SIEMENS

SINUMERIK

SINUMERIK ONE
SINUMERIK Access MyMachine / OPC UA

Configuration Manual

Introduction 1

Safety notes 2

Getting started 3

Setting up of OPC UA server 4

Customer Specific Object Model (CSOM) 5

User administration 6

Functionality 7

OPC UA Advanced 8

Diagnosis 9

Virtual SINUMERIK ONE 10

Technical data 11

Trouble shooting 12

Additional information for OPC UA Advanced A

Valid for:

OPC UA server Version 4.2

07/2024, OPC UA V4.2
A5E51429408B AE

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ **DANGER**
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ **WARNING**
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ **CAUTION**
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ **WARNING**
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens Aktiengesellschaft. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens Aktiengesellschaft
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E51429408B AE
Ⓟ 07/2024 Subject to change

Copyright © Siemens 2019 - 2024.
All rights reserved

Table of contents

1 Introduction 9

1.1 About SINUMERIK 9

1.2 About this documentation 9

1.3 Documentation on the internet 11
1.3.1 Documentation overview SINUMERIK ONE 11
1.3.2 Documentation overview SINUMERIK operator components 11

1.4 Feedback on the technical documentation 12

1.5 mySupport documentation 12

1.6 Service and Support 13

1.7 OpenSSL 14

1.8 Compliance with the General Data Protection Regulation 15

2 Safety notes 17

2.1 Fundamental safety instructions 17
2.1.1 General safety instructions 17
2.1.2 Warranty and liability for application examples 17
2.1.3 Cybersecurity information 17

2.2 OPC UA security notes 19

3 Getting started 21

3.1 General description 21
3.1.1 SINUMERIK OPC UA server 21

3.2 Features 21

3.3 System setup 23

3.4 Compatibility of OPC UA server version and CNC software versions 23

3.5 Supported OPC UA profiles and Conformance units 24

3.6 Reference to OPC UA specification 27

4 Setting up of OPC UA server 29

4.1 Prerequisites 29

4.2 Option OPC UA 30

4.3 Commissioning 30

4.4 Certificate handling 37
4.4.1 Overview 37
4.4.2 Server certificates 39
4.4.3 Client certificates 41
4.4.3.1 Trusted certificates 41
4.4.3.2 Rejected certificates 43

4.5 Testing the connection ... 44
4.6 General OPC UA settings ... 50
5 Customer Specific Object Model (CSOM) ... 53
5.1 Overview ... 53
5.2 Functionalities ... 53
5.3 Workflow for using CSOM in the SINUMERIK OPC UA server ... 54
5.3.1 Overview ... 54
5.3.2 Exporting SINUMERIK model from OPC UA server as OPC UA XML ... 54
5.3.3 Creating a CSOM with SiOME ... 57
5.3.3.1 Overview ... 57
5.3.3.2 Importing SINUMERIK model (XML) ... 58
5.3.3.3 Modeling own object model ... 60
5.3.3.4 Option management in SiOME ... 75
5.3.3.5 Exporting CSOM (XML) ... 80
5.3.3.6 Mapping data types ... 82
5.3.3.7 Mapping of arrays ... 86
5.3.3.8 Supported data types for sinumerik namespace ... 87
5.3.3.9 Modeling rules ... 88
5.3.3.10 Access control with CSOM ... 88
5.3.4 Converting the CSOM from XML to binary ... 89
5.3.5 Importing the CSOM into the SINUMERIK OPC UA server ... 91
5.4 CSOM dialog in SINUMERIK Operate ... 93
5.4.1 Overview ... 93
5.4.2 OPC UA model dialog ... 94
5.4.3 Adding model(s) ... 94
5.4.4 Deleting OPC UA model ... 96
5.4.5 Activating / Deactivating OPC UA model(s) and SINUMERIK namespace(s) ... 98
6 User administration ... 101
6.1 Overview ... 101
6.2 User management ... 102
6.3 Access rights management ... 102
6.4 Allow anonymous access ... 103
6.5 List of access rights ... 106
6.6 Changing access rights for OPC UA configuration screens in SINUMERIK Operate ... 108
6.7 OPC UA configuration folders - service mode access ... 109
7 Functionality ... 113
7.1 Overview ... 113
7.2 Address space model ... 113
7.3 Variable access ... 116
7.3.1 Variable paths for NC access operations ... 116
7.3.2 Variable paths for GUD access operations ... 117
7.3.3 Variable paths for PLC access operations ... 118
7.3.4 Restrict DB or variable access with TIA portal ... 122

7.3.5 Variable paths for machine and setting data ... 123
7.3.6 Finding of OPC UA variables ... 124
7.3.7 Monitored items ... 127

7.4 Alarms ... 128
7.4.1 Overview ... 128
7.4.2 Subscribe / unsubscribe to alarms ... 130
7.4.3 Acknowledgement of alarms ... 132
7.4.4 Sequence description of alarms ... 132
7.4.5 SINUMERIK Alarm object ... 132
7.4.5.1 Description ... 132
7.4.5.2 OPC UA event messages and alarms ... 133
7.4.6 Language of alarms ... 138
7.4.6.1 OPC UA language specification ... 138
7.4.6.2 SINUMERIK language specification ... 138
7.4.6.3 Mapping of SINUMERIK LanguageID with OPC UA LocaleID ... 138
7.4.7 OPC UA alarms and conditions constraints ... 139
7.4.8 OPC UA alarms and conditions client ... 140
7.4.9 OPC UA multi-language alarms and conditions client ... 141

7.5 File system ... 142
7.5.1 Overview ... 142
7.5.2 Prerequisites ... 144
7.5.3 Standard file system support ... 146
7.5.3.1 File transfer with standard methods ... 146
7.5.3.2 File transfer exceeding 16 MB between client and server ... 148
7.5.3.3 Comfort methods for file transfer < 16 MB ... 150

7.6 Select ... 152
7.6.1 Overview ... 152
7.6.2 Description ... 153
7.6.3 Input and output arguments ... 154
7.6.4 Example call ... 156

7.7 Unselect ... 157

7.8 Tool management ... 158
7.8.1 Description ... 158
7.8.2 CreateTool ... 159
7.8.3 DeleteTool ... 160
7.8.4 CreateCuttingEdge ... 162
7.8.5 DeleteCuttingEdge ... 163

8 OPC UA Advanced ... 167

8.1 Introduction ... 167

8.2 Overview ... 167

8.3 Prerequisites ... 168

8.4 General interaction mechanism ... 169

8.5 OPC UA Advanced node ... 170

8.6 OPC UA Advanced use cases ... 170
8.6.1 Overview ... 170
8.6.2 Machine status ... 172

8.6.2.1 Machine status use case ... 172
8.6.3 Tool management ... 172
8.6.3.1 Tool management use case ... 172
8.6.3.2 Tool load use case with Tool Ident Connection ... 174
8.6.3.3 Tool unload use case with Tool Ident Connection ... 175
8.6.3.4 Tool create use case ... 176
8.6.3.5 Tool delete use case ... 178
8.6.4 Production dialog ... 179
8.6.4.1 Production dialog use case ... 179
8.7 OPC UA Advanced logging ... 180
8.8 OPC UA Advanced settings ... 182
8.9 OPC UA Advanced host settings ... 183
9 Diagnosis ... 185
9.1 Overview ... 185
9.2 Status screen ... 186
9.3 Diagnosis screen ... 187
9.4 OPC UA Archiving ... 194
9.5 OPC UA server version ... 203
10 Virtual SINUMERIK ONE ... 205
10.1 Installing OPC UA on Create MyVirtual Machine ... 205
10.2 OPC UA server and virtual SINUMERIK ONE ... 207
11 Technical data ... 209
11.1 Technical data ... 209
12 Trouble shooting ... 211
12.1 Frequently asked questions (FAQs) ... 211
12.2 Reference to OPC UA error code ... 215
A Additional information for OPC UA Advanced ... 217
A.1 PLC interface information ... 217
A.1.1 Symbolic DB ... 218
A.1.2 Request from PLC (byte 0) ... 221
A.1.3 Change trigger (byte 1) ... 222
A.1.4 Machine mode (byte 3) ... 222
A.1.5 RPC SINUMERIK mode (byte 4) ... 223
A.1.6 Magazine number, location number, T number (word 6 - 7, 8 - 9, 10 - 11) ... 224
A.1.7 Number of docking positions (byte 12) ... 224
A.1.8 Machine status (byte 14) ... 224
A.1.9 NC operating mode (byte 15) ... 225
A.1.10 Reserve 1 (word 16 - 17) and reserve 2 (word 18 -19) ... 225
A.1.11 Docking position information of the machine ... 225
A.1.12 Workpiece carrier (byte 22 - 27) ... 228
A.1.13 Machining side (byte 28 - 29) ... 228
A.1.14 Follow-up machining (byte 30) ... 228
A.1.15 Reserve (byte 31) ... 228

A.1.16 HostClientConnected (byte 260) ... 229
A.1.17 PLCReportType (byte 261) ... 229
A.1.18 ToolDataForToolCreate ... 230
A.1.18.1 ToolCreateRequest ... 230
A.1.18.2 DuploNumber ... 230
A.1.18.3 ToolIdentifier ... 230
A.1.18.4 ToolArea ... 230
A.1.18.5 ForceToolNumber ... 230
A.1.18.6 ForceDuploNumber ... 230
A.1.18.7 ToolCreated ... 231
A.1.18.8 ToolNumberCreated ... 231
A.1.18.9 DuploNumberCreated ... 231
A.1.18.10 ToolCreateBusy ... 231
A.1.18.11 ErrorInfo ... 231
A.1.19 ToolDataForToolDelete ... 232
A.1.19.1 ToolDeleteRequest ... 232
A.1.19.2 ToolNumber ... 232
A.1.19.3 DuploNumber ... 232
A.1.19.4 ToolIdentifier ... 232
A.1.19.5 ToolArea ... 232
A.1.19.6 ToolDeleted ... 232
A.1.19.7 ToolDeleteBusy ... 232
A.1.19.8 ErrorInfo ... 232

A.2 Detailed description of OPC UA events, methods and variables ... 233
A.2.1 OPC UA Advanced events ... 233
A.2.1.1 Overview ... 233
A.2.1.2 Subscribing to OPC UA Advanced events ... 234
A.2.1.3 Attributes of BaseEventType ... 234
A.2.1.4 Machine ... 236
A.2.1.5 ToolManagement ... 242
A.2.1.6 WorkpieceCarrier ... 255
A.2.1.7 Error numbers ... 262
A.2.1.8 Example for tool data string ... 265
A.2.2 OPC UA Advanced methods and variables ... 267
A.2.2.1 Overview ... 267
A.2.2.2 Machine ... 267
A.2.2.3 RegisterAdvHost ... 271
A.2.2.4 ToolManagement ... 272
A.2.2.5 Workpiece Carrier ... 281

Index ... 287

Introduction 1

1.1 About SINUMERIK

From simple, standardized CNC machines to premium modular machine designs – the SINUMERIK CNCs offer the right solution for all machine concepts. Whether for individual parts or mass production, simple or complex workpieces – SINUMERIK is the highly dynamic automation solution, integrated for all areas of production. From prototype construction and tool design to mold making, all the way to large-scale series production.

Visit our website for more information SINUMERIK (https://www.siemens.com/sinumerik).

1.2 About this documentation

Content

This configuration manual describes all the information you need to use and configure the SINUMERIK OPC UA server professionally. The SINUMERIK OPC UA server offers the possibility to communicate with SINUMERIK via OPC UA.

These include the following functionalities:

- Customer Specific Object Models (CSOM)
- Reading, writing and subscribing to SINUMERIK variables
- Event-based provision of SINUMERIK alarms and messages from HMI, NC and PLC.
- OPC UA Advanced for machine status, tool management and production dialog.

Validity of the manual

Note

Hardware compatibility

Please note that this manual only describes the communication via the SINUMERIK OPC UA server with a **SINUMERIK 828D control with a PU.5**.

If you have a SINUMERIK 828D control with another/older PPU or PU type, please refer to the Configuration Manual SINUMERIK 840Dsl/828D SINUMERIK Access MyMachine /OPC UA (https://support.industry.siemens.com/cs/ww/en/view/109807257) on the internet.

Target group

This document addresses commissioning engineers, machine tool manufacturers, planners and plant operating companies. The document provides detailed information that commissioning engineers require to setup the SINUMERIK Access MyMachine / OPC UA software.

Standard scope

This documentation only describes the functionality of the standard version. This may differ from the scope of the functionality of the system that is actually supplied. Please refer to the ordering documentation only for the functionality of the supplied drive system.

It may be possible to execute other functions in the system which are not described in this documentation. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

For reasons of clarity, this documentation cannot include all of the detailed information on all product types. Further, this documentation cannot take into consideration every conceivable type of installation, operation and service/maintenance.

The machine manufacturer must document any additions or modifications they make to the product themselves.

Websites of third-party companies

This document may contain hyperlinks to third-party websites. Siemens is not responsible for and shall not be liable for these websites and their content. Siemens has no control over the information which appears on these websites and is not responsible for the content and information provided there. The user bears the risk for their use.

1.3 Documentation on the internet

1.3.1 Documentation overview SINUMERIK ONE

Comprehensive documentation about the functions provided in SINUMERIK ONE Version 6.13 and higher is provided in the Documentation overview SINUMERIK ONE (https://support.industry.siemens.com/cs/ww/en/view/109768483).



You can display documents or download them in PDF and HTML5 format.

The documentation is divided into the following categories:

- User: Operating
- User: Programming
- Manufacturer/Service: Functions
- Manufacturer/Service: Hardware
- Manufacturer/Service: Configuration/Setup
- Manufacturer/Service: Safety Integrated
- Information and training
- Manufacturer/Service: SINAMICS

1.3.2 Documentation overview SINUMERIK operator components

Comprehensive documentation about the SINUMERIK operator components is provided in the Documentation overview SINUMERIK operator components (https://support.industry.siemens.com/cs/document/109783841/technische-dokumentation-zu-sinumerik-bedienkomponenten?dti=0&lc=en-WW).

You can display documents or download them in PDF and HTML5 format.

The documentation is divided into the following categories:

- Operator Panels
- Machine control panels
- Machine Pushbutton Panel
- Handheld Unit / Mini handheld devices
- Further operator components

An overview of the most important documents, entries and links to SINUMERIK is provided at SINUMERIK Overview - Topic Page (https://support.industry.siemens.com/cs/document/109766201/sinumerik-an-overview-of-the-most-important-documents-and-links?dti=0&lc=en-WW).

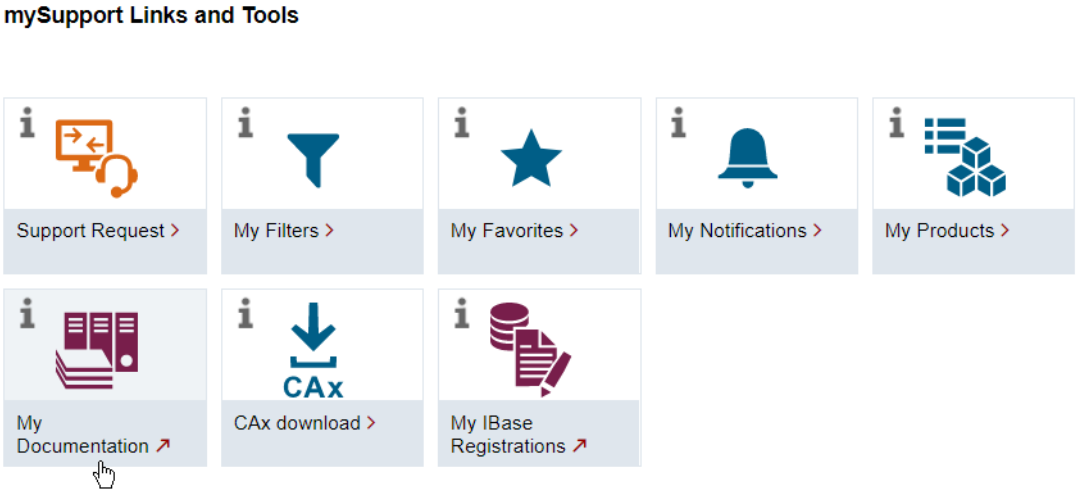
1.4 Feedback on the technical documentation

If you have any questions, suggestions or corrections regarding the technical documentation which is published in the Siemens Industry Online Support, use the link "Provide feedback" which appears at the end of the entry.

1.5 mySupport documentation

With the "mySupport documentation" web-based system you can compile your own individual documentation based on Siemens content, and adapt it for your own machine documentation.

To start the application, click on the "My Documentation" tile on the mySupport homepage (https://support.industry.siemens.com/cs/ww/en/my):



The configured manual can be exported in RTF, PDF or XML format.

Note

Siemens content that supports the mySupport documentation application can be identified by the presence of the "Configure" link.

1.6 Service and Support

Product support

You can find more information about products on the internet:

Product support (https://support.industry.siemens.com/cs/ww/en/)

The following is provided at this address:

- Up-to-date product information (product announcements)
- FAQs (frequently asked questions)
- Manuals
- Downloads
- Newsletters with the latest information about your products
- Global forum for information and best practice sharing between users and specialists
- Local contact persons via our Contacts at Siemens database (→ "Contact")
- Information about field services, repairs, spare parts, and much more (→ "Field Service")

Technical support

Country-specific telephone numbers for technical support are provided on the internet at address (https://support.industry.siemens.com/cs/ww/en/sc/4868) in the "Contact" area.

If you have any technical questions, please use the online form in the "Support Request" area.

Training

You can find information on SITRAIN at the following address (https://www.siemens.com/sitrain).
SITRAIN offers training courses for automation and drives products, systems and solutions from Siemens.

Siemens support on the go



With the award-winning "Industry Online Support" app, you can access more than 300,000 documents for Siemens Industry products – any time and from anywhere. The app can support you in areas including:

- Resolving problems when implementing a project
- Troubleshooting when faults develop
- Expanding a system or planning a new system

Furthermore, you have access to the Technical Forum and other articles from our experts:

- FAQs
- Application examples
- Manuals
- Certificates
- Product announcements and much more

The "Industry Online Support" app is available for Apple iOS and Android.

1.7 OpenSSL

Using OpenSSL

This product can contain the following software:

- Software developed by the OpenSSL project for use in the OpenSSL toolkit
- Cryptographic software created by Eric Young.
- Software developed by Eric Young

You can find more information on the internet:

- OpenSSL (https://www.openssl.org)
- Cryptsoft (https://www.cryptsoft.com)

1.8 Compliance with the General Data Protection Regulation

Siemens observes standard data protection principles, in particular the data minimization rules (privacy by design).

For the SINUMERIK Operate, this means:

The product processes/saves the following personal data:

- FullName (optional)
- User name + password
- UserID
- IP address
- Security events
- Time stamp

It does not involve data from the personal or private sphere.

The above data is required for the user log-in function. The storage of data is appropriate and limited to what is necessary, as it is essential for the identification of the authorized operator. (Mandatory here are: user name + password; the FullName is optional.)

The above data cannot be stored anonymously or pseudonymously, as otherwise the purpose of identifying the operating personnel cannot be achieved.

Our products do not automatically delete the data mentioned above. The data and logs can be deleted manually by authorized personnel.

The above data is secured against loss of integrity and confidentiality by Industry State-of-the-Art Product Security mechanisms.

Safety notes 2

2.1 Fundamental safety instructions

2.1.1 General safety instructions

⚠ WARNING

Danger to life if the safety instructions and residual risks are not observed

If the safety instructions and residual risks in the associated hardware documentation are not observed, accidents involving severe injuries or death can occur.

- Observe the safety instructions given in the hardware documentation.
- Consider the residual risks for the risk evaluation.

⚠ WARNING

Malfunctions of the machine as a result of incorrect or changed parameter settings

As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death.

- Protect the parameterization against unauthorized access.
- Handle possible malfunctions by taking suitable measures, e.g. emergency stop or emergency off.

2.1.2 Warranty and liability for application examples

Application examples are not binding and do not claim to be complete regarding configuration, equipment, or any eventuality which may arise. Application examples do not represent customer-specific solutions, but merely serve to provide assistance with typical tasks.

As the user you yourself are responsible for ensuring that the products described are operated correctly. Application examples do not relieve you of your responsibility for safe handling when using, installing, operating and maintaining the equipment.

2.1.3 Cybersecurity information

Siemens provides products and solutions with industrial cybersecurity functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial cybersecurity concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial cybersecurity measures that may be implemented, please visit
https://www.siemens.com/cybersecurity-industry.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Cybersecurity RSS Feed under
https://new.siemens.com/cert.

Further information is provided on the Internet:

Industrial Security Configuration Manual (https://support.industry.siemens.com/cs/ww/en/view/108862708)

⚠ WARNING

Unsafe operating states resulting from software manipulation

Software manipulations, e.g. viruses, Trojans, or worms, can cause unsafe operating states in your system that may lead to death, serious injury, and property damage.

- Keep the software up to date.
- Incorporate the automation and drive components into a state-of-the-art, integrated industrial cybersecurity concept for the installation or machine.
- Make sure that you include all installed products in the integrated industrial cybersecurity concept.
- Protect files stored on exchangeable storage media from malicious software by with suitable protection measures, e.g. virus scanners.
- Carefully check all cybersecurity-related settings once commissioning has been completed.

2.2 OPC UA security notes

NOTICE

Access rights for SINUMERIK data

OPC UA provides read/write access on data in SINUMERIK. This access might also affect security relevant data.

- You can limit this access on SINUMERIK data by individual read and write permission or by using an own object model and deactivating the SINUMERIK namespace. Please refer to chapter User administration (Page 101), especially chapter "List of access rights".

Note

Communication and system performance

Please note that the OPC UA server is not a real-time enabled process that, depending on client requirements, generates a corresponding communication load in the SINUMERIK system. Increased communication load can have repercussions on system performance.

The SINUMERIK system load may vary with different part programs.

A high system load can have repercussions on communication performance.

Note

Access rights for modifying Machine Builder screen

- Access level "Manufacturer": All the softkeys are enabled in the "Machine Builder" role.
- Access level "Service": You will not be able to modify any settings under "Settings", Certificates, "Models", and "Authentication" softkeys in the "Machine Builder" role.

Getting started 3

3.1 General description

Uniform standard for data exchange

"Industrie 4.0" stands for the intensive utilization, evaluation and analysis of data from the production in IT systems of the enterprise level. PLC programs today already record a wide range of data at the production and process level (pressure values, temperatures and counter readings) and make them available to systems at the enterprise level, for example, to increase the product quality. With Industry 4.0, the data exchange between the production and enterprise levels will increase much faster in the future. However, prerequisite for the success of "Industrie 4.0" is a uniform standard for data exchange.

The **OPC UA (Unified Architecture)** standard is particularly suitable for data exchange across different levels as it is independent from specific operating systems, has secure transfer procedures and better semantic description of the data. OPC UA not only makes data available, but also provides information about the data (e.g. data types). This enables machine-interpretable access to the data.

3.1.1 SINUMERIK OPC UA server

The SINUMERIK OPC UA server offers a communication interface with manufacturer independent standard. The information on SINUMERIK controls can be exchanged with an OPC UA client using this communication interface.

The client is not part of SINUMERIK and is either part of standard software or can be developed as part of individual software. For this purpose a stack for downloading is provided by the OPC foundation.

Some manufacturers provide a software development kit, which can be used to develop an OPC UA client.

3.2 Features

The SINUMERIK OPC UA server provides the possibility to communicate with SINUMERIK via OPC UA. The following functionalities of the OPC UA specification are supported by the server:

- Read, write and subscribe to SINUMERIK variables (NC, PLC) (see chapter Variable access (Page 116))
- Transfer of part programs (see chapter File system (Page 142))
- Support for File and Folder Objects
- Event based provision of SINUMERIK alarms and messages from HMI, NC and PLC (see chapter Alarms (Page 128))

- Methods for selection of part programs from the NC file system and external memory (see chapter Select (Page 152)) and methods for tool management (see chapter Tool management (Page 158))
- Multi language support for the alarm and warning messages.
- The OPC UA server supports customer specific object models (see chapter Customer Specific Object Model (CSOM) (Page 53))
- The OPC UA server supports OPC UA Advanced functionalities, like pallet handling and tool management with CNC controls in your production system (see chapter OPC UA Advanced (Page 167)).

Security settings

The server provides the possibility to communicate in an unencrypted or encrypted way. The following options are possible:

- None
- 128 Bit - Sign (Aes128Sha256RsaOaep)
- 128 Bit - Sign & Encrypt (Aes128Sha256RsaOaep)
- 256 Bit - Sign (Aes256Sha256RsaPss)
- 256 Bit - Sign & Encrypt (Aes256Sha256RsaPss)
- 256 Bit - Sign (Basic256Sha256)
- 256 Bit - Sign & Encrypt (Basic256Sha256)

Deprecated security policies:

- 128 Bit - Sign (Basic128 (RSA15))
- 128 Bit - Sign & Encrypt (Basic128 (RSA15))
- 256 Bit - Sign (Basic256)
- 256 Bit - Sign & Encrypt (Basic256)

Note

These deprecated security policies is likely to be removed in the future hence it is not recommended to use it.

NOTICE
Security risk of no or low encryption During operational process, an encrypted communication must always be used for security reasons.

Furthermore, the SINUMERIK OPC UA server provides the possibility of user administration, which allows to assign access rights for each user individually (see chapter User administration (Page 101)).

3.3 System setup

Accessibility of the server

The accessibility of the server varies in the particular SINUMERIK systems. The following table shows the dependencies of the SINUMERIK systems:

SINUMERIK systems	Accessibility	
SINUMERIK ONE	The OPC UA server needs SINUMERIK Operate and runs on the same target as SINUMERIK Operate. For this reason, system setup depends on whether a Thin Client is used (SINUMERIK Operate runs on NCU) or an IPC with Windows operating system.	
	Thin Client	If a Thin Client is used, the OPC UA server is accessible after successful licensing and activation via X120 and X130 interface of the NCU.
	IPC	If an IPC is used, the OPC UA server is accessible after successful licensing and activation via "eth1" and "eth2" interface of the IPC. In this case the OPC UA server is neither accessible via "eth3" interface of the IPC nor the X120 or X130 interface of the NCU.

Application scenario

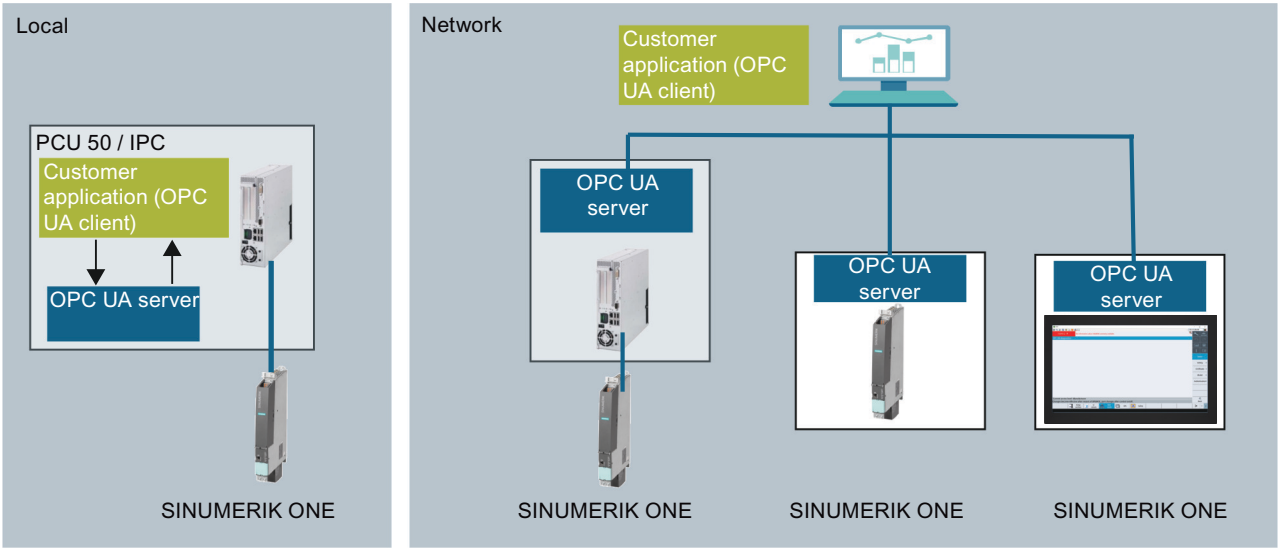


Figure 3-1 Application scenarios

3.4 Compatibility of OPC UA server version and CNC software versions

Therefore the following OPC UA server versions are only compatible with the following SINUMERIK ONE Operate versions:

OPC UA server version	SINUMERIK ONE Operate version
3.1	6.14
3.1 SP1	6.15

OPC UA server version	SINUMERIK ONE Operate version
4.0	6.20, 6.15
4.0 SP1	6.21, 6.20, 6.15
4.1	6.22, 6.21, 6.20, 6.15
4.1 SP1	6.23, 6.22, 6.21, 6.20, 6.15
4.2	6.24, 6.23, 6.22, 6.21, 6.20, 6.15

3.5 Supported OPC UA profiles and Conformance units

As a rule, an OPC UA application or an OPC UA device does not support the entire functional scope of OPC UA. An OPC UA server on an embedded system can, for example, not support any Subscriptions or track changes in the own address space.

Differences in functional scope also apply for clients.

In order to be able to manage differences and be able to make them transparent, profiles were introduced for OPC UA.

In simple terms, profiles define the functionality of an OPC UA application.

Additional details and descriptions on the profiles and facets are available under https://profiles.opcfoundation.org/v104/reporting/.

Structuring of the OPC UA functionality

The smallest functional unit is called a "Conformance Unit" at OPC UA. Conformance Units define a manageable set of functions which are used together and which are therefore are tested as a unit with a compliance test tool.

Several Conformance Units can be summarized into one facet. A facet is a named grouping of related Conformance Units to make the lists of supported Conformance Classes more manageable. Facets define specific facets of a server, e.g. the support of event subscriptions. Conformance Units and Facets together form a "Full-featured" profile. An OPC UA server must support at least one full featured profile. This profile must be implemented with all contained Conformance Units in order for it to be certified.

An OPC UA application can also support several profiles and a profile can, in turn, include other profiles.

The following figure shows the interaction of profiles, facets, and Conformance Units.

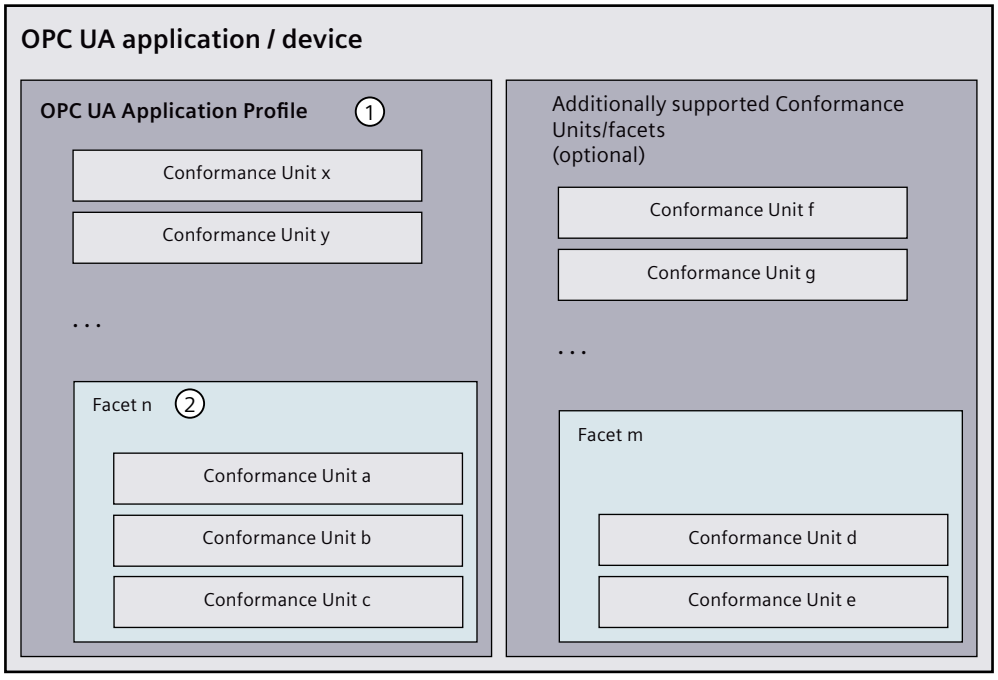


Figure 3-2 OPC UA profiles structure

Checking the profiles for runtime

During the connection establishment OPC UA clients and OPC UA servers exchange their list of supported and used profiles. This mechanism allows the applications to check whether the required features are provided by the communication partner.

Supported OPC UA profiles

The SINUMERIK OPC UA supports the "Embedded 2017 UA Server Profiles" (mandatory conformance units) as shown in the following table.

Conformance group	Conformance unit
Address Space Model	• Address Space Atomicity • Address Space Base
Attribute Services	• Attribute Read • Attribute Write Index • Attribute Write Values

Conformance group	Conformance unit
Base Information	• Base Info Core Structure • Base Info Custom Type System • Base Info Diagnostics • Base Info Engineering Units • Base Info GetMonitoredItems Method • Base Info OptionSet • Base Info Placeholder Modelling Rules • Base Info ResendData Method • Base Info ValueAsText • Base Info Server Capabilities • Base Info Type System
Discovery Services	• Discovery Get Endpoints • Discovery Find Servers Self
Monitored Item Services	• Monitor Basic • Monitor Items 2 • Monitor Items 10 • Monitor Items 100 • Monitor QueueSize_1 • Monitor MinQueueSize_02 • Monitor Value Change • Monitor Items Deadband Filter • Monitor Triggering
Security	• Security None • Security User Name Password • Security Certificate Validation • Security None CreateSession ActivateSession • Security None CreateSession ActivateSession 1.01 • Security Basic 256Sha256 • Security Aes128-Sha256-RsaOaep • Security Aes256-Sha256-RsaPss • Security Administration • Security Encryption Required • Security Signing Required • Security Default ApplicationInstanceCertificate
Session Services	• Session Minimum 1 • Session Minimum 2 Parallel • Session General Service Behaviour • Session Base

Conformance group	Conformance unit
Subscription Services	• Subscription Basic • Subscription Publish Discard Policy • Subscription Minimum 1 • Subscription Minimum 02 • Subscription Publish Min 02 • Subscription Publish Min 05
View Services	• View Basic • View Minimum Continuation Point 01 • View RegisterNodes • View TranslateBrowsePath

3.6 Reference to OPC UA specification

The SINUMERIK OPC UA server is based on OPC foundation (https://opcfoundation.org/)spec. 1.04.

Setting up of OPC UA server

4

4.1 Prerequisites

NOTICE
Protection against security risks To protect industrial plants and systems comprehensively against cyber-attacks, measures must be applied simultaneously at all levels (from the operational level up to the field level, from access control to copy protection). Therefore, before setting up of the OPC UA server, apply the "Defense in Depth" protection concept to avoid security risks in your environment. Ensure that you do not connect the company network to the internet without suitable protective measures. You will find **further information** Configuration Manual Industrial Security on the Defense-in-Depth concept, suitable protective measures and Industrial Security in general in the Configuration Manual Industrial Security (https://support.industry.siemens.com/cs/de/en/view/108862708).

Prerequisites

- OPC UA server requires SINUMERIK Operate.
- OPC UA server requires an OPC UA electronic license (6FC5800-0BP67-0Yx0).
- Make sure that the HMI time is set correctly, since this is a prerequisite for encrypted communication.

4.2 Option OPC UA

Setting the option

1. Set the "Access MyMachine / OPC UA" option via the "Startup > Licenses" operating area.

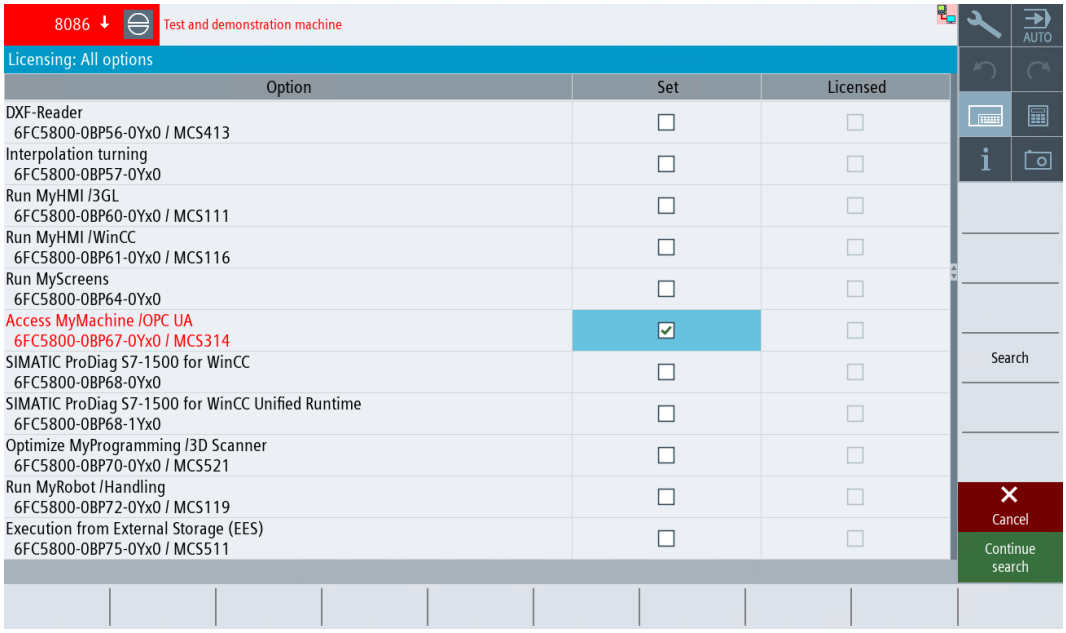


Figure 4-1 Setting the options

4.3 Commissioning

Checking the HMI time

Make sure that the HMI time is set correctly, since this is a prerequisite for encrypted communication.

Note

The certificate needed for secure OPC UA communication is automatically created during the first run-up. The start date of the validity period of the certificate is set to the current date. The validity period is 20 years.

If the SINUMERIK system time is subsequently changed, so that it lies outside the validity period, the secure OPC UA communication does not function (BadCertificateTimeInvalid).

The certificate can also be changed manually, as described in chapter Certificate handling (Page 37).

Executing the OPC UA configuration dialog

1. Start the OPC UA configuration dialog via the operating area "Startup > Network".
2. Press the "OPC UA" softkey.

3. Press the "Setting" softkey. The Settings dialog will appear. Then press the "Change" softkey. Make the necessary settings for connection and activation.

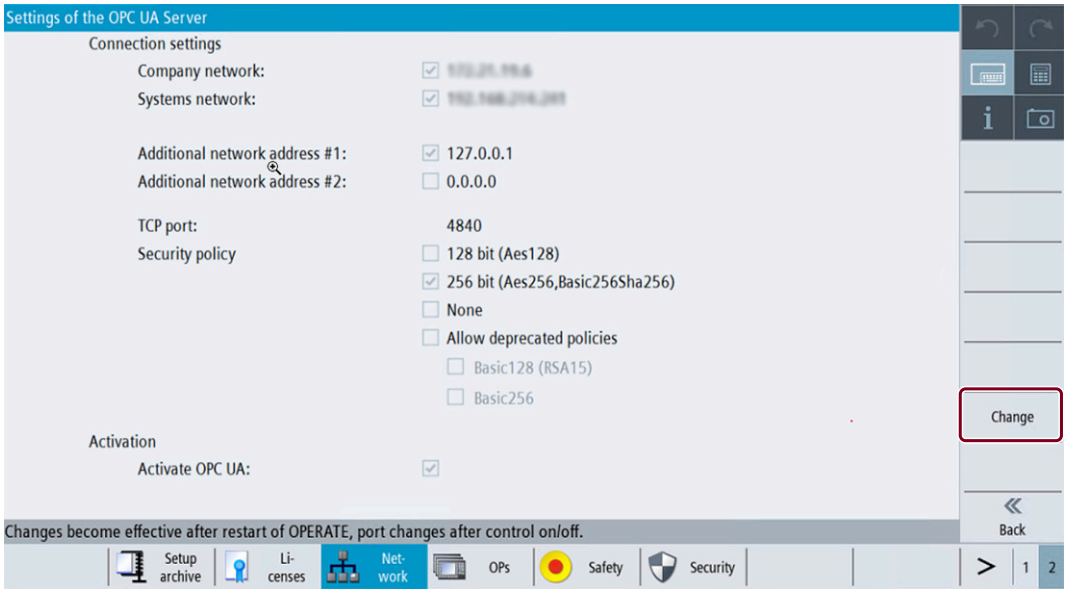


Figure 4-2 Settings of OPC UA Server (with changes)

<table>
<tr><th>Group</th><th>Setting</th><th>Description</th></tr>
<tr><td rowspan="5">Connection settings</td><td>• Company network
• Systems network (machine network)
• Service network</td><td>The available network connections (IP address) on a specific target system (SINUMERIK ONE, and IPC) are shown. The available networks options vary depending on your target system.
• Company network
• Systems network (machine network)
• Service network
For example, IPC, there will be only two networks (company and systems (machine) network) displayed.
OPC UA clients running on the same IPC as the OPC UA server can reach the server through the company or system network IP addresses. The OPC UA server can only be reached through local host address (127.0.0.1) if stated in the list of network addresses.
It is possible to activate or deactivate an interface from OPC UA server point of view.</td></tr>
<tr><td>• Additional network address #1
• Additional network address #2</td><td>All the windows devices will allow two additional IP address so that third interface of IPC can be utilized.
If the additional IP address is not valid then the OPC UA server cannot be reached.
It is possible to activate or deactivate an interface from OPC UA server point of view.</td></tr>
<tr><td>TCP port</td><td>TCP port at which the OPC UA server should be available.
Standard configuration: 4840
Note
The port must also be open in the firewall. For NCU this happens automatically. With IPC the port must be opened manually in the firewall.</td></tr>
<tr><td rowspan="5">Security policy</td><td>You can choose from the following settings, which security policy should be offered from the server:
Setting | Standard configuration
128 bit (Aes128) | Deactivated.
256 bit (Aes256, Basic256Sha256) | Activated.
None | Deactivated.
Allow deprecated policies | Deactivated.
Place the checkmark to activate the deprecated security policies and remove the checkmark to deactivate it.
Deprecated security policies:
• Basic128 (RSA15)
• Basic256</td></tr>
<tr><td>Note
• By default, the strongest security policy is selected (256 bit).</td></tr>
</table>

Group	Setting	Description
		• In the upgrade scenario, the already selected old policies (Basic128 (RSA15), Basic256, and Basic256Sha256) will be continued to be selected along with the "Allow deprecated polices". You are allowed to change the security policies as per the requirement.
Activation	Activate OPC UA	Place the checkmark to activate OPC UA and remove the checkmark to deactivate it.

NOTICE

Security risk due to data manipulation and data sniffing

Anonymous access can be a security risk. Anonymous access should therefore be strictly limited to commissioning.

- For normal operation authentication via username and password or based on certificates should be used (see chapter Certificate handling (Page 37)).

NOTICE

Security risk due to data manipulation and data sniffing

If no message encryption to the client is established, there will be a security risk of data manipulation and data sniffing. It is therefore highly recommended to establish a message encryption to the client.

- Use the highest possible encryption standard (256 bit) to ensure a secure message transfer.

Note

DNS based addressing

If you want to contact the OPC UA server via host name you have to do the following steps:

- Set the host name in SINUMERIK Operate. You can find further information on setting the host name in SINUMERIK Operate in the Commissioning manual "Basesoftware and operating software" (https://support.industry.siemens.com/cs/ww/en/view/109777907).
- Afterwards, go to the certificate dialog of the OPC UA server and renew the server certificate with "DNS only" (see chapter Server certificates (Page 39)).

Now the OPC UA server can be addressed via host name.

For:

- NCU
  - System network is DHCP/DNS server
  - Company network is DHCP/DNS client
- IPC
  - System and company network is DHCP/DNS client
  - Company network is DHCP/DNS client

For addressing the OPC UA server via company network, you must ensure that the central DNS server uses the same host name as given in SINUMERIK Operate.

4. Then press "OK". If you enter a port for the first time, you will receive a safety note.

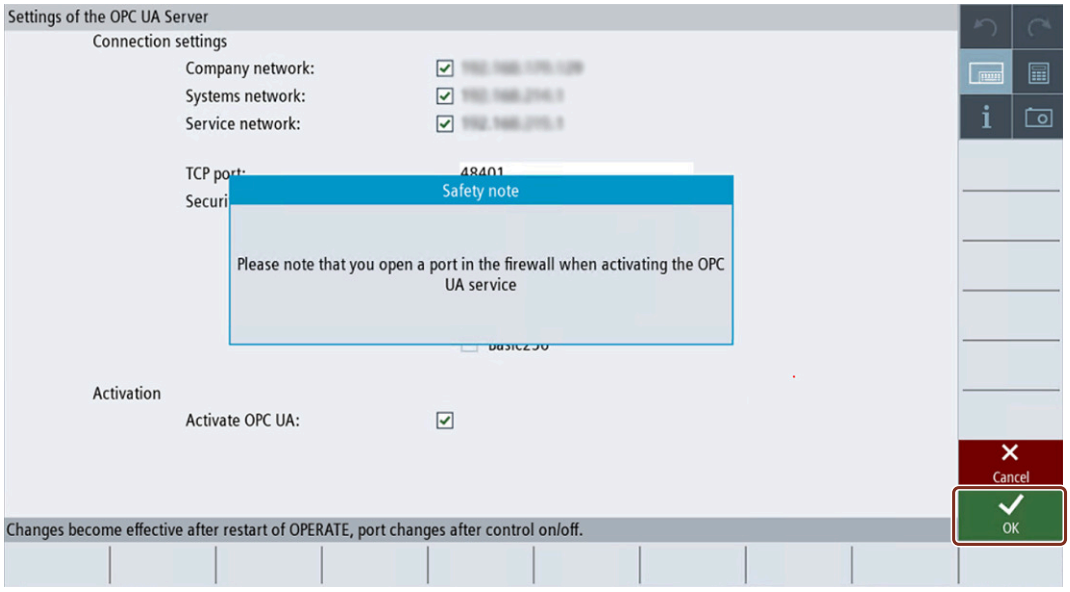


Figure 4-3 Safety note opening TCP port

Note

Port opening on IPC

On first startup of OPC UA server, a windows message appears, asking to confirm the opening of the port.

5. To confirm the opening of the port, press "OK".
6. To perform the authentication settings, press "Back", and then press the "Authentication" softkey. The Authentication dialog box appears.

7. Press the "Change" softkey. Make the necessary settings for authentication.

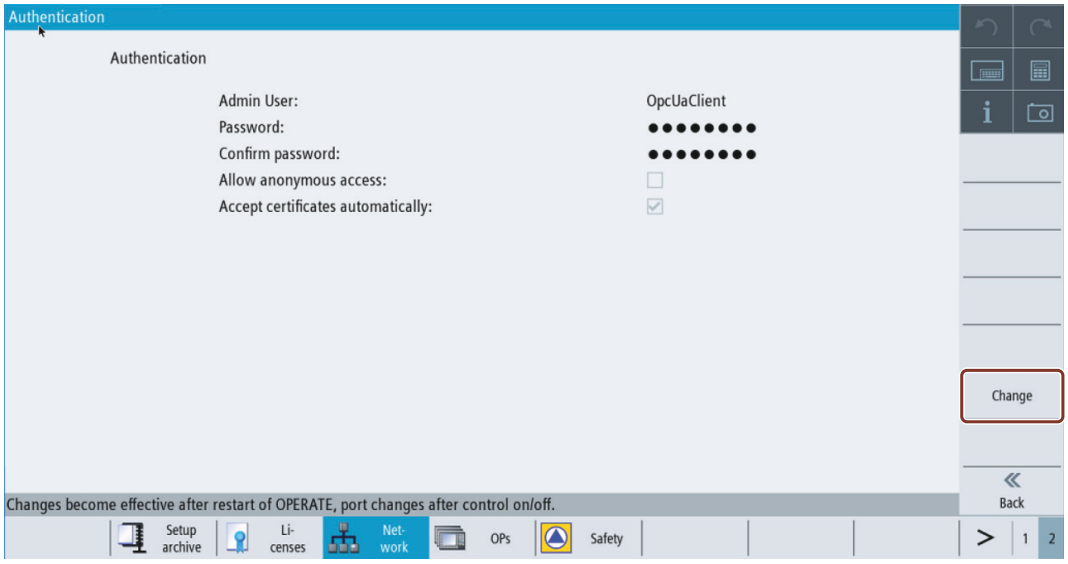


Figure 4-4 Authentication settings of OPC UA server

Group	Setting	Description
Authentication	Admin User	User name of the administrator. The administrator can add or delete users and assign or delete user authorizations.
	Password	Password of the administrator. **Note**: The password must be at least 8 characters long and should not exceed a maximum of 120 characters.
	Confirm Password	Enter the password again for confirmation.
	Allow anonymous access	Standard configuration: Deactivated Anonymous access is only recommended for commissioning.
	Accept certificates automatically	Standard configuration: Activated If this option is set, all client certificates are automatically accepted. For manual acceptance, please refer to chapter Certificate handling (Page 37).

