parameters*:



These settings are described in the programming guide for the PLC Handler; if required, a value can be edited after a mouse-click in the *"value"* column.

6.5.3 Simulation Interface

If the PLC (device) configured in the Flexium Tools project is not available, the OPC Server allows to work in simulation mode.

For this purpose only the symbol file, generated by the CODESYS project, is needed. The symbols exported to this file can be used in the OPC Server, however the values of the symbols will not be accessible in simulation mode.

For simulation with a V3 project, in the PLC settings configuration (see chap. 5.5.2.5) select interface type "SIMULATION3".

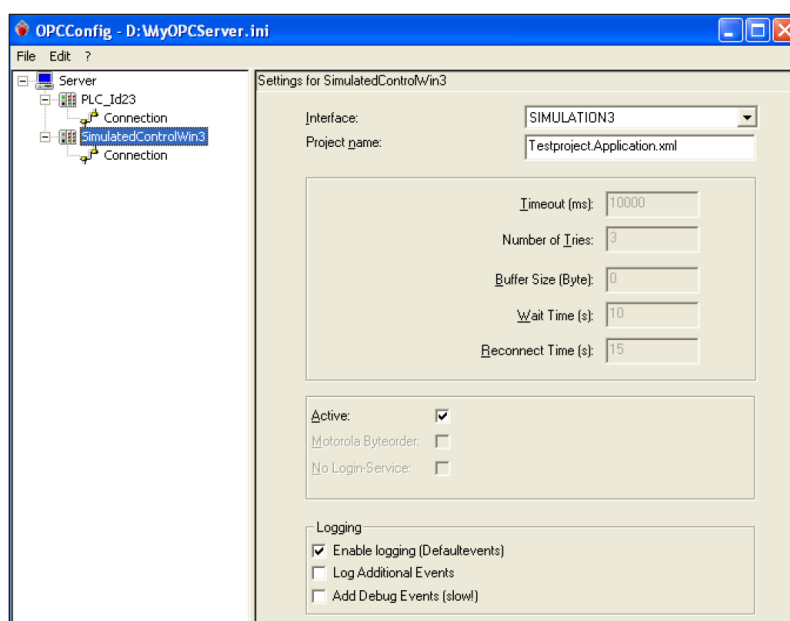
6.5.3.1 SIMULATION3

In the OPC server INI file, a simulation access is configured by selecting the *Interface* SIMULATION3 and by setting the name of the symbol file in *Project name*.

The symbol file is automatically generated by a build command of a CODESYS V3 project when a symbol configuration exists and is stored with the name extension XML next to the project file.

If this file is stored in the OPC server directory, then only the project name has to be specified here. But it can also be copied to any location, then under *Project name* also the directory name has to be specified.

Dialog with configuration of a simulated access to the stock of symbols of *SimulatedControlWinV3*:

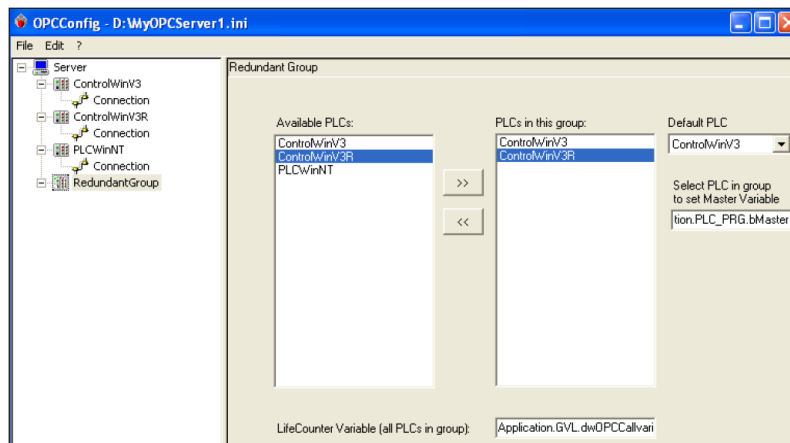


6.5.4 Redundant Group

A *Redundant Group* (also “redundancy group”) serves to ensure that the OPC Server gets values of the desired variables (items). Even if the communication with the primarily desired PLC is not working and is then taken over by a parallel-operated control.