Note

Assigning secure passwords

Observe the following rules when creating new passwords:

- When assigning new passwords, ensure that you do not assign passwords that can be guessed, e.g. simple words, key combinations that can be easily guessed, etc.
- Passwords must always contain a combination of upper-case and lower-case letters as well as numbers and special characters. Passwords must comprise at least eight characters. The server does not support passwords comprising less than eight characters. PINS must comprise an arbitrary sequence of digits.
- Wherever possible and where it is supported by the IT systems, a password must always have a character sequence as complex as possible.

The German Federal Office for IT Security (BSI) (https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/International/GSK_15_EL_EN_Draft.pdf?__blob=publicationFile&v=2) provides additional rules for creating secure passwords.

Programs are available that can help you to manage your passwords. Using these programs, you can encrypt, save and manage your passwords and secret numbers – and also create secure passwords.

Note

If you want to change the password later, you can do this via the OPC UA method "ChangeMyPassword" or in the SINUMERIK Operate screen.

8. If settings are all done, restart is necessary to activate the new settings. Perform a hardware restart on the target system NCU. A restart of the SINUMERIK Operate is necessary on the IPC.

4.4 Certificate handling

4.4.1 Overview

To establish a secure connection between an OPC UA server and a client it is necessary to exchange and trust the certificate of the other communication partner. The exchange is normally done automatically at the first connection attempt between client and server. Nevertheless, there is also the possibility to exchange the certificates manually before the other communication partner is available, e. g for preparing an easy commissioning.

For trusting the certificates there are two possibilities within the server:

- Automatic trusting of new certificates
  If "Accept certificates automatically" is activated in the commissioning dialog, new client certificates are trusted automatically and there is no manual interaction necessary to establish a secure connection.
  This is the most comfortable option, but less secure than the manual trusting, since all certificates will be trusted.
- Manual trusting of certificates (recommended)
  If "Accept certificates automatically" is deactivated in the commissioning dialog the certificates must be trusted manually to establish a secure connection.
  This allows the administrator of the OPC UA server to manually decide, which client can establish a secure connection to the OPC UA server

To have a comfortable way to handle certificates, the OPC UA dialog offers a certificate section, which can be found under the softkey "Certificates".

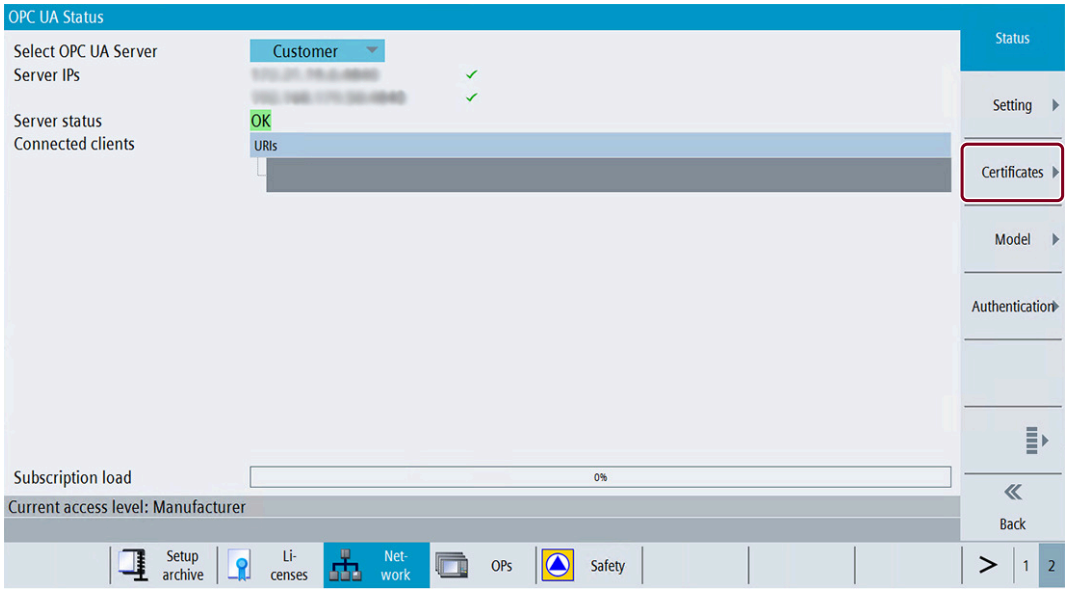


Figure 4-5 Softkey Certificates

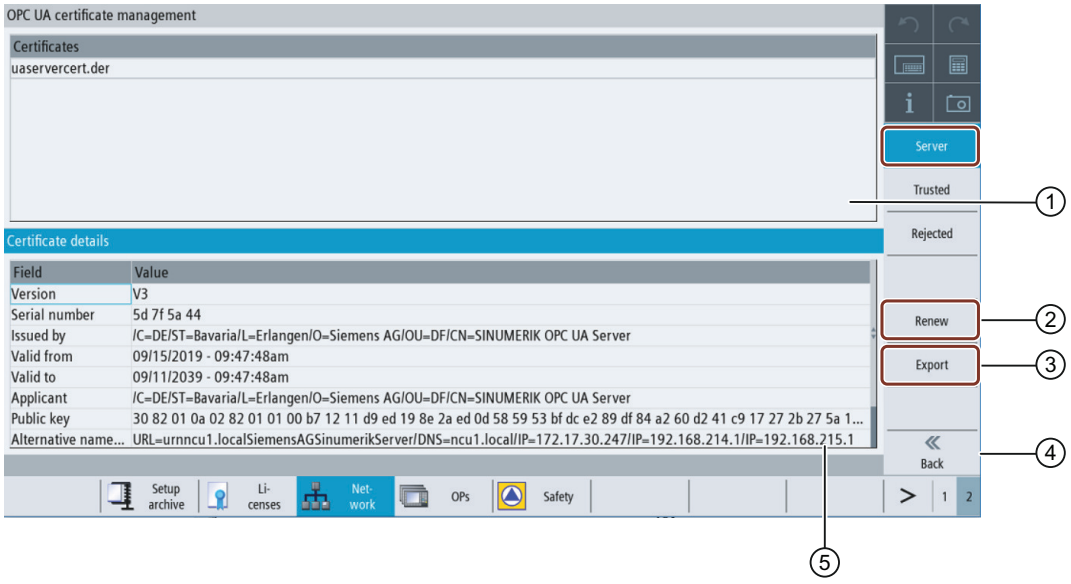
Operations

The Certificate dialog allows the following operations:

- Server certificate
  - Renewal of the server certificate
  - Export of the current server certificate
- Client certificates
  - List of the current trusted certificates
  - List of the rejected client certificates
  - Manual import of a client certificate
  - Deletion of a client certificate
  - Trust a rejected client certificate

4.4.2 Server certificates

Overview



① The name of the OPC UA server certificate is shown in the upper part of the screen.
② You can renew the server certificates.
③ You can export the server certificate to a configured device.
④ You can leave the OPC UA dialogs.
⑤ The details of the server certificate are shown in the lower part of the screen. You can scroll down to see further certificate attributes.

Figure 4-6 Server Certificate

Renewing server certificates

The server certification can be renewed at any time. With the renewal the following things can be specified by the administrator:

- Expiration date of the certificate / validity in years

  Note

  Before using this dialog make sure that the date and the time of SINUMERIK Operate is set correctly, as the certificate will be valid from the current date in SINUMERIK Operate at the time of renewal.

- Decision if IP address and/or DNS name should be mentioned in the server certificate

  Note

  Many clients will need the IP address in the certificate for validation. If the server will be addressed by DNS name (e. g. because the IP address of the OPC UA server changes frequently due to a dynamic assignment by a DHCP server), it is recommended only to include the DNS name in the certificate. Because otherwise the certificate must be renewed and exchanged with every change of the IP address.

To renew a server certificate, proceed as follows:

1. Press the softkey "Renew".
   A pop-up screen will appear that offers two ways of selecting a time period:
   - Select the number of years, the server certificate will be valid
   - Specify a precise date, the server certificate will expire

   Specify also whether the IP address and/or the DNS name should be written in the server certificate.

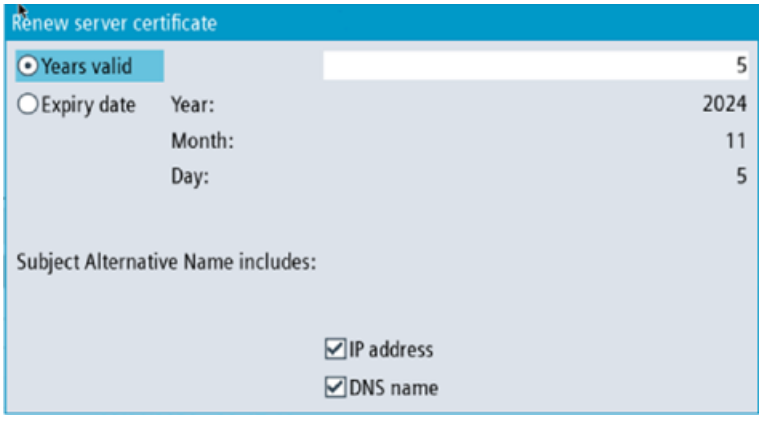


Figure 4-7 Renew server certificate

Pressing the softkey "Cancel" will ignore all input and return to the "Server" dialog.
Pressing the softkey "Ok" will save the input to the system, the currently valid certificate will be deleted and with the next start of SINUMERIK Operate the new certificate gets created.

Exporting server certificates

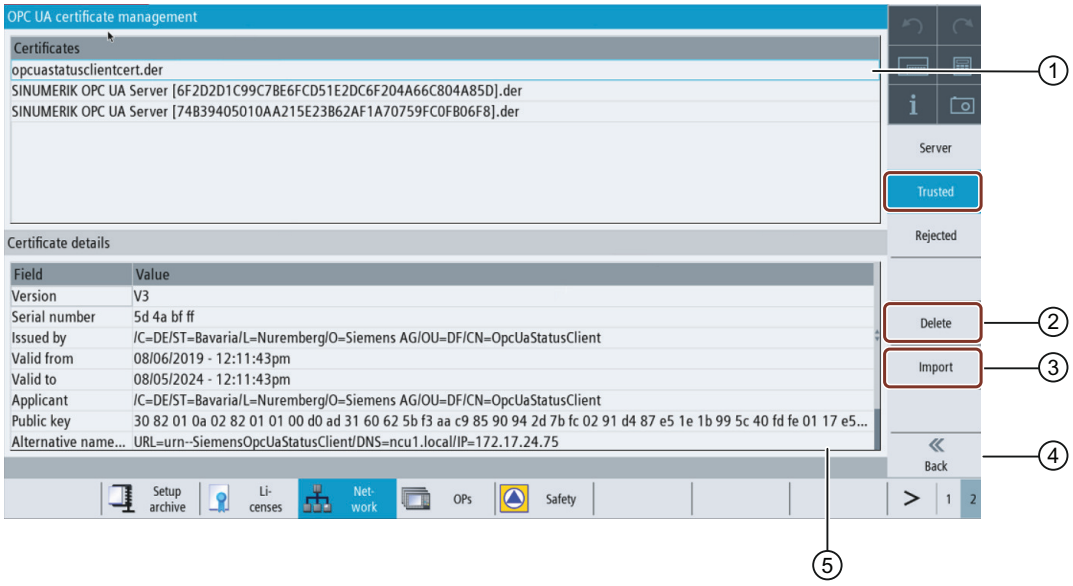
For an offline preparation of the connection to the server, you can export the server certificate. After that the certificate can be imported and trusted on the client side.

1. Press the softkey "Export".
   A pop-up screen will appear showing the SINUMERIK filesystem to export. You can navigate to a location on the SINUMERIK filesystem to export the OPC UA server certificate.
   Pressing the softkey "Cancel" will ignore all input and return to the "Server" dialog.
   Pressing the softkey "Ok" will export the certificate.

4.4.3 Client certificates

4.4.3.1 Trusted certificates

Overview



① The trusted certificates are listed in the upper part of the screen. You can select a certificate using the arrow keys (cursor up/ cursor down).

② You can delete the trusted certificates.

③ You can import a certificate from an USB device.

④ You can leave the OPC UA dialogs.

⑤ The certificate details are shown in the lower part of the screen. To set the focus on the lower part of the screen the softkey "next window" on the keyboard is used.

Figure 4-8 Trusted Certificate

Deleting trusted certificates

1. To manually delete a client certificate, select a certificate in the trusted list and press the softkey "Delete".
   A pop-up screen will appear asking you for confirmation of deletion:



Figure 4-9 Delete certificate

Pressing the softkey "Cancel" will do no action and return to "Trusted" dialog.
Pressing the softkey "Ok" will delete the selected certificate.

Note

After the deletion of the client certificate a connection with OPC UA server can no longer be established by the client of the corresponding certificate.

Importing certificates

To prepare a connection a client certificate can be imported before actually establishing a connection. With the import the certificate is automatically trusted.

1. Press the softkey "Import".
   A pop-up screen will appear showing the SINUMERIK filesystem to import from. You can navigate to a location on the SINUMERIK filesystem to import a certificate to a trusted folder.
   Pressing the softkey "Cancel" will ignore all input and return to the "Trusted" dialog.
   Pressing the softkey "Ok" will import the certificate.

Note

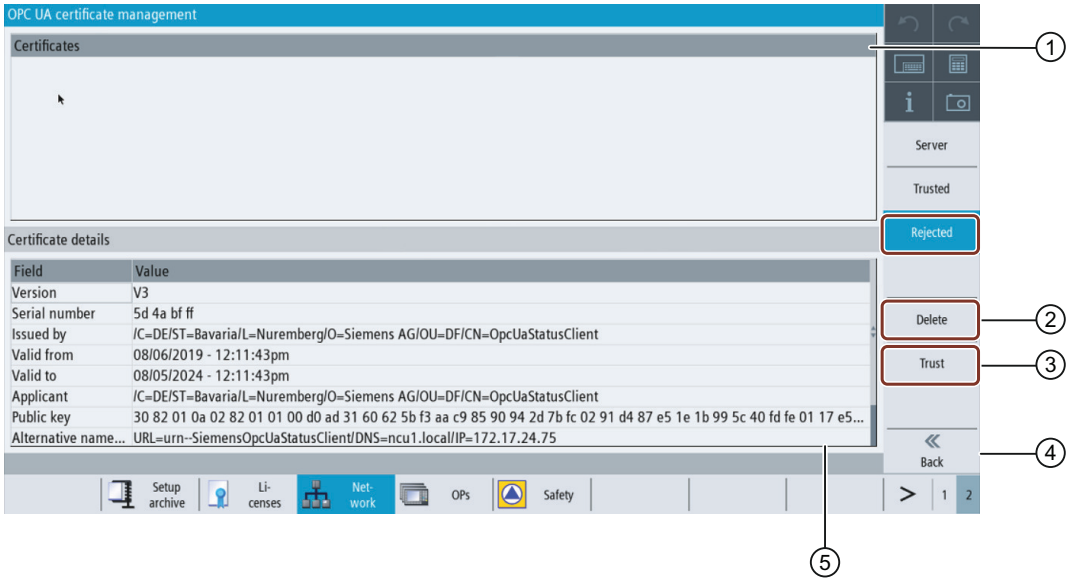
Only certificates with the file extension "*.der" are accepted.

Note

To use a certificate for authentication it is necessary to create a user with the method "AddCertificateUser" first.

4.4.3.2 Rejected certificates

Overview



① The rejected certificates are listed in the upper part of the screen. You can select a certificate using the arrow keys (cursor up/ cursor down).

② You can delete the selected certificate.

③ You can trust the selected certificate.

④ You can leave the OPC UA dialogs.

⑤ The certificate details are shown in the lower part of the screen. To set the focus on the lower part of the screen the softkey "next window" on the keyboard is used.

Figure 4-10 Rejected Certificate

Deleting rejected certificates

1. To manually delete a client certificate, select the certificate in the rejected list and press the softkey "Delete".
   A pop-up screen will appear asking you for confirmation of deletion:



Figure 4-11 Delete rejected certificate

Pressing the softkey "Cancel" will do no action and return to the previous dialog.
Pressing the softkey "Ok" will delete the selected certificate.

Trusting rejected certificates

If the setting "Accept certificates automatically" is deactivated, certificates automatically transferred by a client with the first connection attempt will be treated as untrusted and need to be trusted manually before the connection can be established. In this case, the server will report an error (BadSecurityChecksFailed) on initial connection attempt.

1. To manually trust a client certificate, select the certificate in the rejected list and press the softkey "Trust".
   A pop-up screen will appear asking for confirmation of trusting the certificate.



Figure 4-12 Trust certificate

   Pressing the softkey "Cancel" will return to the "Rejected" dialog.
   Pressing the softkey "Ok" will trust the certificate and move it to the trusted folder.

4.5 Testing the connection

Requirement

To test the connection, you can use the "Sample Applications" of the OPC Foundation (https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-architecture-for-cnc-systems/) under "Developer Tools/Developer Kits/Unified Architecture". It is necessary to register with the OPC Foundation for this.

Note

There are two ways to establish the connection:

- Connection without security
- Connection with the security policy "Aes128Sha256RsaOaep", "Aes256Sha256RsaPss", "Basic256Sha256", "Basic128 (RSA15)" (deprecated), and "Basic256" (deprecated) and the security mode "SignAndEncrypt".

SIEMENS always recommends setting up a connection with security, as only in this way, the confidentiality of the data transmitted can be ensured.

Installation

The "Sample Applications" additionally install a service with the name "OPC UA Local Discovery Server". If you want to locally test the OPC UA connection, i.e. an installation directly on the IPC, you must deactivate this service.

Note

If the service "OPC UA Local Discovery Server" is active, the OPC UA server cannot be started correctly, because it blocks the needed TCP port 4840.

This service has no influence if the "Sample Applications" are installed on a PC in the network. Deactivation is then not necessary.

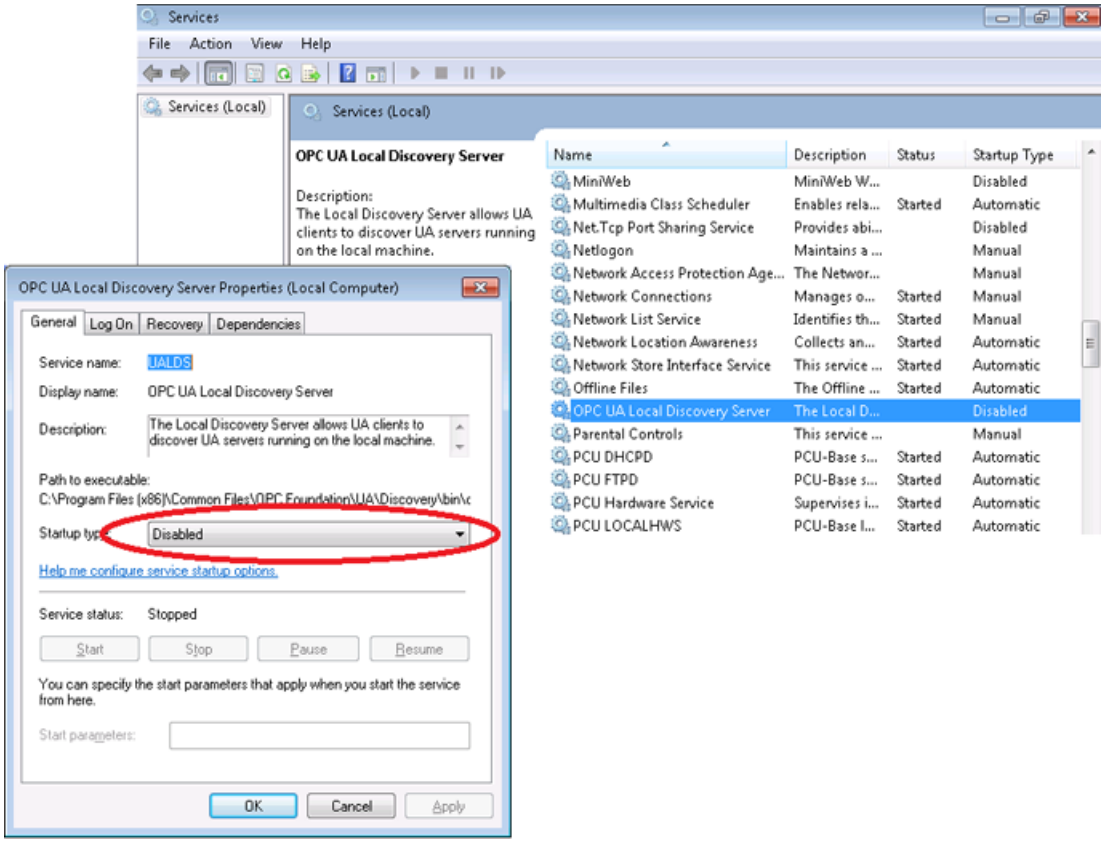


Figure 4-13 Deactivating the "OPC UA Local Discovery Server" service on IPC

Procedure

1. Start the OPC UA "Sample client".

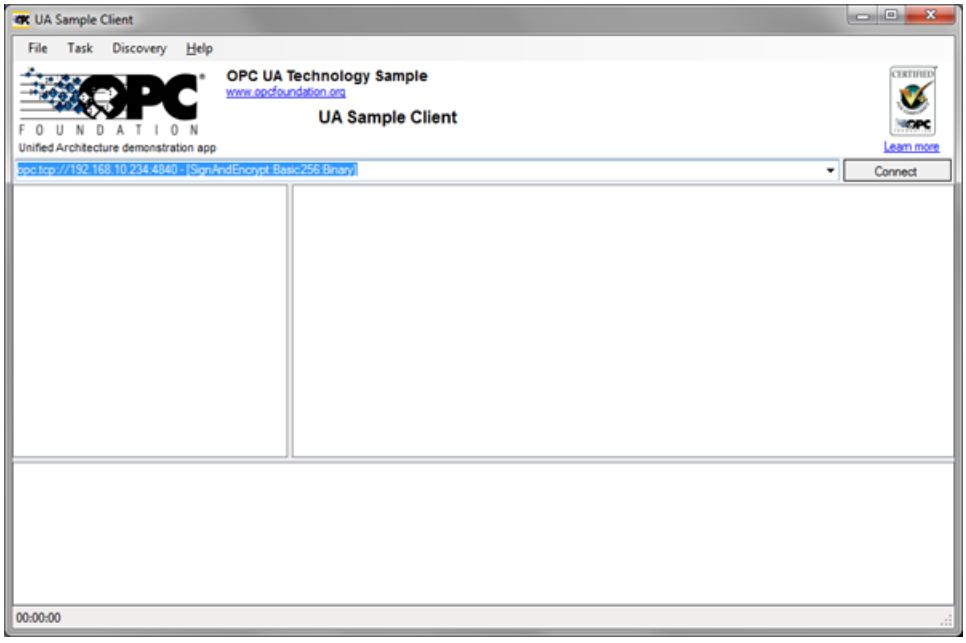


Figure 4-14 Sample Client main window

2. Select the "New" entry from the drop-down list.
   The "Discover Servers" window appears.
3. Enter the IPv4 address of the target system and click "Discover".

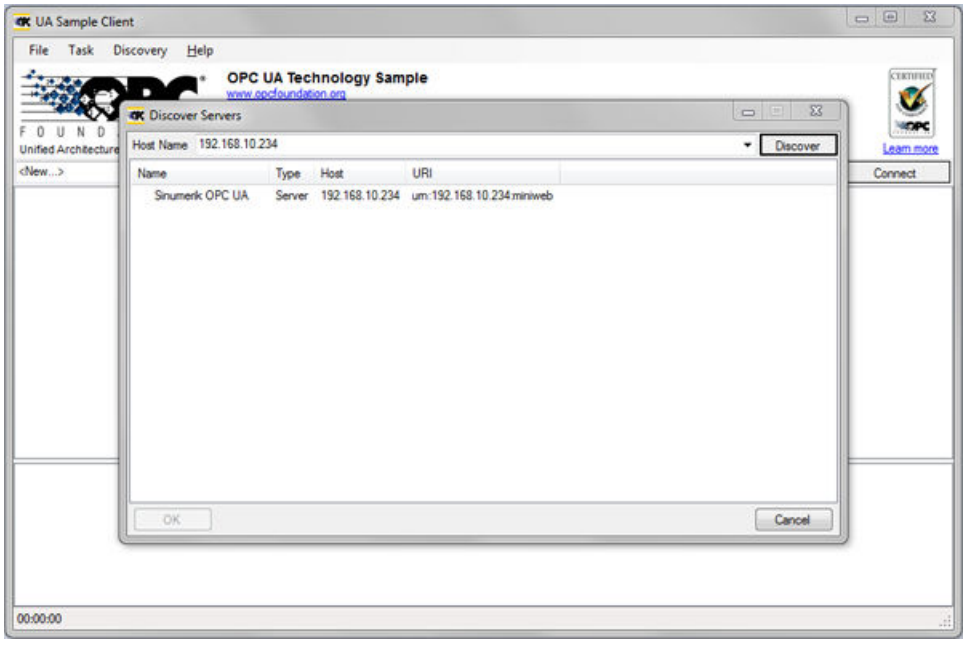


Figure 4-15 Discover servers

4. The SINUMERIK OPC UA server appears in the list. Select the server and confirm with "OK".
5. Return to the main window and click "Connect".

6. To establish a simple connection without security, configure the following settings. After clicking "OK", enter the administrator user assigned when OPC UA was set up and the administrator password. Confirm your settings by clicking "OK".

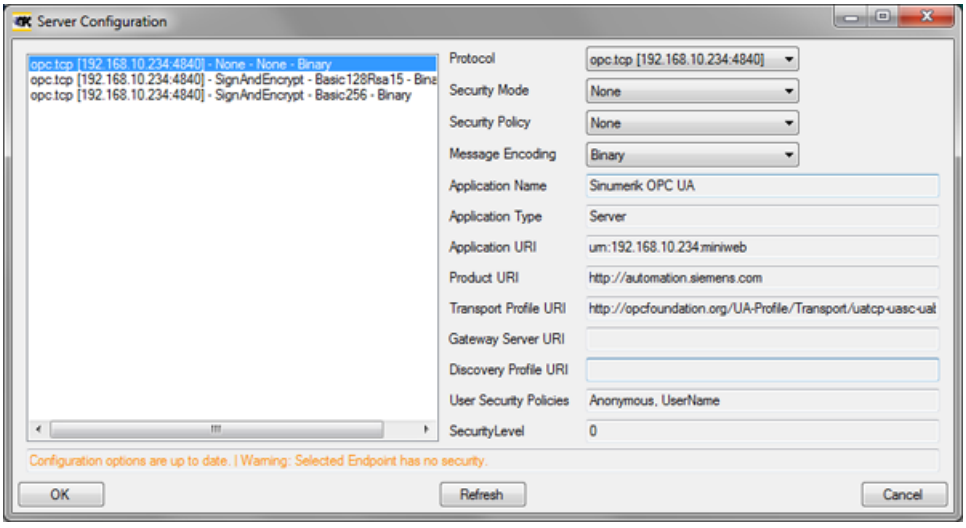


Figure 4-16 Server configuration

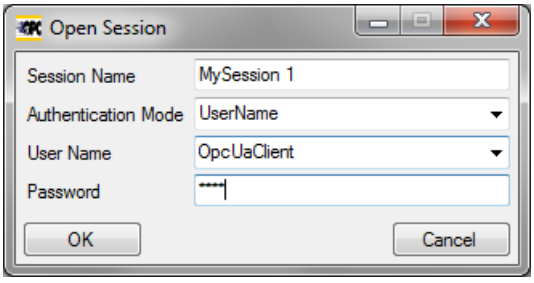


Figure 4-17 User Identity

7. Confirm the prompt asking if you want to trust the transferred certificate with "Yes".

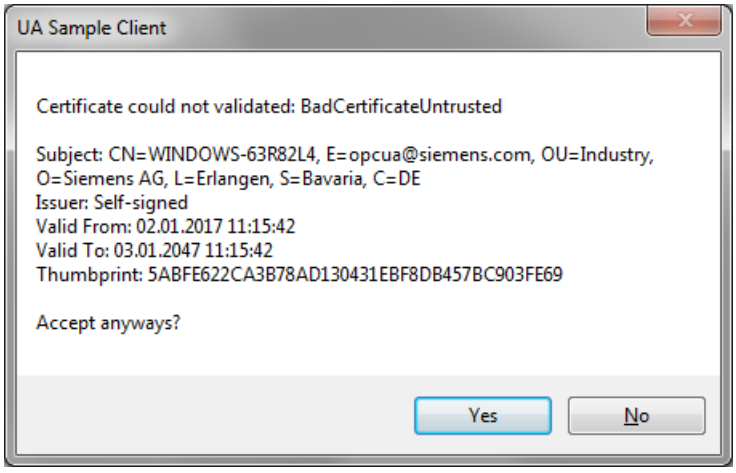


Figure 4-18 Certificate

The connection to the SINUMERIK OPC UA server is now established and the available address space is displayed.

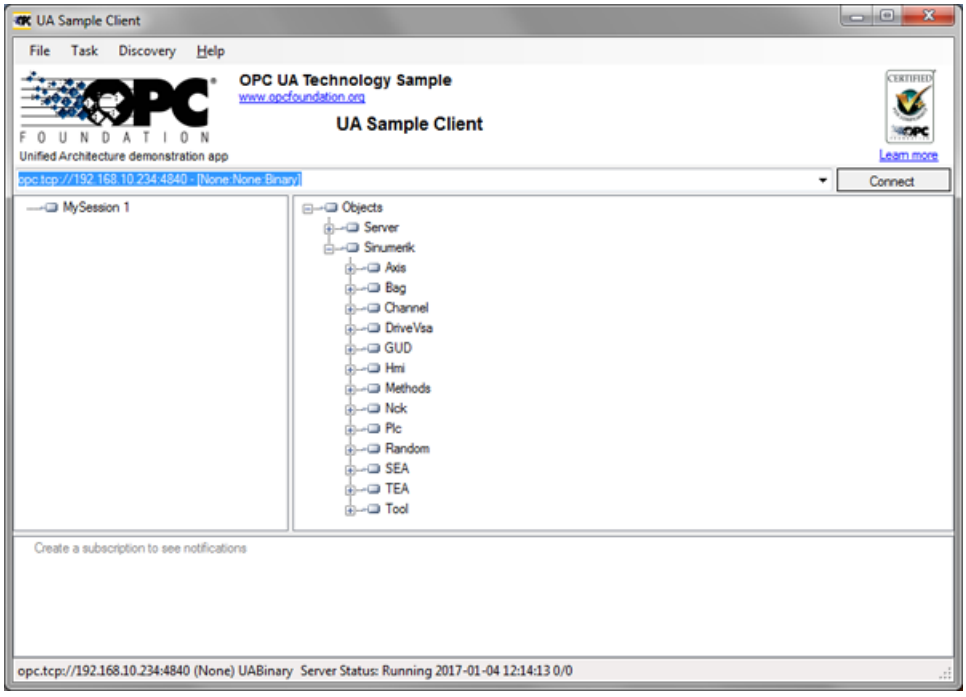


Figure 4-19 Address space of the SINUMERIK OPC UA server

8. Navigate to a nodeID (e.g. R-parameter at Sinumerik > Channel > Parameter > R) and right-click the corresponding entry. You can now test various functions:

– E.g. read, write, and setup monitoring

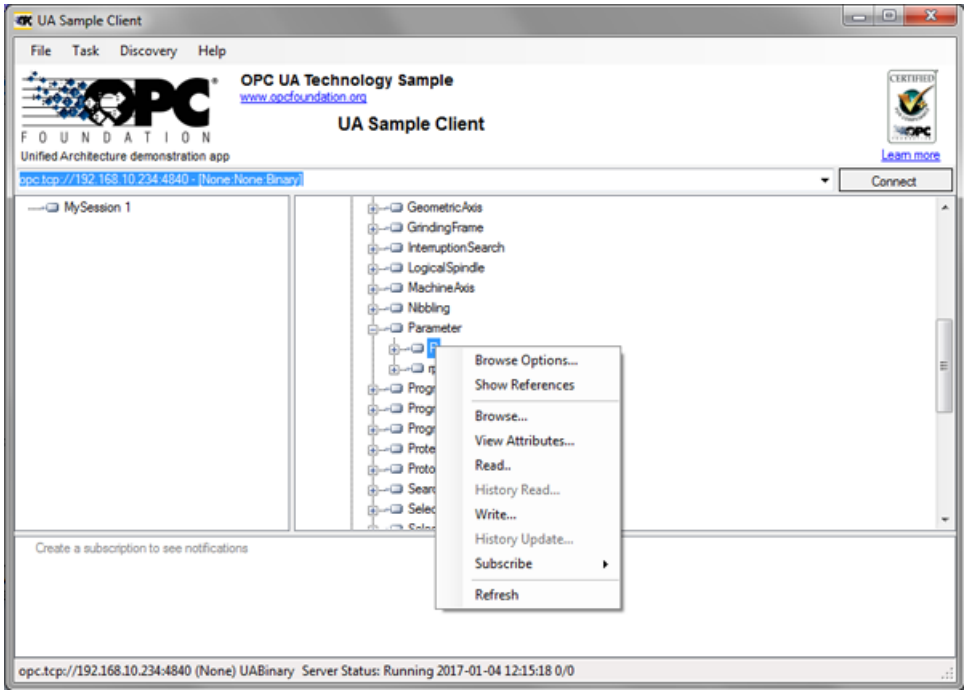


Figure 4-20 NodeID "Sinumerik > Channel > Parameter > R"

– The attributes of a NodeID can be queried via the entry "View Attributes". One of these attributes is the "Value", which provides the corresponding value of R1.

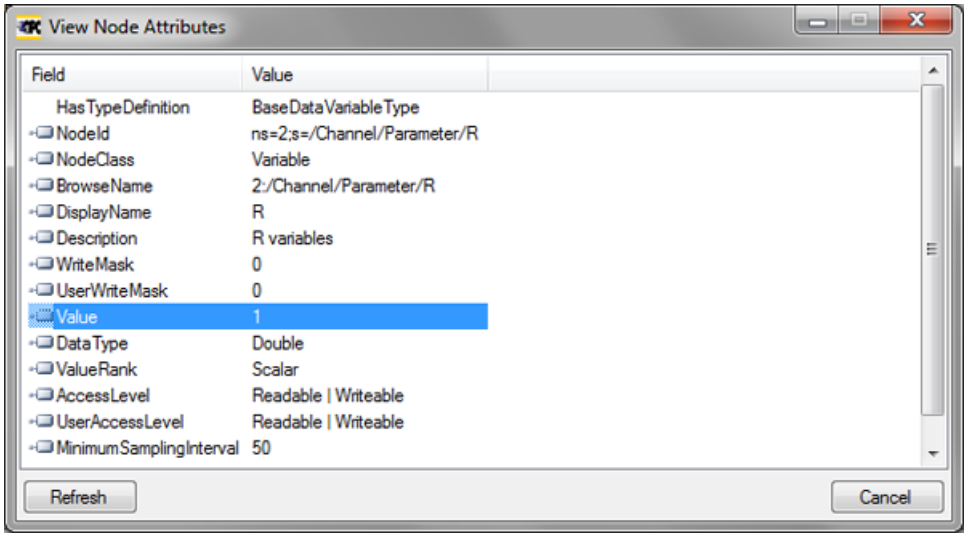


Figure 4-21 Viewing node attributes

4.6 General OPC UA settings

General settings of OPC UA can be set in the file settings.ini. This file is located under the following path:

- OPC UA customer server: /user/sinumerik/hmi/opcua

Absolute DBs in the address tree

You can specify if the absolute DB addresses are to be displayed in the address space.

As a default, this setting is set to `true` which means absolute and symbolic addresses are both visible in the address space.

By setting this value to `false`, you can hide the absolute DB addresses.

Settings.ini

```
[General]
.
.
.
showAbsolutDBs=true
```

Customer server URI

For customer with multiple machines, it is possible to specify a specific customer server URI for each machine. The default value will be empty.

`SiemensAG:SinumerikServer`

Settings.ini

```
[General]
.
.
.
customURI=urn:MyCompany:MyServer
```

Note

Make sure to renew the server certificate (as described in chapter Server certificates (Page 39)) after the entry has been changed.Part program messages as OPC UA alarm

Enabling Part program messages as OPC UA alarm

As a default, this setting is set to `false` which means, Part program messages are sent as `BaseEventType` in OPC UA.

If a customer wants the Part program messages to be sent as alarms in OPC UA, then the following property under the "Events" tag needs to be set to `true` in the settings.ini file. Additionally, it will be sent as event also.

Settings.ini

```
[Events]
.
.
.
PartPrgMsgAsAlarm=true
```

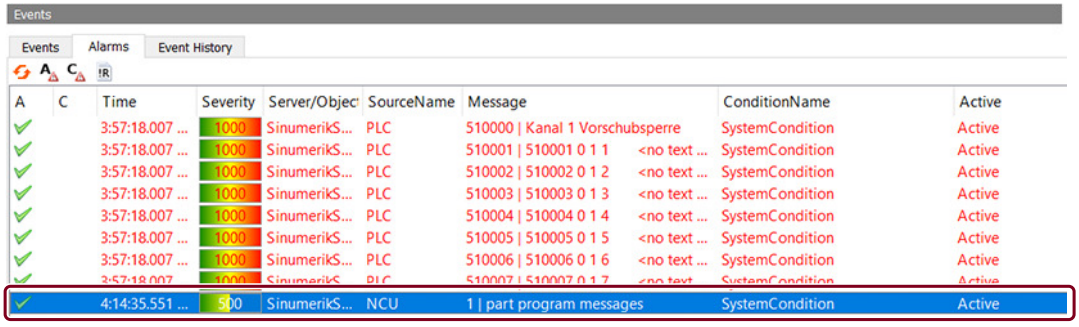


Figure 4-22 PartPrgMsgAsAlarm is True

Hiding OPCUA alarm message ID from the message text

As a default, this setting is set to `true` which means it displays the alarm messages ID combined with message text.

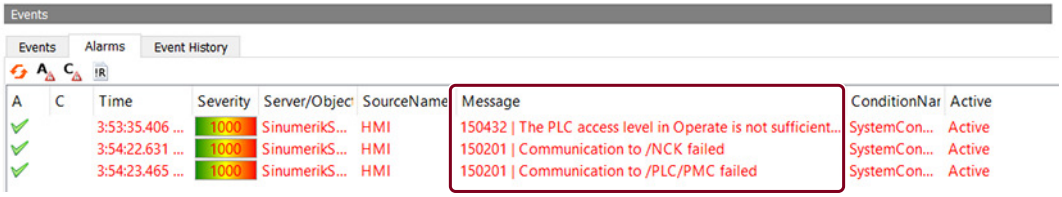


Figure 4-23 DispAlarmIDWithText is True

If a customer wants to hide the alarm message ID and to display only the message text in OPC UA, then the following property under the "Events" tag needs to be set to `false` in the settings.ini file.

Settings.ini

```
[Events]
.
.
.
DispAlarmIDWithText=false
```

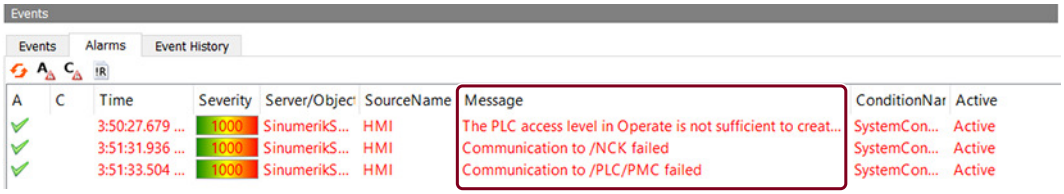


Figure 4-24 DispAlarmIDWithText is False

Customer Specific Object Model (CSOM) 5

5.1 Overview

What is a Customer Specific Object Model (CSOM)?

The primary objective of the OPC UA address space is to provide a standard way for servers to represent objects to clients. The OPC UA object model has been designed to meet this objective. It defines objects in terms of variables and methods. It also allows relationships to other objects to be expressed.

A **C**ustomer **S**pecific **O**bject **M**odel allows a specific view of the OPC UA server to meet the requirements in a customer-specific project. For this purpose, you can specify all required OPC UA nodes in an XML file.

You can use several tools to create your own Object Model. In this manual, all processes and examples are explained on the base of the tool **SiOME** (**S**iemens **O**PC UA **M**odeling **E**ditor).

SiOME

With the help of SiOME, a tool for implementing Customer Specific Object Models or OPC UA companion specifications, you can:

- design information models / address spaces for your OPC UA server
- create new types and instances of OPC UA nodes
- map OPC UA variables to SINUMERIK variables
- enable multiple license in a single custom model (Page 75)

Download link and explanations about SIOME are available here (https://support.industry.siemens.com/cs/de/en/view/109755133).

5.2 Functionalities

What is possible with a CSOM?

Possible application scenarios for a CSOM could be the following:

- Implementation of an own Information model
- Modifying of an own information model:
  - structure
  - display name
  - browse name
  - description

What is not possible?

- Change of data types

CSOM limits

Feature	Value
Maximum number of binary files	25
Maximum number of CSOM namespaces	12[1]
Maximum number of nodes in CSOM	10,000
Maximum number of binary files that can be activated	10

(1) Maximum number of CSOM namespaces can be created is 11, if license namesapce is created. For more information, refer topic Modeling rules (Page 88).

5.3 Workflow for using CSOM in the SINUMERIK OPC UA server

5.3.1 Overview

In order to use a Customer Specific Object Model (CSOM), it is necessary to follow a certain workflow procedure.

The CSOM process workflow described below provides an overview on the necessary process steps. Every process step is explained in greater detail in the following chapters.

CSOM process workflow

The CSOM workflow consists of the following steps:

1. Exporting SINUMERIK model from OPC UA server as OPC UA XML with SINUMERIK Access MyMachine /P2P.
2. Creating a CSOM with SiOME (Page 57).
3. Converting the CSOM from XML to binary (Page 89) with SINUMERIK Access MyMachine / P2P.
4. Importing the CSOM into the SINUMERIK OPC UA server (Page 91) with SINUMERIK Operate.

5.3.2 Exporting SINUMERIK model from OPC UA server as OPC UA XML

SiOME offers the possibility to map variables using drag and drop. In order to provide this usability, SiOME needs to know the SINUMERIK address space of the machine, where the CSOM should be implemented.

Since the address space depends on machine configuration, this address space has to be exported after commissioning of the machine using "SINUMERIK Access MyMachine /P2P".

SINUMERIK Access MyMachine /P2P is reading SINUMERIK address space via OPC UA browse functionality and providing an XML file to be imported into SiOME.

Prerequisites

For exporting SINUMERIK OPC UA model, it is necessary to have SinuReadAll access right.

SINUMERIK Access MyMachine /P2P

SINUMERIK Integrate Access MyMachine /P2P (MLFB 6FC5860-7YC00-0YA0) supports the commissioning of machines with SINUMERIK Operate (software version 2.6 and higher) using a standard Windows PC. Its scope of functions includes the exchange of files between the service PC and the control as well as operation of the HMI user interface. EasyScreen texts, alarm texts, tool management texts and other texts can be edited easily.

Exporting SINUMERIK OPC UA model as OPC UA XML

The following graphic shows the process of exporting the target system address space as OPC UA XML.

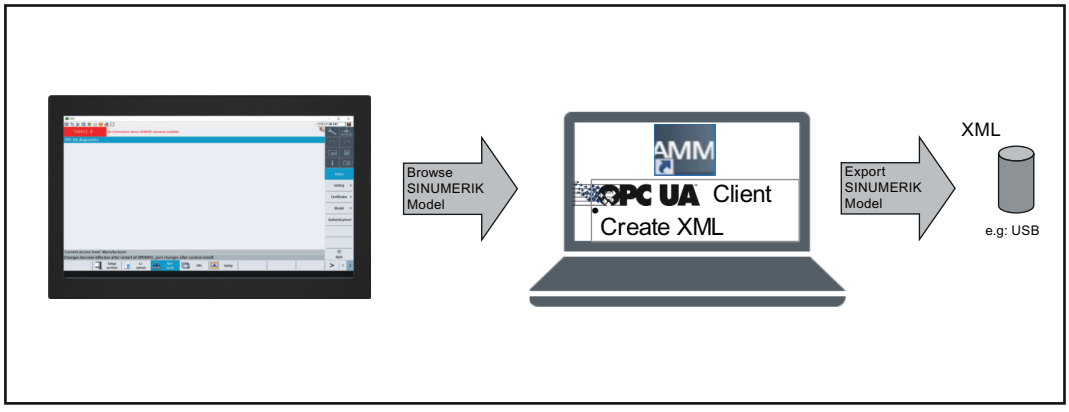


Figure 5-1 Exporting SINUMERIK OPC UA model as XML

Procedure

1. Open the application "SINUMERIK Access MyMachine /P2P".
2. Click on "Tools > Sinumerik Opcua Server Tools > Opcua Server Model Export". A popup window appears.

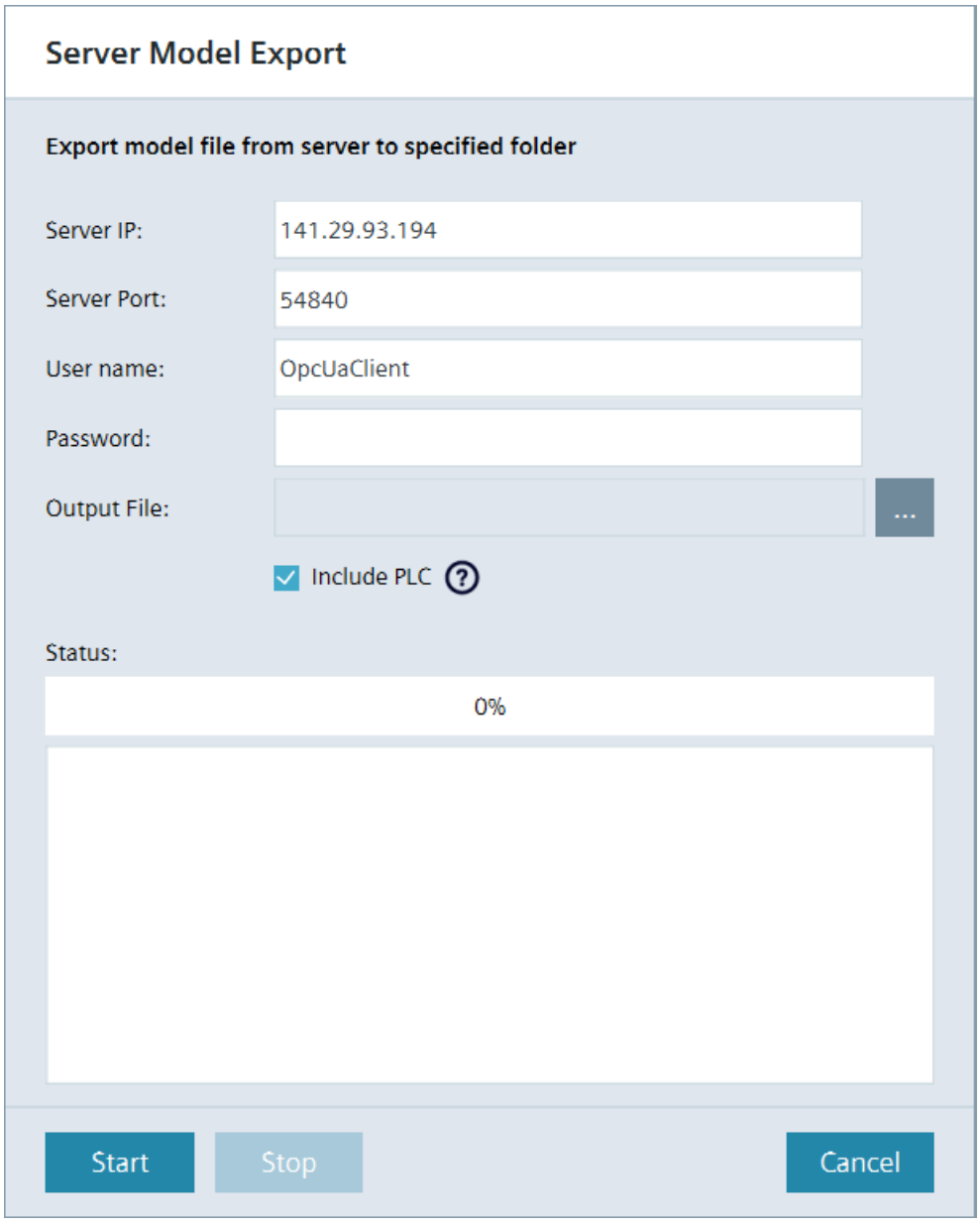


Figure 5-2 Opcua Server Model Export

3. Enter the IP address and the port of the OPC UA server as well as the username and password to access the server. Then specify where the output file should be saved to.

Note

Include PLC

To avoid long loading times, SiOME supports loading the TIA portal project directly.

For SINUMERIK ONE it is therefore recommend to deactivate the checkbox "Include PLC" and load the TIA portal project directly in SiOME.

4. Click "Start" button to generate the xml file. The generated XML file is saved to the specified location.

The generated XML file can then be imported into SiOME tool.

5.3.3 Creating a CSOM with SiOME

5.3.3.1 Overview

SiOME is a free tool from SIEMENS that allows to easily create an OPC UA object model and map the object either to SINUMERIK or SIMATIC S7-1500 variables.

The following chapter shows the engineering workflow for a CSOM with SINUMERIK and SiOME.

This process itself has 3 sub steps:

1. AUTOHOTSPOT
2. Modeling own object model (Page 60)
3. Exporting CSOM (XML) (Page 80)

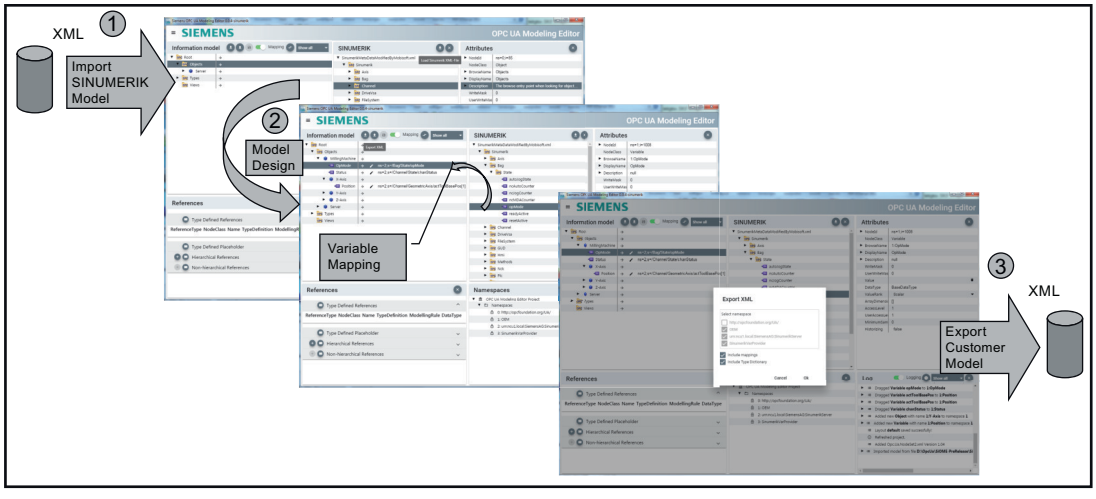


Figure 5-3 Creating a CSOM with SiOME

These 3 sub steps are explained further in the below sections with one application example.

5.3.3.2 Importing SINUMERIK model (XML)

Prerequisite

Exported SINUMERIK model (XML) file from SINUMERIK Access MyMachine /P2P.

Procedure

1. Open the SiOME application.
2. Click "Layout" drop down and select "SINUMERIK" as shown in the below image. The SINUMERIK layout is displayed.

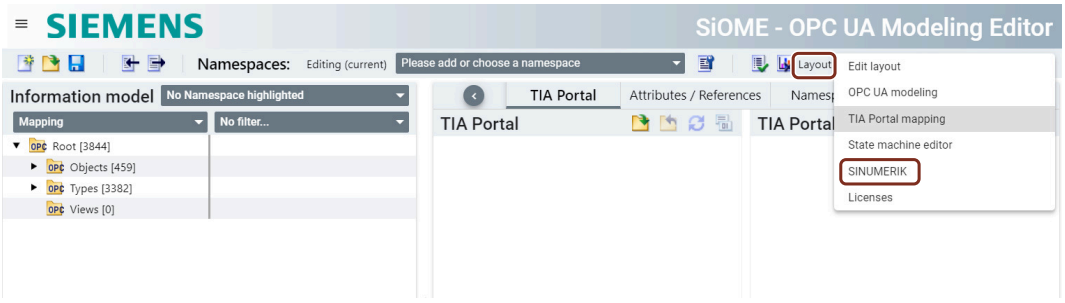


Figure 5-4 SiOME layout option

3. Click the icon in "SINUMERIK" tab to import the SINUMERIK model (XML) as shown in the below image.

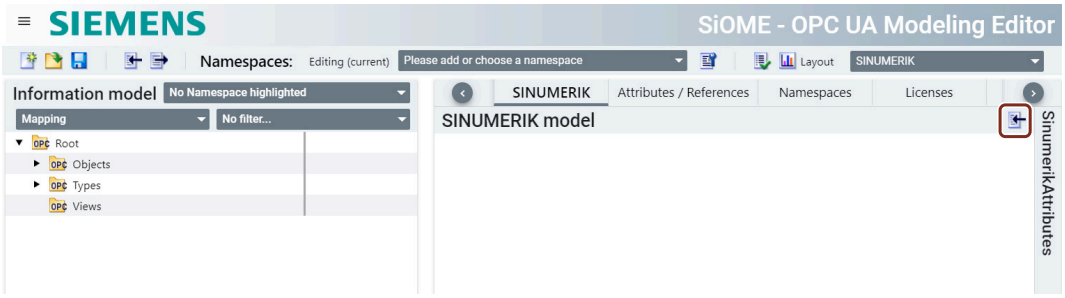


Figure 5-5 SiOME dashboard page

4. A "Open" dialog box appears. Select the exported SINUMERIK model (XML) file and then click "Open".

Result

SINUMERIK tab shows the SINUMERIK browse tree as shown in the below image:

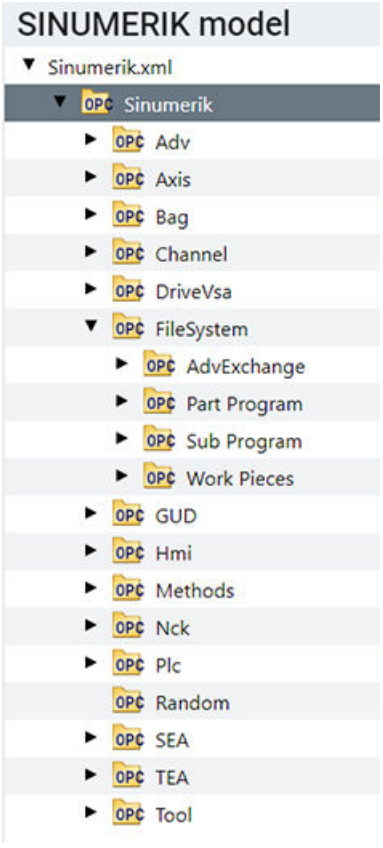


Figure 5-6 SINUMERIK tab with SINUMERIK browse tree

Note

The "Adv" and "AdvExchange" nodes are enbaled only if the OPC UA Advanced option is activated.

5.3.3.3 Modeling own object model

Creating new namespace

To create a new namespace, follow the below steps:

1. Click on "Please add or choose a namespace" and then click "Add New Namespace" as shown in the below image.
2. The "Add Namespace" popup window appears. Enter a name in the "Namespace URI" field and then click "Ok".

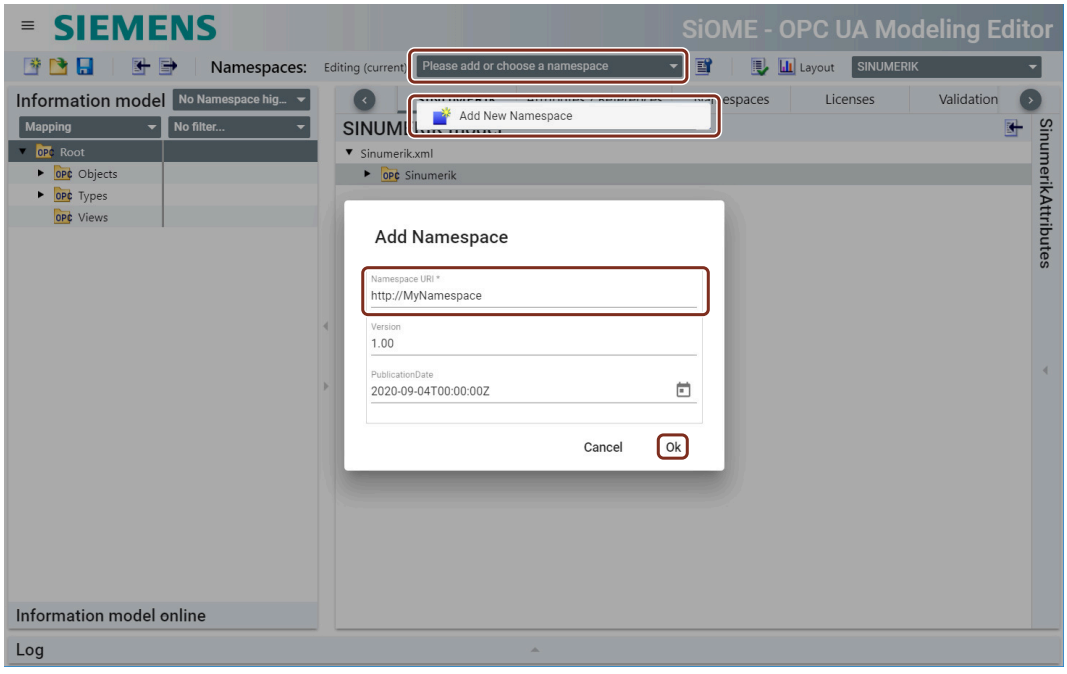


Figure 5-7 Adding new Namespace

Result

The new namespace is added in "Namespaces" tab.

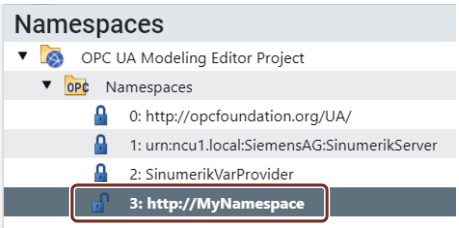


Figure 5-8 Namespace added

Creating a new object

To create a new object, follow the below steps:

1. Right-click on "Objects" node in "Information model" tab and then click on "Add Instance".

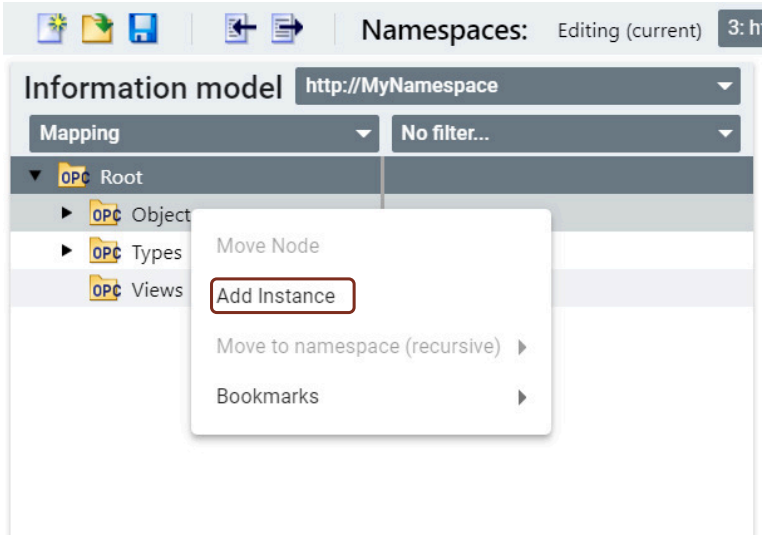


Figure 5-9 Information model tab

2. The "Add Instance" popup window appears. Enter a name in the "Name" field.

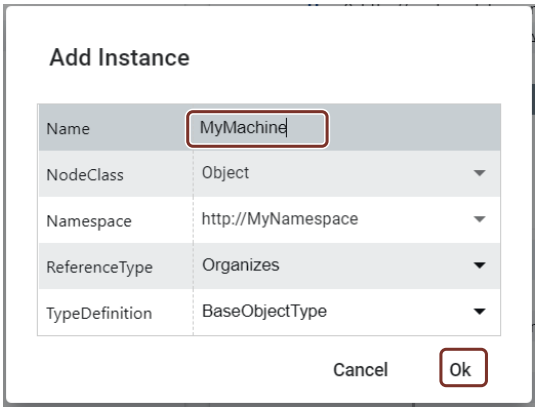


Figure 5-10 Add Instance popup window

3. Verify that newly added namespace is selected in the "Namespace" row and then click "Ok".

Result

The new instance is added under "Objects" node.

Adding a new instance for a variable

To add a new instance for a variable, follow the below steps:

1. Under "Objects" node, right-click on "MyMachine" node and then click on "Add Instance".

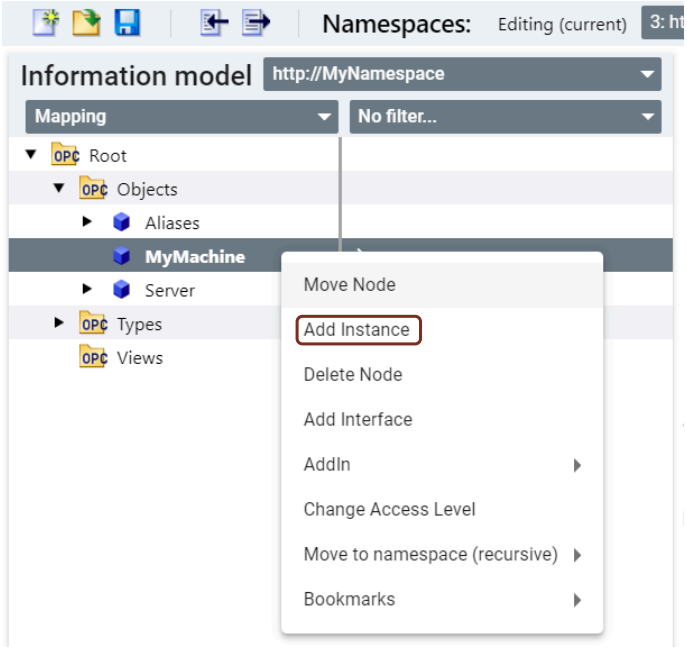


Figure 5-11 Adding a new instance for a variable

2. The "Add Instance" popup window appears. Enter a name in the "Name" field.
3. Select "Variable" from the "NodeClass" drop-down list.
4. Select "Double" from the "DataType" drop-down list and then click "Ok".

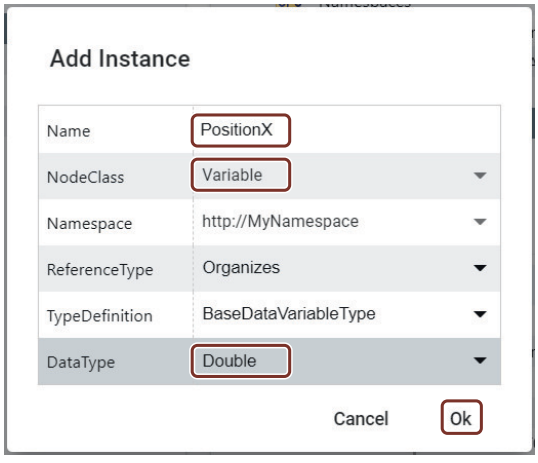


Figure 5-12 Add Instance popup for a variable

Result

The new instance is added for a variable under "MyMachine" node.

Mapping to SINUMERIK data

To map the SINUMERIK data, follow the below steps:

1. Drag and drop the variable from the "SINUMERIK" tab to the variable "PositionX" in the "Information model" tab as shown in the below image.

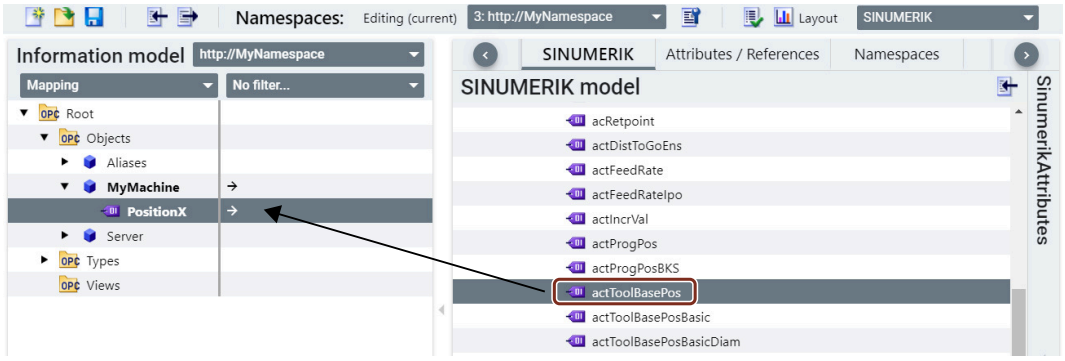


Figure 5-13 Mapping SINUMERIK variable

The variable is mapped in the mapping table in the "Information model" tab.

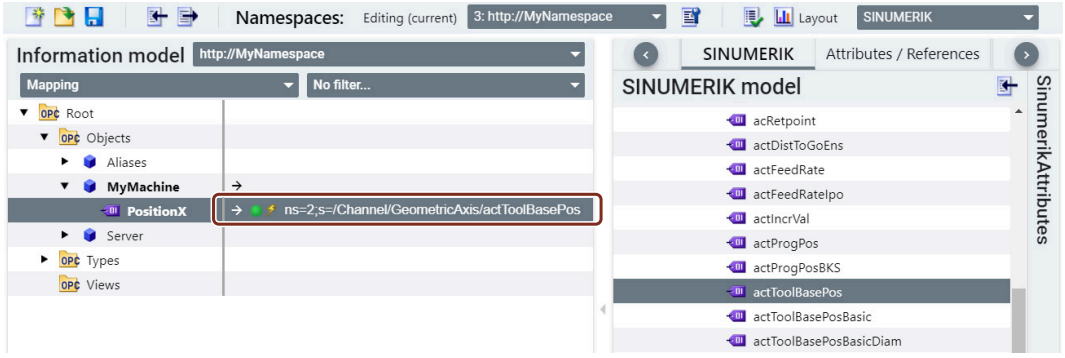


Figure 5-14 SINUMERIK variable mapped in mapping table

2. Add the correct axis index.

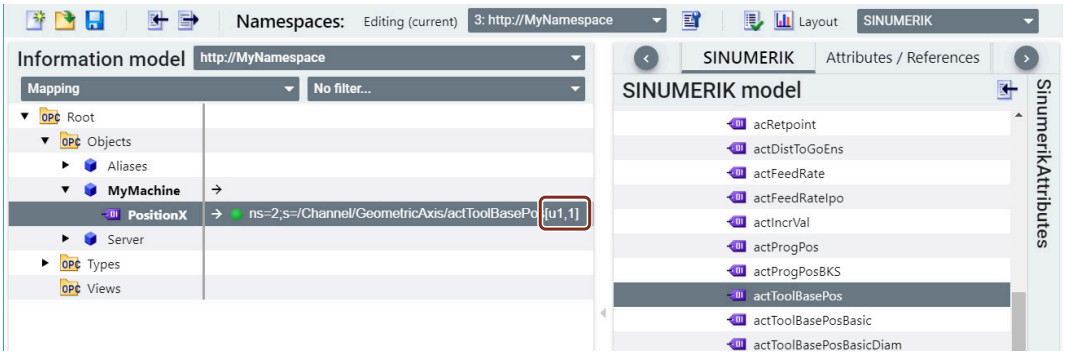


Figure 5-15 Adding axis index

Note

Do not edit the namespace value (for instance, ns=2). If edited, it will not be considered.

Adding a method node in CSOM

Method nodes can be added in two ways:

- Direct drag and drop under custom object.
- Adding a new instance variable and mapping the method from SINUMERIK tab.

Direct drag and drop under custom object

To add a method, press and hold the **ctrl** button and drag and drop the method from the "SINUMERIK" tab to the "MyMachine" node as shown in the below image.

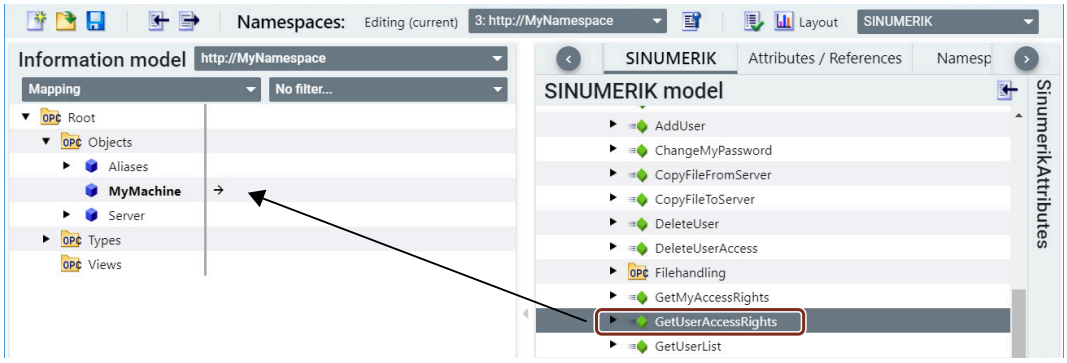


Figure 5-16 Mapping SINUMERIK method

The method is added under "MyMachine" node in the "Information model" tab.

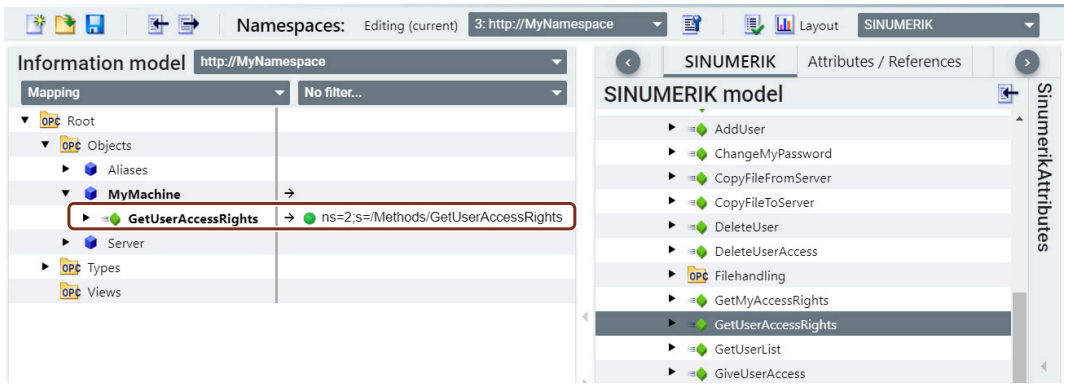


Figure 5-17 SINUMERIK method added

Adding a new instance variable and mapping the method from SINUMERIK tab

To add a new instance for a method, follow the below steps:

1. Under "Objects" node, right-click on "MyMachine" node and then click on "Add Instance".

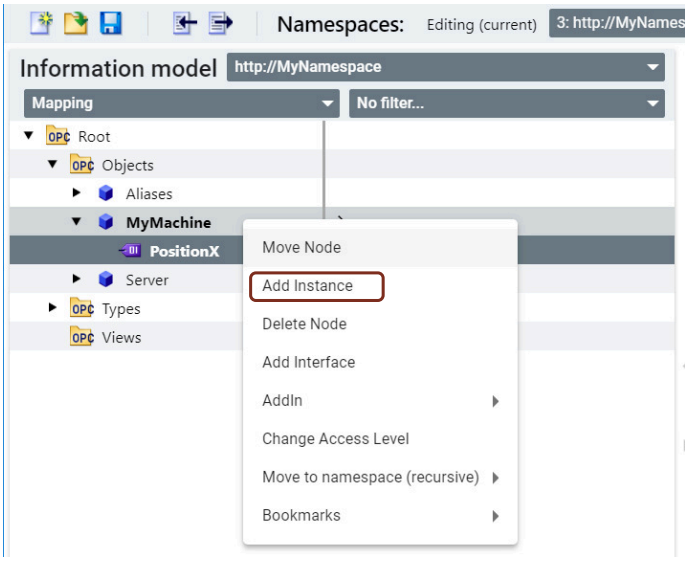


Figure 5-18 Adding a new instance for a method

2. The "Add Instance" popup window appears. Enter a name in the "Name" field.
3. Select "Method" from the "NodeClass" drop-down list and click "Ok".

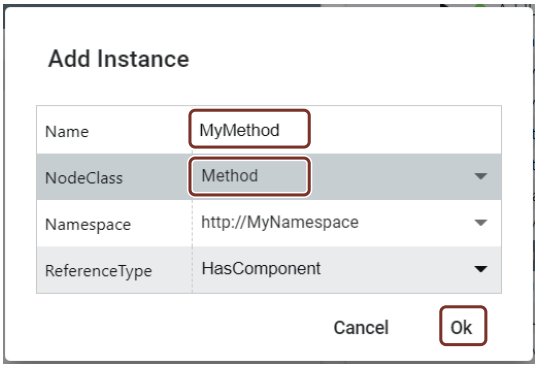


Figure 5-19 Add Instance popup for a method

4. The new instance is added for a method under "MyMachine" node. Expand the "MyMethod" node.

5. Right-click on "InputArguments" node and then click on "Add New Argument".

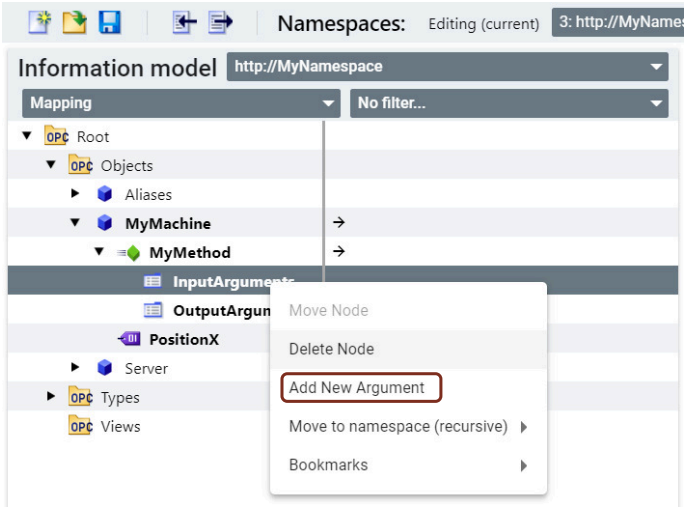


Figure 5-20 Adding a new argument under method

6. Select the newly created argument and then configure the values in the "Attributes / References" tab according to the method argument.

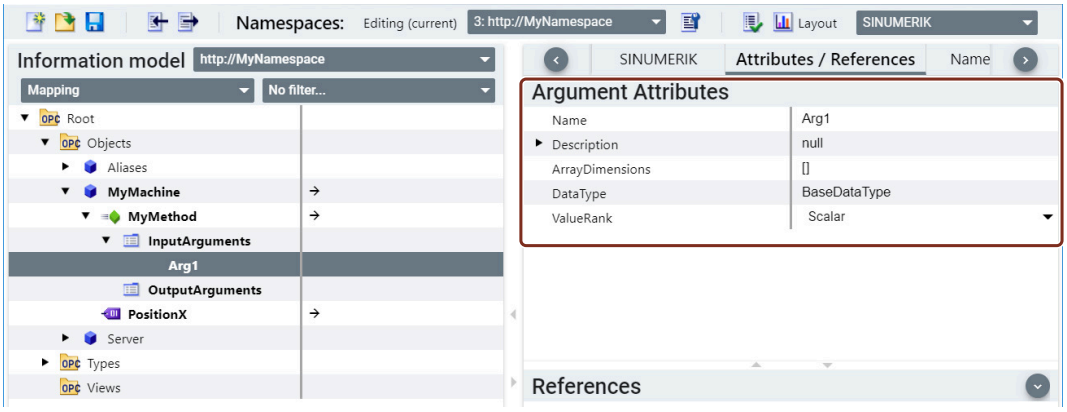


Figure 5-21 Arguments tab

7. Then right-click on "OutputArguments" node and then click on "Add New Argument".

8. Select the newly created argument and then configure the values in the "Attributes / References" tab according to the method argument.
9. Drag and drop the method from the "SINUMERIK" tab to the "MyMethod" node in the "Information model" tab as shown in the below image.

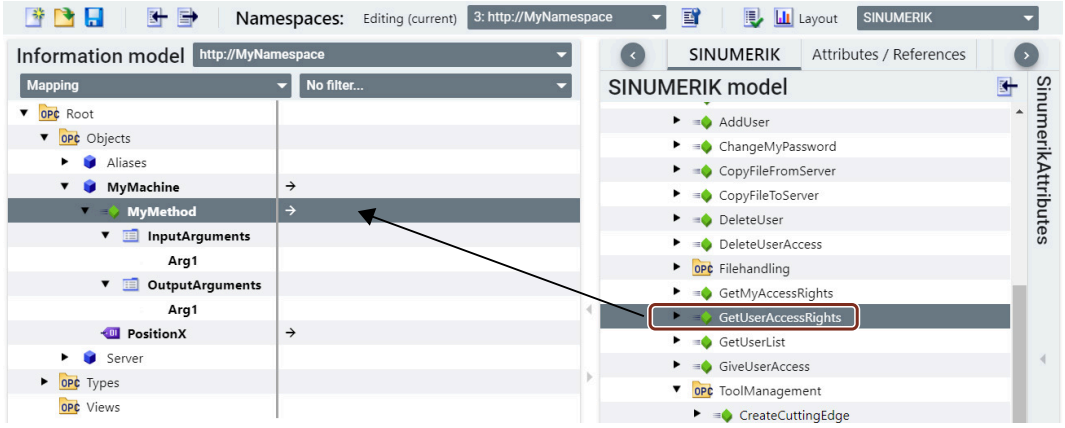


Figure 5-22 Mapping SINUMERIK method

The method is mapped in the mapping table in the "Information model" tab.

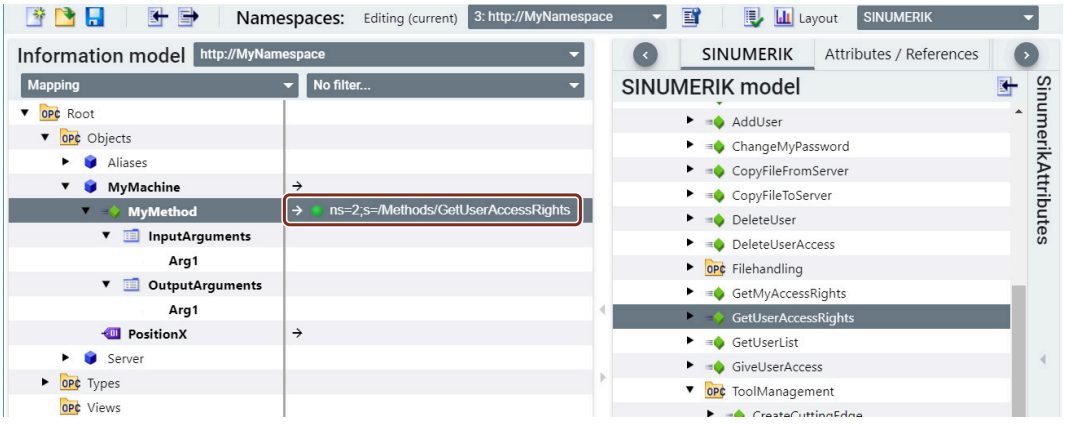


Figure 5-23 SINUMERIK method mapped in mapping table

Note

"CopyFileToServer" and "CopyFileFromServer" methods must be mapped in the same namespace as that of the file system node.

Adding an instance for a SINUMERIK alarm in SiOME

To add an instance for a SINUMERIK alarm, follow the below steps:

1. Under "Objects" node, right-click on "MyMachine" node and then click on "Add Instance".

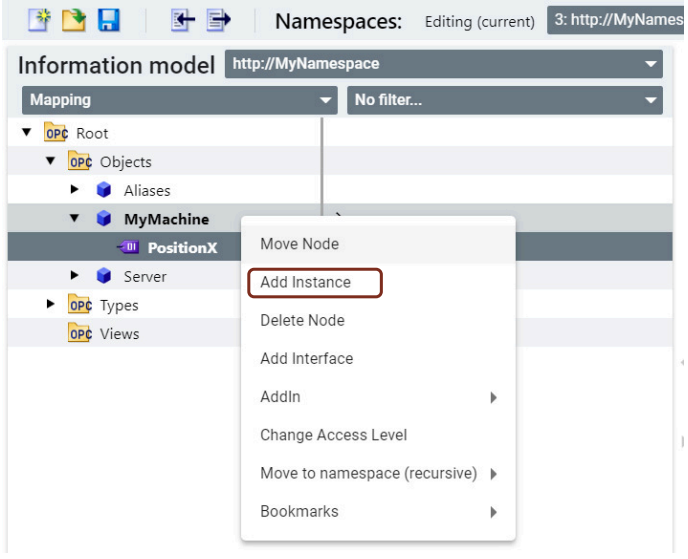


Figure 5-24 Adding a new instance for an alarm

2. The "Add Instance" popup window appears. Enter a name in the "Name" field.
3. Select "Object" from the "NodeClass" drop-down list and then click "Ok".

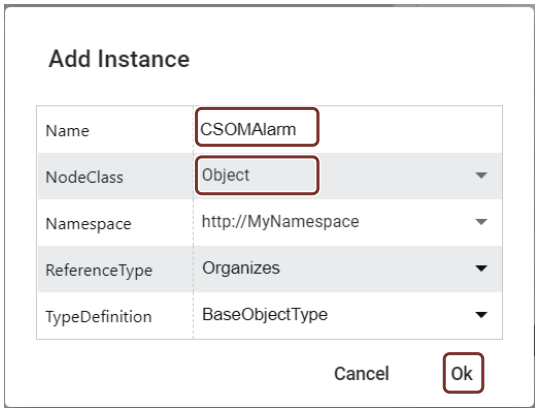


Figure 5-25 Add Instance popup for an alarm

4. The new instance is added for an alarm under "MyMachine" node. Click on "EventNotifier" in the "Additional OPC UA Attributes" tab as shown below.

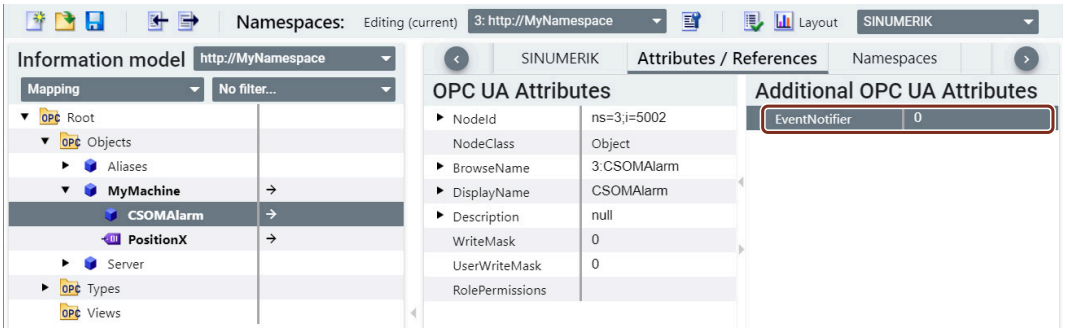


Figure 5-26 New Instance added under a variable for alarm

5. The "Event Notifier" popup window appears. Select "SubscribeToEvents" check-box and click "Ok".

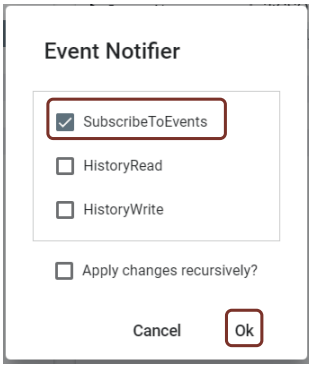


Figure 5-27 Event Notifier popup for an alarm

6. Map the newly created instance with the SINUMERIK node.

Result

An object with event notifier is created and can be used for subscribing alarms.

Adding file system node in CSOM

Before adding a file system node, you need to do the following changes in the settings page in SiOME:

1. Click icon in the SiOME home page and then click "Settings" as shown in the below image.
2. The "Settings" popup window appears. Change the highlighted fields as shown in the below image and then click "Ok". The settings are applied.

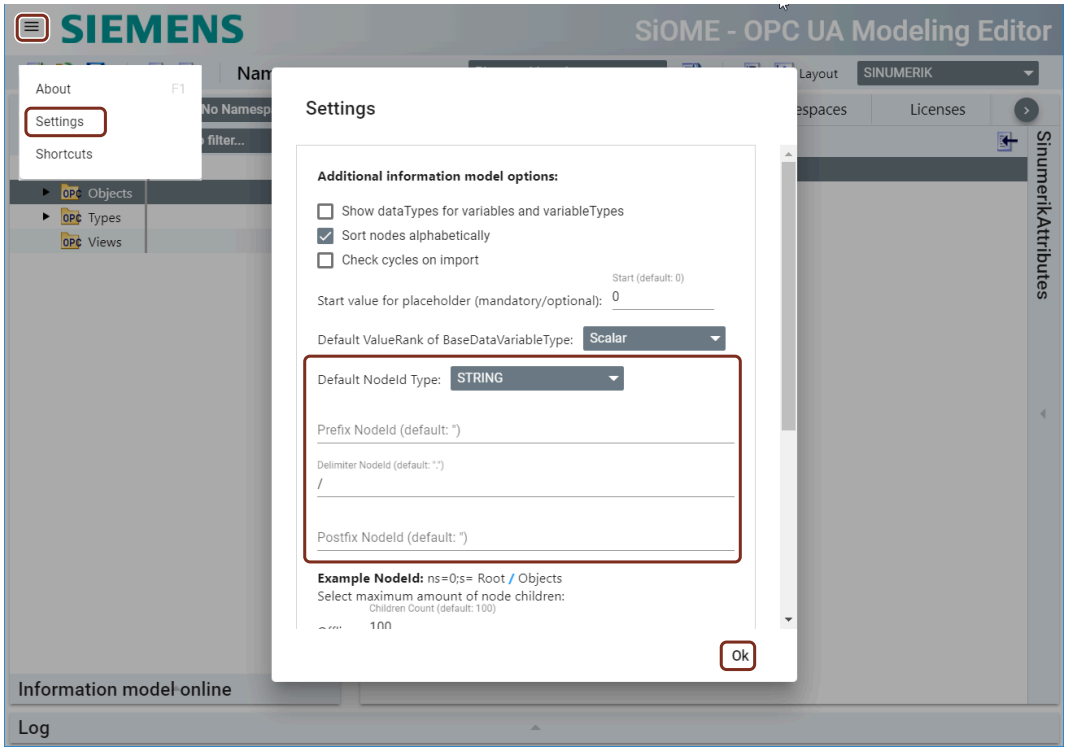


Figure 5-28 SiOME Settings page

To add a file system node, follow the below steps:

1. Click on "Please add or choose a namespace" and then click "Add New Namespace" as shown in the below image.
2. The "Add Namespace" popup window appears. Enter a name in the "Namespace URI" field and then click "Ok".

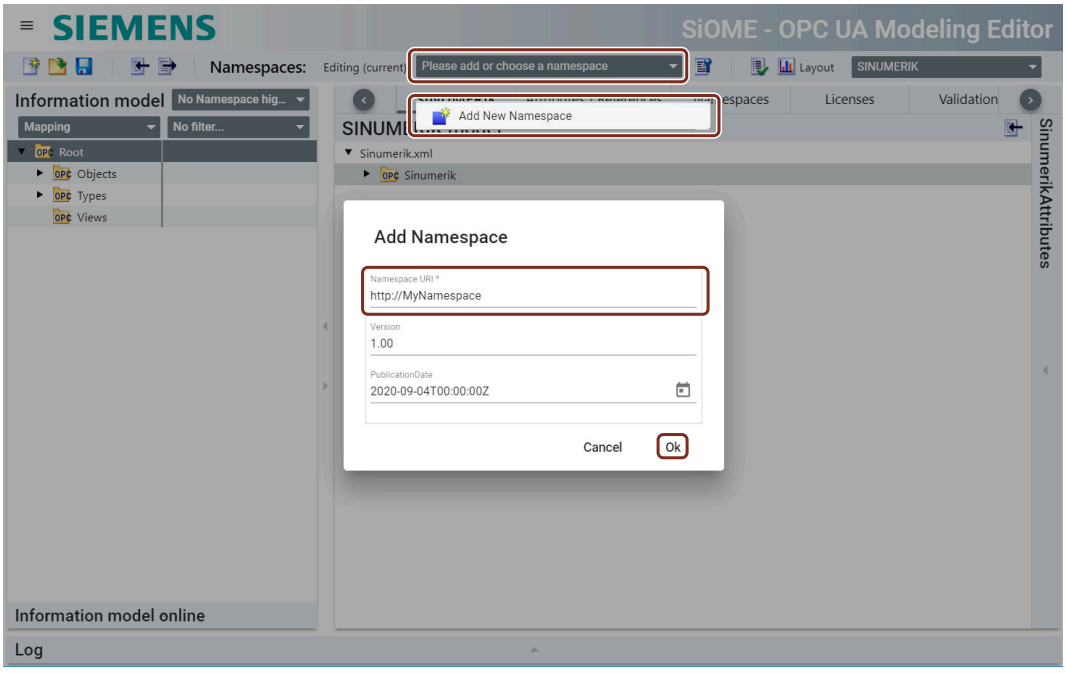


Figure 5-29 Namespaces tab

3. The new namespace is added in "Namespaces" tab. Right-click on "Objects" node in "Information model" tab and then click on "Add Instance".

4. The "Add Instance" popup window appears. Enter a name in the "Name" field.

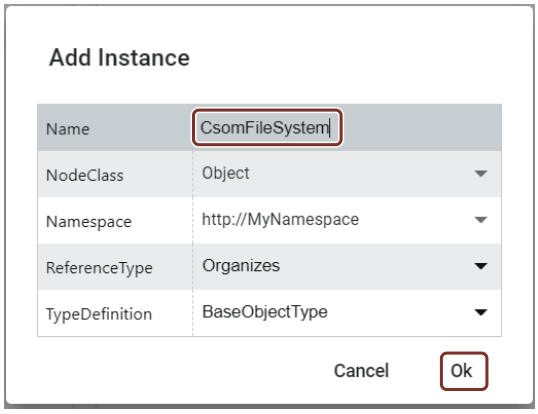


Figure 5-30 Add Instance popup window

5. Verify that newly added namespace is selected in the "Namespace" row and then click "Ok". The new instance is added under "Objects" node.

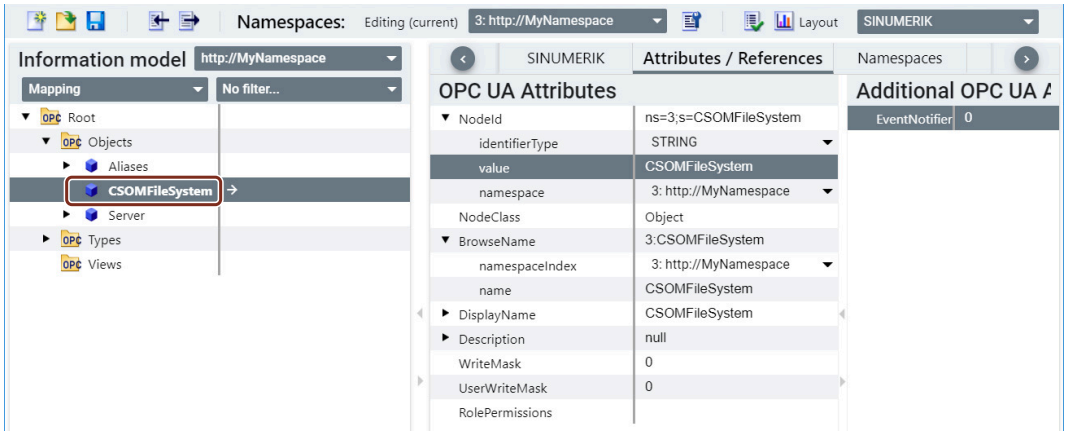


Figure 5-31 New instance is added under object node

Adding a SINUMERIK file directory under file system node

SINUMERIK file directory node can be added in two ways:

- Direct drag and drop under parent node.
- Manually adding a new node of folder type/file directory type under parent node.

Direct drag and drop under parent node

1. To add a SINUMERIK file directory under custom object, press and hold the **ctrl** button and drag and drop the required SINUMERIK file directory from the "SINUMERIK" tab to the "CsomFileSystem" node as shown in the below image.

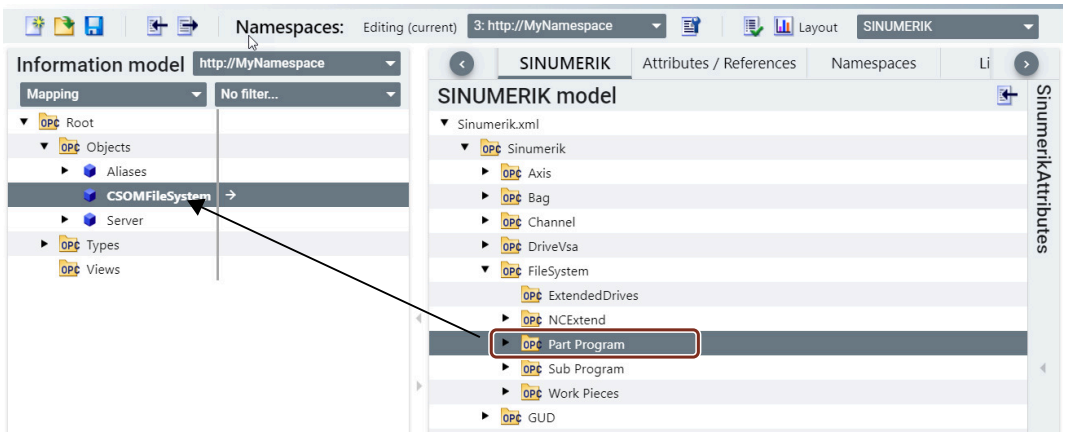


Figure 5-32 Direct drag and drop under custom object

2. The SINUMERIK file directory is added under parent node.

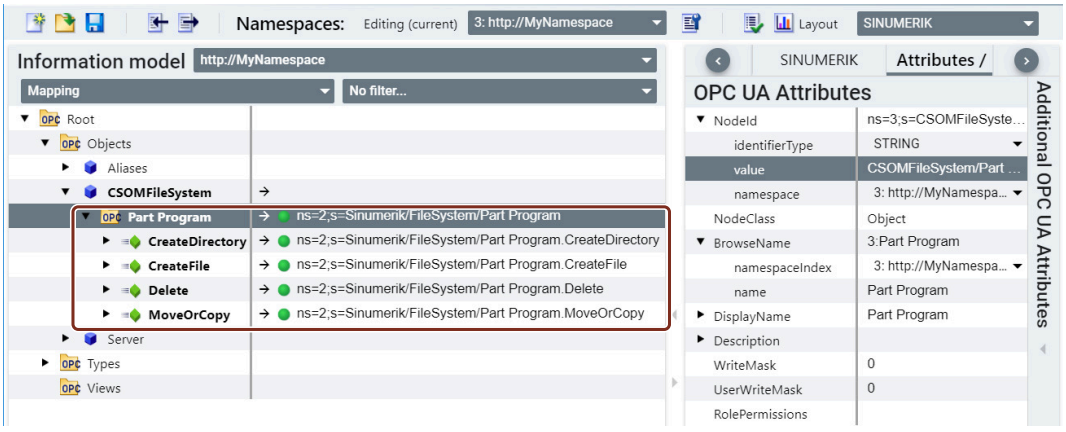


Figure 5-33 SINUMERIK file directory added under parent node

Note

This method is also applicable if you want to add the complete SINUMERIK file system node. See the below image for more information.

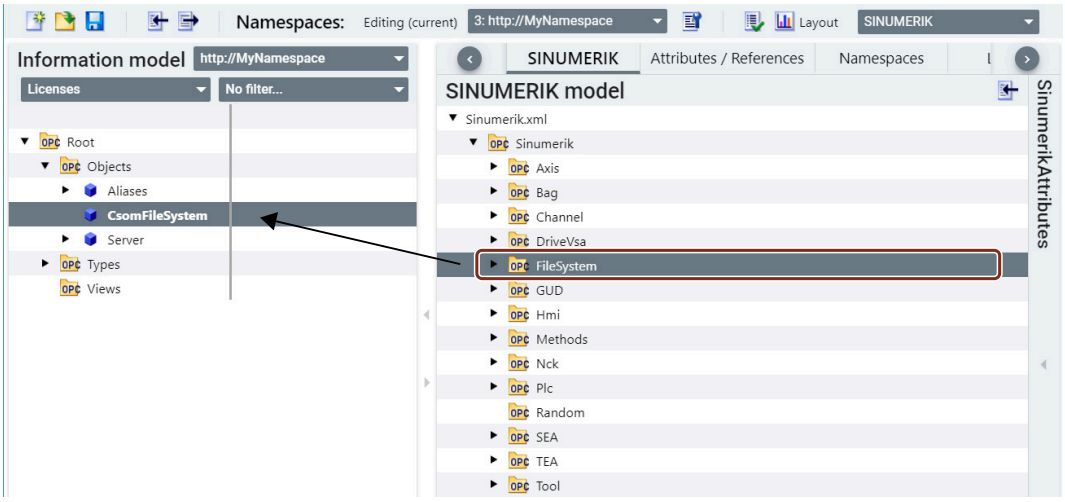


Figure 5-34 Adding complete SINUMERIK file system node

Manually adding a new node of folder type/file directory type under parent node

1. Under "Objects" node, right-click on "CsomfileSystem" node and then click on "Add Instance".
2. The "Add Instance" popup window appears. Enter a name in the "Name" field.
3. Select "FolderType" or "FileDirectoryType" from the "TypeDefinition" drop-down list and then click "Ok".