For this purpose the group defines two PLC connections providing the same set of items. Using a *master* variable, can be used to determine which PLC will be read from.

The needed settings for the OPC server INI file are performed with the tool OPCConfig.exe as follows: Command *Edit/New RedundancyGroup* opens the following dialog for configuring a redundant group.



All PLCs currently configured are listed in the *Available PLCs* window. In order to add a PLC to the redundancy group, select it here and use the *arrow button* to add it in the *PLCs in this group* window.

Regard that all PLCs added to the group must provide identical sets of variables.

From this window, up to two controls can be selected.

Default PLC	Define the PLC from which the OPC Server should read the items, if the check of the Master Variable does not provide a unique one.
Select PLC in group to set Master Variable	Select a PLC in the ‘<i>PLCs in this group</i>’ window and enter an Boolean variable which is used in the application and which should be the “master variable” for this PLC. The OPC Server will check the master variable of each PLC in the group and will read the values from that PLC whose master variable is TRUE. If the master variables do not determine uniquely from which PLC the values should be read, they will be read from the default PLC. The values can only be written if at least one PLC sets the master variable. The name of the master variable in CODESYS follows the following syntax: <Application>.<POU>.<variable> The name of variables in CODESYS V2.3 contains no application name and global variables contain no POU name. So the structure looks like: <POUname>.<variablename> or .<NameGlobalVariable> Note. The CODESYS OPC Server V3 does not work at all with no master variable. If the master variable of the redundant group is missing, this will be documented in the LOG file.

LifeCounter Variable (all PLCs in group)	Via this variable each PLC can check, whether the OPC Server is still active. It must be defined for this purpose in all PLCs of the group and the life counter variable type must be DWORD. The name of the LifeCounter Variable in CODESYS V3 follows the following syntax: <code>Application>.<POU>.<variable></code> The names of variables in CODESYS V2.3 contain no application name and global variables contain no POU name. So the structure looks like: <code><POUname>.<variablename></code> or <code>.<NameGlobalVariable></code>
--	--

1

2

3

4

5

6

6.5.5 Configuration of OPCConfig with OPCConfig.ini

For defining the behavior of the tool OPCConfig it is possible to place the INI file „*OPCConfig.ini*“ in the working directory of the tool.

Under section *[Config]* you can register vendor and product name. Under section *[PLCDefaults]* you can register default settings for the connection to the controller, which the tool will use, if a new controller is added. The *INI-file* is read every startup of *OPCConfig*.

Example

```
[Config]
vendor=3S-Smart Software Solutions GmbH
product=CODESYSOPCDA and CODESYSOPCAE
inifilevalue=<name of the registryvalue, containing the OPServer-inifilepath>
[PLCDefaults]
interface=GATEWAY
active=1    motorola=1
nlogin=0
timeout=12
tries=5
bufferSize=5000
reconnecttime=13
v3gatewayport=1217
v3gateway=localhost
v3address=1234
```

Setting	Description
[Config]	
vendor=3S-Smart Software Solutions GmbH	Vendor
product=CODESYSOPCDA and CODESYSOPCAE	Product name
[PLCDefaults]	
interface=GATEWAY	Default interface
active=1	
motorola=1	
nlogin=0	
timeout=12	
tries=5	
bufferSize=5000	
reconnecttime=13	
v3gatewayport=1217 v3gateway=localhost v3address=0000	Default settings for connection with CODESYS Gateway V3. The settings are used, if a PLC with CODESYS Gateway V3 interface is added.

6.6 Configuration settings in OPCServer.ini

The INI file of the OPC server configuration (*OPCServer.ini*) must be stored in all previous versions of Windows Vista in the installation directory of the OPC server (by default "C:\Program Files\3S CODESYS\CODESYS OPC Server V3").

In all subsequent operating systems of Windows Vista, the file is by default stored in "C:\ProgramData\CODESYSOPC".

Here under the section [Server], the OPC server is configured and under the section [PLC:<PLC-Name>], the communication to that controller is defined.