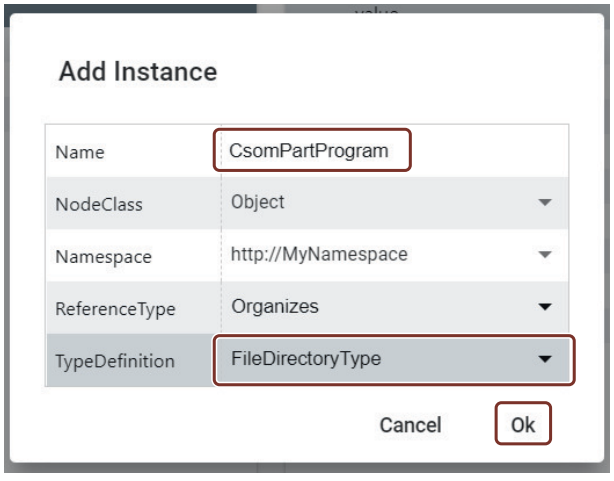


Figure 5-35 Add instance popup_File System

4. The new instance (here "CsomPartProgram") is added under "CsomfileSystem" node.

Note

Under file system node, except extended drives, all other nodes of file system should be of file directory type.

5. Drag and drop the required SINUMERIK file system directory from the "SINUMERIK" tab to the "CsomPartProgram" node as shown in the below image.

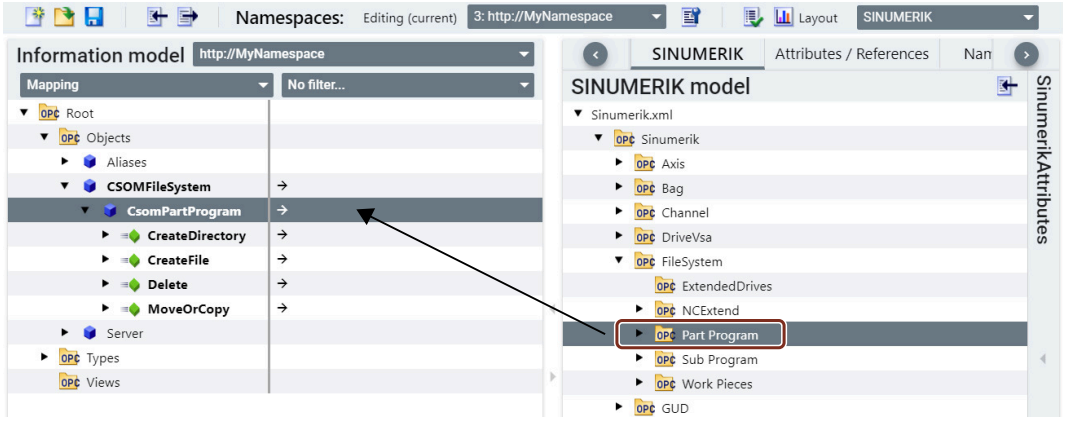


Figure 5-36 Adding a new node of file directory type under custom object

6. The "CsomPartProgram" node is mapped in the mapping table.

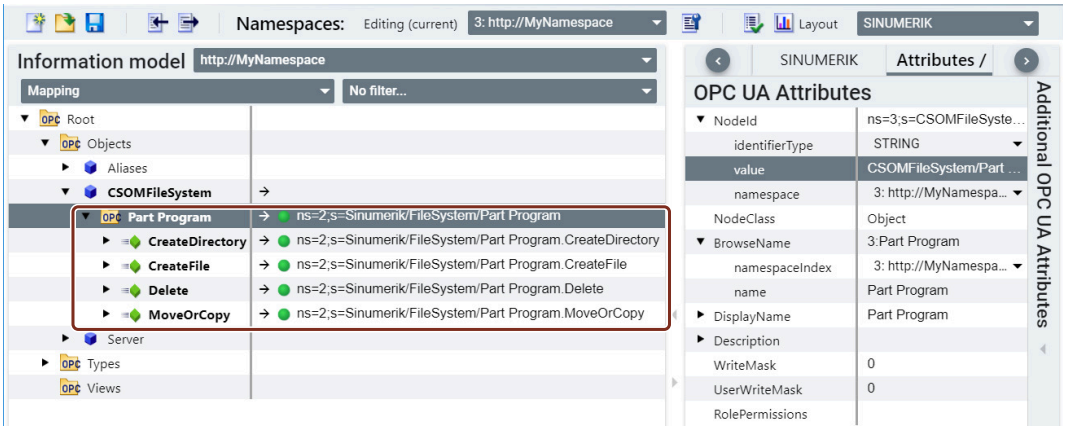


Figure 5-37 SINUMERIK file directory added under custom object

5.3.3.4 Option management in SiOME

Overview

License management is used to enable the availability of nodes within an existing information model based on licenses. These licenses can be used for machine models based on the machine manufacturers need or end user requirements. Any SINUMERIK variable (PLC/NCK) variable can be used by the machine manufacturer to enable the license functionality based on machine model. With the license's functionality user will have a flexibility of displaying different variables and methods according to the license set. Therefore, the SINUMERIK OPC UA Server supports two new use cases in conjunction with a CSOM:

- Individualization of the information model for modular machines on runtime level.
- Allowing a machine manufacturer to make information available within the OPC UA Server at runtime depending on whether a customer has bought a license or not.

This is basically done by engineering licenses within SiOME. To follow an easy to implement solution a "license" is basically any NC / PLC variable (can be chosen freely) in the SINUMERIK system which needs to have a certain value (can be defined in SiOME).

Following are the benefits with license functionality:

- Flexibility to use a single custom model for different machine models
- Flexibility to use a single custom model by OEM for different users
- Flexibility to add the required data set as part of custom model
- Reducing time for both commissioning and creating different models by OEM
- Lot of time can be saved while modeling and maintenance

Adding a license

To enable license options in custom models, follow the below steps:

1. Click "Layout" drop down and select "SINUMERIK". The SINUMERIK layout is displayed.
2. Under "Licenses" tab, click on **+** icon to add licenses, as shown in the below image.

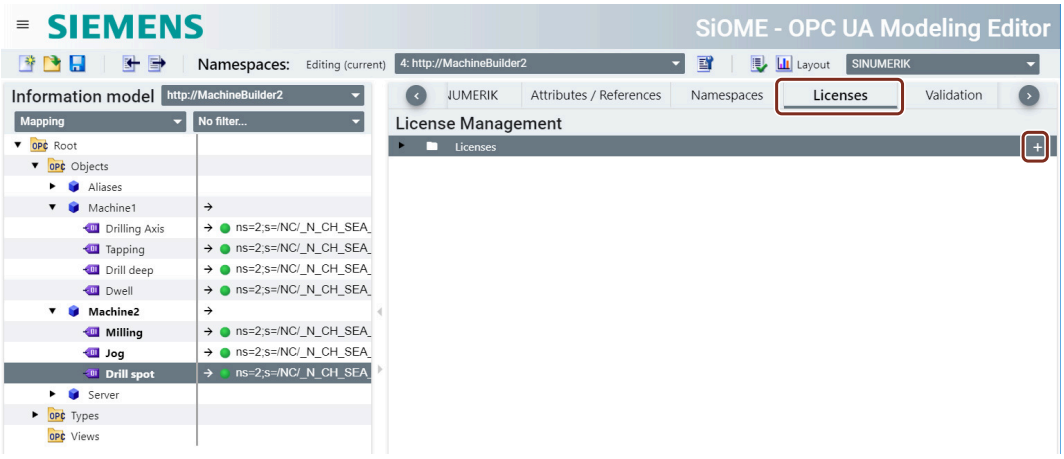


Figure 5-38 Adding licenses in custom models

3. An information popup window appears. By clicking "Ok", a new namespace will be created automatically for SINUMERIK license.

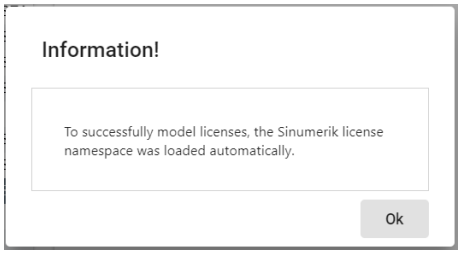


Figure 5-39 Licenses namespace popup window

Note

The SINUMERIK license namespace will get the next available namespace number.

4. The "Add license" popup window appears. Make the necessary settings as described in the below table.

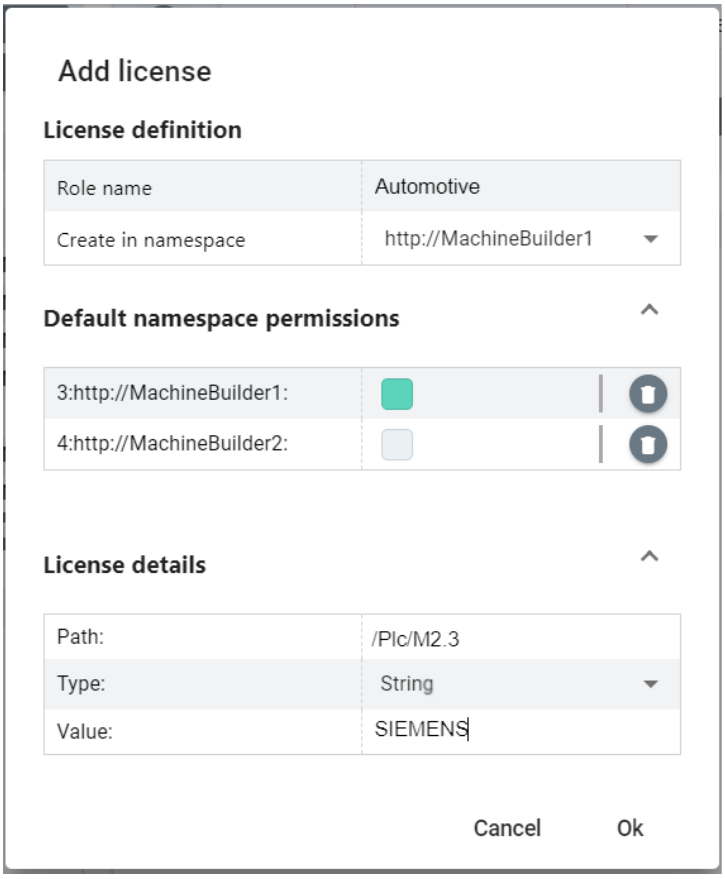


Figure 5-40 Add license popup

Group	Settings	Description	Example
License definition	**Role name**	You can edit the name of the license.	
	Create in name-space	Select the namespace for which you want to add the license from the drop down.	
Default namespace permissions		Select the default namespace. Sets the default license to the selected namespace. If a namespace is selected here, all variables of this namespace will be part of this license by default. If this is not required, then do not select any-thing. It is also possible to deselect the name-space by clicking again on the selection field. Green color indicates, selected and White color indicates, not selected. To remove the selected namespace, click icon.	

Group	Settings	Description	Example
License details	**Path**	Enter the path of the license variable. It is basically the path of the variable within the standard information model of the SINUMERIK node within the OPC UA Server.	Absolute addressing "/Plc/M2.3" Symbolic addressing "/Plc/Memory/activatelicense"
	Type	Select the data type from the drop down list.	String
	Value	Select or enter the value depending on the data type selection. If the variable has the same values as entered here, then the part of the OPC UA browse tree will be available in the OPC UA server, after restart.	"SIEMENS"

Note

The "Value" set for each license should match with the value in specified "Path" during the Operate/HMI startup. In case of mismatch in value, the variables/methods associated with the license will not be displayed in OPC UA address space.

5. Click "OK". The license is added, as shown in the below image.

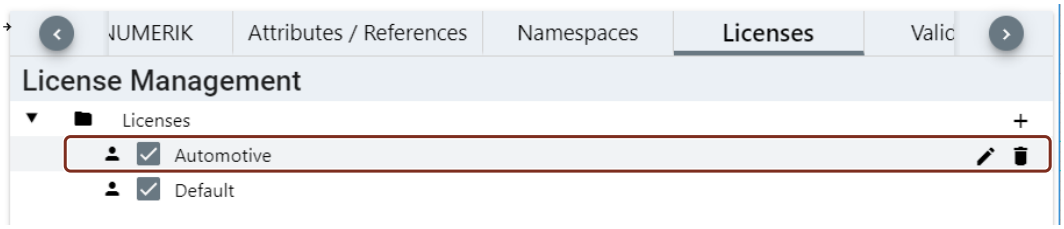


Figure 5-41 License added

If you want to edit the license details, then click icon. If you want to remove the license, then click icon. Add more licenses in the same way, if you need.

Note

SINUMERIK OPC UA server supports up to 100 licenses.

6. Select "Licenses" from the drop down, as shown in the below image.

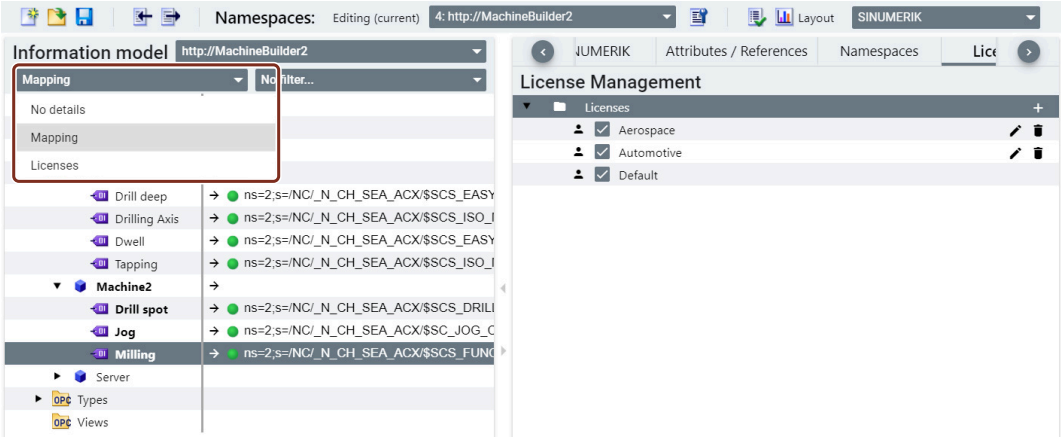


Figure 5-42 Switching from mapping to licenses

Setting licenses for CSOM namespaces

1. Assign the newly created licenses to the variables as per the requirement. The variables in the "Default" license will always be shown in the address space.

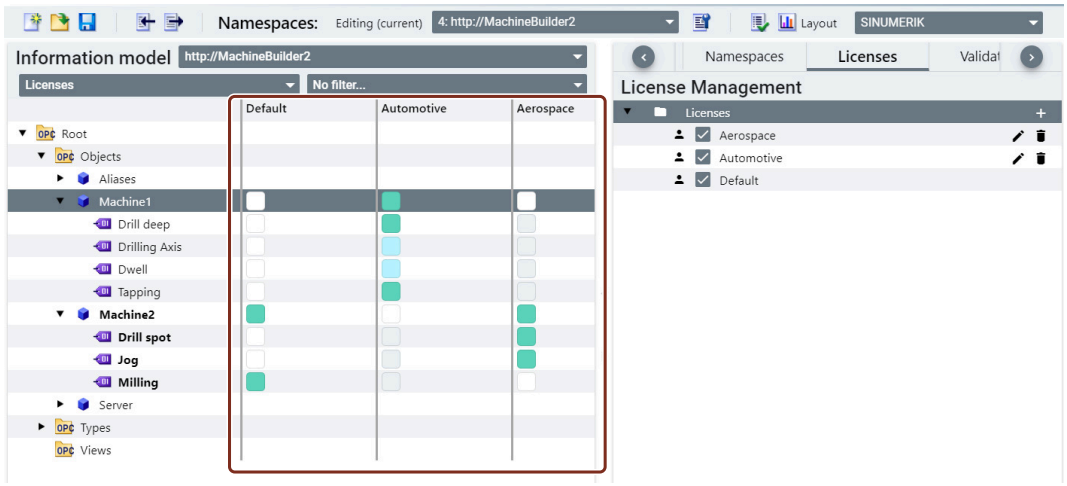


Figure 5-43 Assigning the licenses to variables

Note

After a license is set, restart of the OPC UA Server is required to update the browse tree.

After assigning the licenses to the variables, export the CSOM (XML) as mentioned in the next topic.

5.3.3.5 Exporting CSOM (XML)

To export the CSOM (XML) to your local machine, follow the below steps:

1. Click the Export XML icon as shown in the below image.

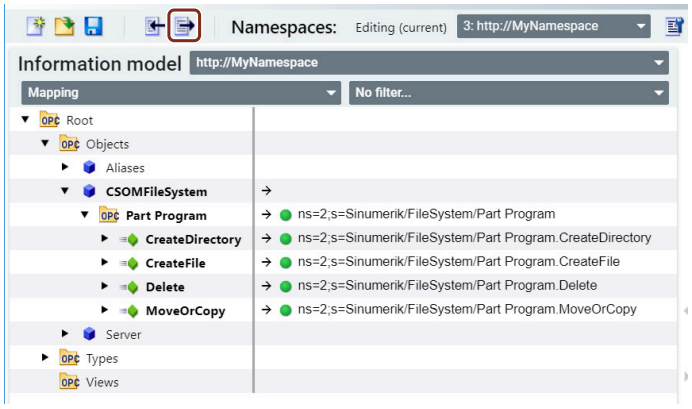


Figure 5-44 Exporting the CSOM

2. The "Export XML" popup appears. Click on browse icon.

3. A "Save as" dialog box appears.
   Select the location to save the exported CSOM (XML) file and then click "Save".

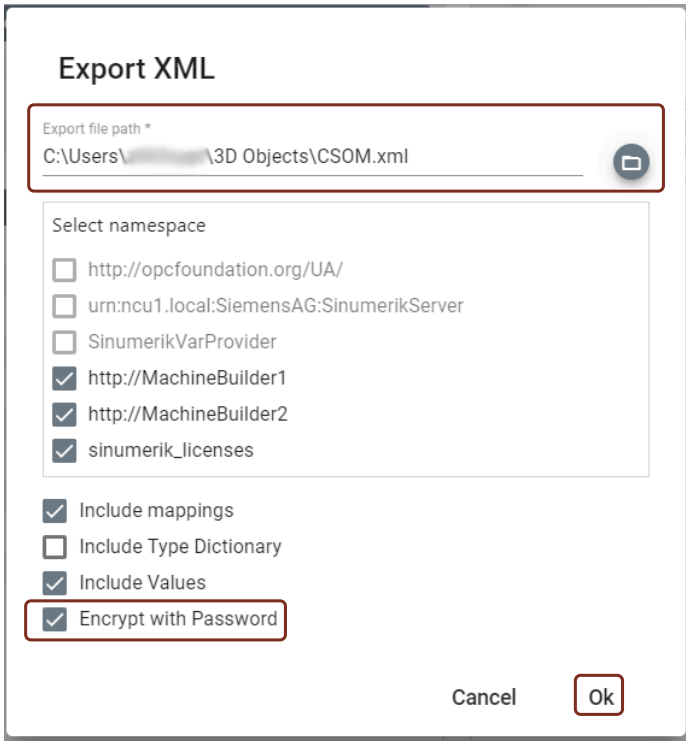


Figure 5-45 Export XML popup

4. If you want to encrypt the CSOM file, then select "Encrypt with Password" check box.
5. Click "OK". A password popup appears.

6. Enter a password. Also remember the password because it is required while converting the CSOM from XML to binary in SINUMERIK Access MyMachine/ P2P.

Note

Passwords must always contain a combination of upper-case and lower-case letters as well as at least one number and one special character. Password must comprise at least eight characters.

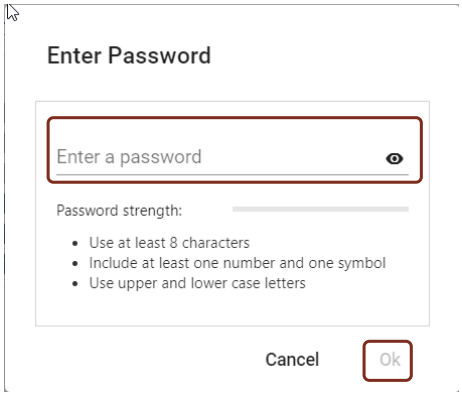


Figure 5-46 CSOM XML encrypt with password popup

7. Click "OK". The CSOM (XML) file is saved in the specified location.

For further steps, continue with the chapter Converting the CSOM from XML to binary (Page 89).

For a more detailed description of SiOME, refer to Industry Online Support (https://support.industry.siemens.com/cs/de/en/view/109755133). You can find a function manual, download link, further explanations and applications examples on modeling and mapping.

Additional information on data types, modeling rules and necessary user access rights for the modeling process are explained in the following chapters:

- Mapping data types (Page 82)
- Modeling rules (Page 88)
- Access control with CSOM (Page 88)

5.3.3.6 Mapping data types

Mapping SINUMERIK data types

The table below shows the compatible SINUMERIK data type for each OPC UA data type.

Assign the data types as shown below (SINUMERIK data type – OPC UA data type). Other assignments are not permitted. You are responsible for the rule-compliant selection and assignment of the data types.

You may find **further information** on mapping of OPC UA data types in the document "OPC UA Information Model for IEC 61131-3".

Table 5-1 Mapping of data types

SINUMERIK data type	OPC UA data type
Bool	Boolean
Character	Byte
Byte	Byte
Word	UInt16
Short Integer	SByte
Doubleword	UInt32
Long Integer	Int64
Float	Float
Real	Float
Double	Double
String	String

Note

The PLC datatype doubleword (e.g. DBD, MD) can be used for datatype “DInt” or “Real”.

For getting “Real” values via OPC UA it is necessary to append the “REAL” specifier to the PLC variable used in the mapping, e.g.: DB100.DBD5:REAL.

Note

PLC datatype "LTime"

The PLC datatype "LTime" cannot be used with SINUMERIK OPC UA server.

Mapping of SIMATIC data types

If you are working with the OPC UA server of S7-1500 CPUs, the table below shows the compatible SIMATIC data type for each OPC UA data type.

Absolute addressing

Table 5-2 Mapping of PLC data types - absolute addressing

Area	Address (IEC) "x" represents the byte offset; "y" the bit number in the byte and "z" the data block number	Permissible data types	OPC UA data type
Output image	Qx.y	BOOL	Boolean
Output image	QBx	BYTE,	Byte
		CHAR	Byte
		STRING	String

Area	Address (IEC) "x" represents the byte offset; "y" the bit number in the byte and "z" the data block number	Permissible data types	OPC UA data type
Output image	QWx	WORD	UInt16
		CHAR	Byte
		INT	Int16
		Date	UIn16
Output image	QDx	DWORD	UInt32
		DINT	Int32
		REAL	Float
		Time	Int32
Data block	DBz.DBXx.y	BOOL	Boolean
Data block	DBz.DBBx	BYTE	Byte
		CHAR	Byte
		STRING	String
Data block	DBz.DBWx	WORD	UInt16
		CHAR	Byte
		Date	UIn16
		INT	Int16
Data block	DBz.DBDx	DWORD	UInt32
		Time	Int32
		DINT	Int32
		REAL	Float
Input image	Ix.y	BOOL	Boolean
Input image	IBx	BYTE	Byte
		CHAR	Byte
		STRING	String
Input image	IWx	WORD	UInt16
		Date	UIn16
		CHAR	Byte
		INT	Int16
Input image	Idx	DWORD	Uint32
		DINT	Int32
		Time	Int32
		REAL	Float
Bit memory	Mx.y	BOOL	Boolean
Bit memory	MBx	BYTE	Byte
		CHAR	Byte
		STRING	String
Bit memory	MWx	WORD	Uint16
		CHAR	Byte
		INT	Int16

Area	Address (IEC) "x" represents the byte offset; "y" the bit number in the byte and "z" the data block number	Permissible data types	OPC UA data type
Bit memory	MDx	DWORD	UInt32
		DINT	Int32
		REAL	Float
PLC time	Clock		UInt16

Symbolic addressing

Table 5-3 Mapping of PLC data types - symbolic addressing

PLC data type	OPC UA built in data types
Bool	Boolean
Byte	Byte
Char	Byte
Wchar	UInt16
String	String
Wstring	String
Time	Int32
Time_Of_Day	UInt32
LTime	Int64
Ltime_Of_Day	UInt64
Date	UInt16
Date_And_Time	DateTime
Sint	SByte
USint	Byte
int	int16
Uint	UInt16
Dint	Int32
UDint	UInt32
Lint	Int64
Ulint	UInt64
Word	UInt16
Dword	UInt32
LWORD	UINT64
Real	Float
LReal	Double

See also

Function Manual S7-1500, ET 200MP, ET 200SP, ET 200AL, ET 200pro, ET 200eco PN Communication (https://support.industry.siemens.com/cs/ww/en/view/59192925/en)

5.3.3.7 Mapping of arrays

To map one-dimensional arrays in SiOME, perform the following settings:

Procedure

1. In the "Attributes/References" tab for the specific variable, set "ValueRank" to "1 Dimension".

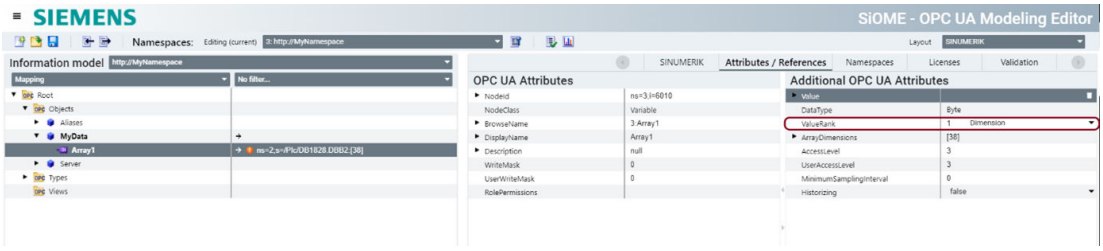


Figure 5-47 Mapping of arrays - ValueRank

2. On clicking "Value" (at the top of the "Attributes/References" tab) a new window appears. Set the first Dimension ([0]) as required (see screenshot [0] = [38]) and confirm with "OK".

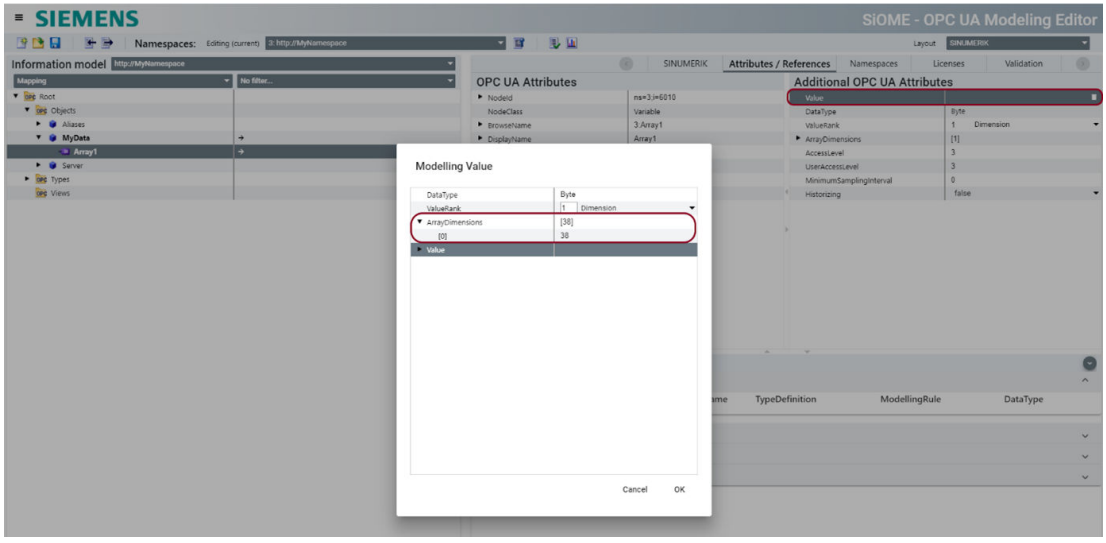


Figure 5-48 Mapping of arrays - Array dimension

3. Then map the node accordingly (taking into account the data type conversions).

Note

- Multi-dimensional arrays are not supported.
- For NC variables the below message appears. This message can be ignored.



5.3.3.8 Supported data types for sinumerik namespace

The following data types for sinumerik namespace are supported by OPC UA server:

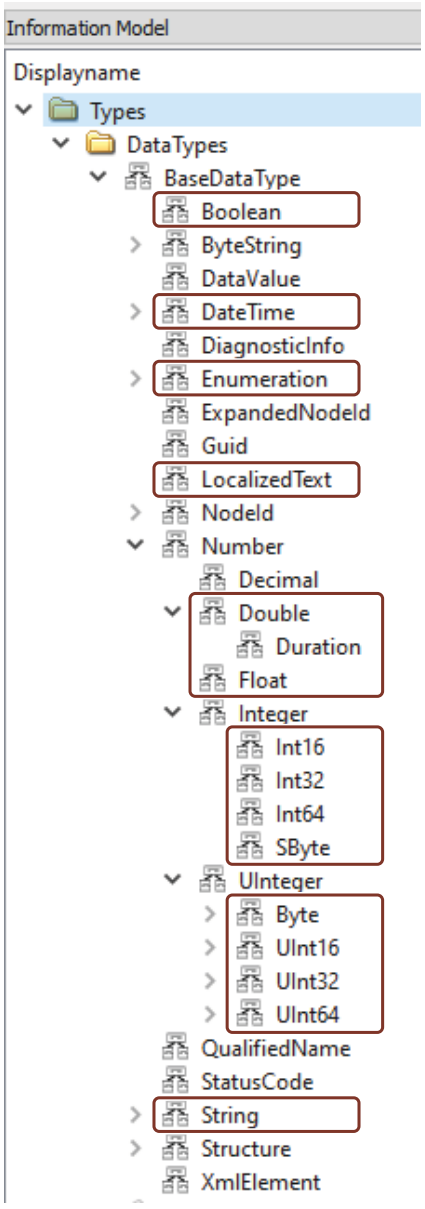


Figure 5-49 Supported datatypes for sinumerik namespace

5.3.3.9 Modeling rules

Following are the rules while modeling the exported SINUMERIK model (XML) with SiOME:

- The exported SINUMERIK Model (XML) should not be used in the "Information Model" window as a base for model creation.
- No new nodes should be added under the below namespaces:
  - 0: http://opcfoundation.org/UA/
  - 1: urn:ncu1.local:SiemensAG:SinumerikServer
  - 2: SinumerikVarProvider
- Up to 10.000 nodes (over all CSOM namespaces) can be modeled within the OPC UA server.
- Total namespaces should not be more than 15 (Including the above namespaces).
- First three namespaces 0, 1 and 2 should not be edited, deleted or change in order.
- Standard folder or file functions names should not be renamed.
- Standard folder or file functions argument should not be modified.
- No new Objects, Variables or Methods should be added under standard folder or file structures.
- Standard folder or files should not be deleted under standard folder or file structures.
- Additional namespaces should have an index of 3 or higher.
- Individual read/write rights can be given to all individual namespaces
- In SiOME, when a new node is created with a "ValueRank" set to "Dimension" then the value attribute of the node should be initialized by clicking on the "Value" in the "Attribute" window.

Note

If customer model is not shown in the browse tree, then refer to the error logs under the folder (..\user\sinumerik\hmi\log\opcua).

5.3.3.10 Access control with CSOM

For the CSOM, the customer has the following possibilities to provide access rights for variables:

1. In SiOME, the CSOM Access level can be defined.
2. In addition, the OPC UA administrator has four new access rights:
   - CsomReadx
   - CsomWritex
   - CsomReadAll
   - CsomWriteAll

To read a variable within a CSOM namespace the OPC UA administrator has to provide the user with the access right "CsomReadx" while "x" stands for the namespace URI.

For example, to read a variable out of namespace index "3", the user needs to get the access right by providing the corresponding namespace URI in namespace index "3". The same is valid for "CsomWritex".

To read all CSOM namespaces at once the OPC UA administrator has to provide the user with the access right "CsomReadAll".

To write all CSOM namespaces at once the OPC UA administrator has to provide the user with the access right "CsomWriteAll".

The following table gives an overview about the interaction between CSOM access level and access rights of target device. A variable that is read only by access rights of target device cannot be written by OPC UA, even if the OPC UA access write is read/write (3).

Table 5-4 Access table

CSOM AccessLevel	CSOM namespace rights	Rights of target device	Result
1 = Read	No access	Read / Write	No access
	Read	Read / Write	Read
	Write	Read / Write	No access
	Read / Write	Read / Write	Read
2 = Write	No access	Read / Write	No access
	Read	Read / Write	No access
	Write	Read	No access
	Write	Read / Write	Write
	Read / Write	Read	No access
	Read / Write	Read / Write	Write
3 = Read/Write	No access	Read / Write	No access
	Read	Read / Write	Read
	Write	Read	No access
	Write	Read / Write	Write
	Read / Write	Read	Read
	Read / Write	Read / Write	Read / Write

5.3.4 Converting the CSOM from XML to binary

The exported CSOM (XML) from SiOME needs to be converted to a binary format that can be read by SINUMERIK OPC UA server. The converted binary file has a more compact format and is therefore controller optimized. For this conversion process SINUMERIK Access MyMachine/ P2P is used.

The following graphic shows the process of converting the CSOM (XML) to binary format.

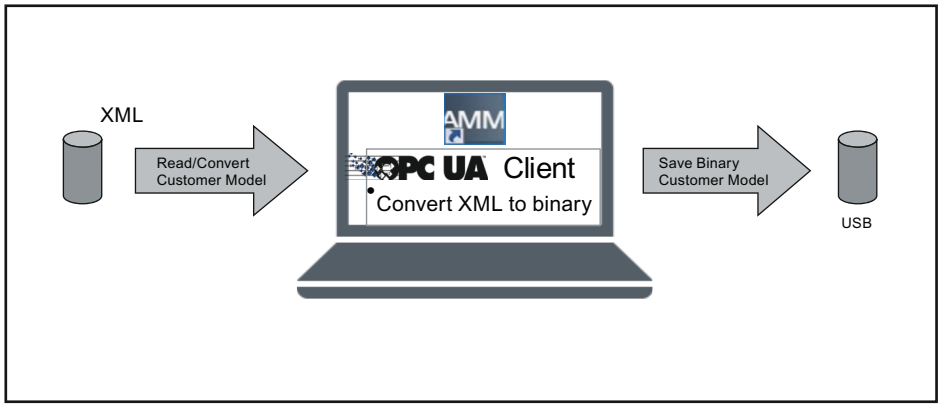


Figure 5-50 Converting the CSOM from XML to binary

Procedure

1. Open the application "SINUMERIK Access MyMachine /P2P".
2. Click "Tools > Sinumerik Opcua Server Tools > Model Binary Converter". A popup window appears.
3. Select the location of the XML file under "Input XML File" and specify the location to save the binary file under "Output BIN File".

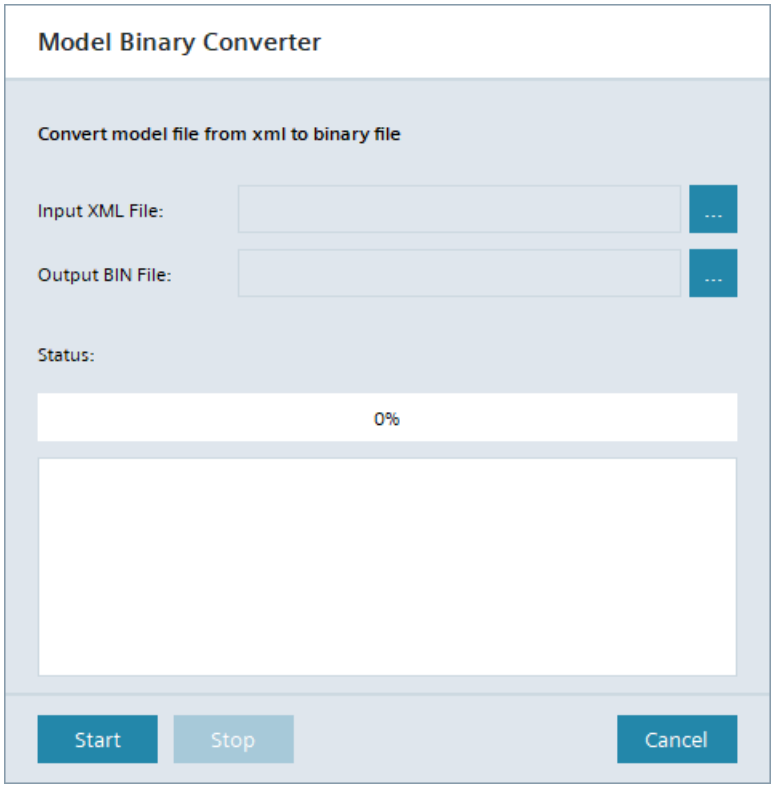


Figure 5-51 Model Binary Converter

4. Click "Start" button to generate the binary file. If the XML file is encrypted, then a password popup appears.

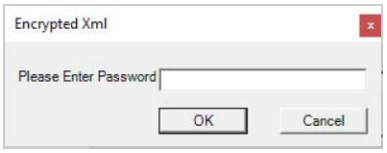


Figure 5-52 Password popup

5. Enter the same password that was set while encrypting the exported CSOM (XML) file from SiOME.
6. Click "OK". If the password is correct, then the binary file will be generated and saved in the specified location.

Result

Imported CSOM XML file is converted to binary format.

5.3.5 Importing the CSOM into the SINUMERIK OPC UA server

After converting the CSOM (XML) to a binary format, it can be imported to the SINUMERIK OPC UA server.

The following image shows the process of importing the CSOM (binary format) to SINUMERIK OPC UA server.

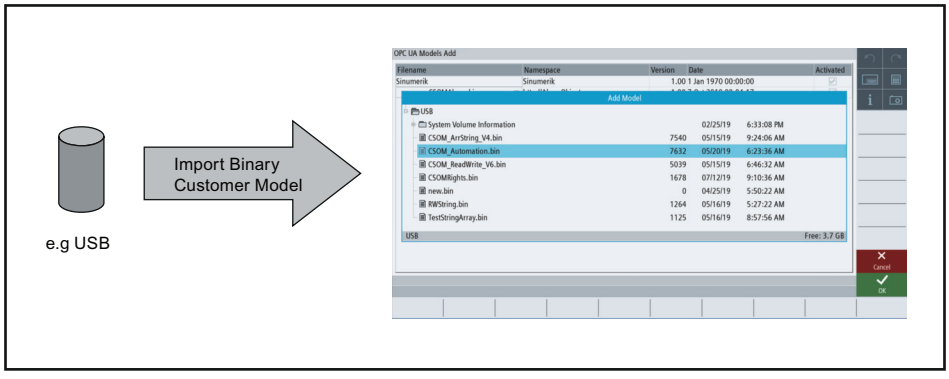


Figure 5-53 Importing the CSOM into the SINUMERIK OPC UA server

Procedure

1. Copy the binary file either to an USB/Networkshare or - alternatively - transfer the file via AMM directly to the control.
   - "**\user\sinumerik\hmi\opcua\models**" for NCU
   - "**C:\Program Files (x86)\Siemens\Motion Control\user\sinumerik\hmi\opcua\modeles**" for PCU/IPC

   Note

   If models folder is not present, then create a folder name **models.**
2. Press the softkey "Add Model" in the OPC UA model dialog and select the saved binary file.
3. Press the softkey "OK".
   The model is visible in the OPC UA model dialog.
4. Then press the softkey "Change" and activate the customer specific model.
5. Restart the OPC UA server by restarting SINUMERIK OPC UA server.

Result

The CSOM is now accessible in SINUMERIK Operate.

Note

For accessing data via an OPC UA client, it is necessary to have appropriate access rights (see chapter "List of access rights").

See also

List of access rights (Page 106)

5.4 CSOM dialog in SINUMERIK Operate

5.4.1 Overview

To have a comfortable way to work with a customer specific object model, the OPC UA dialog offers a special section, which can be found under the softkey "Model".

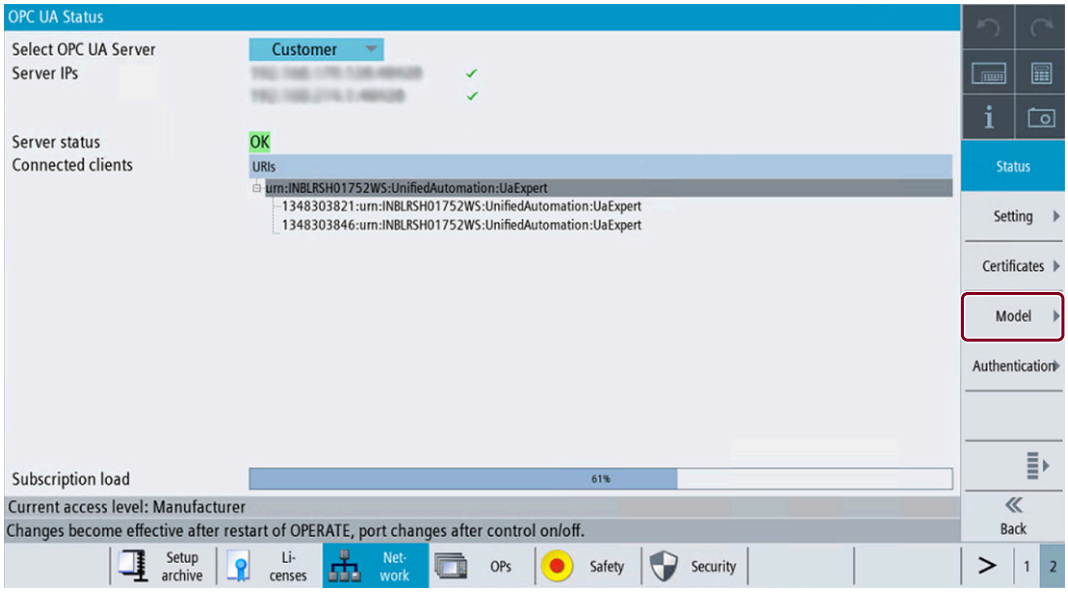


Figure 5-54 Softkey Model

The SINUMERIK Operate dialog has the following functionality:

- Import of CSOM binary file(s) from USB/Networkshare or via AMM
- View the filename of the selected binary file(s)
- Expand and view the namespaces of the selected binary file(s)
- Deleting the selected binary file(s)
- Activate the selected binary file(s)
- Deactivate the selected binary file(s)

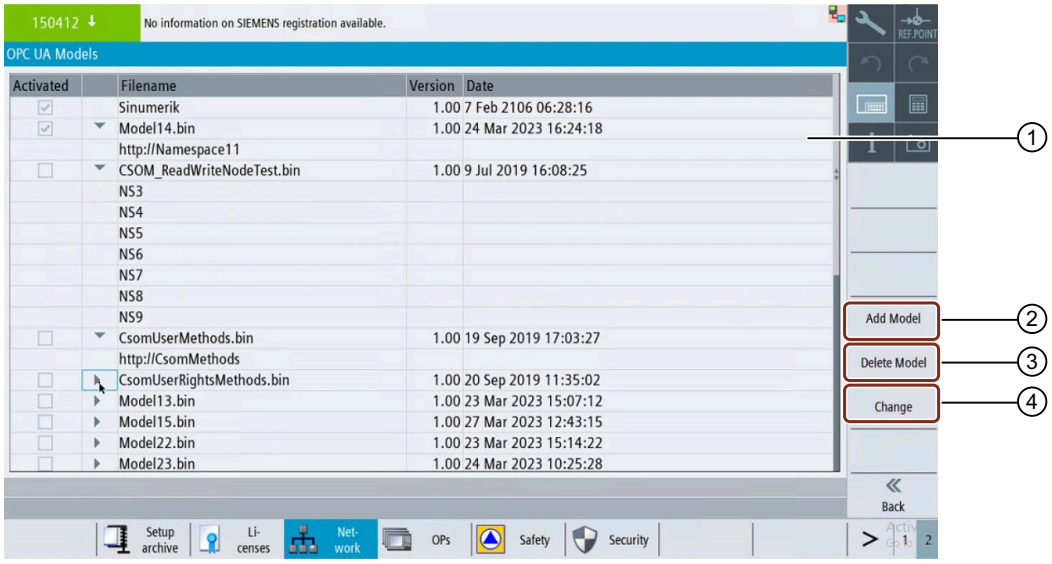
Note

The changes are visible only after restarting the OPC UA server.

5.4.2 OPC UA model dialog

Overview

Press the "Model" softkey to get the below screen and perform the following operations.



① The OPC UA model dialog shows the activated CSOMs and also displays the following information:

- Filename of the CSOM(s)
- The namespace(s) of the CSOM(s)
- The version of the CSOM(s)
- The date of the import
- Status of activation

② You can add one or multiple new CSOM(s)

③ You can delete a CSOM

④ You can change one or multiple CSOM(s): status can be activated/deactivated

Figure 5-55 OPC UA Models

5.4.3 Adding model(s)

Note

The "Add Model" option is visible with user access rights, but manufacturer's access rights are required to add models.

Procedure

1. To add one or multiple customer specific object model(s), press the softkey "Add Model".

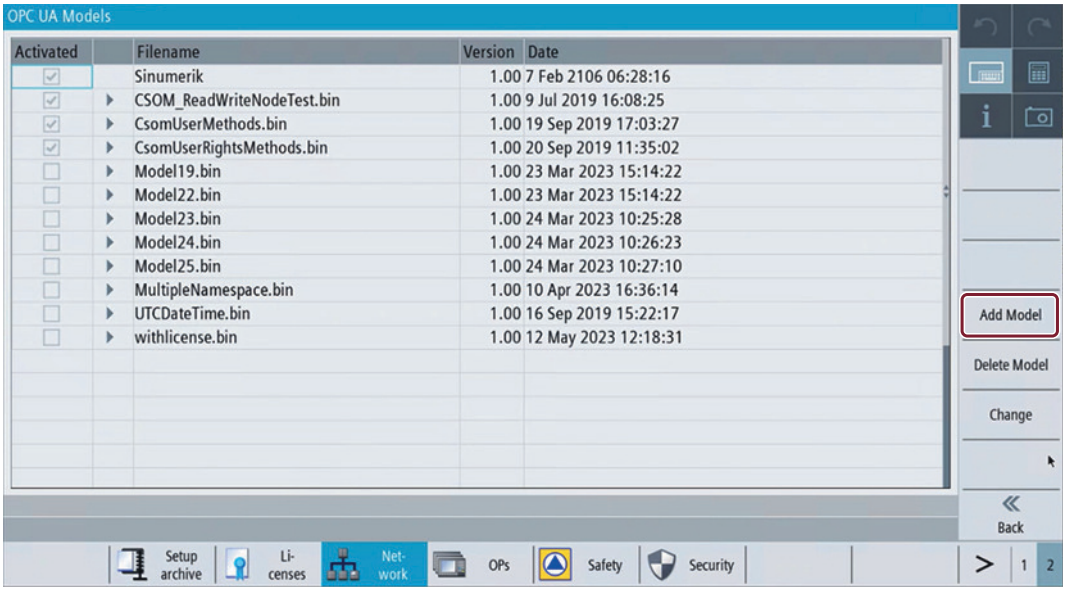


Figure 5-56 Softkey Add Model

The "Add Model" popup screen appears.