6.6.1 Configuration of the OPC server under section [Server]

Settings, only important for the OPC server, are listed under the section [Server] in the file "OPCServer.ini". The entries listed here may partly be selected in the *OPCConfig tool*. See chap.5.5.2.4 "Settings for OPC server".

Setting	Value (example)	Description	Necessary
PLCs	=1	Number of controls.	Yes
PLCx	=<PLC-Name>]	This variable will assign the name of a controller. The x in the name stands for the index x from 0 to number of controllers -1 .	Yes
logevents	=1	Turning on the logging mechanism.	No
logfilter	=63	Filter for the log file.	No
memloginterval	=1	Interval in seconds at which the log entries of the memory consumption are generated. If it is 1, every second a log entry is generated. It is only used if the log entries or the memory consumption is enabled in logfilter.	No
ReadCyclesAfterWrite	=2	On each written item the OPC Server sends a callback to the client. In the past, fast changing variables did not update their value for each callback, and didn't update the client. In order to ensure an update anyway, you can use the "ReadCyclesAfterWrite" setting: It specifies the number of read-cycles after the last writing after which the values in the client should get updated in any case, even if they have not been changed. (Standard: The Callback will be generated after 2 read cycles. For any other value, the INI file has to be edited manually.)	No
UseColonAsPlcDivider	=1	Here a value can be defined. If the value is 0, the items remain unchanged. If the value is set, the first "." in the item name is replaced by a colon.	No
Updaterate	=200	Update rate in milliseconds.	Yes
GroupUpdateBehaviour	=1	In this case, data changed callbacks for the activated/deactivated item are generated. If it is 0, only callbacks of all items in the group are produced, every time an item (or more items) within a group is (are) activated/deactivated.	No

The following table lists the possible options for the logfilter.
If multiple entries are to be activated, these add up and enter the result in 'logfilter'.

Logfilter	Description
15 (16#0000000F)	Generate standard log entries.
32 (16#00000020)	Generate additional log entries.
48 (16#00000030)	Generate debug log entries.
512 (16#00000200)	Generate log entries about the memory consumption. The entry will be created every "memloginterval" seconds.

6.6.2 Configuration of the communication to the controllers

Under the section *[PLC:<PLC-Name>]* the communication settings are done for the related controls.
This information is passed to the underlying *PLCHandler*.

All settings particular the special are described in detail in "*PLCHandler Programming Guide*".

At least partially the entries can be selected using the tool *OPCConfig*. See chap. 5.5.2.5 "*Settings for connection to <PLC>*".

The most important entry here is the address of the controller. It depends on the interface type, which settings have to be set.

Example for interface type GATEWAY3

Setting	Description
gateway=Tcp/Ip	CODESYS Gateway V3 settings.
gatewayaddress=localhost	IP Address.
gatewayport=1217	Port.
Device=TCP/IP	CODESYS Control V3 settings.
parameters=2	Parameter to set address and port.
parameter0=Address	
value0=050C	Here insert CODESYS PLC address. It must be identical to the node address, as defined in the programming system.
parameter1=Port	
value1=11740	Here insert the port address.

6.6.3 Example of an INI file for the V3 interface

Setting	Description
[Server]	Settings for the OPC Server.
PLCs=1	Connection to two PLCs is intended.
PLC0=PLC1	Name of the first PLC: PLC1.
logevents=1	Switch-On of the log-mechanism.
Logfilter=15	Creation of default log entries.
ReadCyclesAfterWrite=2	Herewith after each write request of the client, after having written a value, for this value a callback will be caused. This is done in order to by all means pass the current value once more after "2" update cycles. This is needed for quickly changing values.
UseColonAsPlcDivider=0	Here a value can be specified. If it is not zero, the first "." (e.g. the one after the PLC-name) is superseded by a colon.
updaterate=200	To determine the configurable update rate.
[PLC:PLC1]	Settings for PLC1.
interfacetype=GATEWAY3	Herewith via the GATEWAY a connection with the PLC will be established.
active=1	Active.
timeout=2000	Timeout.
tries=3	Number of trials.
Project=OPC.Device. Application.xml	Name and path of the generated symbol file. In case of simulation this file is concerned. If there is no path this file has to be in the OPC server directory.
waittime=10	Waiting time.
reconnecttime=5	Reconnect time.
gateway=Tcp/Ip gatewayaddress=localhost gatewayport=1217	Settings for the connection to PLC_A (Connection between client and local Gateway via TCP/IP)
device=TCP/IP parameters=1 parameter0=Address value0=050C	Parameter for the connection to the target device, where PLC1 is running (dialog <i>Communication Parameters</i> in the tool OPCConfig). The value to be entered under "value0" is the CODESYS node address and must be identical to the node address in the programming system.

6.6.4 Configuration of 'Redundant Group'

If you want to join controllers to redundant groups, do this comfortable up to two controllers with the configuration tool *OPCConfig*.

If you want to extend the redundant group, you can do this with corresponding entries in OPCServer.ini for up to five controllers by editing the INI file manually.

At runtime, all controls operate in parallel and all of them are capable to send the values of the symbols to the client. In fact, they are read from the client, where the master variable is TRUE.

If this is not clearly possible, it is read out from the default-PLC. If the controller, which is read from, fails, then the **next best** controller is used. This happens, when the quality of the symbols gets BAD. This is as long possible as controls are available with symbols in quality GOOD. With an increasing life counter it is controllable, whether the OPC server still operates.

All controls have to be configured as described, by defining the number and the names of the controls in the section [server] and by defining the associated communication in the section [PLC:<PLCName>]. For redundant groups the section [Redundancy] is required with defining the number of the desired groups. Then for each group a section [RedGroup:x] is required. Here the group name, the number of controls, the default-plc, all control names, the master variables and the name of the live counter variable have to be defined. Up to five controls can be joined to a redundant group.

Example for configuration of a redundant group in a INI file

Setting	Description
[Server]	
PLCs=5	
PLC0=ControlWinV3	
PLC1=controlWinV3R0	
PLC2=controlWinV3R1	
PLC3=controlWinV3R2	
PLC4=controlWinV3R3	
[PLC:ControlWinV3]	In this section the communication is defined.
interfacetype=GATEWAY3	
active=1	
logfilter=15	
logevents=1	
timeout=10000	
tries=3	
waittime=15	
reconnecttime=60	
gateway=Tcp/Ip	
gatewayaddress=localhost	
gatewayport=1217	
parameters=1	
parameter0=Address	
value0=ABCD	Here you have to specify the node address of the control as it is defined (in the Gateway dialog) in CODESYS V3.

[PLC:ControlWinV3R0]	In this section the communication is defined.
interfacetype=GATEWAY3	
active=1	
logfilter=15	
logevents=1	
timeout=10000	
tries=3	
waittime=15	
reconnecttime=60	
gateway=Tcp/Ip	
gatewayaddress=localhost	
gatewayport=1217	
parameters=1	
parameter0=Address	
value0=ABCE	Here you have to specify the node address of the control as it is defined (in the Gateway dialog) in CODESYS V3.
...	Here you have to edit the sections for the further controls.
[Redundancy]	All controls of a redundant group have to have the same symbol stock!
NumOfGroups=1	Number of redundant groups.
[RedGroup:0]	
Name=RedundantGroup	Name of the redundant group.
NumPLCs=5	Number of controls in the redundant group.
DefaultPLC=0	Default-plc is PLCName0
PLCName0=ControlWinV3	Name of PLCName0
PLCName1=ControlWinV3R0	Name of PLCName1
PLCName2=ControlWinV3R1	Name of PLCName2
PLCName3=ControlWinV3R2	Name of PLCName3
PLCName4=ControlWinV3R3	Name of PLCName4
VarMaster0=Application. PLC_PRG.bMaster	Mastervariable of control PLCName0
VarMaster1=Application. PLC_PRG.bMaster0	Mastervariable of control PLCName1
VarMaster2=Application. PLC_PRG.bMaster1	Mastervariable of control PLCName2

1

2

3

4

5

6

VarMaster3=Application. PLC_PRG.bMaster2	Mastervariable of control PLCName3
VarMaster4=Application. PLC_PRG.bMaster3	Mastervariable of control PLCName4
LifeCounter=Application. GVL.OPCalive	Life counter variable of the OPC server.