2. Select the binary file(s) from the SINUMERIK file system.

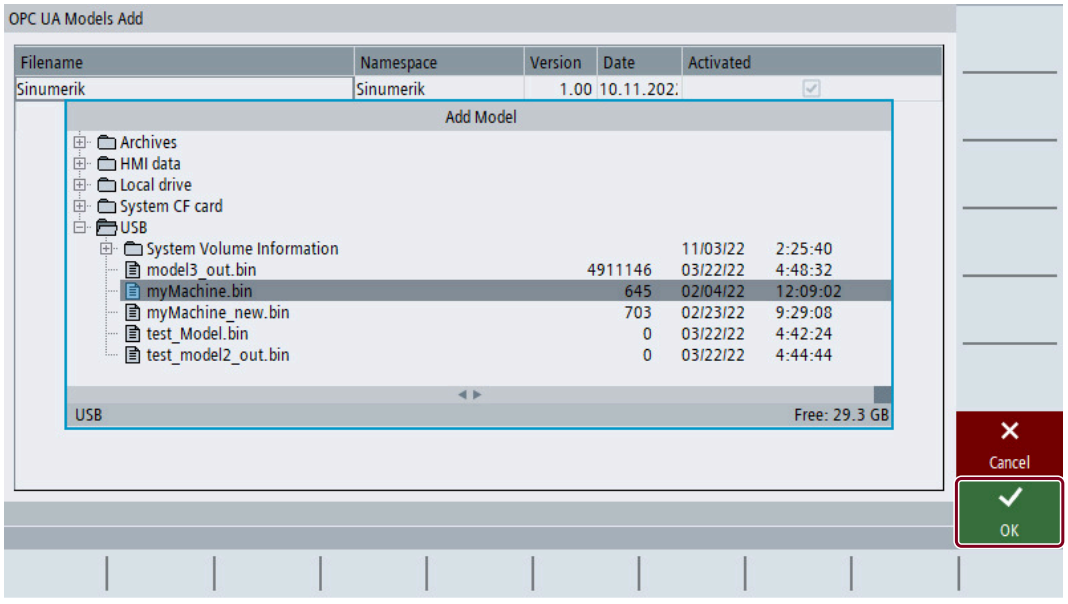


Figure 5-57 OPC UA Models Add

3. Press softkey "OK" to add the selected binary file(s).
   Press softkey "Cancel" to return to "OPC UA Models" screen.

5.4.4 Deleting OPC UA model

Procedure

1. To delete one customer specific object model, press the softkey "Delete Model".

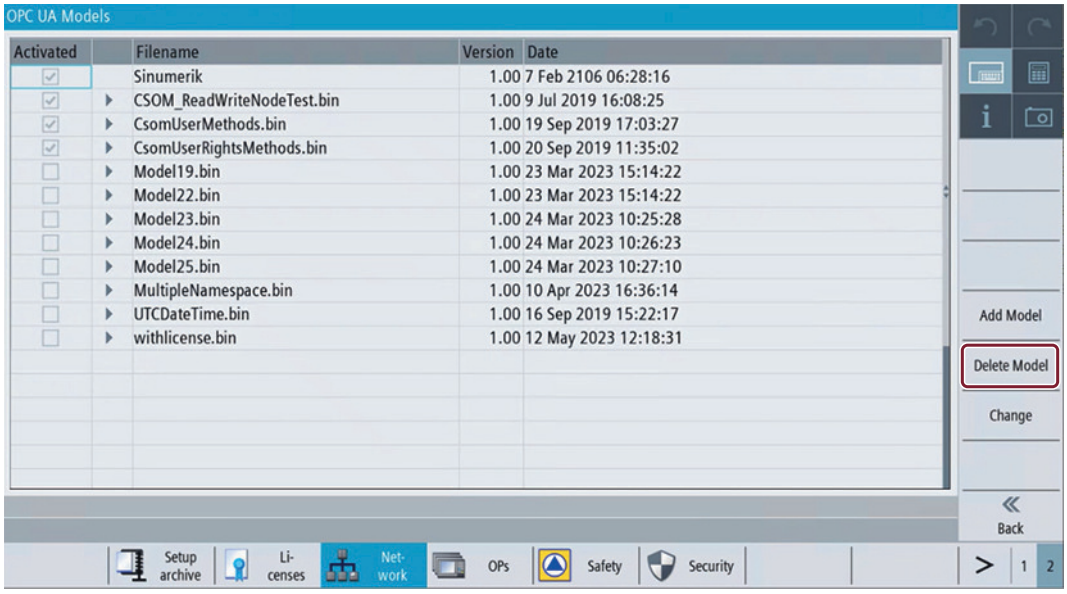


Figure 5-58 Softkey Delete Model

2. Select the model you want to delete from the displayed list and press softkey "OK".

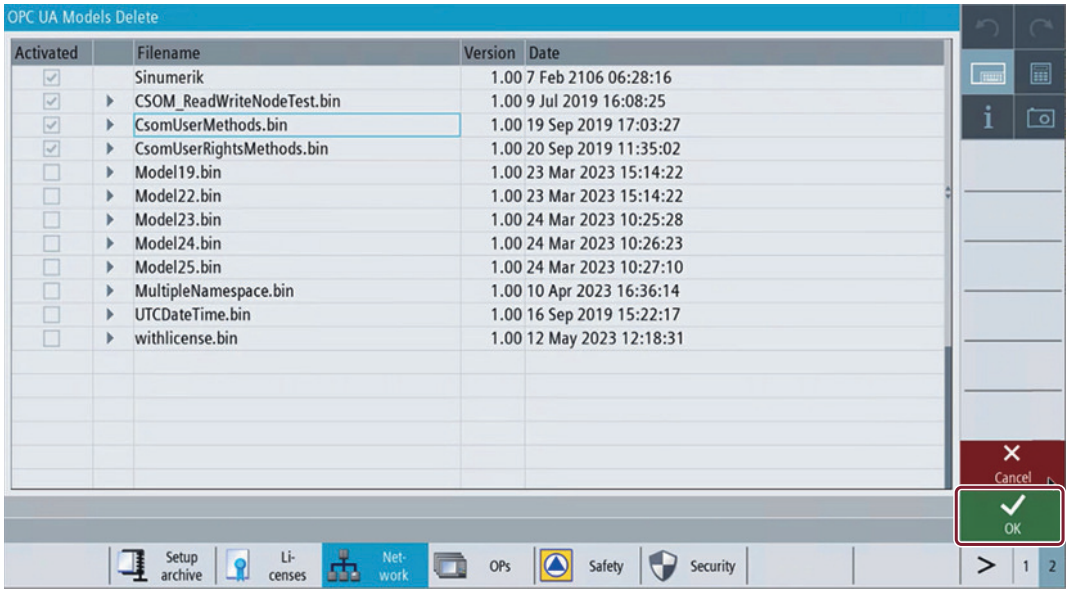


Figure 5-59 OPC UA Models Delete

3. A pop-up screen is displayed asking to confirm the delete action:

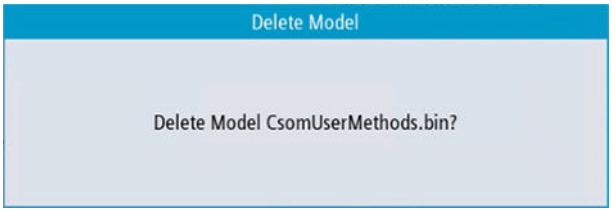


Figure 5-60 Delete Model popup

Press softkey "OK" to delete the model.
Press softkey "Cancel" to return to the "OPC UA Models" screen.

5.4.5 Activating / Deactivating OPC UA model(s) and SINUMERIK namespace(s)

Procedure

1. Press the softkey "Change".

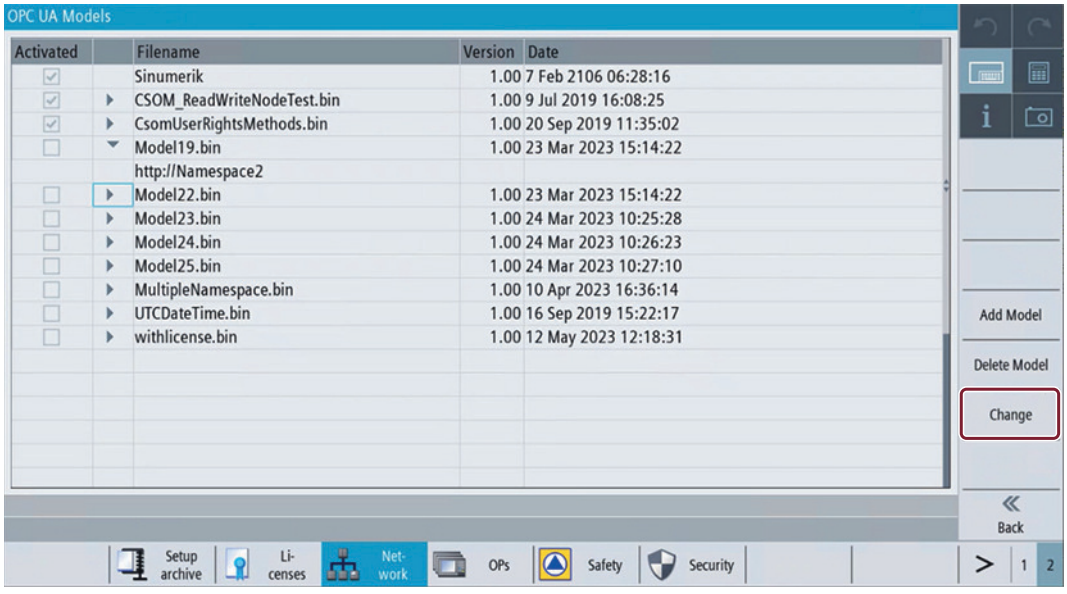


Figure 5-61 Softkey Change

2. The following screen appears where you can perform the following functions:
   - Selecting one or multiple CSOM file(s) from the list.
   - Activating one or multiple CSOM file(s) by selecting the check box.
   - Deactivating one or multiple CSOM file(s) by clearing the check box.

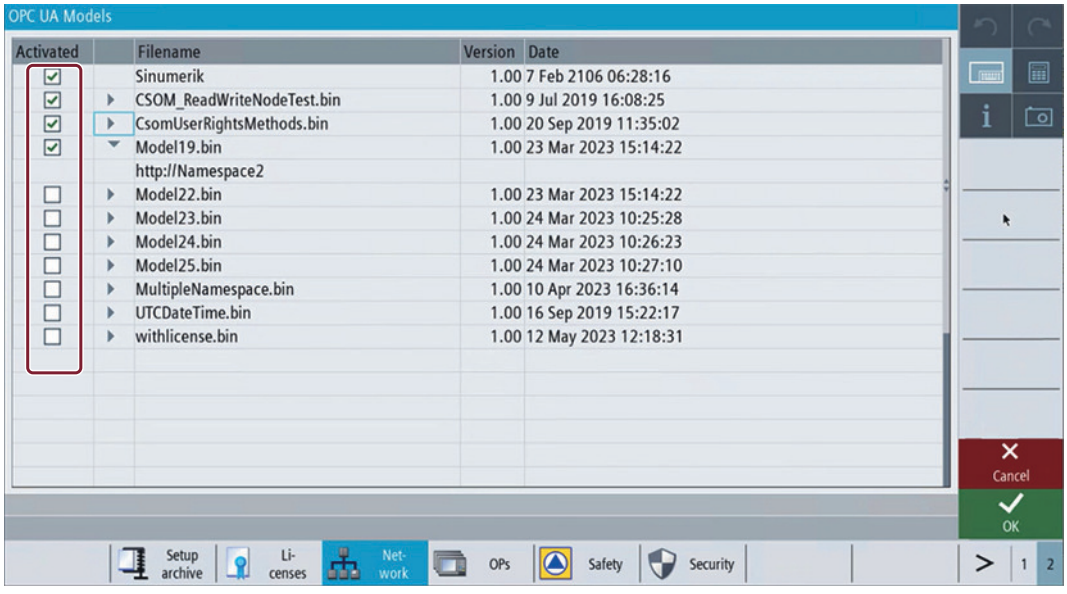


Figure 5-62 OPC UA Models Select

Pressing the softkey "Cancel" will do no action and return to "OPC UA Models" screen.

Pressing the softkey "OK" will apply the changes.

Activation with Namespace URI

The conditons for activating the CSOM files are described in the following table:

Condition	Description
Activating multiple CSOM files with same namespace	Activating multiple binary files is not possible for CSOM files which have same namespace URI, A popup message displays "Selected model file contains the same Namespace URI as in <filename.bin>. By clicking "OK", the currently activated file will be deactivated.
Namespace count limit reached	Maximum number of namespaces that can be activated cannot exceed 12.
Max. bin files that can be activated	Maximum number of CSOM files that can be activated is 10.
Max. bin files that can be loaded	Maximum number of CSOM files that can be loaded is 25.
Activating with namespace URI	With new installations of OPC UA server version 4.1, only access rights with namespace URI is allowed.

Note

Namespaces can be viewed as mentioned below:

- Namespace URI of a CSOM file is visible by expanding the CSOM file in the OPC UA models screen.
- Namespace URI can be viewed in the SIOME application Layout "SINUMERIK > Namespaces Tab".

Activating CSOM files created in OPC UA versions <= 4.0

When a binary file created in OPC UA version <= 4.0 is loaded in OPC UA version 4.1, the following is the behavior they exhibit:

- The namespace indexing works only if one CSOM file is activated. User can apply CsomReadx and CsomWritex rights, whereas "x" stands for the namespace index number (3-12).

The access rights changes from Namespace index to Namespace URI when any of the following operations are performed:

- If you remove the user access by invoking DeleteUserAccess method, then CsomReadx and CsomWritex access rights are removed.
- When you activate two or more CSOM files, the existing CsomReadx and CsomWritex rights are deleted. After the server restart, access rights can be given/deleted only with Namespace URI.

User administration

6

6.1 Overview

The admin can add/delete users and access rights via OPC UA methods provided by the server. Therefore a connection with a generic client must be established, using the admin credentials.

Users and access rights can then be assigned using the following OPC UA server methods:

- Add users (AddUser, AddCertificateUser)
- Delete users (DeleteUser)
- List users (GetUserList)
- Change password (ChangeMyPassword)
- Give access rights (GiveUserAccess)
- Remove access rights (DeleteUserAccess)
- List access rights (GetMyAccessRights, GetUserAccessRights)

NOTICE
Misuse of access rights As an administrator, you are fully responsible for the administration of users and their access rights. Any error in the administration process can lead to the misuse of access rights.

Note

Anonymous connection

You can also establish an anonymous connection during commissioning, if this setting is active, but the methods will not be available (feedback: "BadRequestNotAllowed").

Note

Anonymous user

Anonymous users don't have any access (Read/Write) rights after installation. As an administrator, you have to set these access rights explicitly.

Note

Administrator has only read rights

Note that the administrator has only read rights per default. Other rights need to be set explicitly.

Note

You can only add or remove users or access rights if you are connected as administrator. If you call the methods with a different user, you will receive the message "BadInvalidArgument".

6.2 User management

A new user created with the "AddUser" or "AddCertficateUser" function has no access rights at all. The user administrator has the responsibility for the user management and the associated access rights. All users must use a secure password.

Table 6-1 Methods for user administration

Method	Description	
AddUser	Creates a new user for accessing OPC UA. **Input arguments:**	
	UserName	User Name
	Initially, the password of the new user is the user name. It should then be changed using the method "ChangeMyPassword".	
AddCertificateUser	Creates a new user for accessing OPC UA via certificate authentication. **Input arguments:**	
	UserName	user, certificate is issued to
	CertficateData	Certificate(.der) as byte string
DeleteUser	Deletes a user who was added previously using the method "AddUser" and "AddCertificateUser". **Input arguments:**	
	UserName	User Name
	The administrator user, created when OPC UA was set up, cannot be deleted.	
GetUserList	The administrator can read the list of all users. **Input arguments:**	
	-	List of users
ChangeMyPassword	Changes the password for the **connected** user. **Input arguments**:	
	OldPwd	Current password
	NewPwd1	New password
	NewPwd2	New password (security prompt)
	Important! Whereas the methods "AddUser", "DeleteUser", "GiveUserAccess" and "DeleteUserAccess" can only be called up if the user is connected as the administrator, the user must connect as the corresponding user in order to change the password.	

6.3 Access rights management

After setting up the OPC UA components, the administrator user has read access to all data ("SinuReadAll") but no write access. These access rights must be set explicitly.

The administrator can also add the user access rights for individual PLC DBs.

Table 6-2 Methods for user administration

Method	Description	
GetMyAccessRights	The currently **connected** user can read his access rights. **Input Arguments:**	
	-	Rights
GetUserAccessRights	The administrator can read the access rights of another user. **Input Arguments:**	
	User name	Rights
DeleteUserAccess	Deletes the specified access rights for a user. **Input Arguments:**	
	User	A user whose access rights are to be deleted
	Realm	The access rights to be deleted as a string. If a user wants to delete several access rights, they must be separated by a semicolon.
For all possible realm strings, see chapter "List of access rights". Example: DeleteUserAccess("John","PlcReadDB100") Admin wants to delete read rights of user "John" for PLC data block DB100.		

Major access rights versus minor access rights

A possibility to reset all user access rights is to use the general OPC UA access rights behavior that by deleting a major access right all minor access rights will be deleted too. So if you have granted several read rights for special users to read certain data blocks beforehand, you can reset all of these access rights by deleting "SinuReadAll".

See also

List of access rights (Page 106)

6.4 Allow anonymous access

To enable anonymous access in OPC UA client, follow the procedure:

Note

Anonymous access must be used only when the system does not support user access. For security reasons it is not recommended to use anonymous access.

Procedure

1. From the OPC UA configuration dialogs (Page 30), activate the checkbox for "Message encryption None" and press "OK" to save the changes.

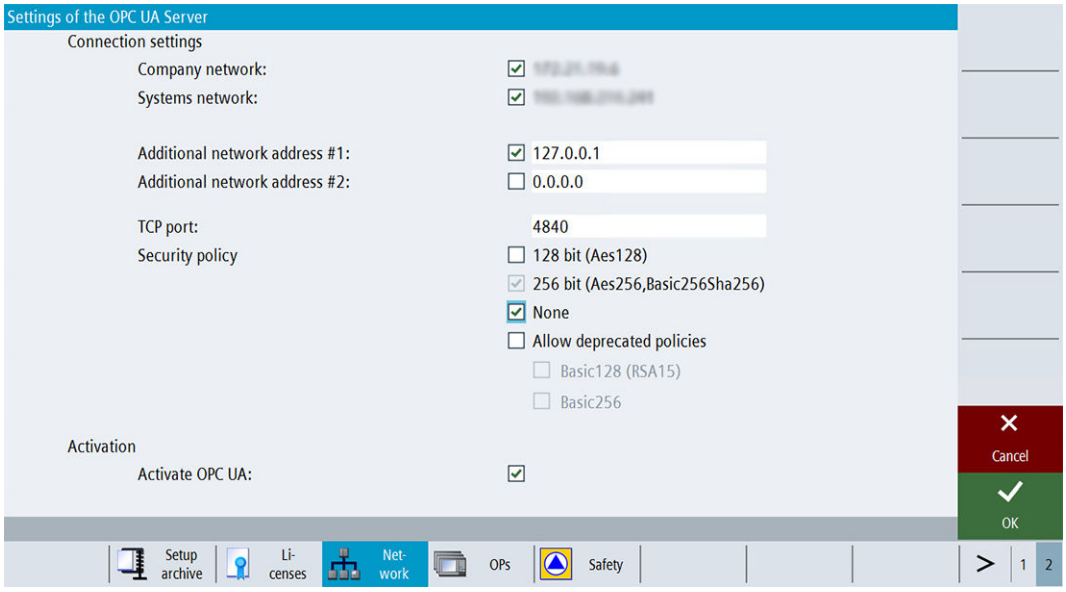


Figure 6-1 Settings of the OPC UA server: Message encryption None

2. Open the authentication settings by pressing the authentication softkey and activate the ckeckbox for "Allow anonymous access".

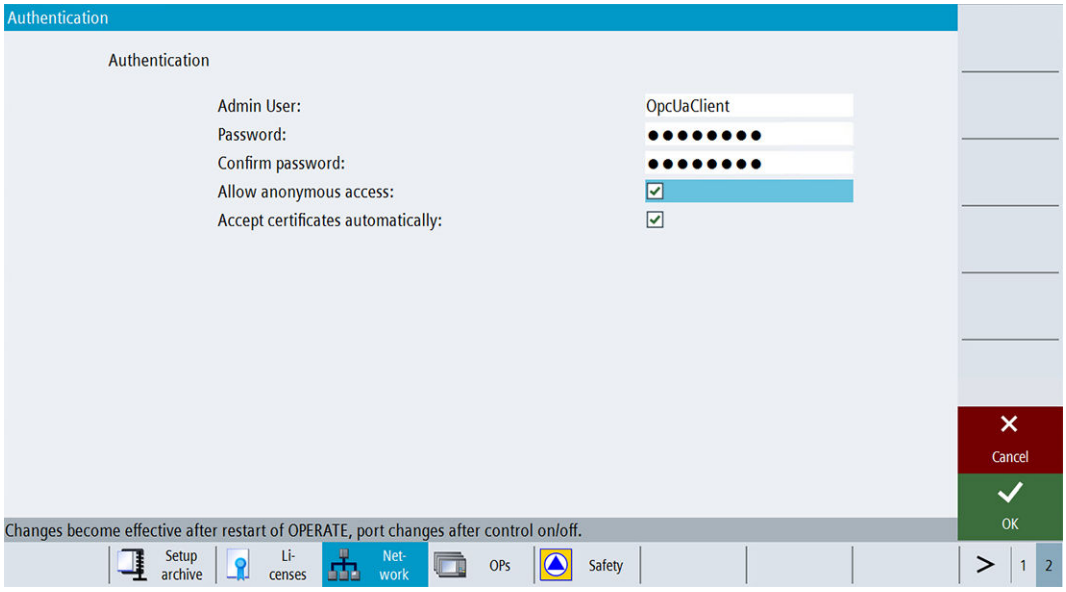


Figure 6-2 Authentication: Allow anonymous access

3. Press "OK" to save the configuration.
4. Start an OPC UA client and connect as OPC UA administrator.

5. Execute from browse tree "Objects > Sinumerik > Methods" the method "GiveUserAccess" and provide the following parameters (case sensitive):
   - User name: "anonymous"
   - User Rights: "SinuReadAll"

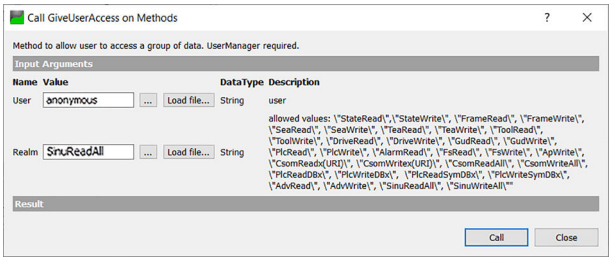


Figure 6-3 GiveUseAccess

6. Restart SINUMERIK Operate for the changes to reflect.

6.5 List of access rights

Below is the list of access rights a user is assigned:

Table 6-3 List of access rights

Method	Description	
GiveUserAccess	Sets the specified access rights for a user. The access rights below can be combined in any combination. **Input Arguments:**	
	User	User name which is to given the access rights
	Realm	The access rights to be set as a string. If a user wants to set several access rights, they must be separated by a semicolon.
	Some possible realm strings are:	
	"StateRead"	Status data - NC, channel, axis, read access
	"StateWrite"	Status data - NC, channel, axis, write access
	"FrameRead"	Zero offsets, read access
	"FrameWrite"	Zero offsets, write access
	"SeaRead"	Setting data, read access
	"SeaWrite"	Setting data, write access
	"TeaRead"	Machine data, read access
	"TeaWrite"	Machine data, write access
	"ToolRead"	Tool and magazine data, read access
	"ToolWrite"	Tool and magazine data, write access, Tool management methods
	"DriveRead"	Drive data, read access
	"DriveWrite"	Drive data, write access
	"GudRead"	User data, read access
	"GudWrite"	User data, write access
	"FsRead"	File system, read access
	"FsWrite"	File system, write access
	"PlcRead"	PLC, read access
	"PlcWrite"	PLC, write access
	"PlcReadSymDBx"	PLC, read access for symbolic address (x indicates the name of the symbolic address)[1]
	"PlcWriteSymDBx"	PLC, write access for symbolic address (x indicates the name of the symbolic address)[1]
	"AlarmRead"	Allows to subscribe to alarms
	"RandomRead"	Random, read access
	"RandomWrite"	Random, write access
	"SinuReadAll"	All of the read access operations mentioned
	"SinuWriteAll"	All of the write access operations mentioned
	"ApWrite"	Allows to call method "Select"
	"CsomReadx"	CSOM read access (x indicates the namespace URI)
	"CsomWritex"	CSOM write access (x indicates the namespace URI)
	"CSOMReadAll"	CSOM read access to read all CSOM namespaces

Method	Description	
	"CSOMWriteAll"	CSOM write access to write all CSOM namespaces
	"ADVRead"	Monitoring ADV events, Calling ADV request methods
	"ADVWrite"	Access for OPC UA Advanced methods which modify the system (write).
	Examples: • GiveUserAccess ("MyUser", "GudRead; PlcWrite") Sets the read access for user data for the "MyUser" user and sets the write access for the PLC. • GiveUserAccess ("John","PlcReadDB100") Admin gives read rights to user "John" for PLC data block DB100.	

1) If the symbolic DB is optimized then the access is given only to the symbolic DB address. If the symbolic DB is non-optimized then the access is given to both Symbolic DB address and absolute DB address. For example:

In case of optimised -- PlcReadSymDBSiemensBit

In case of non-optimised -- PlcReadSymDBSiemensBit + PlcReadDB256.

2) CSOM read and write access rights is sufficient to access the CSOM namespace. It overwrites all the other user rights. Therefore, no other additional rights are needed for reading or writing in CSOM address space (for example, PLC read access rights is not needed to read the PLC data in CSOM).

Note

CSOM access is not covered by SinuReadAll / SinuWriteAll, but must be assigned individually.

6.6 Changing access rights for OPC UA configuration screens in SINUMERIK Operate

It is possible for the machine manufacturer to change the access rights for the OPC UA server configuration screens according to the protection level of the configuration.

Therefore the machine builder can change the access level needed to change, for example, the authentication information for the admin or the CSOM within the OPC UA server.

The access rights for OPC UA configuration screens in SINUMERIK Operate can be adjusted in the following path:

- for NCU "*/card/user/sinumerik/hmi/opcua/cfg/opcuiaccess.conf*"
- for PCU/IPC "*C:\Program Files (x86)\Siemens\MotionControl/user/sinumerik/hmi/opcua/cfg/opcuiaccess.ini*"

Note

Bear in mind that changing access rights can allow the customer to change essential configurations of the OPC UA server, especially gaining access to variables that the machine manufacturer may not want to offer.
Access rights for the SINUMERIK Operate screens can only be changed with the manufacturer's access rights.

Procedure

1. Open the file "opcuiaccess.conf" on your embedded controller.
2. Change the access right for the desired OPC UA functionality. The access rights mirror the SINUMERIK Operate access levels from 1 to 7.

Access level	Protected by	Area
1	Password: SUNRISE (default value)	Manufacturer
2	Password: EVENING (default value)	Service
3	Password: CUSTOMER (default value)	User
4	Keyswitch 3	Programmer, machine setter
5	Keyswitch 2	Qualified operator
6	Keyswitch 1	Trained operator
7	Keyswitch 0	Semi-skilled operator

3. Save the file again.

6.7 OPC UA configuration folders - service mode access

On Windows platforms (PCU/IPC), it is recommended to restrict access to the OPC UA configuration folders (../**user**/sinumerik/hmi/opcua).

When any user (administrator or Operate user) login to PCU/IPC, then by default they get access to the OPC UA configuration folders.

To restrict the access of other users to the OPC UA configuration folder, the administrator should set the access rights accordingly. The administrator should remove other users access to the OPC UA configuration folders. To do so follow the below steps:

Procedure

NOTICE
OPC UA does not work Please note that this procedure only applies to "Users" and not to the "AUDUSER", since otherwise OPC UA does not work.

1. Right-click on the OPC UA configuration folder and then click "Properties". The "opcua Properties" popup window appears.
2. Click the "Security" tab.

3. To remove other users access to the OPC UA configuration folders, select the user and click "Advanced" as shown in the below image.

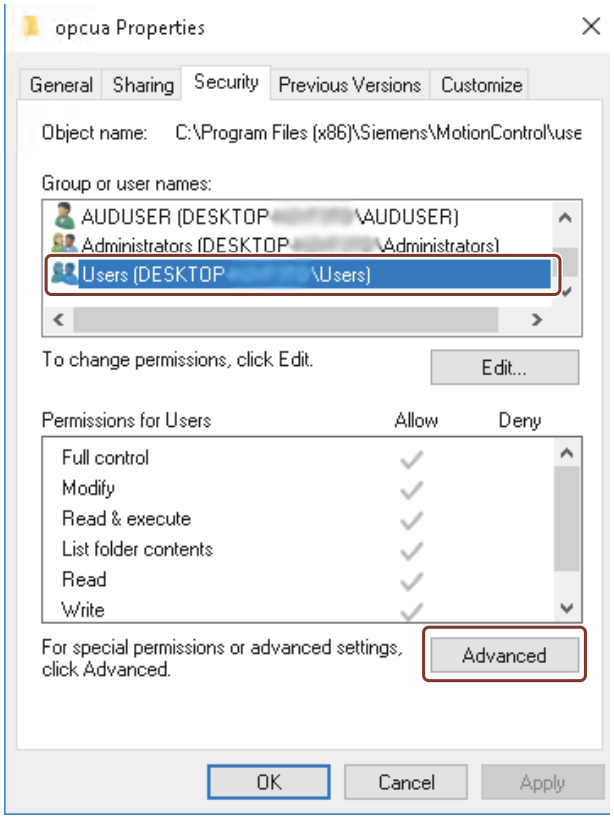


Figure 6-4 OPC UA configuration folder_Properties

4. The "Advanced security settings for opcua" popup window appears. Select the user and click "Disable inheritance" as shown in the below image.

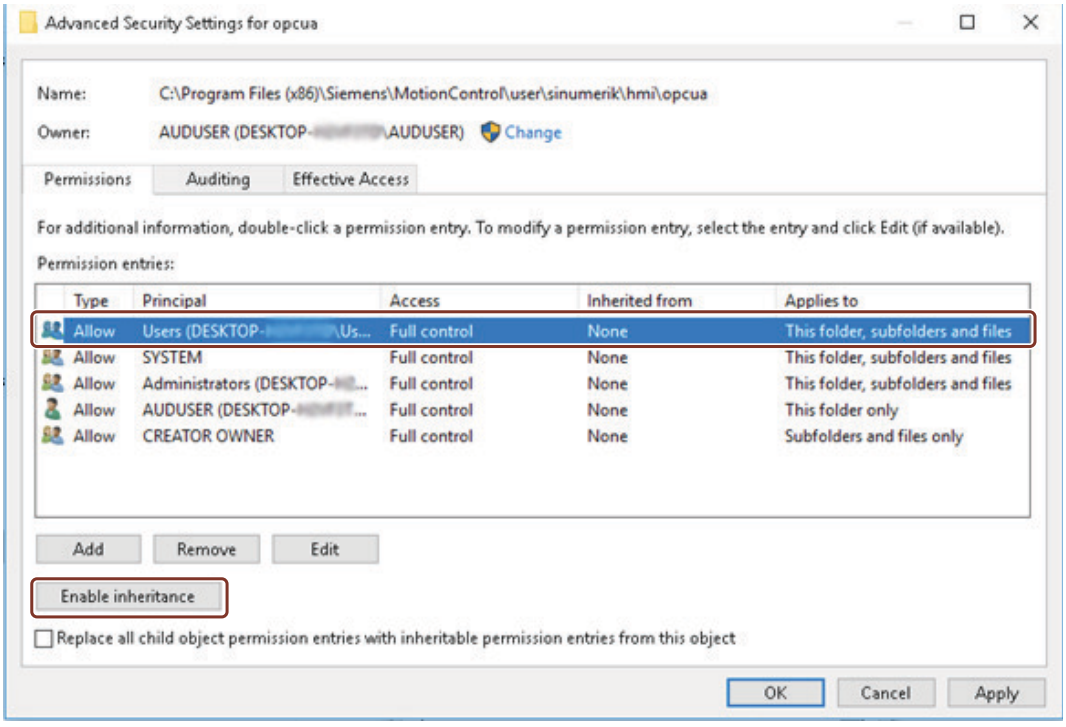


Figure 6-5 Advanced security settings for OPC UA configuration folder

5. The "Block Inheritance" popup window appears. Confirm by clicking "Convert inherited permissions into explicit permissions on this object".

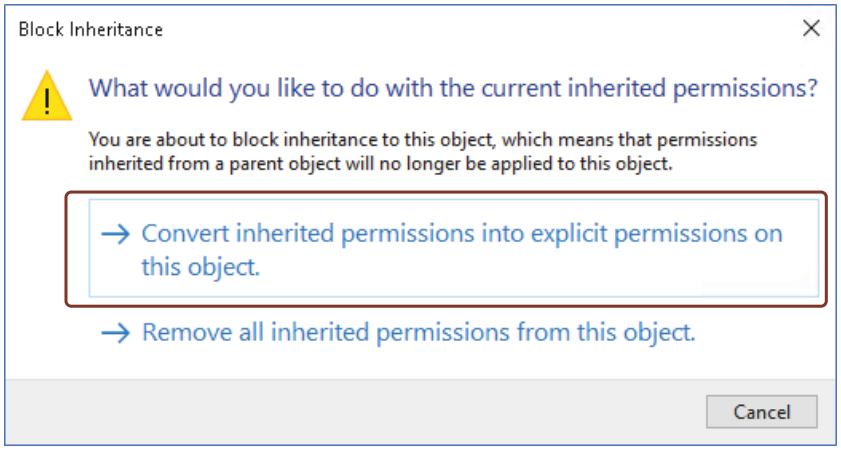


Figure 6-6 Block Inheritance for OPC UA configuration folder

6. Click "Remove" and then click "OK". The other user access to the OPC UA configuration folder is now removed as shown in the below image.

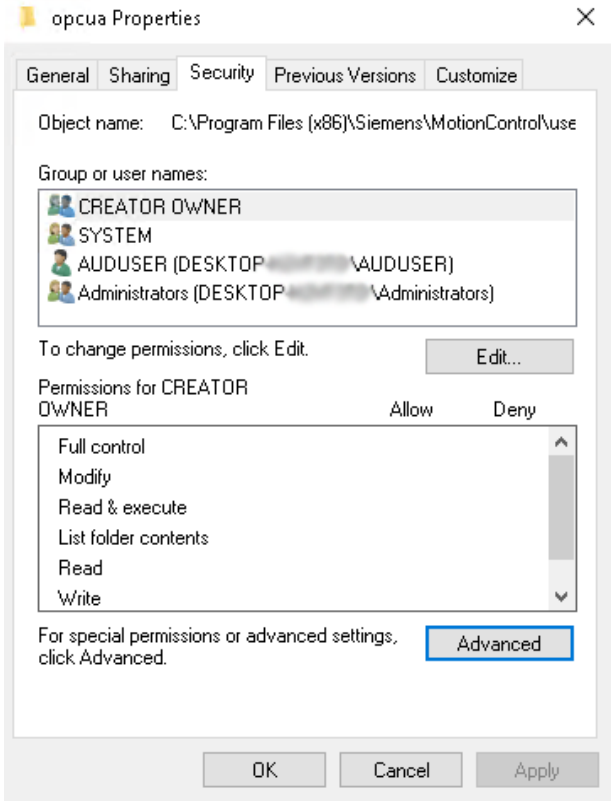


Figure 6-7 OPC UA configuration folder_Properties_user access removed

Functionality 7

7.1 Overview

Overview

The OPC UA server provides the possibility to communicate with SINUMERIK via OPC UA. The following functionalities of the OPC UA specification are supported by the server:

- **Data Access:**
  Read, write and subscribe to SINUMERIK variables (NC, PLC)
- **Alarms & Conditions:**

  Event based provision of SINUMERIK alarms and messages from HMI, NC and PLC
- **Methods:**
  User management, file transfer, tool management and program selection

This chapter describes the address space of the OPC UA server and gives further information how to address some SINUMERIK specific values. Especially since a lot of SINUMERIK values are stored in arrays or matrices.

Furthermore you can find description on the SINUMERIK alarm object and how to get the alarms from the server.

At the end of this chapter explanation on how users can transfer files from or to the server using the SINUMERIK file system.

7.2 Address space model

Address space model

If the OPC UA server is browsed, the available address space is mapped under the "Sinumerik" node.

Global User Data (GUD) can be found under the "/Sinumerik/GUD" node.

The PLC data can be found under the "/Sinumerik/Plc" node.

Machine data can be found under the node "/Sinumerik/TEA".

Setting data can be found under the node "/Sinumerik/SEA".

Observe the following while browsing:

- In the address space of the NC, the displayed variables always represent only the first parameter of the corresponding unit.
  Example:
  The R parameters can be found under "Sinumerik > Channel > Parameter > R". The corresponding identifier is called "/Channel/Parameter/R", which is finally mapped to "/Channel/Parameter/R[u1, 1]". If you want to access other parameters, you need to specify the corresponding index in brackets, e. g. "/Channel/Parameter/R[u2,56]".
- In the address space of the PLC, the displayed variables represent the access format that has to be extended accordingly. For accessing symbolic addresses use the variable path as shown in the OPC UA browse tree. For accessing absolute addresses the appropriate byte no. needs to be appended to the base address from the OPC UA browse tree.
  Example:
  The variable "/Plc/MB" is in the address space and is mapped to "/Plc/MB0". For accessing further bytes this variable must be extended by the appropriate byte number, e.g. "/Plc/MB6". For an example of symbolic PLC access, refer to chapter Variable paths for PLC access operations (Page 118).
- The address space of the NC also contains variables that are not available in a corresponding machine configuration. These variables return "BadAttributeIdInvalid" as value.

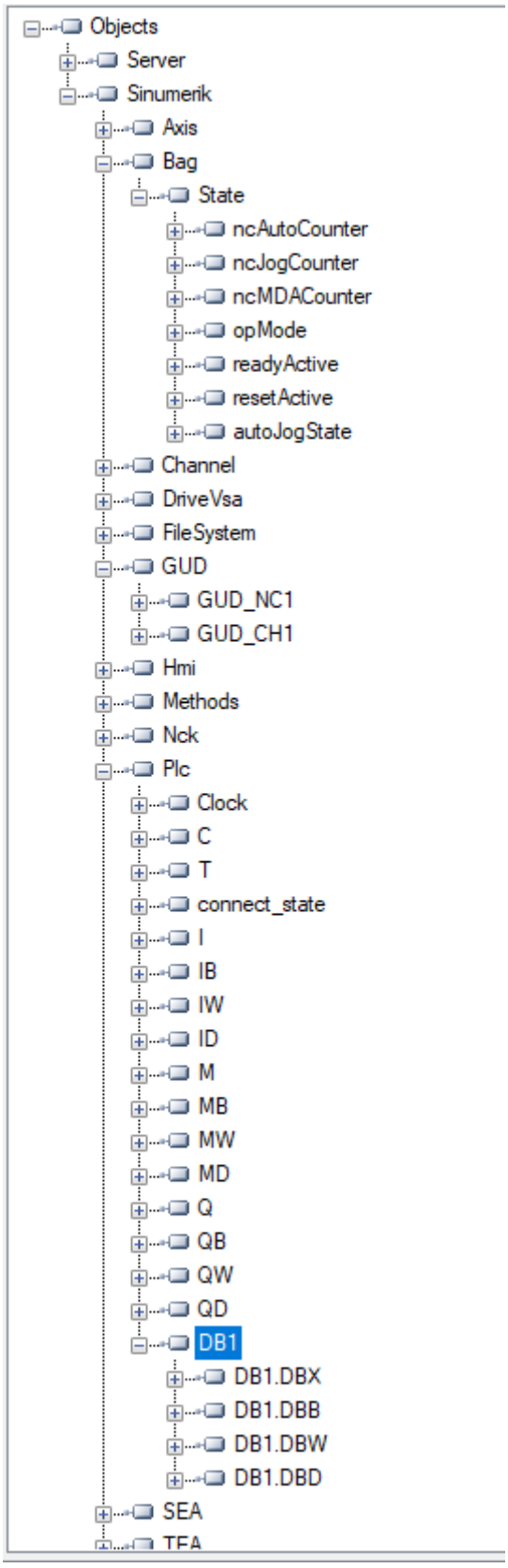

Objects
Server
Sinumerik
Axis
Bag
State
ncAutoCounter
ncJogCounter
ncMDACounter
opMode
readyActive
resetActive
autoJogState
Channel
DriveVsa
FileSystem
GUD
GUD_NC1
GUD_CH1
Hmi
Methods
Nck
Plc
Clock
C
T
connect_state
I
IB
IW
ID
M
MB
MW
MD
Q
QB
QW
QD
DB1
DB1.DBX
DB1.DBB
DB1.DBW
DB1.DBD
SEA
TEA

Figure 7-1 Browsing

7.3 Variable access

7.3.1 Variable paths for NC access operations

Note

You have to pay attention to the correct upper-case and lower-case of the "nodeID". The respective identifier of the "nodeID" provides information on the correct notation.

Variable access

The variable paths for NC access are stored in the address space of the SINUMERIK Operate OPC UA server.

You can obtain **further information** from the List Manual for 840D sl and 828D "NC variables and interface signals" (https://support.industry.siemens.com/cs/de/en/view/109769139).

Attribute	Value
NodeId	NodeId
NamespaceIndex	2
IdentifierType	String
Identifier	/Channel/Parameter/R
NodeClass	Variable
BrowseName	2, "/Channel/Parameter/R"
DisplayName	"en_us", "R"
Description	"en_us", "R"
WriteMask	0
UserWriteMask	0
Value	
SourceTimestamp	1/30/2014 3:17:25.822 PM
ServerTimestamp	1/30/2014 3:17:25.822 PM
SourcePicoseconds	0
ServerPicoseconds	0
Value	66
DataType	Double
NamespaceIndex	0
IdentifierType	Numeric
Identifier	11
ValueRank	-1
ArrayDimensions	BadAttributeIdInvalid
AccessLevel	Readable, Writeable
UserAccessLevel	Readable, Writeable
MinimumSamplingInterval	50
Historizing	false

Figure 7-2 Identifier for R parameter

The displayed NC variables always represent only the first parameter of the corresponding NC data area (channel, TO area, mode group).

Example

Syntax of the R parameter is as follows: R[Channel,Parameter]

The R parameters are found under the identifier "/Channel/Parameter/R", which is eventually mapped to "/Channel/Parameter/R[u1, 1]". If you want to access other parameters, you must correspondingly extend the identifier, for example "/Channel/Parameter/R[u2, 56]".

Table 7-1 Examples of variable paths (NC access operations)

Variable path	Description
/Channel/Parameter/R[u1,10]	R parameter 10 in channel 1
/Channel/Parameter/R[u1,1,5] /Channel/Parameter/R[u1,1,#5]	R parameter array R parameters 1 to 5 in channel 1
/Channel/GeometricAxis/name[u2,3]	Name of the 3rd axis in channel 2
/Channel/GeometricAxis/actToolBasePos[u1,3]	Position of the 3rd axis in channel 1

Note

Please keep in mind that with array access only max 149 parameters are allowed in one access operation (for example /Channel/Parameter/R[u1, 1, #149]).

7.3.2 Variable paths for GUD access operations

GUD variables can be found in the OPC UA server under the "/Sinumerik/GUD" node.

The displayed GUD variables always represent only the first parameter (for GUD arrays) of the first NC channel (for channel-dependent GUD variables). If you want to access a different parameter of a GUD array or a different channel, you must extend the identifier accordingly for the NC access.

GUD arrays are 1-indexed for access, and access is always one-dimensional. This means, the index must be calculated for multi-dimensional arrays.

Example 1: One-dimensional array, NC-global GUD array

"UGUD.DEF" file

```
DEF NCK INT ARRAY[2]
M17
```

Access is performed as follows:

```
ARRAY[0] → /NC/_N_NC_GD3_ACX/ARRAY[1]
ARRAY[1] → /NC/_N_NC_GD3_ACX/ARRAY[2]
```

Example 2: Two-dimensional array, channel-dependent GUD array

"UGUD.DEF" file

```
DEF CHAN INT ABC[3,3]
M17
```

Access is performed as follows:

```
ABC[0,0] → /NC/_N_CH_GD3_ACX/ABC[u1, 1]
ABC[0.1] → /NC/_N_CH_GD3_ACX/ABC[u1, 2]
ABC[0.2] → /NC/_N_CH_GD3_ACX/ABC[u1, 3]
ABC[1.0] → /NC/_N_CH_GD3_ACX/ABC[u1, 4]
ABC[1.1] → /NC/_N_CH_GD3_ACX/ABC[u1, 5]
ABC[1.2] → /NC/_N_CH_GD3_ACX/ABC[u1, 6]
ABC[2.0] → /NC/_N_CH_GD3_ACX/ABC[u1, 7]
ABC[2.1] → /NC/_N_CH_GD3_ACX/ABC[u1, 8]
ABC[2.2] → /NC/_N_CH_GD3_ACX/ABC[u1, 9]
```

7.3.3 Variable paths for PLC access operations

PLC variables can be found in the OPC UA server under the node "/Sinumerik/Plc".

SINUMERIK 840D sl and SINUMERIK 828D support addresses only, e.g.: MB5, DB21.DBB5.

SINUMERIK ONE offers in addition symbolic access to variables.

For non optimized datablocks, these datablocks will be shown symbolically and absolutely in the browse tree (twice). For optimized datablocks, these datablocks will only be shown symbolically.

Note

Switching off absolute addressing for non optimized datablocks

On adding the item "AbsoluteDBNoBrowsable = false" in [General] section of the file "settings.ini", the "classic" datablocks will not be displayed in browse tree with their absolute DB numbers. Only symbolic names will appear in browse tree.

Default setting is "true". Then both symbolic names and absolute DB no. will be shown in browse tree.

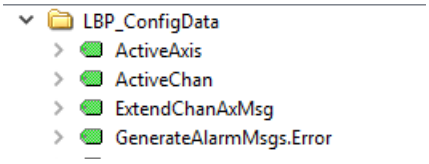
The settings file path is:

- for NCU will be "/user/sinumerik/hmi/opcua/cfg/settings.conf"
- for PCU/IPC "C:\Program Files(x86)\Siemens\MotionControl/user/sinumerik/hmi/opcua/cfg/settings.ini"

For accessing **symbolic addresses** use the variable path as shown in the OPC UA browse tree.

Example

In “Plc” node you can find symbolic datablocks, e.g.: LBP_ConfigData



For reading variables please refer to the corresponding symbolic variable in the browse tree.

Single values of an array can be accessed by the exposed array members.

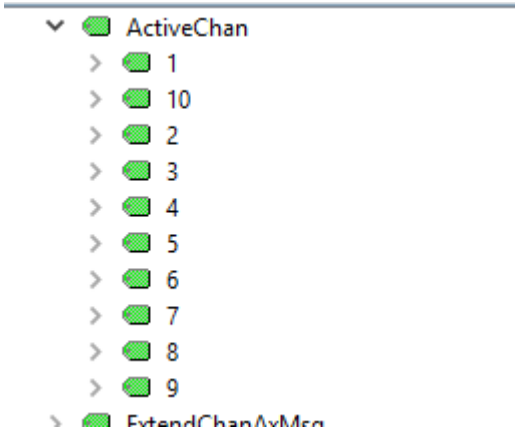


Figure 7-3 Exposed array members of variable "ActiveChan"

For accessing **absolute addresses** the appropriate byte no. needs to be appended to the base address from the OPC UA browse tree.

Example

For accessing MB5 the access string from OPC UA browse tree "/Plc/MB" must be appended by the appropriate byte number: "/Plc/MB**5**".

This applies accordingly to datablock addresses: "/Plc/Db21.DBB5".

Access formats for absolute addresses

The various access formats are shown in the following table. They need to be prefixed with "/Plc/".

Note

The data type is converted during access with the OPC UA data access interface. Refer to the following table for the data type conversions.

Table 7-2 PLC syntax

Area	Address (IEC)	Permissible data types	OPC UA data type
Output image	Qx.y	**BOOL**	Boolean
Output image	QBx	**BYTE**, CHAR, STRING	UInt32 String
Output image	QWx	**WORD**, CHAR, INT,	UInt32 Int32
Output image	QDx	**DWORD**, DINT, REAL	UInt32 Int32 Double
Data block	DBz.DBXx.y	**BOOL**	Boolean
Data block	DBz.DBBx	**BYTE**, CHAR, STRING	UInt32 String

Area	Address (IEC)	Permissible data types	OPC UA data type
Data block	DBz.DBWx	**WORD**, CHAR, INT	UInt32 Int32
Data block	DBz.DBDx	**DWORD**, DINT, REAL	UInt32 Int32 Double
Input image	Ix.y	**BOOL**	Boolean
Input image	IBx	**BYTE**, CHAR, STRING	UInt32 String
Input image	IWx	**WORD**, CHAR, INT	UInt32 Int32
Input image	IDx	**DWORD**, DINT, REAL	UInt32 Int32 Double
Bit memory	Mx.y	**BOOL**	Boolean
Bit memory	MBx	**BYTE**, CHAR, STRING	UInt32 String
Bit memory	MWx	**WORD**, CHAR, INT	UInt32 Int32
Bit memory	MDx	**DWORD**, DINT, REAL	UInt32 Int32 Double
Counters	Cx	-	Byte
Timers	Tx	-	UInt32
PLC time	Clock	-	UInt16

Notes regarding the table:

- "x" represents the byte offset; "y" the bit number in the byte and "z" the data block number.
- The data type in bold characters is the default data type and does not have to be specified. The specifications DB2.DBB5.BYTE and DB2.DBB5 are equivalent.
- Square brackets are used to access arrays, e.g. "/Plc/DB5.DBW2:[10]" (word array of length 10).
- Access to STRING arrays ("/Plc/DB123.DBB0:STRING[5]") and bit arrays ("/Plc/DB123.DBX0.0:[5]") are not supported.

Examples of variable paths (PLC access operations)

Table 7-3 Examples of variable paths (PLC access operations)

Variable path	Description
/Plc/M5.0	Memory bit 0 at byte offset 5
/Plc/DB5.DBW2	Word (16-bit) at byte offset 2 in data block 5
/Plc/DB8.DBB2:STRING	UTF8 string beginning at byte offset 2 in data block 8
/Plc/DB8.DBW2:[10]	Array of 10 words beginning at byte offset 2 in data block 8
/Plc/DB100.DBB1	Byte at byte offset 1 in data block 100
/Plc/DB2.DBD0:REAL[10]	Array of 10 double words (32-bit) beginning at byte offset 0 in data block 2, which are formatted as a floating-point number

Note

- Timers can only be read. A timer is active if it contains a value other than 0.
- If the data type CHAR or STRING is used in conjunction with a byte access, UTF8 characters are read, but if either data type is used in conjunction with a word access, UTF16 characters are read.
- Variables of the STRING type contain the maximum length in the first byte and the actual length in the second byte. When strings are written, the actual length is adapted accordingly. The maximum length is not changed.
- For the STRING data type in conjunction with a byte access (e.g. "/Plc/DB99.DBB0:STRING"), the maximum string length is 255 characters. As a result of the UTF8 formatting, for some characters (e.g. for the "µ"), two bytes are required so that the maximum string length is correspondingly reduced.
- Only one-dimensional arrays are supported.

7.3.4 Restrict DB or variable access with TIA portal

Prerequisites

You can generally control whether a data block from the PLC can be reached by OPC UA via global settings in the TIA portal:

- Presets for **new data blocks**
  - Go to settings "PLC programming > general".
  - Select the check box ""Set "DB accessible from OPC UA" for new data blocks".

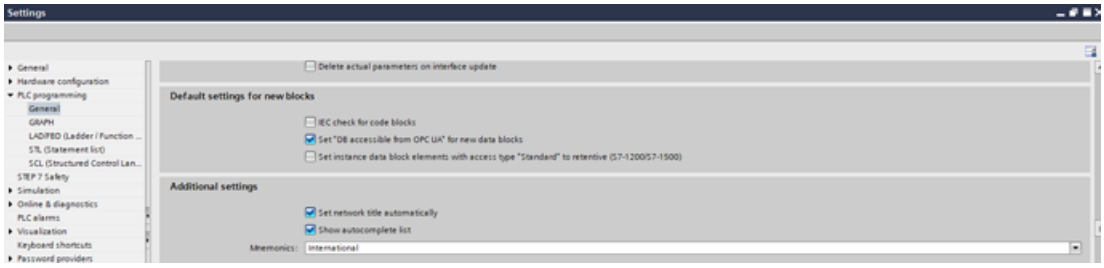


Figure 7-4 TIA portal DB accessible from OPC UA for new data blocks

- Presets for **specific data blocks**
  - Select the specific data block and open the context menu with the mouse.
  - Click on "Settings".
  - In the index "General" click on "Attributes".
  - Select the checkbox "Data blocks accessible from OPC UA".

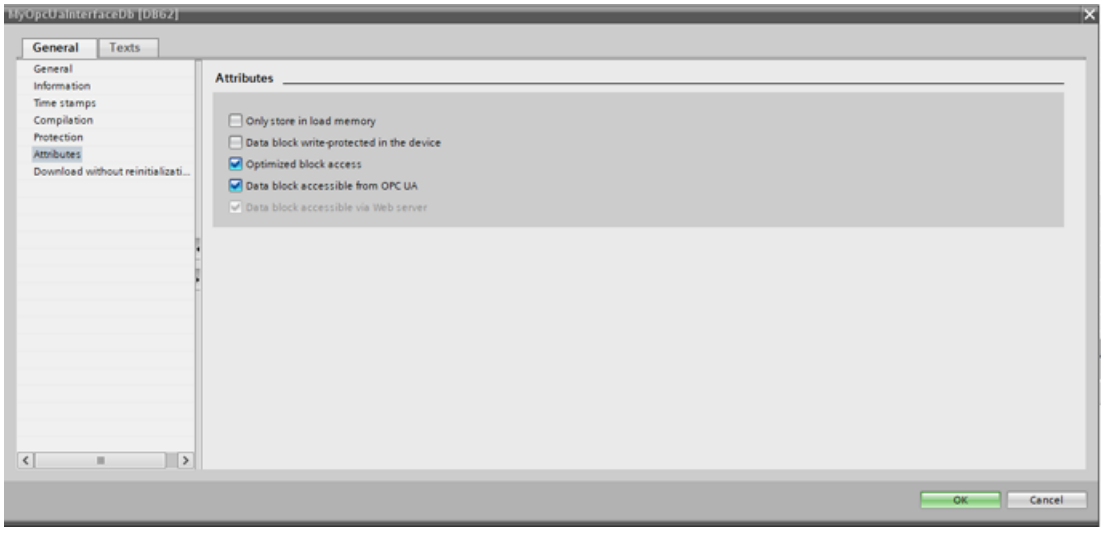
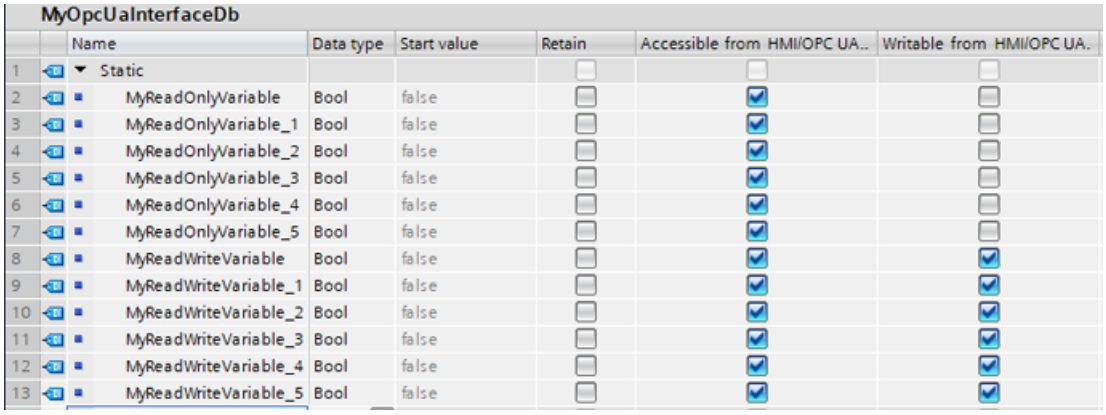


Figure 7-5 TIA portal DB accessible from OPC UA

PLC tag table

PLC tag tables in the TIA portal contain the definitions of the PLC tags and symbolic constants that are valid throughout the CPU. A PLC tag table is created automatically for each CPU used in the project. With the help of the PLC tag table, you can see and change whether a variable is accessible or writeable from OPC UA.



MyOpcUaInterfaceDb

	Name	Data type	Start value	Retain	Accessible from HMI/OPC UA..	Writable from HMI/OPC UA.
1	Static					
2	MyReadOnlyVariable	Bool	false		☑	
3	MyReadOnlyVariable_1	Bool	false		☑	
4	MyReadOnlyVariable_2	Bool	false		☑	
5	MyReadOnlyVariable_3	Bool	false		☑	
6	MyReadOnlyVariable_4	Bool	false		☑	
7	MyReadOnlyVariable_5	Bool	false		☑	
8	MyReadWriteVariable	Bool	false		☑	☑
9	MyReadWriteVariable_1	Bool	false		☑	☑
10	MyReadWriteVariable_2	Bool	false		☑	☑
11	MyReadWriteVariable_3	Bool	false		☑	☑
12	MyReadWriteVariable_4	Bool	false		☑	☑
13	MyReadWriteVariable_5	Bool	false		☑	☑

Figure 7-6 TIA portal tag table

Column	Explanation
Accessible from HMI/OPC UA	Indicates whether OPC UA can access this variable during runtime.
Writable from HMI/OPC UA/Web API	Indicates whether or not the tag can be written from OPC UA during runtime.

Note

Please note that by deactivating the checkbox "Accessible from HMI/OPC UA" or "Writeable from HMI/OPC UA", the variable can also not be addressed by the programming package (3GL).

7.3.5 Variable paths for machine and setting data

The variable paths for machine and setting data are stored in the address space of the OPC UA server under the nodes "/Sinumerik/TEA" and "/Sinumerik/SEA". Pay attention to the correct upper-case and lower-case of the "nodeID". The respective identifier of the "nodeID" provides information on the correct notation.

The displayed machine and setting variables always represent only the first parameter of the corresponding data area (channel, axis).

Table 7-4 Examples of variable paths (machine and setting data)

Variable path	Description
/NC/_N_CH_TEA_ACX/$MC_CHAN_NAME	Channel name of channel 1
/NC/_N_CH_TEA_ACX/$MC_CHAN_NAME[u2]	Channel name of channel 2

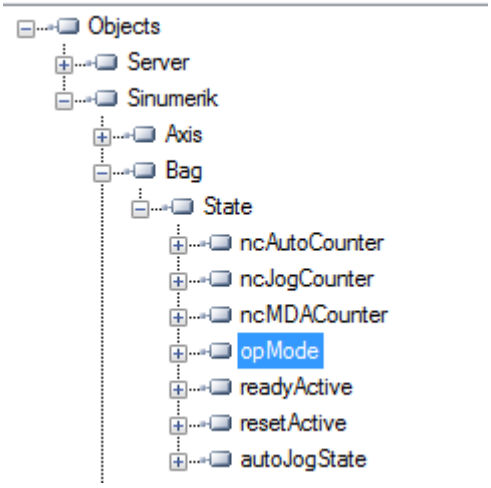
Machine data arrays are 1-indexed for access.

7.3.6 Finding of OPC UA variables

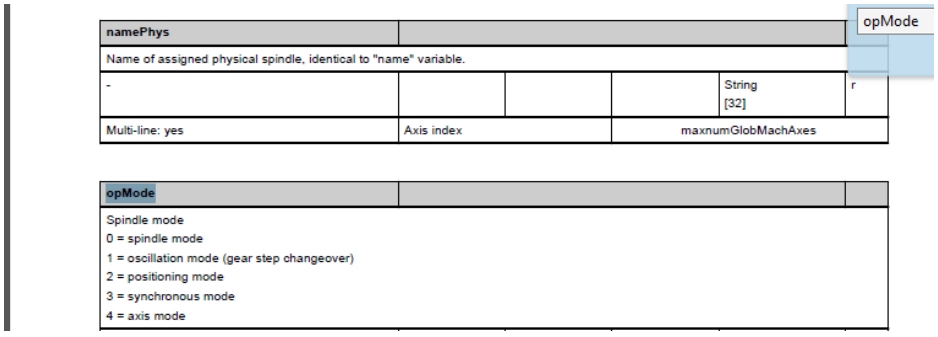
You can find **further information** on variable documentation in the list manual NC variables and interface signals (https://support.industry.siemens.com/cs/de/en/view/109769139)

Example 1: Finding an OPC UA variable in the variable documentation

You want to find the variable "opMode" in folder "/Bag/State".



1. Refer to the document mentioned above. Search for "opMode".

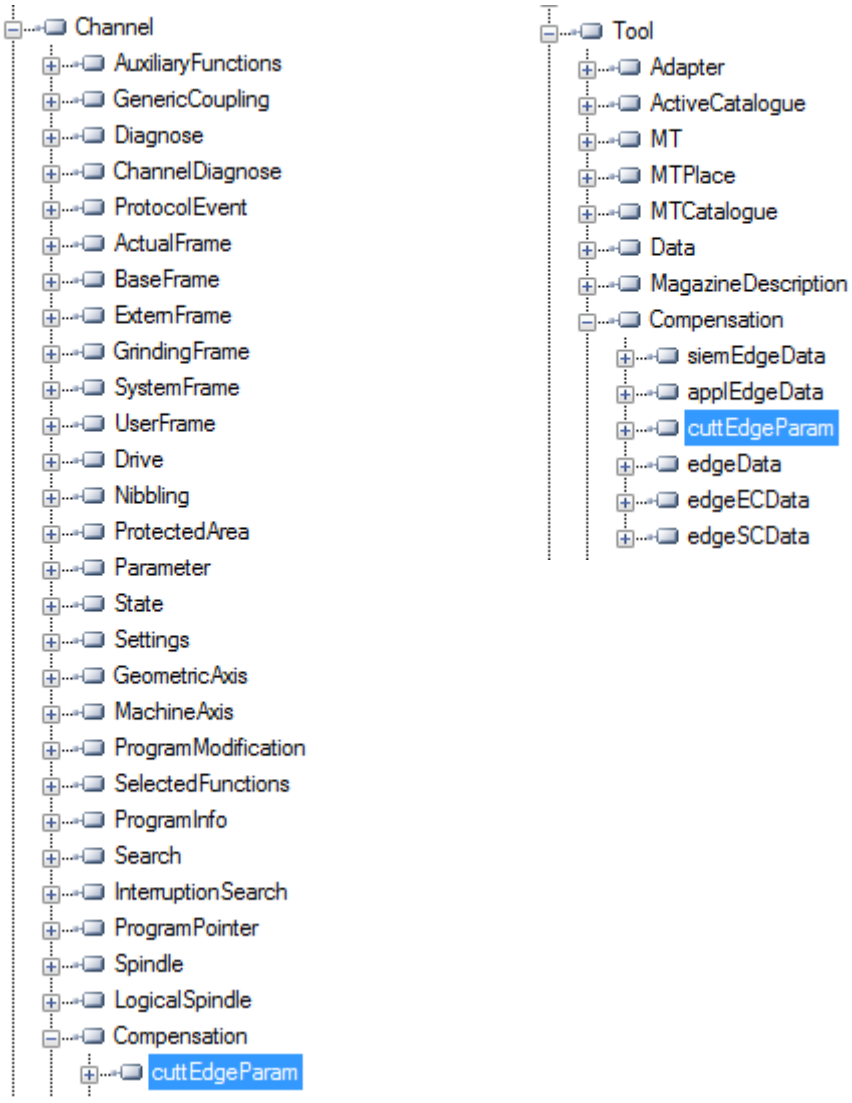


namePhys					opMode
Name of assigned physical spindle, identical to "name" variable.					
-				String [32]	r
Multi-line: yes	Axis index		maxnumGlobMachAxes		

opMode		
Spindle mode 0 = spindle mode 1 = oscillation mode (gear step changeover) 2 = positioning mode 3 = synchronous mode 4 = axis mode		

Example 2: Finding an OPC UA variable occurring in different folders in the variable documentation

You want to find the variable "cuttEdgeParam" which occurs in the folder "/Channel/Compensation" and "/Tool/Compensation".



1. At the beginning of each chapter for variable sections, you find the information "OEM-MMC: LinkItem" specifying "/ToolCompensation/".

   3.7.2 Area T, Block TO : Tool edge data: Offset data

   OEM-MMC: Linkitem /ToolCompensation/...

   The data module TO is organized as a 2-dimensional variable array.

2. Refer to the document and search for "ChannelCompensation" and then navigate manually to the requested parameter "cuttEdgeParam".

cuttEdgeParam	$TC_DPx[y,z]			
Compensation value parameters for a tool edge				
mm, inch or user-defined	0		Double	wr
Multi-line: Yes	(EdgeNo - 1) * numCuttEdgeParams + ParameterNo	numCuttEdgeParams * numCuttEdges		

Example 3: Finding a variable from documentation on OPC UA client

You want to find the variable "cuttEdgeParam" in the Tool edge data section.

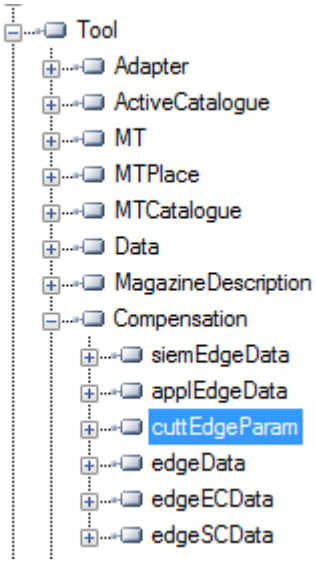
1. At the beginning of each chapter of the variable documentation you find the information "OEM-MMC: LinkItem" specifying here "/ToolCompensation/".

   3.7.2 Area T, Block TO : Tool edge data: Offset data

   OEM-MMC: Linkitem /ToolCompensation/...

   The data module TO is organized as a 2-dimensional variable array.

2. Therefore you will find the variable "cuttEdgeParam" in the OPC UA Browse Tree in the folder "Tool", subfolder "Compensation".



7.3.7 Monitored items

An OPC UA client can subscribe to a selection of nodes of interest and let the server monitor these items. Only in case of changes, e.g. to their values, the server notifies the client about such changes. This mechanism reduces the amount of transferred data immensely. Besides the reduction of bandwidth this mechanism introduces further advantages and is the recommended mechanism to “read” information from a UA server.

A client can subscribe to different types of information provided by an OPC UA server. The purpose of a subscription is to group these sources of information, called monitored items, together, forming a piece of information called a notification.

A subscription consists of at least one monitored item, which has to be created within the context of a session and can be transferred to another session. To create a session, a secure channel between the client and the server has to be established.

There are two different types of “changes” a client can subscribe to when adding monitored items to the subscription:

- subscribe to data changes of Variable Values (Value attribute of a Variable)
- subscribe to Events of Objects (EventNotifier attribute of an Object)

Publish interval

Clients define MonitoredItems to subscribe to data and Events. Each MonitoredItem identifies the item to be monitored and the Subscription to use to send Notifications. The item to be monitored may be any Node Attribute.

Notifications are data structures that describe the occurrence of data changes and Events. They are packaged into NotificationMessages for transfer to the Client. The Subscription periodically sends NotificationMessages at a user-specified publishing interval, and the cycle during which these messages are sent is called a publishing cycle." (see OPC UA Part 4 - Services 1.04 Specification.pdf (https://opcfoundation.org/))

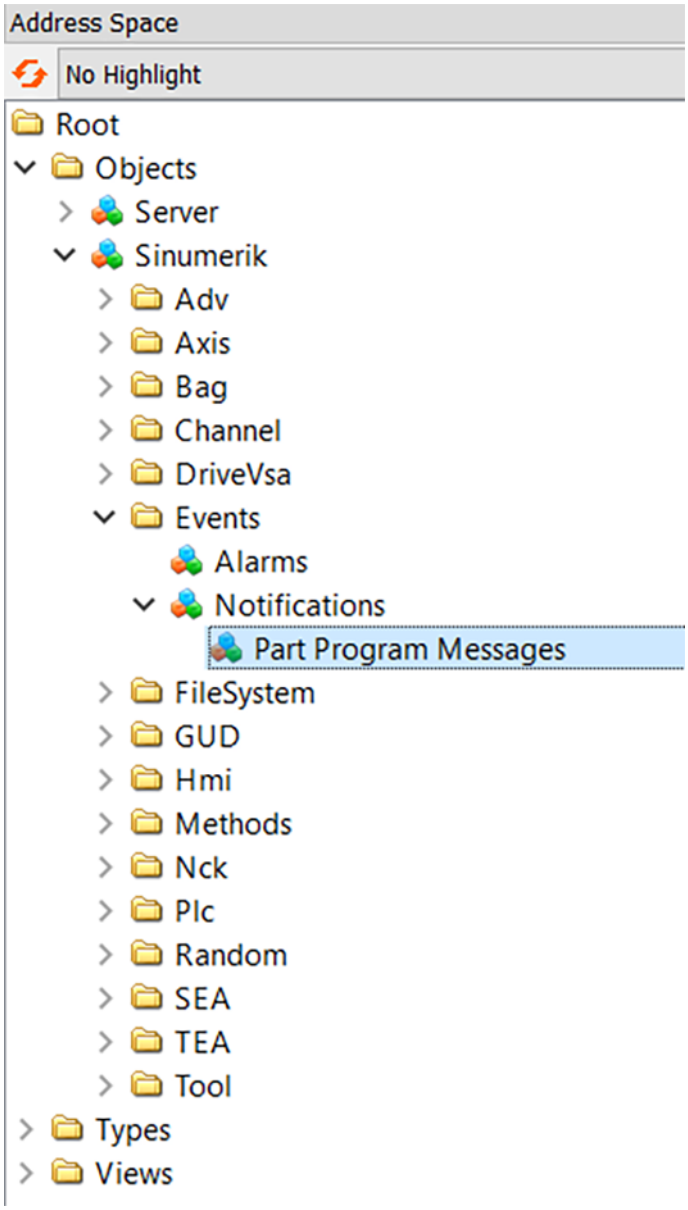
Sampling interval

Each MonitoredItem created by the Client is assigned a sampling interval that is either inherited from the publishing interval of the Subscription or that is defined specifically to override that rate. [...] The sampling interval indicates the fastest rate at which the server should sample its underlying source for data changes. (see OPC UA Part 4 - Services 1.04 Specification.pdf (https://opcfoundation.org/))

7.4 Alarms

7.4.1 Overview

Any OPC UA client supporting Alarms & Conditions connected to the OPC UA server can subscribe to alarms to get the notifications of alarms.
All OPC UA Clients that have subscribed for SINUMERIK alarms will be provided with an alarm as soon as it becomes active. Also if the alarm becomes inactive, the status of the corresponding alarm/s will be updated automatically.
Alarms and Conditions support subscription of all the pending and active alarms of the SINUMERIK system.



When "SINUMERIK" or "Events" node is subscribed then both alarms and Part program messages are sent to OPC UA. If only alarms node is subscribed then only alarms are sent to OPC UA. If "Notifications" or "Part Program Messages" node is subscribed then only Part program messages are sent to OPC UA.

Part program messages are supported as an event.

Note

If a customer wants the Part program messages to be sent as alarms in OPC UA, then the property under the "Events" tag needs to be set to `true` in the settings.ini file. Additionally, it will be sent as event also. For more information, refer "Enabling Part program messages as OPC UA alarm (Page 50)" topic.

The OPC UA Server provides all alarms that will be provided by the SINUMERIK AlarmService:

- HMI alarms
- NCK alarms including drive alarms
- PLC alarms (SINUMERIK user alarms)
- PLC1500 alarms (system diagnosis, S7 motion control, security, Program_Alarm, ProDiag, supervisions und interlocks for GRAPH)

Multi language support for the alarms and warnings messages are supported and the required alarm language can be selected during session creation in OPC UA Client. If the desired language is not supported in the operate, the default English language is supported. The SINUMERIK Alarm object is of the "CNCAlarmType" which is defined in the Companion Specification "OPC UA Information Model for CNC Systems (http://opcfoundation.org/UA/CNC/)".

Part program messages

Part program messages are considered as events in OPC UA.

Note

If a customer wants the Part program messages to be sent as alarms in OPC UA, then the property under the "Events" tag needs to be set to `true` in the settings.ini file. Additionally, it will be sent as event also. For more information, refer "Enabling Part program messages as OPC UA alarm (Page 50)" topic.

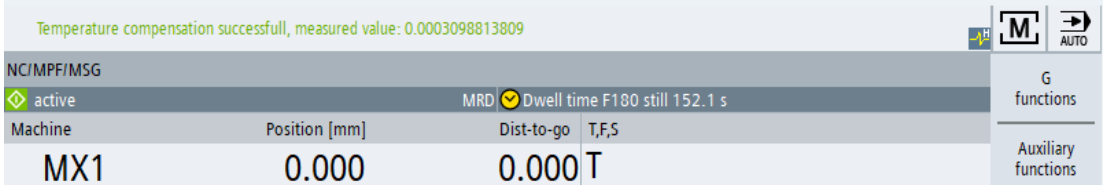


Figure 7-7 Part program messages

They will not be reported in an alarm subscription. In order to have access to part program messages, use the variable path: "/Channel/ProgramInfo/msg[u <channel number>]".

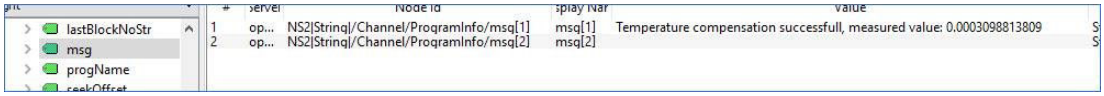


Figure 7-8 Variable path

7.4.2 Subscribe / unsubscribe to alarms

Subscribe to alarms

The SINUMERIK Alarm Event object is connected to the SINUMERIK node. To receive the alarms, an event subscription must be placed at the SINUMERIK node. The following example describes how to receive the alarms using the OPC UA Foundation Client:

1. Open the "Quickstart Alarm Condition Client".

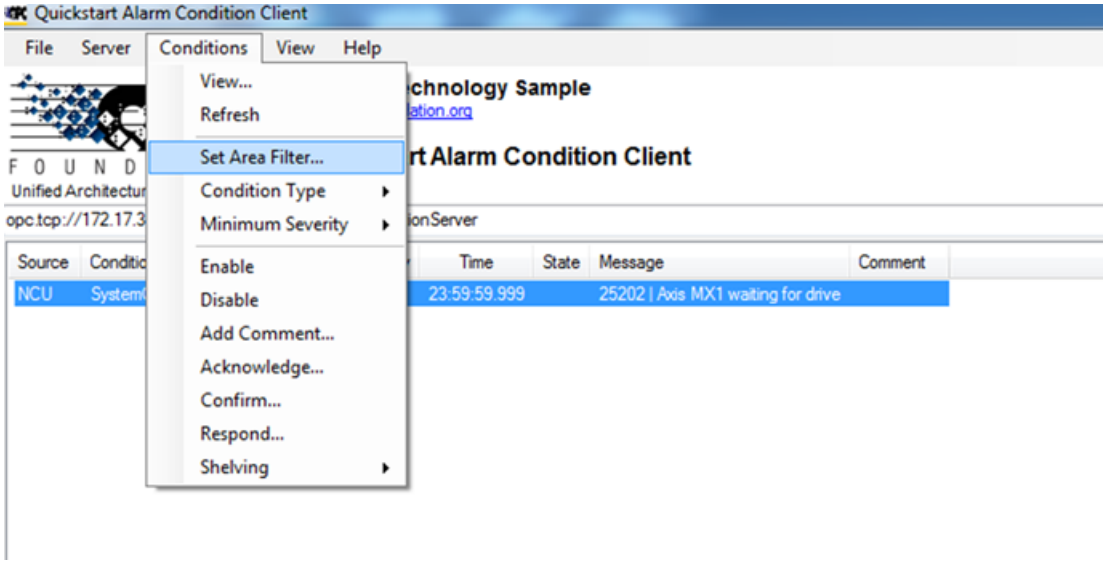


Figure 7-9 Alarm Condition Client

2. Click "Conditions > Set Area Filter...". The "Select Area" window appears.

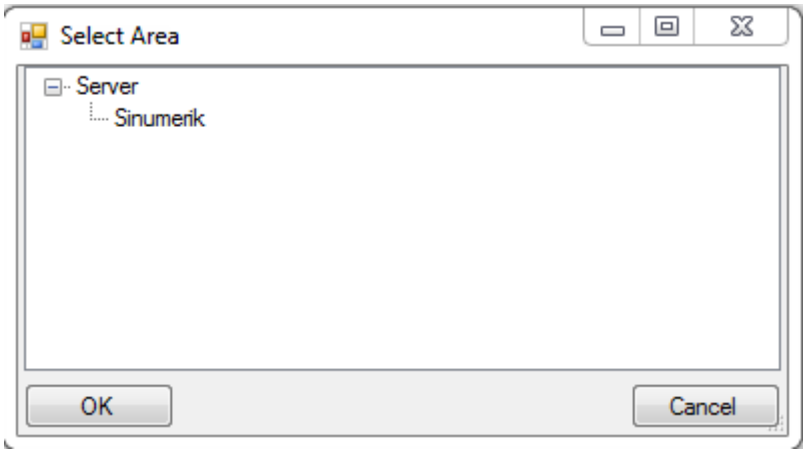


Figure 7-10 The Select Area Window

3. Select "Sinumerik".
4. Click "OK".

The alarms will be displayed on the screen.

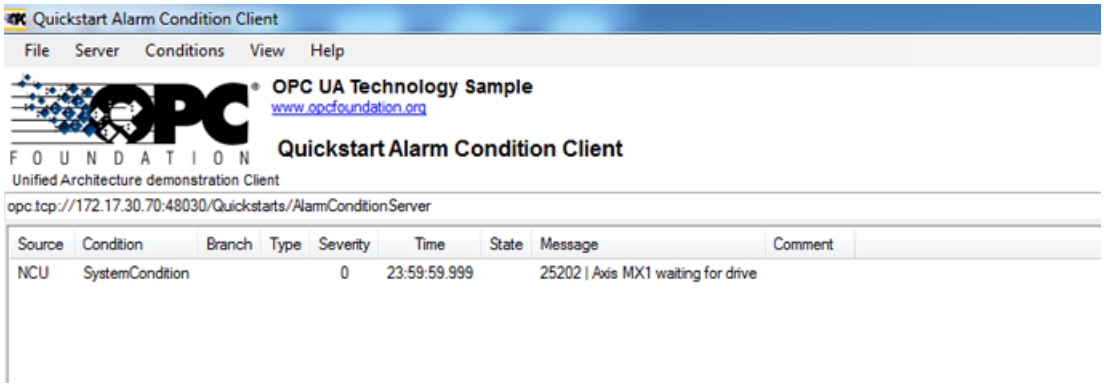


Figure 7-11 Alarm List

Unsubscribe to alarms

1. Click "Conditions > Set Area Filter...". The "Select Area" window appears.
2. Right click on "Sinumerik" and select "Remove Monitored Item" to unsubscribe the server from the Quickstart Alarm Condition Client.

7.4.3 Acknowledgement of alarms

Below is the snippet for acknowledging the alarms using the OPC Foundation .Net Standard SDK client:

//C# code snippet for Alarm Acknowledgement

```
// step 1:Enter the conditionID{AlarmID_SourceID_InstanceID}(eg:- "ns=2;s=150202_0_1")
Console.WriteLine("Enter the Condition ID of the alarm that needs to be acknowledged: ");
string conditionID = Console.ReadLine();
// Remove "ns=2;s=" from the condition ID
string condID = conditionID.Substring(conditionID.LastIndexOf(";s=") + 3);

// Step 2: Enter the comment for acknowledgement(optional)
Console.WriteLine("Enter the Acknowledgement comment(if any): ");
string comment = Console.ReadLine();
// get the Alarm Id from the condition ID
string[] AlarmID = conditionID.Split('_');
// convert Alarm Id(string) to byte array
byte[] byteConditionID = Encoding.ASCII.GetBytes(AlarmID[0]);
LocalizedText lt = new LocalizedText("en-GB", comment);

// Step 3: Call a method to ACK
NodeId ObjectID = new NodeId(conditionID, 2);
// 9111 is the OpcUaId for Acknowledgement method
NodeId MethodNodeID = new NodeId(9111, 0);
// Call Acknowledgement method via Session object
Session.Call(ObjectID, MethodNodeID, byteConditionID, lt);
```

7.4.4 Sequence description of alarms

The OPC UA server automatically sends an object of the "CNCAlarmtype" to the OPC UA client containing the single alarm which has just been triggered.

The OPC UA server automatically resends an object of the "CNCAlarmtype" with the same content as when the corresponding alarm was triggered, except a change in the status.

To get all the active alarms, the client has to subscribe to the Sinumerik node.

7.4.5 SINUMERIK Alarm object

7.4.5.1 Description

Every variable or object in the address space of an OPC UA server is called a node. Every node has a server unique node id, its symbolic name, addressing information inside the address model and some other attributes.

Events are by themselves not visible as nodes in the address space. They can only be received via objects. Not all objects can signal events. Whether an object can signal events is specified at the object by the EventNotifier attribute. Only objects where this attribute has been set can be specified in the Event Monitored Item and received in Clients Events.

The Server Object serves as root notifier, that is, its EventNotifier Attribute shall be set providing Events. However Server object will not be allowed to subscribe for the Events. Only the "Sinumerik" Object node is accessible and can subscribe to the events.

7.4.5.2 OPC UA event messages and alarms

Access to alarms

User access right is required to subscribe the Events of the Sinumerik object. User access right with access permission has to be set to “SinuReadAll” or “AlarmRead”.The access right is provided using Method Call “GiveUserAccess” as shown below.

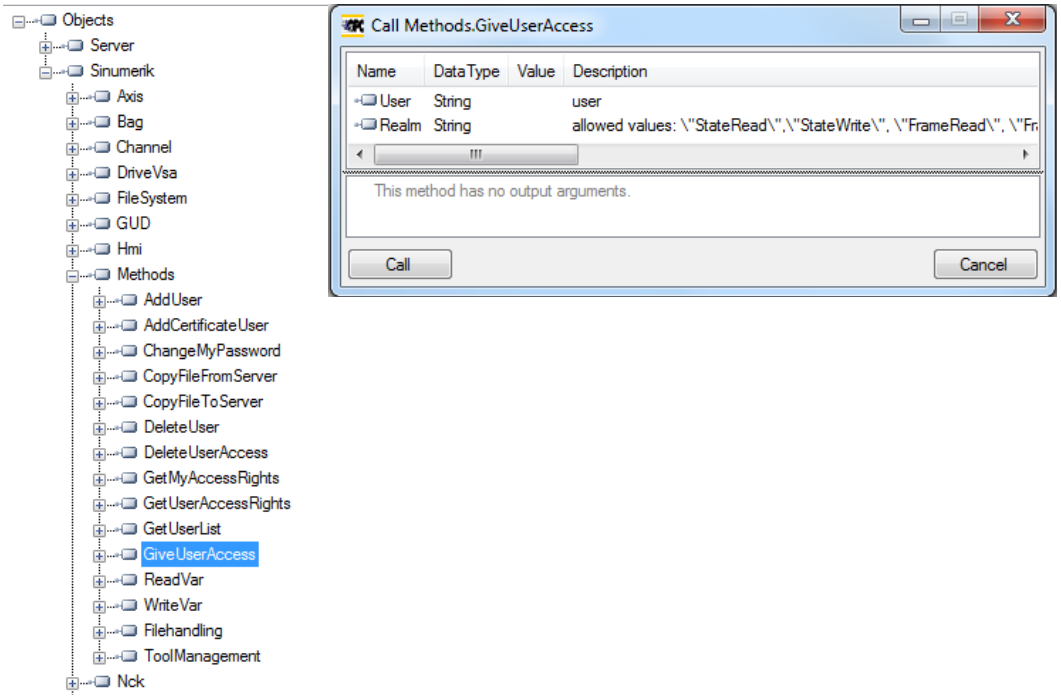


Figure 7-12 Alarm access rights

If the client does not have the access with “SinuReadAll “or “AlarmRead” and user tries to subscribe to the Events, server will return error code with “BadUserAccessDenied”.

Event types

The SINUMERIK Alarm object is of the “CNCAlarmType” which is defined in the Companion Specification “OPC UA Information Model for CNC Systems (http://opcfoundation.org/UA/CNC/)”.

The root of the derivation hierarchy is the BaseEventType. The types for Alarms and Conditions are available below the ConditionType. The Application-specific event types such as CncAlarmType can be derived. The CncAlarmType extends the DiscreteAlarmType.

An alarm is composed of various nested or parallel state machines. Monitoring can generally be enabled or disabled. If monitoring is enabled, the alarm can be active or otherwise inactive. Acknowledgment, confirm and comments of alarms is currently not supported.

The basic type for all condition objects is the condition type. It is derived from BaseEventType. All mechanisms for alarm processing work even without the condition objects are contained in the address space.

If a condition object changes one or several states, the server sends an event with the requested event fields to the client. So only the alarms, where a status change happens after the connection is established, will be sent. To receive all currently active alarms the refresh method can be used.

CncAlarmType

The CncAlarmType, which is specified in the Companion Specification "OPC UA Information Model for CNC Systems" is derived from the DiscreteAlarmType, which is defined by the OPC Foundation.

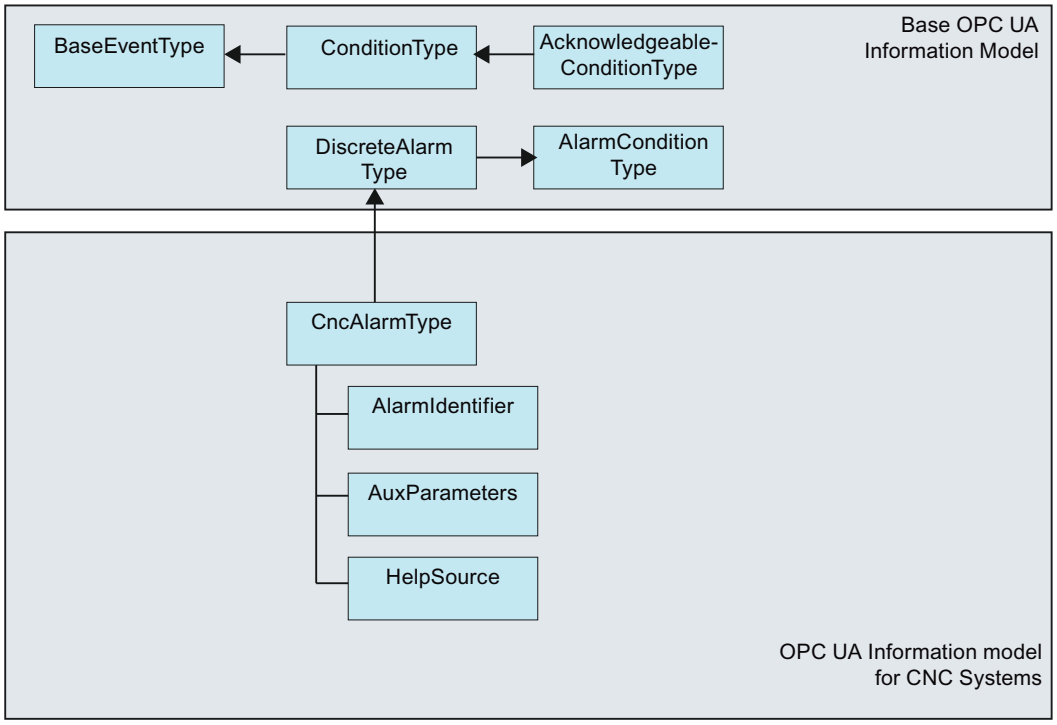


Figure 7-13 OPC UA Information Model for CNC Systems

Description of the CncAlarmType

Since the CncAlarmType is derived from a number of types as you can see in Figure 5-8, it does not only contain the three attributes AlarmIdentifier, AuxParameters and HelpSource, but also all the other attributes which are inherited from the objects.

Attributes of BaseEventType

Attribute	Data type	Mapping with respect to SINUMERIK	M/O	Description
EventId	String	Unique node id generated from SINUMERIK system.	M	EventId is generated by the server to uniquely identify a particular Event Notification. The EventId shall always be returned as value and the server is not allowed to return a StatusCode for the EventId indicating an error.
EventType	NodeId	It is always set to 'CncAlarmType'.	M	The EventType shall always be returned as value and the server is not allowed to return a StatusCode for the EventType indicating an error.
SourceNode	NodeId	Alarm source identifier provided by SINUMERIK system.	M	SourceNode identifies the Node that the Event originated from. If the Event is not specific to a Node, the NodeId is set to null.
SourceName	String	Supported alarm source names are HMI, NCK, and PLC.	M	SourceName provides a description of the source of the Event. This could be the string-part of the DisplayName of the Event source using the default locale of the server. If it is not possible for a CNC system to provide this information in detail, the SourceName should provide the main component responsible for this alarm (e.g. CNC, PLC, or even Channel).
Time	UtcTime	Alarm time stamp	M	Time provides the time of the Event occurred. Once set, intermediate OPC UA servers shall not alter the value.
ReceiveTime	UtcTime	Alarm time stamp of the server.	M	ReceiveTime provides the time the OPC UA server received the Event from the underlying device of another server.
Message	Localized Text	Reading attributes via (SLAE_EV_ATTR_MSG TEXT)	M	Alarm Message provides a human readable and localizable text description of the Event.
Severity	UInt16	Reading attributes via (SLAE_EV_ATTR_SEVE RITY)	M	Severity of the event message. The range of values of the severity is from 1 to 1000, where 1000 corresponds to the highest severity.
LocalTime	TimeZoneDataType	Offset and the DaylightSavingInOffset flag	O	LocalTime is a structure containing the Offset and the DaylightSavingInOffset flag. The Offset specifies the time difference (in minutes) between the Time Property and the time at the location in which the event was issued. If DaylightSavingInOffset is - TRUE: Standard/Daylight savings time (DST) at the originating location is in effect and Offset includes the DST correction. FALSE: The Offset does not include DST correction and DST may or may not have been in effect.

Severity of Alarms

SINUMERIK system uses four severity levels (e.g. Information, Warning, Error, and Part program messages). The table below shows the values at SINUMERIK system and its mapping in OPC UA server/client:

Severity Level	SINUMERIK System	OPC UA server/client
Information	0-1	1
Warning	2-999	500
Error	1000	1000
Part program messages	1	100

Additional attributes of the ConditionType

Attribute	Data type	Mapping with respect to SINUMERIK	M/O	Description
ConditionClassId	NodeId	Unique node id	M	String NodeID SystemConditionClassType
ConditionClassName	String	Set to "SystemConditionClassType"	M	SystemConditionClassType
ConditionName	String	Set to "SystemCondition".	M	ConditionName identifies the Condition instance that the Event originated from. It can be used together with the SourceName in a user display to distinguish between different Condition instances.
Retain	Boolean	True when the alarm is active. False otherwise.	M	Information whether or not the alarm shall be displayed. This is set to true by default.

Quality	String	According to SINUMERIK quality attribute, below string will be set:
		• BAD
		• GOOD
		• UNCERTAIN
			M	The quality provides information about the reliability of an alarm.
				Possible values of SINUMERIK:
				`AlarmQuality.QUALITY_BAD = 0`
				`AlarmQuality.QUALITY_GOOD = 192`
				`AlarmQuality.QUALITY_UNCERTAIN = 64`

LastSeverity	UInt16	Reading attributes via(SLAE_EV_ATTR_SEVERITY)	M	LastSeverity provides the previous severity of the ConditionBranch. Initially this Variable contains a zero value; it will return a value only after a severity change. The new severity is supplied via the Severity Property which is inherited from the BaseEventType.
BranchId	NodeId	Null	M	BranchId is Null for all Event Notifications that relate to the current state of the Condition instance.
Comment	LocalizedText	User can set the comments	M	The value of this variable can be set.
ClientUserId	String	Null	M	The value of this variable is set to null.
Enable		Not supported	M	Servers do not expose Condition instances in the AddressSpace.

Attribute	Data type	Mapping with respect to SINUMERIK	M/O	Description
Disable		Not supported	M	Servers do not expose Condition instances in the AddressSpace.
AddComment		Not supported	M	Not supported and the result code should return Bad_MethodInvalid.
ConditionRefreshMethod			None	When the method is called up, an event with the current state is triggered for the calling client for all conditions. Only those conditions are updated for which the Retain flag has been set.

Additional attributes of the AcknowledgeableConditionType

Attribute	Data type	Mapping with respect to SINUMERIK	M/O	Description
AckedState	Localized text	True / False	M	AckedState when FALSE indicates that the Condition instance requires acknowledgement for the reported Condition state. When the Condition instance is acknowledged, the AckedState is set to TRUE.
ConfirmedState	LocalizedText	True / False	O	ConfirmedState indicates whether it requires confirmation.
EnabledState	Localized text	True / False	M	Always set to true
Acknowledge	Localized text	True / False	M	Acknowledge will be set by the server to TRUE or FALSE.
Confirm			O	The Confirm Method is used to confirm an Event Notifications for a Condition instance state where ConfirmedState is FALSE. Normally, the NodeId of the object instance as the ObjectId is passed to the Call Service. However, some Servers do not expose Condition instances in the AddressSpace. Therefore all Servers shall also allow Clients to call the Confirm Method by specifying ConditionId as the ObjectId. The Method cannot be called with an ObjectId of the AcknowledgeableConditionType Node.

Additional attributes of the CncAlarmType

The CNCAlarmType is defined in the VDW Companion Specification "OPC UA Information Model for CNC Systems".

Attribute	Data type	Mapping with respect to SINUMERIK	M/O	Description
AlarmIdentifier	String	Unique Alarm id.	M	Unique alarm number. This mapped to Alarm ID.
AuxParameters	String	All available (out of 10) parameters will be displayed in '' separated value.	M	10 Auxilliary parameter values provided by SINUMERIK system.

7.4.6 Language of alarms

7.4.6.1 OPC UA language specification

The OPC UA server has a built-in data type "LocalizedText" to store the language specific alarm text. This data type defines a structure containing a string in a locale-specific translation specified in the identifier for the locale. The elements are defined in the table below :-

Name	Type	Description
LocalizedText	structure	
text	String	The localized text.
locale	LocaleId	The identifier for the locale (e.g. “en-US”).

The "LocaleId" is a simple data type that is specified as a string that is composed of a language component and a country/region component as specified by IEEE 754-1985 (http://standards.ieee.org/findstds/interps/index.html), IEEE Standard for Binary Floating-Point Arithmetic. The <country/region> component is always preceded by a hyphen.

The format of the LocaleId string is shown below:

<language>[-<country/region>]

- <language> is the two letter ISO 639 code for a language
- <country/region> is the two letter ISO 3166 code for the country/region

You can find **further information** in the specification **OPC UA Part3 - Address Space Model 1.03 Specification.pdf**

7.4.6.2 SINUMERIK language specification

The SINUMERIK system currently supports 31 languages which are mentioned below. These languages are identified by the 3-letter abbreviation that follows Microsoft conventions.

Note

In the list of languages that are mentioned, not every language is supported always.

7.4.6.3 Mapping of SINUMERIK LanguageID with OPC UA LocaleID

Mapping of the SINUMERIK LanguageID with the OPC UA specific LocaleId for each of the supported languages.

Language	SINUMERIK LanguageID	OPC UA Specific LocaleId
German - Germany	deu	de-DE
English - United Kingdom	eng	en-GB
Chinese (Simplified)	chs	zh-CHS
Chinese (Traditional)	cht	zh-CHT
Czech - Czech Republic	csy	cs-CZ

Language	SINUMERIK LanguageID	OPC UA Specific LocaleId
Danish – Denmark	dan	da-DK
Bulgarian - Bulgaria	bgr	bg-BG
Greek – Greece	ell	el-GR
Spanish – Spain	esp	es-ES
Finnish – Finland	fin	fi-FI
French – France	fra	fr-FR
Hindi – India	hin	hi-IN
Croatian – Croatia	hrv	hr-HR
Hungarian – Hungary	hun	hu-HU
Indonesian – Indonesia	ind	id-ID
Italian – Italy	ita	it-IT
Japanese - Japan	jpn	ja-JP
Korean – Korea	kor	ko-KR
Malay – Malaysia	msl	ms-MY
Dutch - The Netherlands	nld	nl-NL
Polish – Poland	plk	pl-PL
Portuguese - Brazil	ptb	pt-BR
Romanian - Romania	rom	ro-RO
Russian – Russia	rus	ru-RU
Slovak – Slovakia	sky	sk-SK
Slovenian – Slovenia	slv	sl-SI
Swedish – Sweden	sve	sv-SE
Tamil – India	tam	ta-IN
Thai – Thailand	tha	th-TH
Turkish – Turkey	trk	tr-TR
Vietnamese - Vietnam	vit	vi-VN

In the above list “OPC UA Specific LocaleId” is used by the OPCU UA client to connect with the server.

7.4.7 OPC UA alarms and conditions constraints

Below are the features which are not supported in this version:

- Acknowledgements and confirmation of the alarms.
- Only alarm text will be available in localized text. All other attributes will be available in English only.

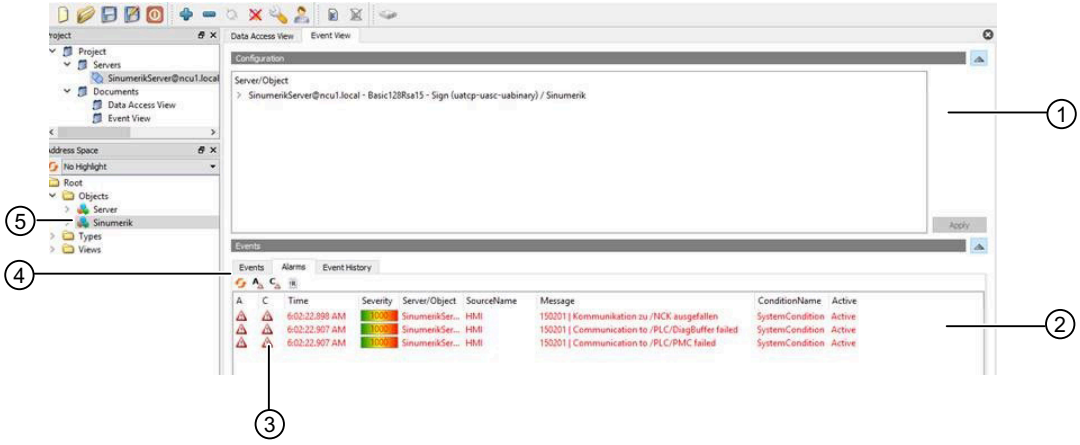
See also

General OPC UA settings (Page 50)

7.4.8 OPC UA alarms and conditions client

User interface

The figure and table below describe the user interface of the UaExpert client example with which the information of the namespace of an OPC UA server can be conveniently accessed.