6.6.5 Information on the changed logfilter

The values of the INI file setting '*logfilter*=' created with a previous version of the OPC server than 3.4.0.0 have to be corrected for all subsequent versions.

Valid values before V3.4.0.0	Valid values from V3.4.0.0	Description
10 (16#0000000A)	15 (16#0000000F)	Creation of default log entries.
-47 (16#ffffffd1)	47 (16#0000002F)	Creation or default log entries and additional entries.
-1 (16#fffffff)	63 (16#0000003F)	Creation or default log entries, additional entries and debug entries.

6.6.6 Configuration of simulation

If there is not a control with a running application, the resulting lack of connection from PLC to OPC server can be replaced by a static access to a file that contains all the symbol information.

See chap. 5.5.3, "*Simulation Interface*".

If it is a Gateway V3 connection, which is to be replaced, then under section [*<PLCName>*] the interface SIMULATION3 has to be set and in project the directory and the name of the symbol file has to be set. The directory need not to be set, if the file is stored in the same directory as *OPCServer.ini*.

Setting	Description
[PLC:SimulatedControlWinV3]	
interfacetype=SIMULATION3	
project=OneSymbol.Device.Application.xml	Name of the symbol file (with extension XML) and eventually the directory name.

These settings can be done with the tool *OPCConfig.exe*. See chap. 5.5.3.1 "SIMULATION3".

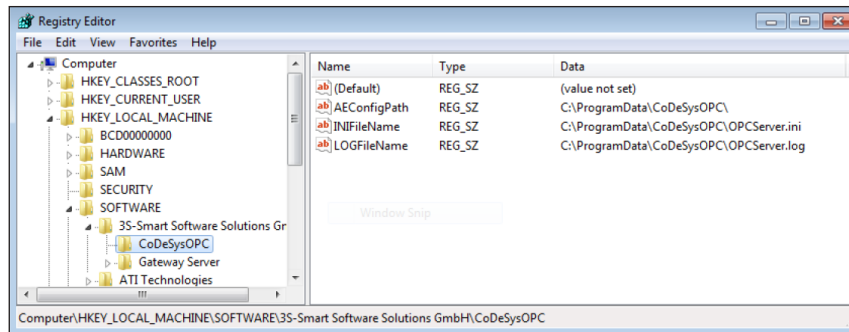
6.6.7 Changing directory and name of INI and LOG file

If there is a desire of special directories and names for the INI file and the LOG file, then the associated entries in the windows registry have to be changed.

The database can be accessed with *regedit.exe*. The entries are found under `[HKEY_LOCAL_MACHINE\SOFTWARE\3S-Smart Software Solutions GmbH\CODESYSOPC]`.

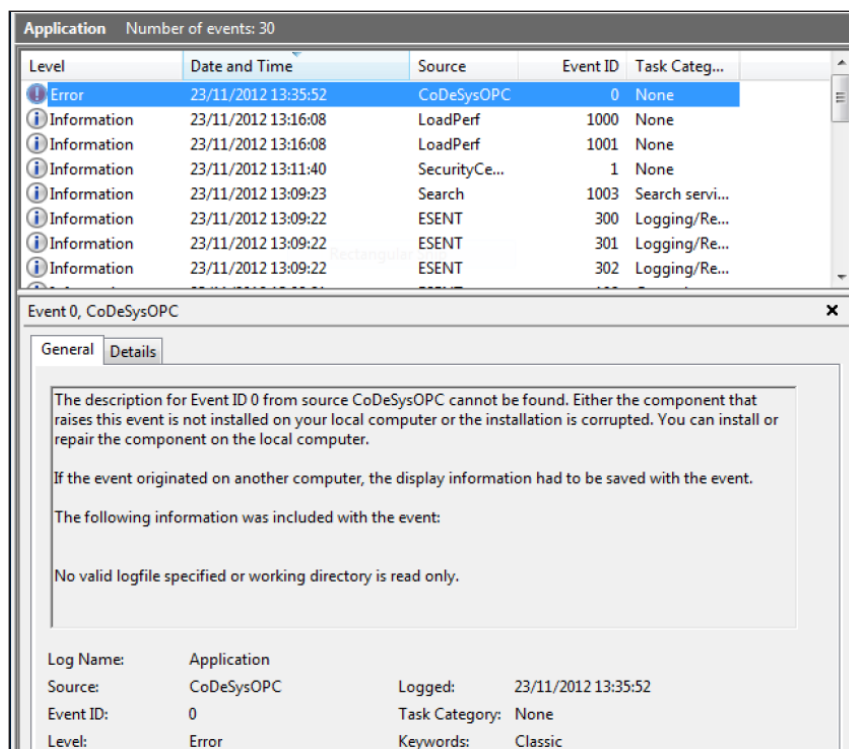
Under the name *“INIFileName”* there are stored the directory and name of the configuration file of the OPC server and under the name *„LOGFileName”* the directory and name of the LOG file is stored.