① The Alarm window

② Displays the received events with preconfigured event fields. The standard event fields are:

- In the Events tab: Time, ReceiveTime, Severity, SourceName, Message, EventType and SourceNode
- In the Alarms tab: AcknowledgeState, Time, Severity, SourceName, Message, ConditionName, ActiveState and Retain Flag

③ In the first column of the Alarm tab, a symbol indicates whether an event has already been Acknowledged. (red flag: unacknowledged, green checkmark: acknowledged)

④ The Alarm / Event Subscription View

⑤ Alarm / Event Instances:

The user needs to subscribe to these instances (by dragging or by configuring).

Figure 7-14 User interface UaExpert client

7.4.9 OPC UA multi-language alarms and conditions client

The OPC UA client must explicitly provide the OPC UA specific language "LocaleId" to change the alarm texts. Below is an example of changing the client language using OPC UA foundation stack client.

```
//Create and connect session
var preferredLocalesList = new List<String>();
preferredLocalesList.Insert(0, "de-DE");

Session  mSession = Session.Create(
    ApplicationConfig,
    mEndpoint,
    true,
    "MySession",
    60000,
    UserIdentity,
    preferredLocalesList //preferred locale list
    );
```

Figure 7-15 OPC UA multi-language alarms and conditions client using OpcUa foundation .Net Client

In the case of UaExpert client proceed as follows:

1. Open the “Configure UaExpert” window under "Settings" Tab in the client
2. Provide the OPC UA specific "LocaleId" as value for the parameter "General.LocaleId".
3. Then connect to the server.

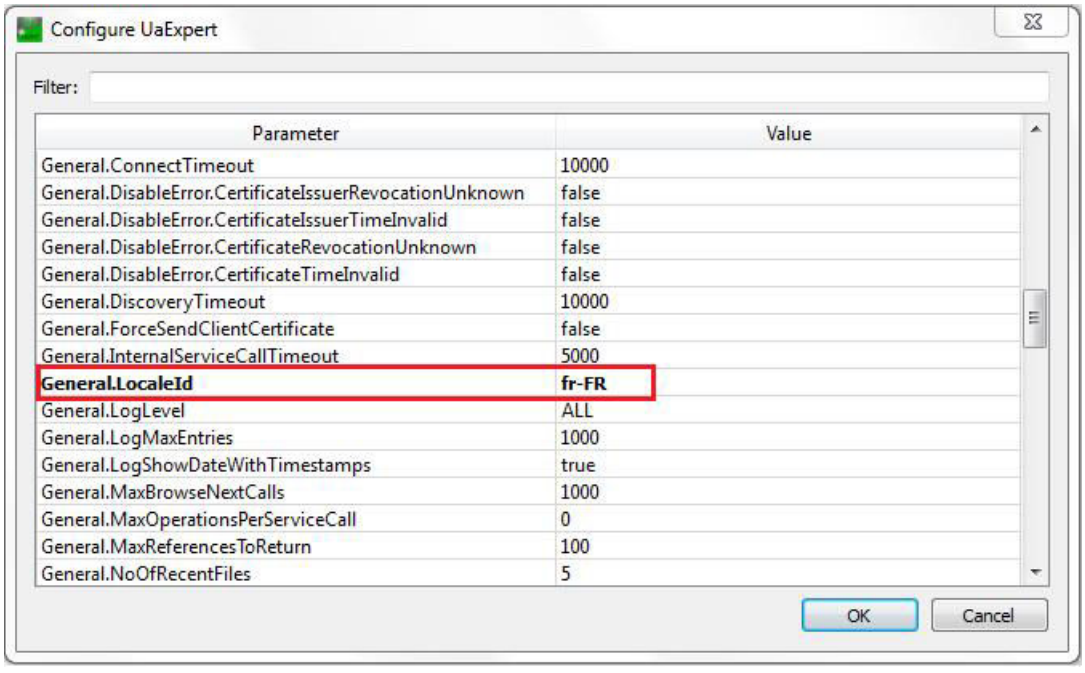


Figure 7-16 Client User Interface for changing Session Language

Language	OPC UA Specific LocaleId
German - Germany	de-DE
English - United Kingdom	en-GB
Chinese (Simplified)	zh-CHS
Chinese (Traditional)	zh-CHT
Czech - Czech Republic	cs-CZ
Danish – Denmark	da-DK
Bulgarian - Bulgaria	bg-BG
Greek – Greece	el-GR
Spanish – Spain	es-ES
Finnish – Finland	fi-FI
French – France	fr-FR
Hindi – India	hi-IN
Croatian – Croatia	hr-HR
Hungarian – Hungary	hu-HU
Indonesian – Indonesia	id-ID
Italian – Italy	it-IT
Japanese - Japan	ja-JP
Korean – Korea	ko-KR
Malay – Malaysia	ms-MY
Dutch - The Netherlands	nl-NL
Polish – Poland	pl-PL
Portuguese - Brazil	pt-BR
Romanian - Romania	ro-RO
Russian – Russia	ru-RU
Slovak – Slovakia	sk-SK
Slovenian – Slovenia	sl-SI
Swedish – Sweden	sv-SE
Tamil – India	ta-IN
Thai – Thailand	th-TH
Turkish – Turkey	tr-TR
Vietnamese - Vietnam	vi-VN

7.5 File system

7.5.1 Overview

SINUMERIK OPC UA supports the standard OPC UA file and folder objects, which allows transfer of files as well as the manipulation of the file systems.

Furthermore, the server offers 2 comfort methods to copy NC part programs from the OPC UA client to the OPC UA server and vice versa. Due to the nature of the method this comfort

methods are limited to a file size of 16 MB. For bigger files please use the file and folder objects as described in chapter File transfer exceeding 16 MB between client and server (Page 148).

Operations

This allows an OPC UA client to use the following operations within the part of the SINUMERIK file system:

1. Create files/directories
2. Copy files/directories
3. Moving files/directories
4. Deleting files/directories
5. Renaming files/directories

File system

The standard OPC UA file system is placed in the SINUMERIK folder and the file structure of the NCU is as shown below:

1. Part Programs
2. Sub Programs
3. Work Pieces
4. NCExtend (External SD Card/internal SD Card)

   Note

   For NCU, external SD Card/internal SD Card is supported.

   For IPC, SSD/Harddisk is supported.

5. ExtendedDrives (USB/Networkshare)

   Note

   The ExtendedDrives folder will only be displayed if there are external drives available. Please keep in mind that a licence my be required to use these external drive.

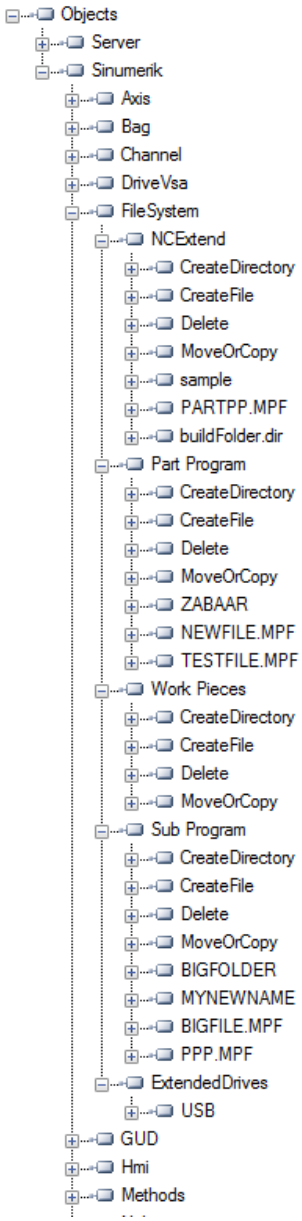


Figure 7-17 The file system

7.5.2 Prerequisites

The OPC UA server allows the OPC UA client to support the transfer of files between the client and the server.

As a user, you will require user access rights to access these files from the server. The access rights are provided using the "GiveUserAccess" method. The following access rights can be provided for the file system (also see chapter List of rights):

- FsRead for the standard file system methods like Open, GetPosition, Read as well as the CopyFileFromServer method.
- FsWrite for the standard file system methods like CreateDirecotry, CreateFile, Delete, MoveOrCopy, Write, SetPosition, Close as well as the CopyFileToServer method.

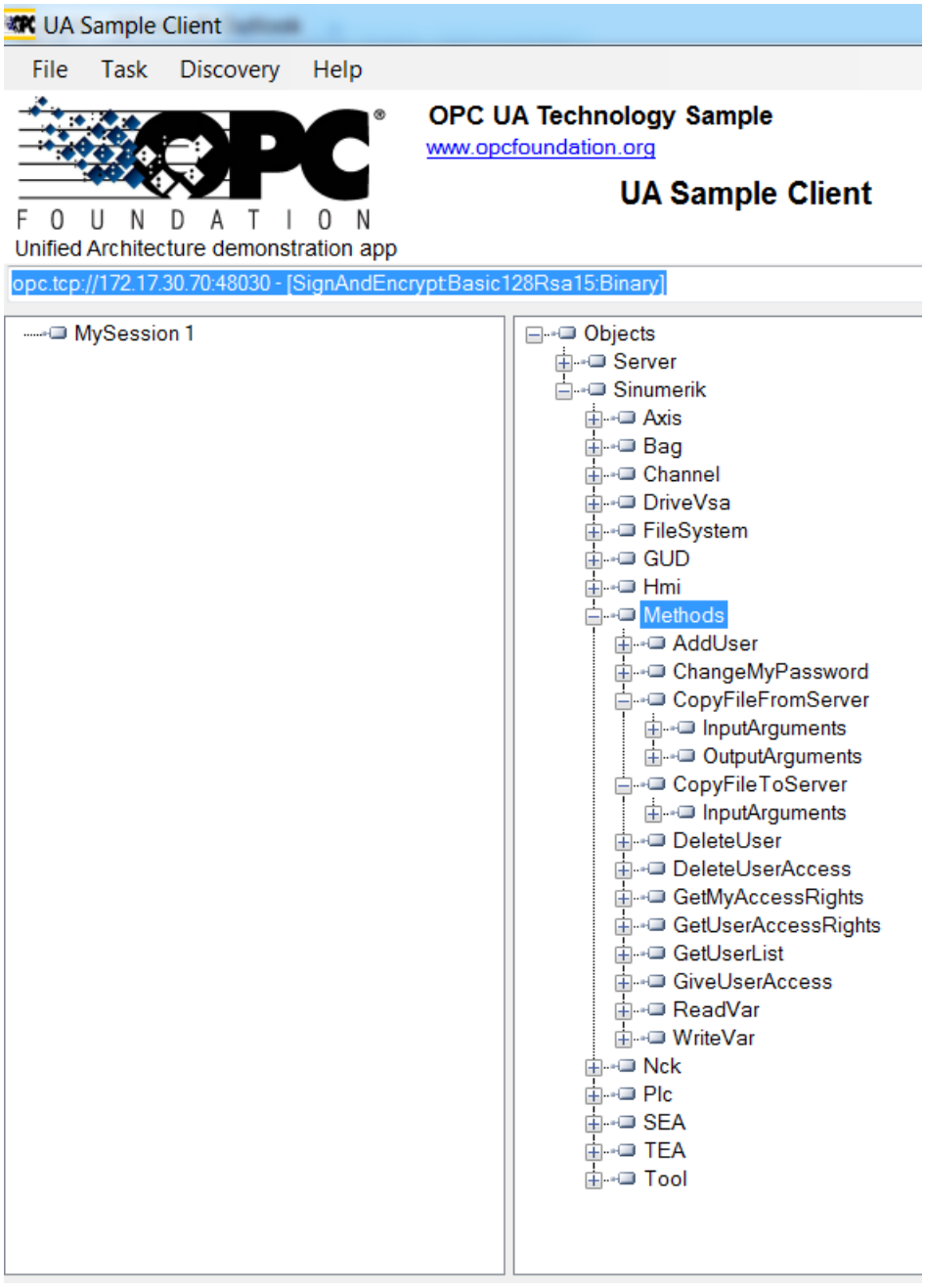


Figure 7-18 Standard Method

7.5.3 Standard file system support

7.5.3.1 File transfer with standard methods

The SINUMERIK OPC UA server supports the "FileType"/"FolderType" as described in the OPC UA Specification Part 5, which allows manipulating files and folders via OPC UA.

Folder methods

The folders, "Part Programs", "Sub Programs", "Work Pieces", "NCExtend", "ExtendedDrives", and "AdvExchange" are of the "FolderType", which contain the following methods:

Note

Copy/Move folder without .dir extension

OPC UA does not support Copy/Move action of a folder wihout .dir extension to the Part program memory using Copy or Move method.

Rename the folder with .dir extension and use Copy or Move method in OPC UA.

Note

- The "AdvExchange" folder is visible only when the OPC UA Advanced license is activated.
- File system rights are required to access AdvExchange folder.
  - File system read to view the AdvExchange folder's content.
  - File system write to delete the Adv Exchange folder's content.

Method/Attribute	Description
CreateDirectory	To create new folders under parent folder.
CreateFile	To create new file under parent folder.
Delete	To delete folder and file under parent folder.
MoveOrCopy	To copy or move files from source to destination within server filesystem.

You can create, delete, move or copy folders and files using the above methods. When you create a new folder using "CreateDirectory"', a new node will be created with "FolderType" and name provided by the user in OPC UA client. This folder contains all methods and attributes specified in above table.

The node in the address space, under which the "CreateDirectory" method is called, is the "parent" node of the new folder node.

The folder methods exist under all folder type objects in the file system. Please note that you must always call the method under the direct parent node of the file or folder.

For the methods "Delete" and "MoveOrCopy" you must always provide the full identifier of the node to be moved, copied or deleted.

Whenever you create a new file using the method "CreateFile", a new node will be created with "FileType" with a user provided name. This file again contains all methods and attributes specified in the table above. The node in address space, under which the "CreateFile" method

is called, is the "parent" node of the new file node. For specific information for the described methods, check the Typedefinition in the OPC UA Specification Part 5.

Examples for the usage of the folder methods

Name	Signature	Usage
CreateDirectory	[in] String directoryName [out] NodeId directoryNodeId	Call from parent folder/directory, e.g.: Sinumerik/FileSystem/Work Pieces
CreateFile	[in] String fileName [in] Boolean requestFileOpen [out] NodeId fileNodeId [out] UInt32 fileHandle	Call from parent folder/directory, e.g.: Sinumerik/FileSystem/Part Program File-name including extension, e.g.: myPart-Prog.mpf
Delete	[in] NodeId objectToDelete	Call from parent folder/directory, e.g.: Sinumerik/FileSystem/Part Program
MoveOrCopy	[in] NodeId objectToMoveOrCopy [in] NodeId targetDirectory [in] Boolean createCopy [in] String newName [out] NodeId newNodeId	Call from parent folder/directory of object to move/copy, e.g.: copy a part program the parent folder is: Sinumerik/File-System/Part Program

Note

For further details on methods and method signatures please refer to OPC UA Specification Part 5.

File methods

All files which are in the above mentioned folders will be of the "FileType" type and consist of the following methods and properties:

Method/Attribute	Description
Open	Opens the file either in read/write mode.
Read	Reads contents of the file.
Write	Writes data to the file. (if write permission is available)
Close	Closes the file. (succeeds if file is open)
GetPosition	Gets the position of current position of file pointer while file read/write operation.
SetPosition	Sets the position of current position of file pointer while file read/write operation.
OpenCount	Gives the number of file open instances.
Size	Gives the file size details.
UserWritable	Set to true if current user has access to modify the content of the file.
Writable	Set to false if the file is read only.

Whenever you create a new file using the method "CreateFile", a new node will be created with "FileType" type with a user provided name. This file again contains all methods and attributes specified in the table above. The node in address space, under which the "CreateFile" method is called, is the "parent" node of the new file node. For specific information for the described methods, check the type definition in the OPC UA Specification Part 5 Annex C.

Note

No multiple extensions supported

The methods "CreateFile","CopyFileToServer", "CopyFileFromServer" and "MoveOrCopy" will not support files with multiple extensions (i.e. test.mpf.mpf).

7.5.3.2 File transfer exceeding 16 MB between client and server

For file transfer, the OPC UA specification v1.0X, Part 5, Annex C offers the use of file and folder object methods.

How can a file transfer be implemented in a client using the file and folder object methods?

The basic idea is to open a file and copy the content from location A to B and then close the file.

Therefore to copy a file from the SINUMERIK to an OPC UA client the client needs to do the following:

- Open the file on the SINUMERIK via the Open method,
- then pass the received handle to the Read method and operate on arrays of bytes,
- then close the file via the Close method.

For the other direction, the file has to be created on the SINUMERIK file system first, using the Create method.

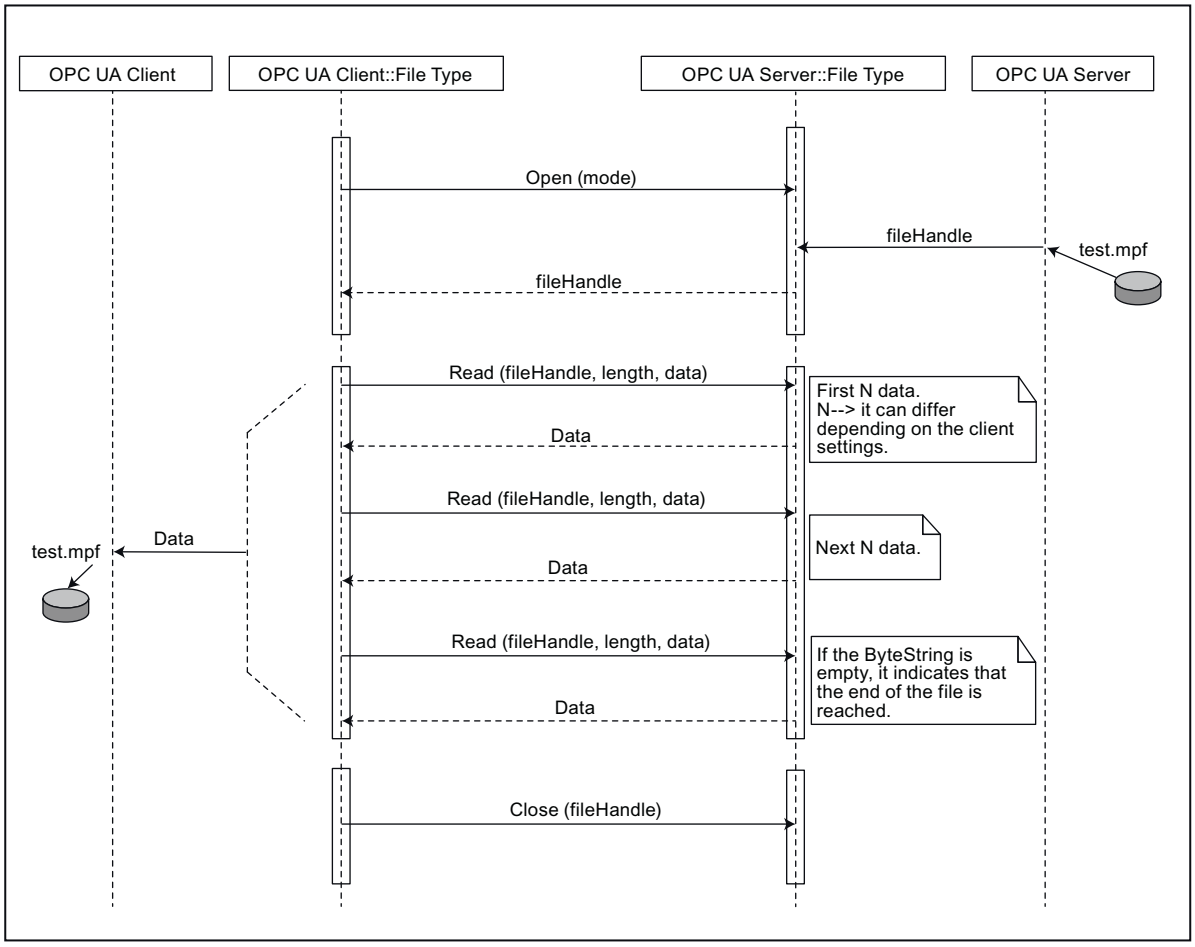


Figure 7-19 File transfer from server to client using standard file system methods

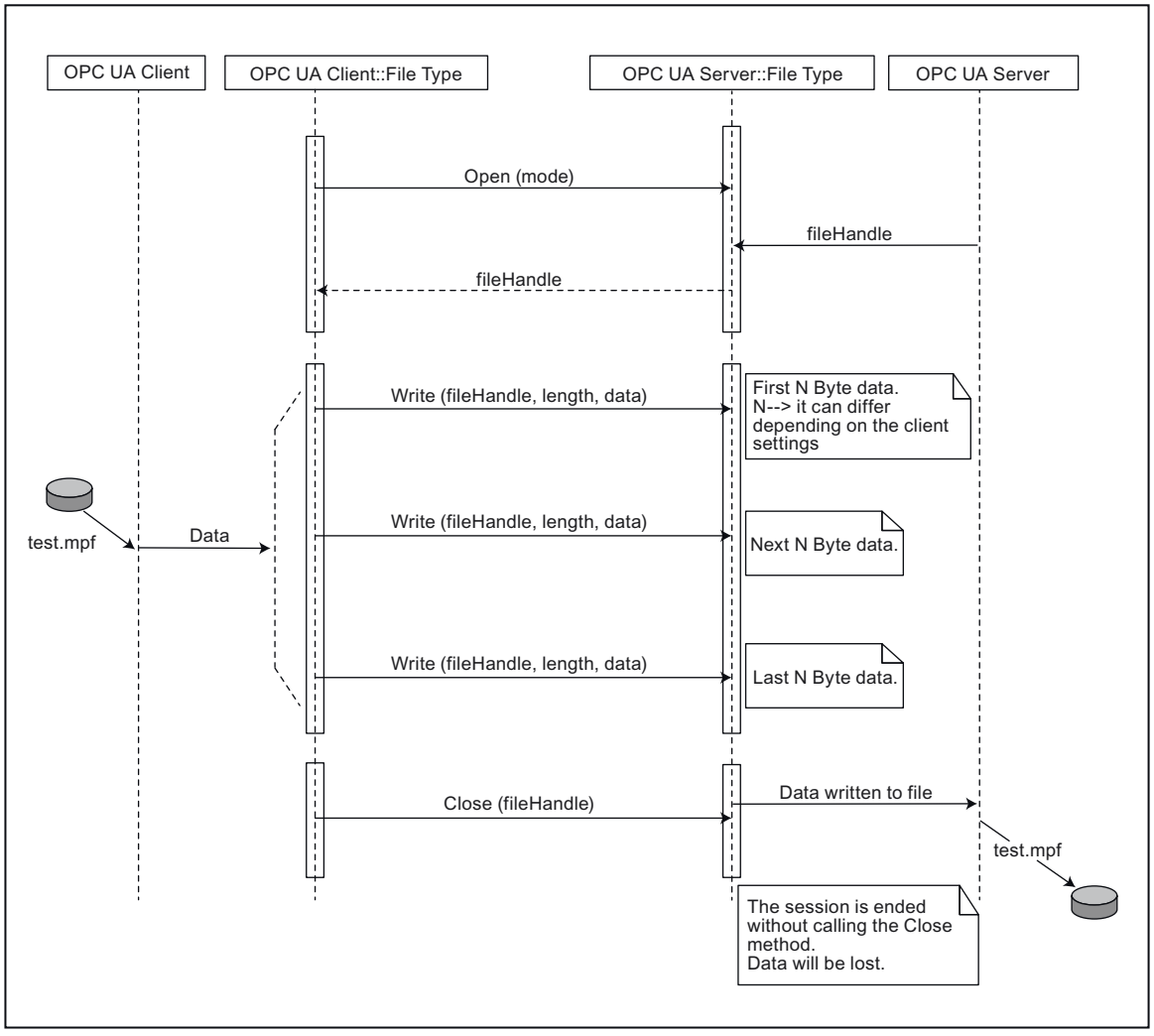


Figure 7-20 File transfer to server from client using standard file system methods

7.5.3.3 Comfort methods for file transfer < 16 MB

In addition to the standard file system, two additional methods are provided to transfer files from server to client and vice versa.

Note

With these methods, you will be able to transfer maximum of 16 MB by default, depending on the client settings. As the maximum ByteString and message size depends on the server and client-side stack limit. For file transfer more than 16 MB, it is recommended to use OPC UA standard file transfer methods.

Files without extension are not supported.

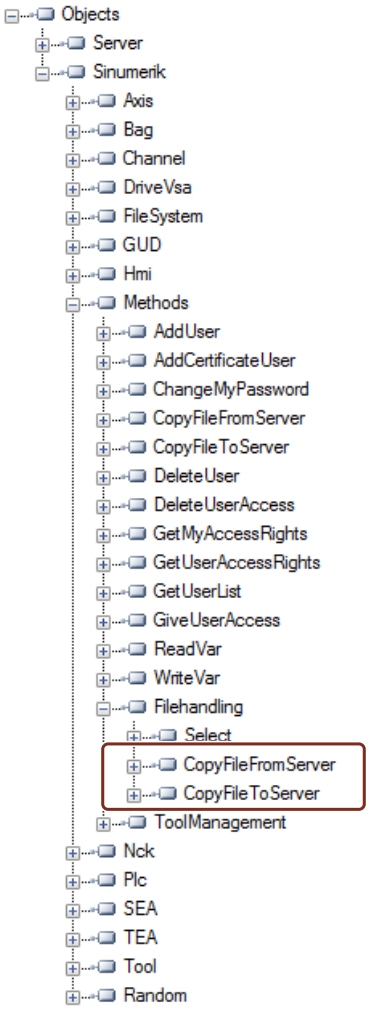


Figure 7-21 Methods for the file transfer

Procedure and Example

1. CopyFileFromServer:
   – Allows copying file from SINUMERIK OPC UA server to client location.
   – The user shall provide the name of the file with full path to be copied.
   – On completion of the file transfer, an appropriate message will be displayed.

Type	Data type	Argument	Description
Input parameter	string	SourceFile	Name of the file need to be copied with absolute path.
Output parameter	ByteString	Data	Raw file data

2. CopyFileToServer:
   – Allows copying a client file to a specified SINUMERIK NC memory location.
   – The user shall select the file to be transferred and specify the location on server.

Type	Data type	Argument	Description
Input parameter	string	TargetFilename	Target file name with absolute path
Input parameter	ByteString	Data	Raw file Data
Input parameter	Boolean Overwrite	Overwrite	True: Overwrite the file if already exists. False: File will not be overwritten.

For example:

The complete path of the files can be provided as below:

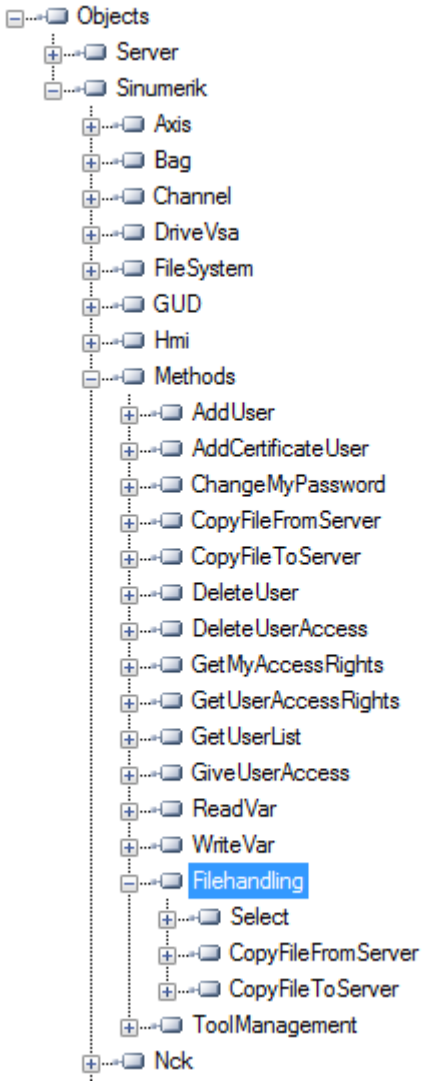
- Sinumerik/FileSystem/Part Program/partprg.mpf
- Sinumerik/FileSystem/Sub Program/subprg.spf
- Sinumerik/FileSystem/Work Pieces/wrkprg.wpf
- Sinumerik/FileSystem/NCExtend/Program.mpf
- Sinumerik/FileSystem/ExtendedDrives/USBdrive/Q3.mpf

7.6 Select

7.6.1 Overview

The "Select" method is provided under "Methods > Filehandling" in the address space, which allows the selecting of a part program from the NC file system. You can call this method and select the file to be executed by providing the node identifier of the file in address space and the channel number.

By calling this method, you can only select the program for execution and not start the execution of the program itself.



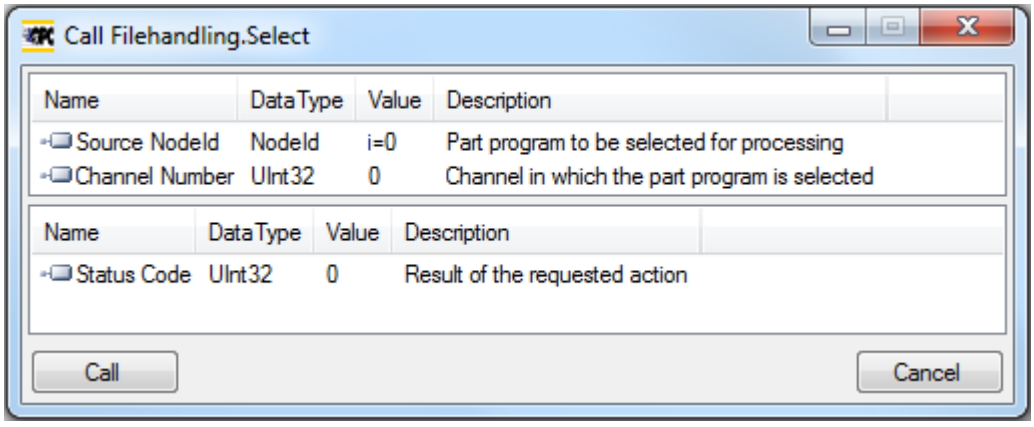
7.6.2 Description

You are allowed to select the part program file for execution from the NC file system and external memory, which includes "local drive", "USB" and "network share". As part of the file system feature, the NC file system is exposed in the OPC UA address space.

There are two input values to be provided to call the "Select" method.

- Node identifier of the file to be selected for execution.
- Channel number.

Each part program file on the file system is associated with a node identifier in the OPC UA address space and is provided as the input. Only one part program can be selected for a channel. An error will be displayed otherwise.



Status code is an output parameter which indicates the error code in case of failures.

7.6.3 Input and output arguments

Signature of the method "Select" is as follows:

```
Select (
[in] string SourceFileNodeId,
[in] int32 ChannelNumber,
[out] int32 Status Code)
```

Argument	Description
SourceFileNodeId	Represents the node identifier of the file with absolute path (which is selected for execution).
Channel Number	A number which represents the channel to be used while program execution.

Prerequisites

- Channel to be used during program execution must be in the state "Reset".
- User with "**ApWrite**" access right can call "Select" method. If the user does not have the access "**ApWrite**" and tries to call "Select" method, it fails and server will return with OpcUa status "**BadUserAccessDenied**".

Note

The access right for the user is provided using the "GiveUserAccess" method.

Status Code of the method call

The following table gives details on values and description on the status of the "Select" method call. As part of output argument, the result code (value) is displayed in the OPC UA client.

Status Code (value)	Description
0	Successful
1	Channel does not exist
2	Part Program cannot be found
3	Channel is not in Reset
4	Target rejected requested action.

Note

No file restriction

Notice that a file with any extension is allowed to be selected through OPC UA "Select" method. OPC UA does not restrict selecting files with any file extension.

Joblists cannot be selected.

OPC UA Status

The following table gives details on values and description of the OPC UA method call status:

Result	Description
Succeeded	Method is executed with success/failure.
OpcUa_BadInvalidArgument	Invalid inputs are provided.
OpcUa_BadUserAccessDenied	User does not have permission to invoke the method.

7.6.4 Example call

Procedure

1. Look for the NodeID of the particular part program you want to select (for example "NC_PROG1.MPF").
2. Navigate in the "File System" node until you reach the particular file.

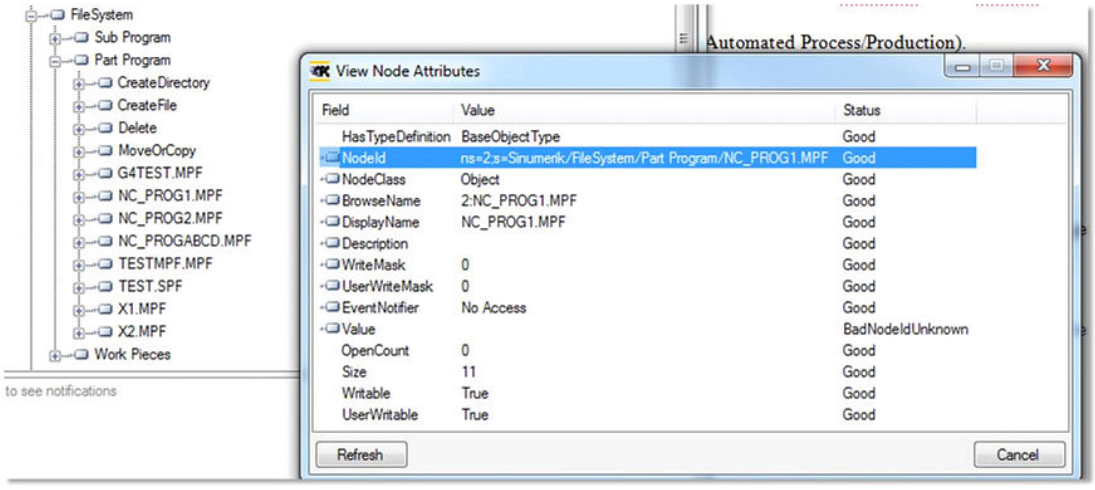


Figure 7-22 Finding of NodeID

3. Specify the NodeID and the channel number in the call of the method.

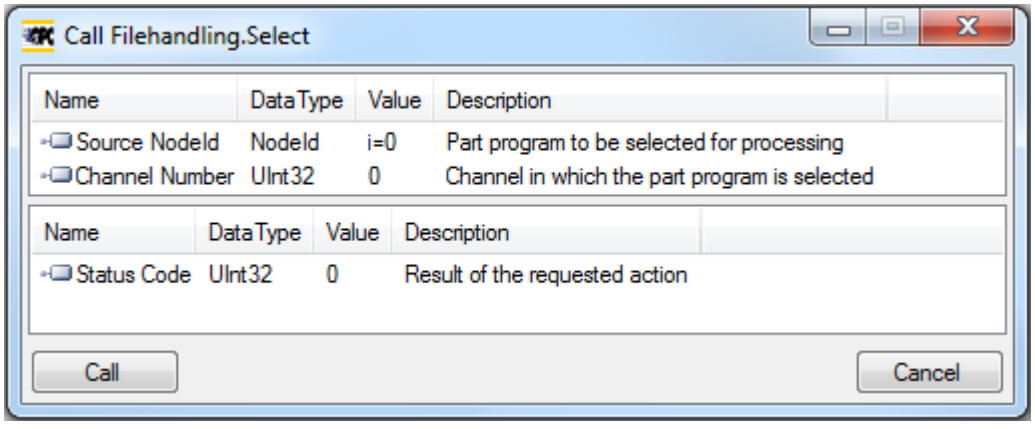


Figure 7-23 Arguments of select method

The particular part program will be selected.

7.7 Unselect

The Unselect method is provided under "Sinumerik > Methods > Filehandling" in the address space, which allows to unselect a part program per channel. You can call this method by providing the channel number.

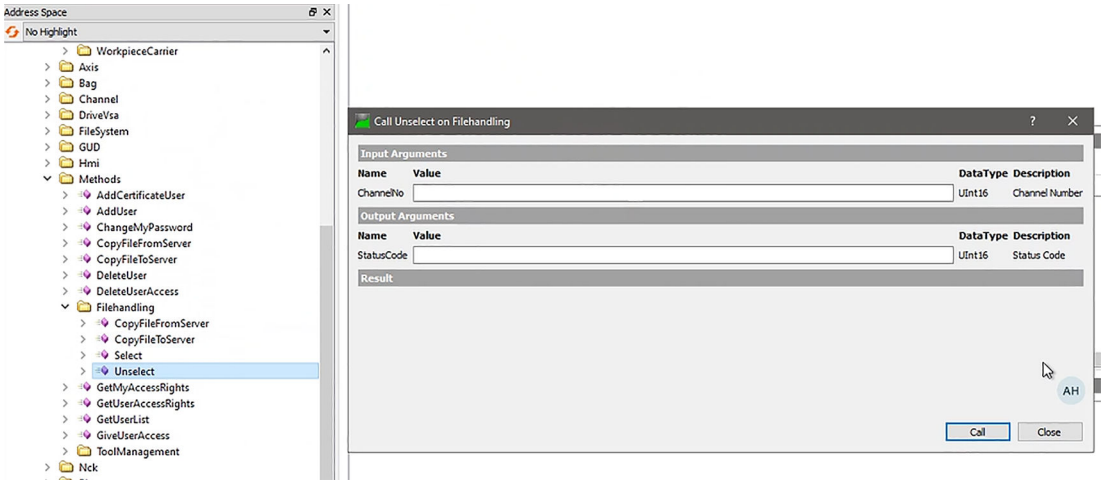


Figure 7-24 Calling Unselect method

Structure and arguments

Signature of the method Unselect is as follows:

```
OpcUa_StatusCode Unselect(OpcUa_UInt16 channelNo, OpcUa_UInt16
statusCode)
```

Argument	Description
channelNo [in]	Number of the channel

As an additional output argument, the OPC UA Status code will be provided.

OPC UA Status code

The following table gives details on values and description of the OPC UA method call status:

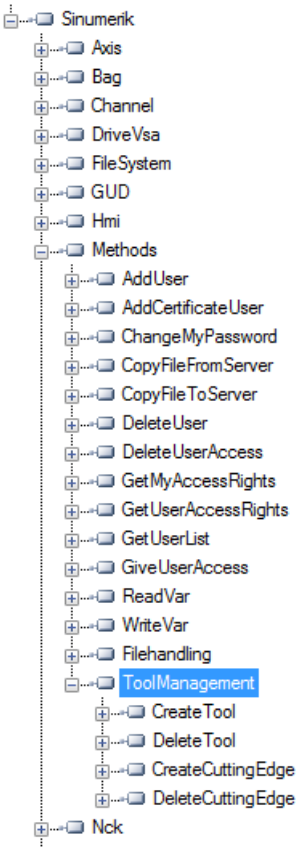
Result	Description
0	good
1	Channel not found/available
2	No program selected
3	Program running/not reset
4	Request rejected
5	Unknown error

7.8 Tool management

7.8.1 Description

The OPC UA server supports the creation and deletion of tools and cutting edges. The methods for this operation can be found under "Sinumerik > Methods > ToolManagement" folder. Following are the four methods present in "ToolManagement" folder:

- CreateTool
- DeleteTool
- CreateCuttingEdge
- DeleteCuttingEdge



Example calls

For example calls of the provided methods, please refer to the shown screenshots of OpcFoundation Client.

Prerequisites

User with "ToolWrite" access right can call "ToolManagement" methods. If the user does not have the access "ToolWrite" and tries to call "ToolManagement" methods, it fails and server will return with OpcUa status "**BadUserAccessDenied**".

Note

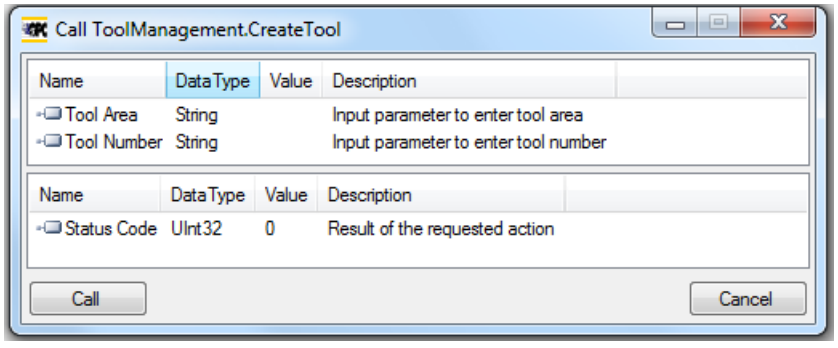
The access right for the user is provided using the "GiveUserAccess" method.

7.8.2 CreateTool

The "CreateTool" method is used to create a new tool with a special T-number in Tool List section of the SINUMERIK, and appears under the folder “Methods/ToolManagement”. The CreateTool method does not contain the settings of tool parameters. The tool parameters e.g.: tool type, cutting edge date etc. are set via data access functions.

The CreateTool method has two input parameters and one output parameter.

```
Signature:
CreateTool(
[in] string ToolArea
[in] string ToolNumber
[out] Uint32 StatusCode
)
```



The following table will give details about the parameters of the method:

Type	Parameters	Description
Input	Tool Area	Input parameter to enter tool area.
Input	Tool Number	5 digit number given to the created tool. For range of number please refer to 828D or 840D sl documentation respectively.
Output	Status Code	A number which gives a feedback if the method was executed successfully or not.

The method returns a value which indicates whether the creation was successful or not. If the creation was not successful the return value will give information about the reason of the failure.

Status code

The status code is the result of the requested action which is a number as shown in the table below:

Status Code	Reason
0	OK.
1	Tool area does not exist.
2	Tool number out of range.(Reason wrong parameter)
3	Tool number exists already.
4	Maximum number of tools reached.

Method Result Codes

Result	Description
Succeeded	Method executed with success/failure reason.
BadInvalidArgument	Arguments provided are not correct.
BadUserAccessDenied	"ToolWrite" access is not provided.

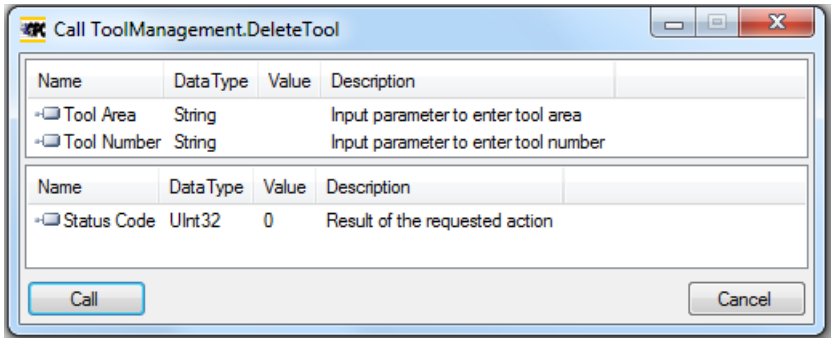
7.8.3 DeleteTool

The "DeleteTool" method is used to delete an existing tool in Tool List section of the SINUMERIK, and appears under the folder “Methods/ToolManagement”.

The method deletes the tool with all cutting edges in all data blocks where it is saved.

```
DeleteTool(
[in] string ToolArea
[in] string ToolNumber
```

```
[out] Uint32 StatusCode
)
```



The following table will give details about the Parameters of the method:

Type	Parameters	Description
Input	Tool Area	Input parameter for the end user to enter tool area.
Input	ToolNumber	5 digit number which is to be deleted. For range of number please refer to 828D or 840D sl documentation respectively.
Output	StatusCode	A number which gives a feedback if the method was executed successfully or not.

The method returns a value which indicates whether the delete was successful or not. If the delete was not successful the return value will give information about the reason of the failure.

Status code

If the deletion of the tool was not successful the return value will give information about the reason of the failure which are explained in the table below.

StatusCode	Description
0	OK.
1	Tool area does not exist.
2	Tool number out of range.(Reason wrong parameter)
3	Tool does not exist.
6	Tool active.(Reason tool in use)

Method Result Codes

The Result return "Succeeded" when the method is correctly executed and the *StatusCode* gives the reason of Success/Failure.

It returns "BadInvalidArgument", if inputs are not according to OPC UA standards.

Result	Description
Succeeded	Method executed with success/failure reason.
BadInvalidArgument	Arguments provided are not correct.
BadUserAccessDenied	"ToolWrite" access is not provided.

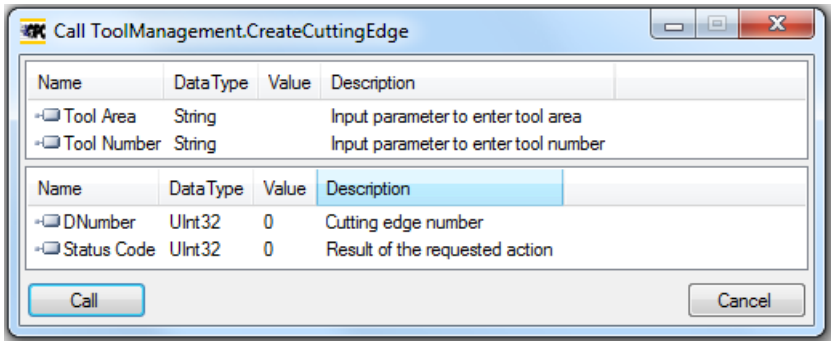
7.8.4 CreateCuttingEdge

The "CreateCutting Edge" method is used to create a new cutting edge of an existing tool in “Tool List” section of the SINUMERIK. The next superior free D number will be created.

The ”CreateCuttingEdge” method appears under the folder “Methods/ToolManagement”. This method does not contain the settings of cutting edge parameters.

The CreateCuttingEdge method has two inputs and two output parameters.

```
Signature:
CreateCuttingEdge(
[in] string ToolArea
[in] string ToolNumber
[out] Uint32 DNumber
[out] Uint32 StatusCode
)
```



The following table will give details about the parameters of the method:

Type	Parameters	Description
Input	Tool Area	Input parameter to enter tool area.
Input	Tool Number	5 digit number which is to be deleted. For range of number please refer to 828D or 840D sl documentation respectively.

Type	Parameters	Description
Output	DNumber	Cutting Edge Number of the tool.
Output	Status Code	A number which gives a feedback if the method was executed successfully or not.

The method returns a value which indicates whether the creation was successful or not. If the creation was successful the DNumber under which the new cutting edge was created will be returned. If the creation was not successful the return value will give information about the reason of the failure.

Status code

The status code is the result of the requested action and is represented by a number, as shown in the table below:

Status Code	Reason
0	OK.
2	Tool number out of range.
4	Maximum cutting edges reached no more cutting edges.
5	There is no tool for which edge can be created. (Reason wrong tool area or tool number)

Method Result Codes

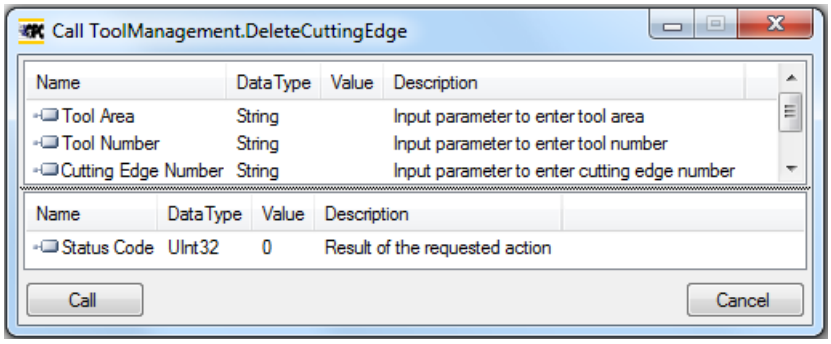
Result	Description
Succeeded	Method executed with success/failure reason.
BadInvalidArgument	Arguments provided are not correct.
BadUserAccessDenied	"ToolWrite" access is not provided.

7.8.5 DeleteCuttingEdge

The "DeleteCuttingEdge" is used to delete a cutting edge of an existing tool in "Tool List" section of the SINUMERIK. This method appears under the folder "Methods/ToolManagement".

The DeleteCuttingEdge method has three input and one output parameters.

```
Signature:
DeleteCuttingEdge(
[in] string ToolArea
[in] string ToolNumber
[in] string CuttingEdgeNumber
[out] Uint32 StatusCode
)
```



Following table will give details about the Parameters of the method:

Type	Parameters	Description
Input	Tool Area	Input parameter to enter tool area.
Input	Tool Number	Tool number of an existing tool whose cutting edge is to be deleted.
Input	Cutting Edge Number	5 digit number which is to be deleted. For range of number please refer to 828D or 840D sl documentation respectively.
Output	Status Code	A number which gives a feedback if the method was executed successfully or not.