With the command *New* in the context menu of the *“Registry Editor”* the values can be changed.



In case the entry *LOGFileName* is invalid, an entry to the Windows Event logger is created.

Example of an error log in the Windows Event Viewer:



6.7 Connection to OPC servers on other computers with DCOMCNFG.EXE

This setting is only to be done, if the OPC client cannot configure a remote connection to an OPC Server! The following description relates on a Windows operating system in German!

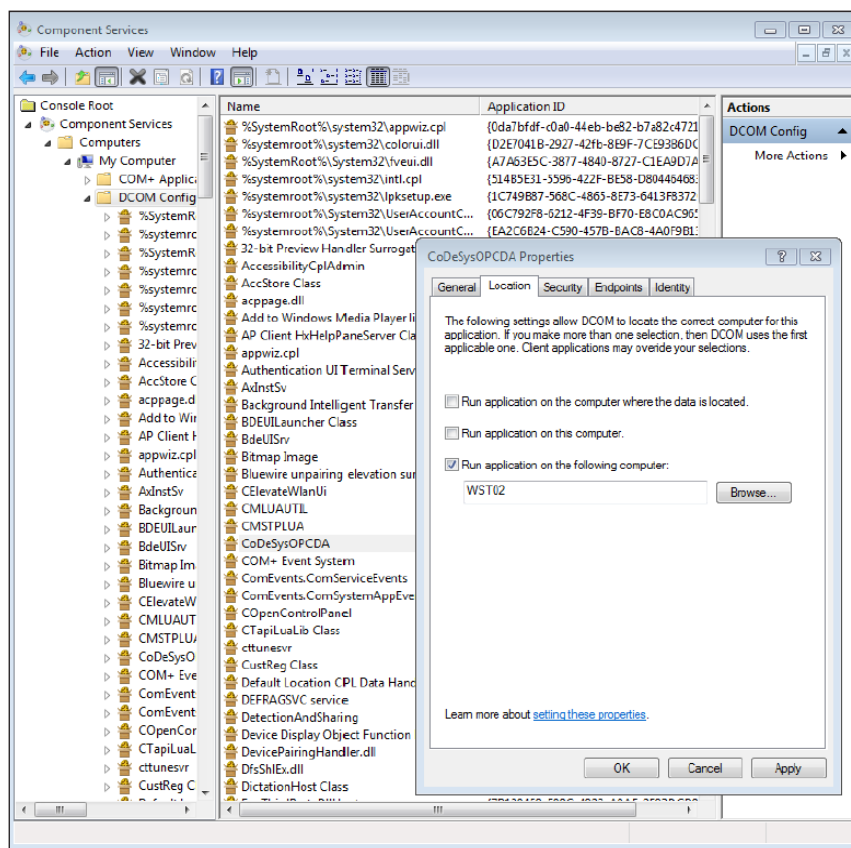
With *DCOMCNFG.EXE*, which is founded in the system directory, you can configure an OPC server, which operates on a remote PC with connection.

With double-click on *DCOMCNFG.EXE* the dialog Component Services opens.

Under *Console Root->Component Services ->Computers->My Computer->DCOM-Configuration* there is a list of services. With selection of *CODESYSOPCDA* you can command *Properties* by the context menu and the dialog *CODESYSOPCDA Properties* opens.

Select tab *Location* and there select the option *Run application on the following computer*.

Here you can insert the desired remote PC.



In order to make this connection work, there must be an OPC server running also on the local computer.

Windows 7 64Bit

If the operating system Windows 7 64Bit is used, then the above procedure is not possible, because *DCOMCNFG* has not registered the component service *CODESYSOPCDA*.

By calling '*MMC comexp.msc / 32*' you can register the service and thus a remote connection to an OPC server can be established.

6.8 Error Diagnosis

6.8.1 Means of error diagnosis

1. OPC Server LOG file.
2. Second (test-)client for reproducing the error.
3. V2.3 Interface:
Gateway Inspector with trace and logging functions (for sophisticated communication errors).

6.8.2 Brief checklist

Basically check the following points:

1. PLC running?
Gateway installed and running?
Symbol in task bar in the right hand corner active?
2. Installation and Registration of the OPC Server OK?
Takes place automatically by executing:
WinCODESYSOPC /Install (installation + registration) resp.
WinCODESYSOPC /RegServer (only registration).
3. Preparation of the project resp. settings in the CODESYS programming system OK?
V3 Interface:
 - Symbols created?
 - Project saved? Built and download done?V2.3 Interface:
 - Project options: Dump symbol entries activated?
 - Communication parameters: selected Gateway channel OK?
 - Project saved, built, download done?
4. Check the connection and the server settings: *OPCConfig.exe*:
 - Connection: Entry must correspond with the valid Gateway (set up in the CODESYS communication parameters) If no communication setting is defined the *OPC-Server* ends with *STATE_TERMINATE*.
 - Usual server settings: Motorola byte order: no (Attention when using 68K, 8051, PowerPC, only concerns V2.3), set up public groups, synchronous initialization, notification on quality change, update rate 200ms, waiting time target system 60s, reconnect interval 5s, communication timeout 3s, buffer size: 0 or fitting for target system (only concerns V2.3).
 - Save -> New Configuration becomes effective when new connection to the OPC Server is established.
5. Check, whether there are symbol entries disposable: *DiagnosticOPCClient.exe* (this diagnostic client is not part of the installation but can be delivered on demand).
 - Establish connection (*Connect OPC Server*).
 - Click on *Private Groups* in right hand side window with right mouse button.
 - Confirm dialog with *OK*.
 - Click on the group entry which appears now with right mouse button *Add all items* all generated symbol entries (variables) should appear

6.8.3 Error cases

Error case: No connection to the PLC (Status NO_CONFIG)

To be checked:

- Is the PLC running?
- Are the communication parameter in the OPC Server OK?
- Is the communication protocol for the PLC valid?
- Is the PLC available on the defined communication channel?
(Might be tested with **ping** via TCP/IP.)
- Is the cable to the PLC OK?
- If a remote gateway is used, is it started?

Error case: OPC server is connected, but no/incorrect PLCs are listed

If a connection to the OPC server is established but the PLCs are not listed in the way they defined in the initialization file of the *OPCserver*, check if the *INI-file* is in the correct location and has the correct name as it is defined in the registry ("*OPCServer.ini*").

Error case: No symbols available (Status NO_CONFIG)

To be checked:

V3.x Interface:

- Have symbols been exported?
Check this in file `<project name>.<device name>.<application name>.xml`

V2.3 Interface:

- Is option Dump symbol entries activated in the CODESYS programming system in *Project->Options*, and are variables selected for this purpose?
(Possibly check the SYM file.)
- Is a symbol file (*.sdb) available in the gateway files directory or on the PLC?
- Corresponds the *Project-ID* of the project in the programming system with that specified in the symbol file? (Possibly before the restart of the PLC a further download has been done without creating also a new boot project).

Error case: Items Quality "BAD"

Possible reasons:

- Communication error occurs with PLC while OPC Server is running.
- Communication error occurs during downloading a new program to the PLC.
- Group or item is inactive.



- The OPC Server will automatically terminated as soon as all clients have closed their connection to the server.
- The OPC Server is displayed as process in Windows Task Manager.

Page deliberately empty

Published by NUM Spa

Via F. Somma 62
20012 Cuggiono (MI) (Italy)
Tel. : +39 02 979 691
website: www.num.com

Printed December 2024