The method should return a value which indicates whether the delete was successful or not. If the delete was not successful the return value should give information about the reason of the failure.

Status code

The status code is the result of the requested action which is a number as shown in the table below:

Status Code	Reason
0	OK
2	Tool number out of range.
4	Cutting edge does not exist.
5	There is no tool for which edge can be deleted (Reason wrong tool area or tool number)
6	Tool active. (Reason tool in use)
7	The first cutting edge cannot be deleted.

Method Result Codes

Result	Description
Succeeded	Method executed with success/failure reason.
BadInvalidArgument	Arguments provided are not correct.
BadUserAccessDenied	"ToolWrite" access is not provided.

OPC UA Advanced

8

8.1 Introduction

The SINUMERIK Access MyMachine /OPC UA - Advanced enables you to further automate production processes, like pallet handling and tool management with CNC controls in your production system. OPC UA Advanced in SINUMERIK ONE is the successor of SINUMERIK CMI (SINUMERIK Create MyInterface). The communication between the CNC control and the host computer of the production system takes place via a defined, open communication interface. The interface to the PLC is similar to CMI via a PLC interface, which is defined in this documentation.

The OPC UA Advanced option (Access MyMachine /OPC UA - ADV) is an add-on license to SINUMERIK AccessMyMachine /OPC UA. To use the OPC UA Advanced functionalities, you require SINUMERIK AccessMyMachine /OPC UA - ADV license in addition to SINUMERIK AccessMyMachine /OPC UA.

8.2 Overview

With OPC UA Advanced option the following functionalities are available for integration into an automation environment:

- Machine status
  - Event based notification about machine coordinate system status changes
- Workpiececarrier handling
  - Event based notification about workpiece carrier status changes
  - Workpiece carrier status can be set by the host via a method
- Tool management
  - Event based notification about tool status changes
  - Method based access to the data of an individual tool
  - Complete tool data can be provided via an OPC UA method / event.
  - Complete magazine data can be provided via an OPC UA method / event.
  - Tools can be loaded / unloaded using Tool Ident Connection as similar to CMI.
  - Tool create and delete via an OPC UA method / event (without Tool Ident Connection)
  - Force unload tool status will be provided without host computer intervention.
- Selected part program name can be read via a method.

The active data exchange between the host computer and the CNC controls also enables the following:

- Higher machine utilization through production planning, control and analysis
- Reduction of machine downtimes through transmission of maintenance data
- Logging

These advantages contribute to increased productivity.

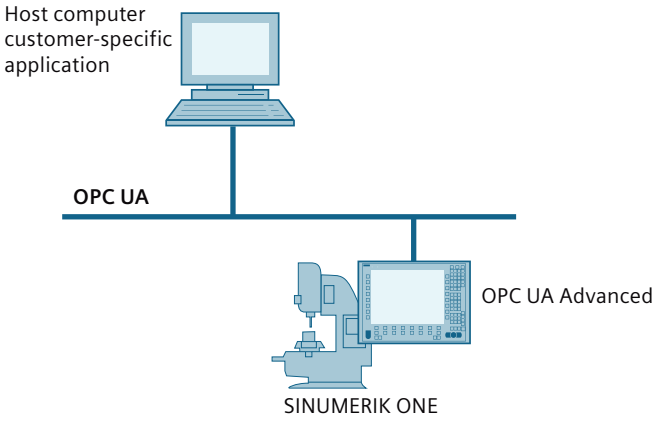


Figure 8-1 Application scenario - OPC UA Advanced

8.3 Prerequisites

- OPC UA license Access MyMachine /OPC UA (6FC5800-0BP67-0YB0)
- OPC UA Advanced license Access MyMachine /OPC UA - ADV (6FC5800-0BP85-0YB0)
- Tool Ident Connection (6FC5800-0BP52-0Yx0/MCS513)

8.4 General interaction mechanism

OPC UA Advanced functionality is based on methods and events to ensure simple integration in the host applications. The following figure shows an example of the information flow between OPC UA client and OPC UA server in connection with the MachineStatusEvent:

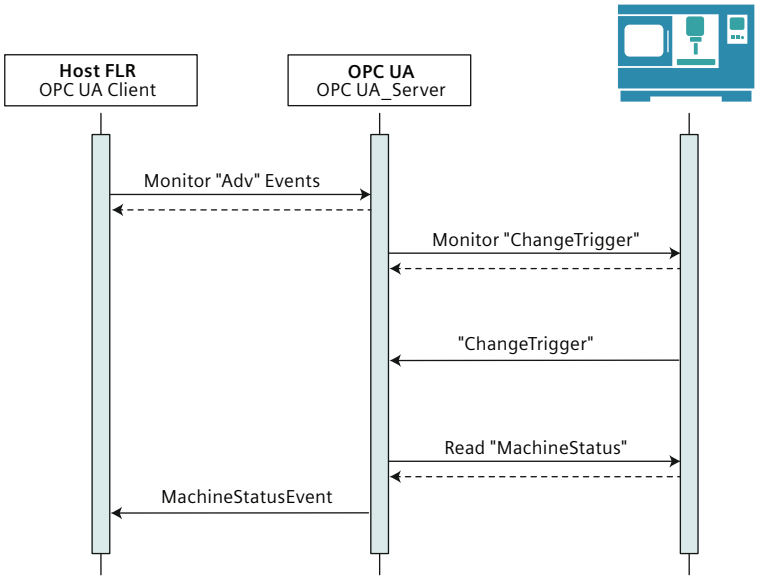


Figure 8-2 General interaction mechanism

8.5 OPC UA Advanced node

After the OPC UA Advanced option is enabled, the "Adv" node and the "AdvExchange" node are available under the SINUMERIK address space. The "Adv" node contains the following sub-nodes:

- Machine
- ToolManagement
- WorkpieceCarrier

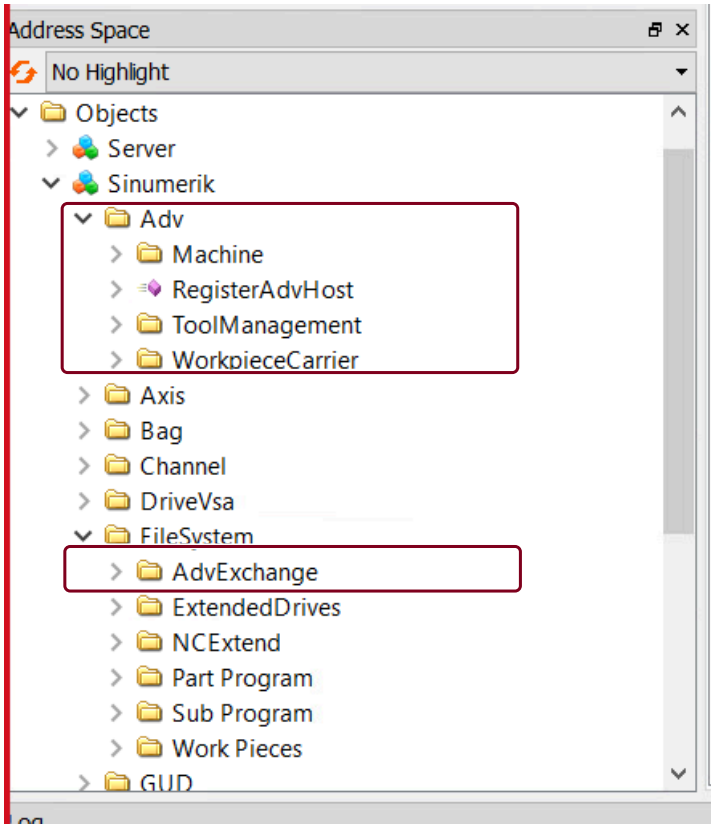


Figure 8-3 Adv node

8.6 OPC UA Advanced use cases

8.6.1 Overview

The following use cases help you to find your individual use, the corresponding solution OPC UA offers, the trigger which is needed for starting the use case and the result which comes out. If you like more detailed information about the triggering events and methods, you can refer to linked chapter.
The logical interaction sequence of each OPC UA Advanced use case is described in the further course of this chapter.

OPC UA Advanced use cases

Use case	Solution OPC UA	Trigger	Result
Machine status	• MachineStatusEvent for changes in Machine Status (triggered by PLC or method) • Method to request MachineStatusEvent	• RequestMachineStatusEvent() (Page 267)	• MachineStatusEvent (Page 236)
Tool management Information	• ReportToolEvent for changes in Tool Management (triggered by PLC)	• RequestToolDataByMagazinPlace() (Page 272) • RequestToolDataByToolName() (Page 273) • RequestToolDataByToolNo() (Page 275)	• ReportToolEvent (Page 242) • ReportToolErrorEvent (Page 246)
	• Method to actively request tool data via host systemOutput is an event which indicated that a file is available to read by host system	• RequestToolDataFile() (Page 277) • RequestMagazinDataFile() (Page 276)	• DataFileEvent (Page 248)
Tool load	• Method to load a tool (only with ToolIdent) Tool data is provided as a string	• AddToolRequestEvent() (Page 250) • AddToolResponse() (Page 278)	• ReportToolEvent (Page 242) • ReportToolErrorEvent (Page 246)
Tool unload	• Method to unload a tool (only with ToolIdent) Tool data is provided as a string	• RemoveToolRequestEvent() (Page 253) • RemoveToolResponse() (Page 280)	• ReportToolEvent (Page 242) • ReportToolErrorEvent (Page 246)
Tool creation	• Method to create a tool	• AddToolRequestEvent() (Page 250) • AddToolResponse() (Page 278)	• ReportToolEvent (Page 242) • ReportToolErrorEvent (Page 246)
Tool deletion	• Method to delete a tool	• RemoveToolRequestEvent() (Page 253) • RemoveToolResponse() (Page 280)	• ReportToolEvent (Page 242) • ReportToolErrorEvent (Page 246)
Production dialog	• WorkPieceCarrierEvent for changes in the Production Dialog (triggered by PLC) Method to Set Values within the Production Dialog	• RequestWpcStatusByDockingIndex() (Page 281) • RequestWpcStatusByWpcName() (Page 282) • RequestAllWpcStatus() (Page 283) • SetWorkpieceCarrierStatus() (Page 284)	• WorkpieceCarrierEvent (Page 255) • WorkpieceCarrierErrorEvent (Page 258)
Logging	Logging functionality to have an overview about the raised events and methods and their status.	-	Log file under the following path: ../user/sinumerik/hmi/log/opcua

Further CMI functionality which is covered via OPC UA

Use case	Solution OPC UA	For further information see chapter...
Alarms	• CNCAlarmEvent will be send to client for new alarms.	Alarms (Page 128)
Program transfer	• Implementation of OPC UA FileDirectoryType Manipulation of files via standard OPC UA methods	File transfer with standard methods (Page 146)
Program selection	• Possible via OPC UA method	Select (Page 152) Unselect (Page 157)
List of NC programs	• Accessible via OPC UA address space	Address space model (Page 113)
Variable read/write	• OPC UA Read/Write	Variable access (Page 116)

8.6.2 Machine status

8.6.2.1 Machine status use case

The Machine status functionality provides the host system with the machine status with the same information provided in Create MyInterface. MachineStatusEvent is triggered by the PLC or by the host computer via a method.

Prerequisite

To receive machine status, the client must be subscribed either to "ADV" node or "Machine" node.

RequestMachineStatus method

1. The MachineStatusEvent can actively be requested on the host system by calling the method `RequestMachineStatus(OpcUa_UInt32 orderNo)`.
2. After invoking the method, a MachineStatusEvent is triggered which includes the order number provided.

8.6.3 Tool management

8.6.3.1 Tool management use case

The PLC can inform the host system via the DB Interface about changes in the tool management. When PLC triggers the interface, a ReportToolEvent is provided containing the corresponding tool data.

Prerequisite

To receive notification on the tool data, including ReportToolEvent and ReportToolErrorEvent, the client must be subscribed either to "ADV" node or "ToolManagement" node.

ReportToolEvent / ReportToolErrorEvent

The ReportToolEvent / ReportToolErrorEvent provides the host with tool management information that is triggered either by the PLC interface to indicate changes or by a request method invoked by the client.

- `void ReportToolEvent(OpcUa_UInt32 orderNo, ToolReportType toolReportType, OpcUa_String toolData)`
- `void ReportToolErrorEvent(OpcUa_UInt32 orderNo, ToolReportType toolReportType, OpcUa_Int errorNo, OpcUa_String errorInfo)`

Requesting individual tool data by the host system

The tool data of a particular tool can be requested by the host system via methods that are found under ToolManagement.

Invoking one of these methods triggers a ReportToolEvent containing the requested data as event payload. If there is an error, a ReportToolErrorEvent is triggered.

The method to request depends on the tool information available (Tool number or tool name or magazine place). Below are the methods:

Request tool data methods

- `RequestToolDataByToolNo(OpcUa_UInt32 orderNo, OpcUa_UInt16 toArea, OpcUa_UInt32 toolNo, OpcUa_String ToolSelection)`
- `RequestToolDataByToolName(OpcUa_UInt32 orderNo, OpcUa_UInt16 toArea, OpcUa_String toolIdent, OpcUa_UInt16 duploNo, OpcUa_String ToolSelection)`
- `RequestToolDataByMagazinePlace(OpcUa_UInt32 orderNo, OpcUa_UInt16 toArea, OpcUa_UInt16 magazineNo, OpcUa_UInt16 magazinePlace, OpcUa_String ToolSelection)`

Requesting full tool / magazine data by the host system

The complete tool data / magazine data can be requested with the two methods provided that generates a file which can be transferred to the host computer via OPC UA file methods. An event is triggered by OPC UA server to indicate the completion of the generated file which contains the order number and the file name. The file can be found under the ADV Exchange node under the FileSystem node.

1. After invoking one of the below methods, OPC UA server will create a file containing the tool / magazine data and save it to "AdvExchange" folder in the "FileSystem" node.
2. A DataFileEvent is then triggered to the host system containing the order number, corresponding file name and status of the executed request. The file format will be as follows:
   - <ordernumer>_ToolData.ini
   - <orderNumber>_MagzineData.ini

3. The host can now read the file from "AdvExchange" using FileDirectoryType methods or CopyFileFromServer method.
4. After the files is read, it must be deleted by the host system from "AdvExchange", using FileDirectoryType methods.

Note

Filetransfer rights for the Filesystem (FsRead/FsWrite) are necessary. AdvExchange stores up to 20 files after which the file with oldest timestamp gets deleted.

With each system reboot, the AdvExchange is cleared.

Request Magazine / Tool Data File methods

- `RequestMagazineDataFile(OpcUa_UInt32 orderNo,OpcUa_UInt16 magazineNo)`
- `RequestToolDataFile(OpcUa_UInt32 orderNo,OpcUa_UInt16 toAreaNo)`

DataFileEvent

To receive the DataFileEvent the client must subscribe either to "ADV" node or "ToolManagement" node.

- `void DataFileEvent(OpcUa_UInt32 orderNo, OpcUa_String filename, OpcUa_StatusCode status, OpcUa_string errorinfo)`

8.6.3.2 Tool load use case with Tool Ident Connection

Prerequisites

- Precondition for using the tool unloading function on a PCU 50 or industrial PC is that SINUMERIK Operate is installed with a Tool Ident Connection license enabled. For a complete description configuration and activation of Tool Ident Connection please refer to Function Manual SINUMERIK ONE Tool Management (https://support.industry.siemens.com/cs/document/109777604/sinumerik-one-tool-management?dti=0&lc=en-AT), chapter 10. You can find detailed information on configuration of DB19 and error handling in this manual.
- To receive notification on the tool loading functionality, including ReportToolEvent and ReportToolErrorEvent, the client must be subscribed either to "ADV" node or "ToolManagement" node.

Tool load with Tool Ident Connection

1. A tool can be loaded into a machine with tool data provided by the host using the Tool Ident in combination with OPC UA Advanced. Tool loading process is triggered by setting K Code 4 in the DB19 in the PLC.
2. The OPC UA server raises an AddToolRequestEvent to the host to provide tool data for the tool to load.
   - `void AddToolRequestEvent(OpcUa_UInt32 orderNo, ToolRequestType toolRequestType, OpcUa_UInt32 duploNo, OpcUa_String toolIdent)`

3. Then the host can provide the tool data for the tool to load by invoking the following method:
   - `OpcUa_StatusCode AddToolResponse(OpcUa_UInt32 orderNo, OpcUa_String toolData, OpcUa_UInt32 errorNo, OpcUa_String errorInfo)`

Note

Timeout

If the host does not respond to an event sent by the SINUMERIK Tool Management, the operation is aborted.

The duration for timeout can be configured in settings.ini file.

4. The `OrderNo` entered in the AddToolResponse must be the same as the `OrderNo` of AddToolRequestEvent.
5. If `errorNo` is 0 and `errorInfo` is empty, the user accepts to load the tool, the OPC UA server initiates the tool loading process and loads the tool to magazine. After successful tool load a ReportToolEvent will be sent to the host.
6. If the host does not want the tool to be loaded, host must set a `errorNo` to a value between "1 and 255" and can provide error information in the `errorInfo`, then the tool will not be loaded and a ReportToolErrorEvent will be sent.

8.6.3.3 Tool unload use case with Tool Ident Connection

Prerequisites

- Precondition for using the tool unloading function on a PCU 50 or industrial PC is that SINUMERIK Operate is installed with a Tool Ident Connection license enabled. For a complete description of configuration and activation of Tool Ident Connection please refer to Function Manual SINUMERIK ONE Tool Management (https://support.industry.siemens.com/cs/document/109777604/sinumerik-one-tool-management?dti=0&lc=en-AT), chapter 10. You can find detailed information on configuration of DB19 and error handling in this manual.
- To receive notification on the tool unloading functionality, including ReportToolEvent and ReportToolErrorEvent, the client must be subscribed either to "ADV" node or "ToolManagement" node.

Tool unload with Tool Ident Connection

1. A tool can be unloaded from a machine with the Tool Ident Connection by setting command 5 in the DB19 in the PLC.
2. The OPC UA server raises a RemoveToolRequestEvent to signal the host that a tool shall be unloaded.
   - `void RemoveToolRequestEvent(OpcUa_UInt32 orderNo, ToolRequestType toolRequestType, OpcUa_UInt32 duploNo, OpcUa_String ToolIdent, OpcUa_UInt32 toolNumber, OpcUa_String toolData)`

3. The host can then acknowledge the tool unload by the invoking the following method:
   - `OpcUa_StatusCode RemoveToolResponse(OpcUa_UInt32 orderNo, OpcUa_String toolData, OpcUa_UInt32 errorNo, OpcUa_String errorInfo)`

Note

Timeout

If the host does not respond to an event sent by the SINUMERIK Tool Management, the operation is aborted.

The duration for timeout can be configured in settings.ini file.

4. The `OrderNo` entered in the RemoveToolResponse must be the same as the `OrderNo` of RemoveToolRequestEvent.
5. If `errorNo` is 0 and `errorInfo` is empty, the host accepts to unload the tool and the OPC UA server initiates the tool unloading process. After successful tool unload a ReportToolEvent is sent to the host.
6. If the host does not want to unload the tool and chooses a `errorNo` between "1 to 255", the tool will not be unloaded and a ReportToolErrorEvent will be sent to the host. The reason for not loading can be described in the `errorInfo`.
7. After calling the method the tool will be unloaded by TDI and the server will send a ReportToolEvent to the host with the latest tool data.

Note

If ForceUnload=true

If the parameter `ForceUnload` in the configuration file "Settings.ini" is set to `true`, steps 3 to 6 are skipped and tool will be unloaded without acknowledgment from host.

"Settings.ini" is located under the following path:

- /user/sinumerik/hmi/opcua

8.6.3.4 Tool create use case

Prerequisite

To receive notification on the tool creation functionality, including ReportToolEvent and ReportToolErrorEvent, the client must be subscribed either to "ADV" node or "ToolManagement" node.

ToolCreateRequest

With the tool Create use case the PLC can trigger the creation of a new tool by requesting the tool data from a host computer.

1. The PLC triggers the loading process by providing the variables DuploNumber, ToolIdentifier, Toolarea and optionally ForceToolNumber, ForceDuploNumber in the OPC UA Adv symbolic DB and the PLC sets ToolCreateRequest=1, to initiate a tool create request to the server.
2. The OPC UA server sets ToolCreateBusy=1 after it receives the tool create request.
3. OPC UA server sends a AddToolRequestEvent to the host to request tool data.
   - `void AddToolRequestEvent(OpcUa_UInt32 orderNo, ToolRequestType toolRequestType, OpcUa_UInt32 duploNo, OpcUa_String toolIdent)`
4. Then the host can provide the tool data to `CreateTool` by invoking the following method:
   - `OpcUa_StatusCode AddToolResponse(OpcUa_UInt32 orderNo, OpcUa_String toolData, OpcUa_UInt32 errorNo, OpcUa_String errorInfo)`

   Note

   Timeout

   If the host does not respond to an event sent by the SINUMERIK Tool Management, all the bits are reset and the operation is aborted.

   The duration for timeout can be configured in settings.ini file.

5. The `OrderNo` entered in the AddToolResponse must be the same as the `OrderNo` of AddToolRequestEvent. ToolData contains a string with the corresponding tool data, `errorNo` needs to be set to "0" an `errorInfo` shall be empty.
6. Then the SINUMERIK OPC UA server will create a tool and after creating the tool successfully a ReportToolEvent will be sent to the host and in the PLC interface
   The host can abort the creation by entering a suitable `errorNo` and a corresponding `errorinfo`, the tool will not be created and the errorNo and errorInfo will be provided to the PLC DB to inform the PLC about the reason of failure.

In the ideal tool creation process the toolData will be read by the SINUMERIK system and a tool will be created with the tool number given in the tool data in the defined toolArea. However, if this tool number already exists, the next free number is used.

Special case ForceToolNumber / ForceDuploNumber

In the ideal tool creation process the toolData will be read by the SINUMERIK system and a tool will be created with the next available tool number in the defined toolArea. If the host wants to enforce a specific tool number or duplo number, the PLC variables "ForceToolNumber" and "ForceDuploNumber" can be set.

1. The PLC variable ForceToolNumber (see chapter Symbolic DB (Page 218)) and/or ForceDuploNumber are to be set, these PLC variables can be specified in the `OpcUa_String toolData`.
2. Host can then provide the tool data to `CreateTool` by invoking the following method:
   - `OpcUa_StatusCode AddToolResponse(OpcUa_UInt32 orderNo, OpcUa_String toolData, OpcUa_UInt32 errorNo, OpcUa_String errorInfo)`
3. After creating the tool with specific tool number or duplo number successfully, a ReportToolEvent will be sent to the host.
   If the tool number or duplo number given in the toolData string is already present in the SINUMERIK Tool Management system with another tool, then a ReportToolErrorEvent will be sent, and the tool won't be created.

8.6.3.5 Tool delete use case

Prerequisite

To receive notification on the tool deletion functionality, including ReportToolEvent and ReportToolErrorEvent, the client is subscribed either to "ADV" node or "ToolManagement" node.

RemoveToolRequest

You can choose to delete a tool with host interaction (by actively calling a method) or without host interaction (only triggered by the PLC).

1. The PLC updates the variables ToolNumber or ToolIdentifier and DuploNumber along with ToolArea in the OPC UA Adv symbolic DB and sets ToolDeleteRequest=1, to initiate a `DeleteTool` request to the server.
2. The OPC UA server sets ToolDeleteBusy=1 after it receives the tool delete request.
3. OPC UA server reads the updated variables from PLC.
4. The OPC UA server raises a RemoveToolRequestEvent to signal the host that a tool can be deleted.
   - `void RemoveToolRequestEvent(OpcUa_UInt32 orderNo, ToolRequestType toolRequestType, OpcUa_UInt32 duploNo, OpcUa_String toolIdent, OpcUa_UInt32 toolNumber, OpcUa_String toolData)`

5. Then the host can provide the tool data to `DeleteTool` by invoking the following method:
   - `OpcUa_StatusCode RemoveToolResponse(OpcUa_UInt32 orderNo, OpcUa_UInt32 errorNo, OpcUa_String errorInfo)`

Note

Timeout

If the host does not respond to an event sent by the SINUMERIK Tool Management, all the bits are reset and the operation is aborted.

The duration for timeout can be configured in settings.ini file.

6. The `OrderNo` entered in the RemoveToolResponse must be the same as the `OrderNo` of RemoveToolRequestEvent.
7. If `errorNo` is "0" and `errorinfo` is empty, the host accepts to delete the tool. OPC UA server initiates the tool delete process, after deleting the tool successfully a ReportToolEvent will be sent to the host.
8. The host can abort the DeleteTool by entering a suitable `errorNo` and a corresponding `errorinfo`, the tool will not be deleted and a ReportToolErrorEvent will be sent. The tool will not be deleted and the `errorNo` and `errorInfo` are provided to the PLC DB to inform the PLC about the reason of failure.

Note

If `ForceDelete` in the configuration file `"Settings.ini"` is set to "True".

The interaction with the host is avoided, thereby steps 3 to 7 are skipped.

8.6.4 Production dialog

8.6.4.1 Production dialog use case

Prerequisite

To receive notification on the WorkpieceCarrierEvent, the client must be subscribed either to "ADV" node or "WorkpieceCarrier" node.

WorkpieceCarrierEvent

The WorkpieceCarrierEvent is triggered by the PLC or by the host via a request method.

- `void WorkpieceCarrierEvent(OpcUa UInt32 orderNo, OpcUa UInt32 dockingStationNo, dockingStationStatus, OpcUa_StatusCode status, OpcUa_String name, machiningside, followUpMachining)`
- `void WorkpieceCarrierErrorEvent(OpcUa UInt32OrderNo, OpcUa UInt32 errorNo)`

Request workpiece carrier status

1. The WorkpieceCarrierEvent can also be actively requested by the host system, by calling the following methods:
   - `RequestWpcStatusByDockingIndex (OpcUa_UInt32 orderNo, OpcUa_UInt16 index)`
   - `RequestWpcStatusByWpcName (OpcUa_UInt32 orderNo, OpcUa_String name)`
   - `RequestAllWpcStatus (OpcUa_UInt32 orderNo)`
2. After calling a method successfully a WorkpieceCarrerEvent is triggered, which includes the order number provided and the information about the requested workpiece carriers.

Set workpiece carrier status

1. The workpiece status can be set by the host system via the following method: `SetWorkpieceCarrierStatus(OpcUa_String wpcName,WorkpieceCarrierStatus status)`

Host system shall only set "state 8"(ProgramSelectionComplete), all other states are provided by the PLC.

WorkpieceCarrierStatus	
0	Reset
1	NewWpc
2	MachiningPlanned
4	PrepareProgramSelection
8	ProgramSelectionComplete
16	MachiningInProgress
32	MachiningFinished
64	MachiningAborted
128	BufferingOnly

8.7 OPC UA Advanced logging

OPC UA Advanced also offers a logging functionality to have an overview about the raised events and methods and their status. Logging is only available for ADV functionality.

The following information is being written into the log file:

- All ADV events sent to host, including their parameters.
- All ADV requests (methods) called from host, including their parameters.

Structure of the log file

The log file is structured and contains the following values:

- Date
- Time (incl. ms)
- Initiation by the <Client> or by the <Server>
- Thread ID
- "Adv" node activity

Example for log file

```
2022-02-10T14:18:09.146Z <Client> tid=10168 Subscribed for: AdvStatus
2022-02-10T14:18:46.730Z <Server> tid=6772 WorkpieceCarrierEventType triggered by PLC:
orderNo:0, dockingStationNo:0, status:8, name:WST01, machiningSide:0,
followUpMachininig:0, reserved:0
2022-02-10T14:18:55.667Z <Server> tid=6220 ReportToolEventType triggered by PLC:
orderNo:0, toolNo:0, placeNo:0, tooldata:
2022-02-10T14:19:09.392Z <Server> tid=5460 MachineStatusEventType triggered by PLC:
oderNo:0, sinumerikMode:0, hostMode:0, plcMode:0, machineStatus:1, NCProgram:Sinumerik/
FileSystem/Sub Program/FILE111.MPF, reserveByte1:0, reserveByte2:0, resByteString:0
```

Activating / Deactivating Adv logging

Adv Logging functionality can be activated or deactivated via the Diagnosis > ADV log screen (Page 187). As a default setting logging is deactivated.

- Setting is stored in settings.ini ../user/sinumerik/hmi/opcua

Note

Logging is implemented for commissioning and troubleshooting purposes.

Parameters of the ADV log

The parameters of the log functionality can be changed via the Diagnosis screen (Page 187).

Parameters	Value	
	Min	Max
Number of log files	3	10
Number of record in the log file	10,000	99,999

Note

Maximum values reached

If the maximum number of log files is reached and each file contains the maximum number of records, the oldest log file will automatically be deleted and new empty log file will be created.

Location of log file

The naming of the log file depends on the OPC UA server. You can find the log file "adv.log“ under the following path:

../user/sinumerik/hmi/log/opcua

8.8 OPC UA Advanced settings

Specific settings of OPC UA Advanced can be set in the file settings.ini under the section `[Adv]`. This file is located under the following path:

- /user/sinumerik/hmi/opcua

Name of the symbolic address

You can specify the name of your symbolic DB. As a default this setting is `OpcUaAdv`.

Settings.ini

```
[Adv]
.
.
.
SymbolicDBName=OpcUaAdv
```

Unload without host response

You can specify whether a tool will be unloaded without acknowledgment from the host. Per default this setting is `false`. For a more detailed description of this parameter see chapter Tool unload use case with Tool Ident Connection (Page 175).

Settings.ini

```
[Adv]
.
.
.
ForceUnload =false
```

Delete without host response

You can specify whether a tool will be deleted without acknowledgment from host. Per default this setting is `false`. For a more detailed description of this parameter see chapter Tool delete use case (Page 178).

Settings.ini

```
[Adv]
.
.
.
ForceDelete =false
```

Timeout value in seconds

Timeout value in seconds, if you do not call tool methods which requested by the machine via event. Per default this setting is 300 seconds.

Settings.ini

```
[Adv]
.
.
.
ToolOpsTimeout=300
```

Note

Make sure to renew the server certificate (as described in chapter Server certificates (Page 39)) after the entry has been changed.

8.9 OPC UA Advanced host settings

Specific host settings of OPC UA Advanced are set in the file advhostsettings.ini. This file is located under the following path:

- /user/sinumerik/hmi/opcua

Setting AdvHost name

After successful request of the method "RegisterAdvHost", the defined user in the "advhostsettings.ini" file is saved and registered. If the file "advhostsettings.ini" does not exist, it will be created, when calling the method for the first time. If you want to delete the already registered user and register a new one, then you have to delete the entry in this file manually, before calling the method again.

advhostsettings.ini

```
[AdvHost]
.
.
.
HostUserName = UserName
```

Diagnosis

9

9.1 Overview

Overview

The OPC UA server offers a variety of diagnostics information, as described in the OPC UA Standard Part 5 - "Information Model", Chapter 6.

This diagnostics information can be found under the Server Node:

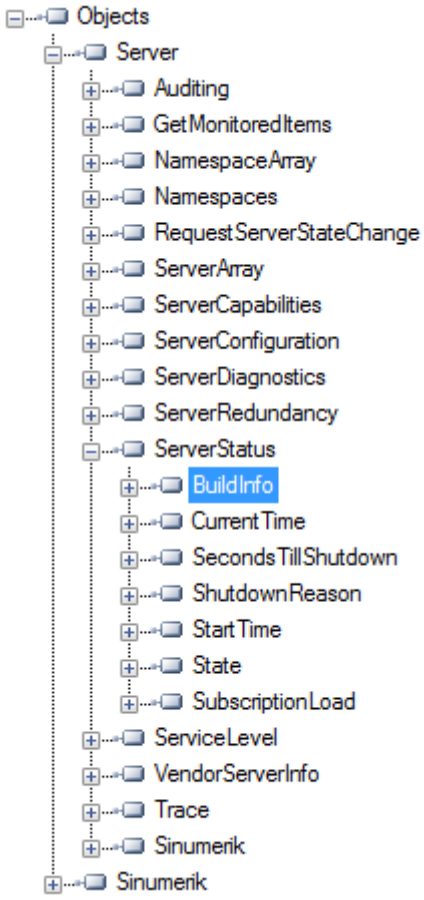


Figure 9-1 Diagnostics Information - Server Node

9.2 Status screen

Status screen

Additional to the server status information available via OPC UA, there is a SINUMERIK Operate screen, which shows the actual status of the OPC UA server.

To open the status screen, select the operating area "Startup > Network" in SINUMERIK Operate, then press the "OPC UA" softkey. The OPC UA status screen is the first screen to be displayed.

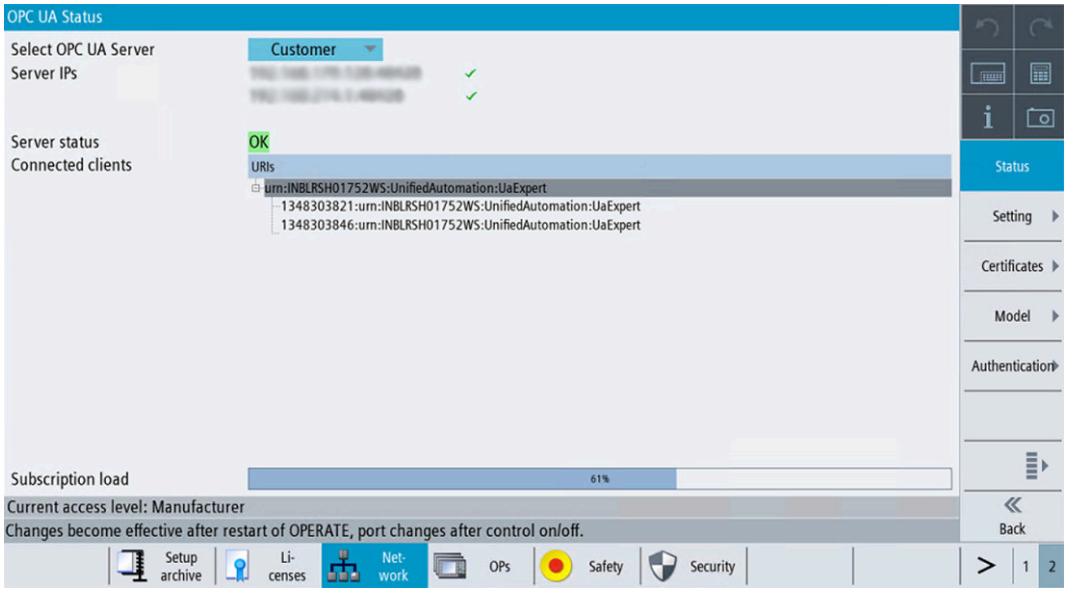


Figure 9-2 Status

Value	Description	Further explanation
Server IPs	Server IPs and ports of the company network, systems network or service network where the OPC UA server is accessible	If the server IP is connected, then a green tick is displayed in front of the IP. If the server IP is disconnected, then a red cross is displayed in front of the IP.
Server status	Possible status of the server: • Ok (server up and running) • Not activated (OPC UA server deactivated) • No connection can be set up (error within the OPC UA server) • No more sessions possible. All sessions are in use by other clients. The status screen cannot create a session.	There are too many sessions used by other clients. External clients are allowed to create 10 sessions with SINUMERIK ONE. The session limitation is 11 in the configuration file, to have one more session for the status client.

Value	Description	Further explanation
Connected clients	Clients which are connected to the server **Example**: • MD1EXMQC: remote PC of the client • SiemensAG:OpcUaTestsApp: URN of the application of the remote PC • 10788... Session ID • OpcUaTestConsole: Session Name	
Subscription load	Utilized capacity of the OPC UA server regarding possible subscriptions (see chapter, Technical data (Page 209)), not the overall load.	

9.3 Diagnosis screen

Overview

Diagnosis screen offers support for troubleshooting and service for OEM with SIEMENS. Diagnosis functionalities are used only for service purpose.

From the Diagnosis screen user can:

- Activate or deactivate the OPC UA Server logging.
- Configure OPC UA Server logging.
- Reset the OPC UA Server to factory settings.
- Export the diagnosis data to an external data storage (for example, USB/Networkshare).
- Activate or deactivate OPC UA Advanced logging and change parameters.

Note

Diagnosis softkey is visible by default only for manufacturer, service and user access levels.

OPC UA Diagnosis

1. Press the "OPC UA" softkey.
2. Press the extended softkey as shown in the below image.

Note

If the extended softkey is disabled then it means the OPC UA Server license is not active.

3. Press the "Diagnosis" softkey.

Note

The "Diagnosis" softkey will not be visible if OPC UA Server is not actiavted. User need to activate the OPC UA Server from the OPC UA settings screen (Page 30).

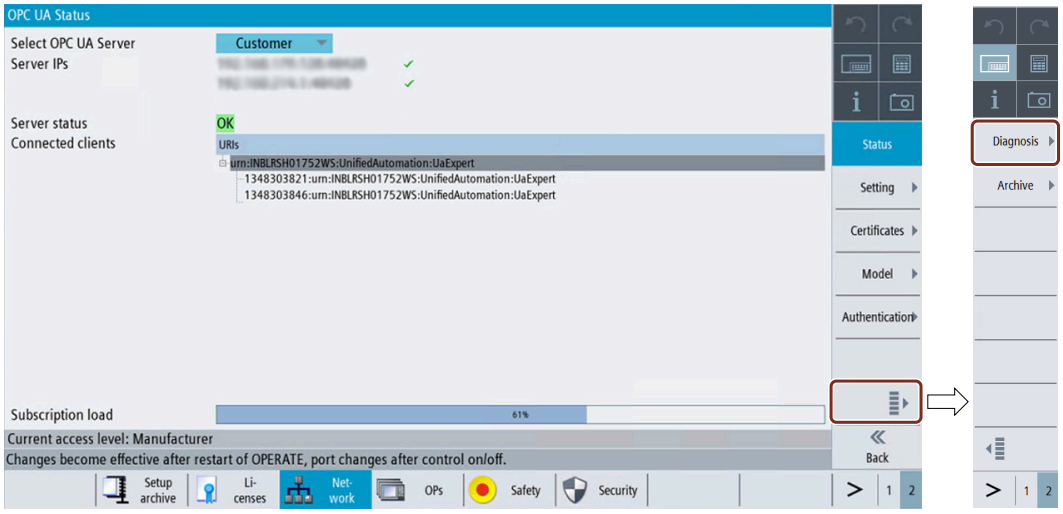


Figure 9-3 Softkey Diagnosis

4. The "OPC UA Diagnosis" dialog will appear. Then press the "Change" softkey.

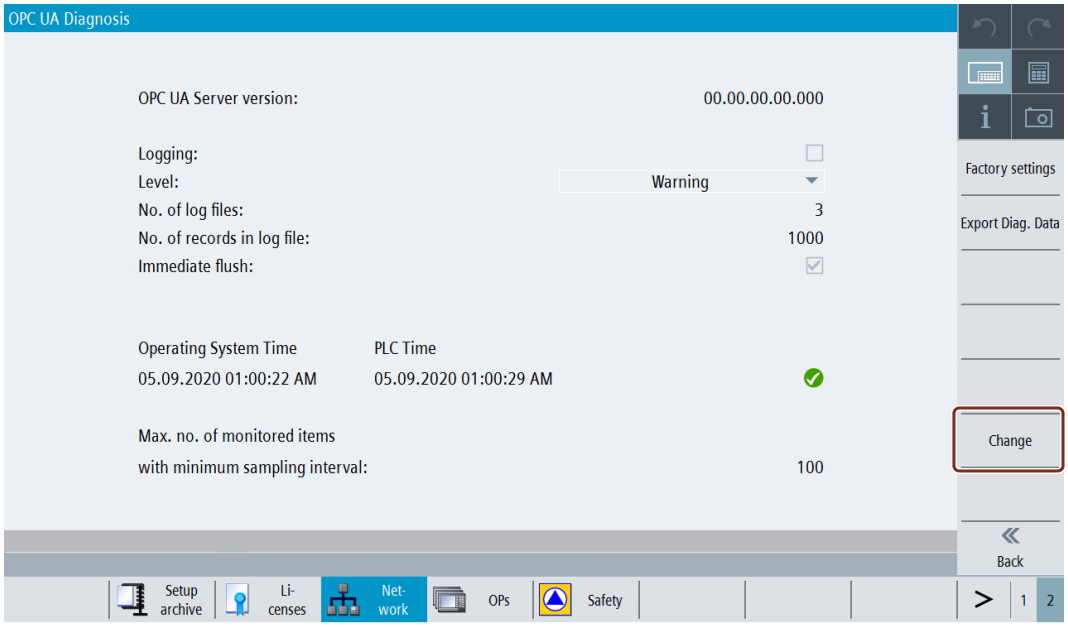


Figure 9-4 Softkey Change

5. Make the necessary settings for logging.

Note

The changes become effective only after the restart of the SINUMERIK Operate.

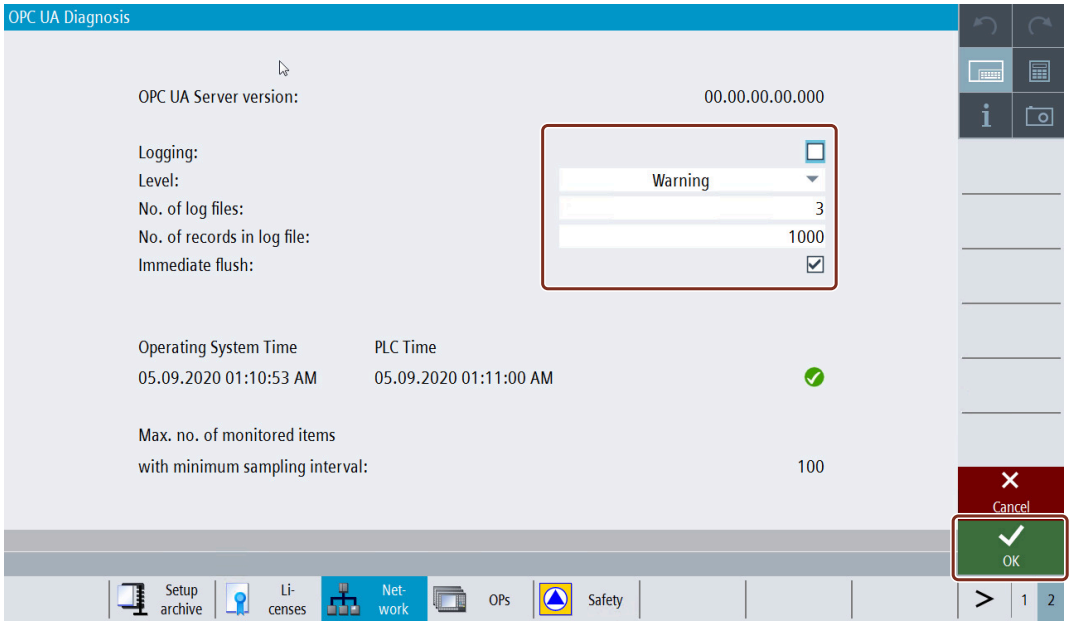


Figure 9-5 Activate Logging option

Field	Description
"OPC UA Server Version"	Shows the OPC UA Server version installed.
"Logging"	Place the checkmark to activate logging and remove the checkmark to deactivate it. By activating logging, it generates the log of the OPC UA Server.
"Level"	Select the type of level from the drop down list. The drop down list has the following types of level: • None • Error • Warning (by dafault) • System • Info • Debug • Content • All The log will be generated based on the selection of level type. No log will be generated if you have activated logging and selected "None" from the list.
"No. of log files"	Enter the no. of log files between 3 - 10. If the user selects 10, and the 10th log file is already generated, then it will replace the 1st log file.

Field	Description
"No. of records in log file"	Enter the no. of records in log file between 1000 - 99999. If the user selects 99999, and the 99999th record is already generated, then it will replace the 1st record.
"Immediate flush"	Place the checkmark to activate the immediate flush and remove the checkmark to deactivate it. By activating immediate flush, every record will be written immediately into the log file. This option should be used only for logging without delays.
"Operating System Time and PLC Time"	Operating System and PLC date and time are displayed. The tick icon indicates that the PLC time is in sync with the operating system (HMI Operate) time. If the date and time is not in sync, icon is shown and it indicates that time should be updated.
"Max. no. of monitored items with minimum sampling interval"	This field provides information about the max. no. of items which can be monitored with minimum sampling interval".

6. Then press "OK".
7. The changes become effective only after the restart of the SINUMERIK Operate.

Note

If the OPC UA server logging is activated for more than 30 days, then it will be automatically disabled on the next HMI restart.

Resetting the OPC UA Server to factory settings

This helps the user to reset the OPC UA configuration of the SINUMERIK device to its initial state. This would be helpful when the current OPC UA configuration is not valid and needs to be reset.

User shall use this as the last option to reconfigure the OPC UA server to its initial state. All settings, IP, Port, and certificates will be deleted and the default values will be restored.

After the restart of the SINUMERIK Operate, OPC UA server need to be activated again with necessary changes.

Note

On factory reset, model settings are set to default and model settings cannot be changed with user access level.

Note

Make sure to have an OPC UA archive before starting the process of resetting the OPC UA server to factory settings.

1. Press the "OPC UA" softkey.
2. Press the extended softkey.

3. Press the "Diagnosis" softkey. The "OPC UA Diagnosis" dialog will appear.
4. Then press the "Factory settings" softkey.

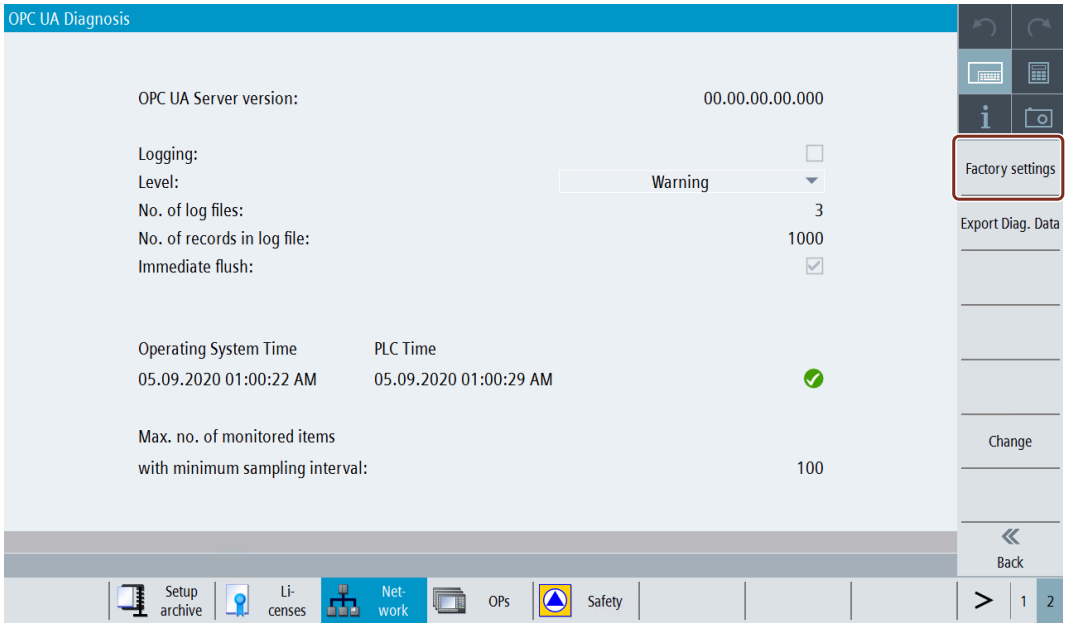


Figure 9-6 Softkey Factory Settings

5. The "Factory settings" popup window will appear.

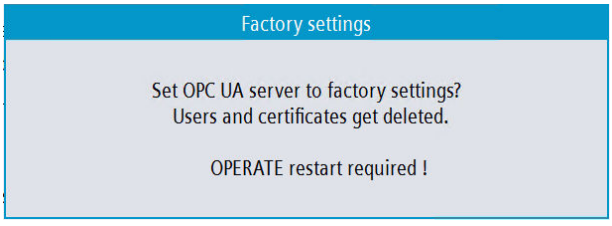


Figure 9-7 Factory settings popup

Note

If the user performs factory settings, the OPC UA Server will be reset to default setting. User need to enable the OPC UA Server again from the OPC UA settings screen (Page 30).

6. Then press "OK". A status message is shown at the bottom, "changes become effective after the restart of OPERATE".

Exporting diagnosis data

This feature exports the OPC UA configuration, log files and other diagnostic information from the SINUMERIK system, onto the selected location on SINUMERIK file system.

This can be used to analyze details of behavior of the system or root cause of the situation in case of problems.

1. Press the "OPC UA" softkey.
2. Press the extended softkey.

3. Press the "Diagnosis" softkey. The "OPC UA Diagnosis" dialog will appear.
4. Then press the "Export Diag. data" softkey.

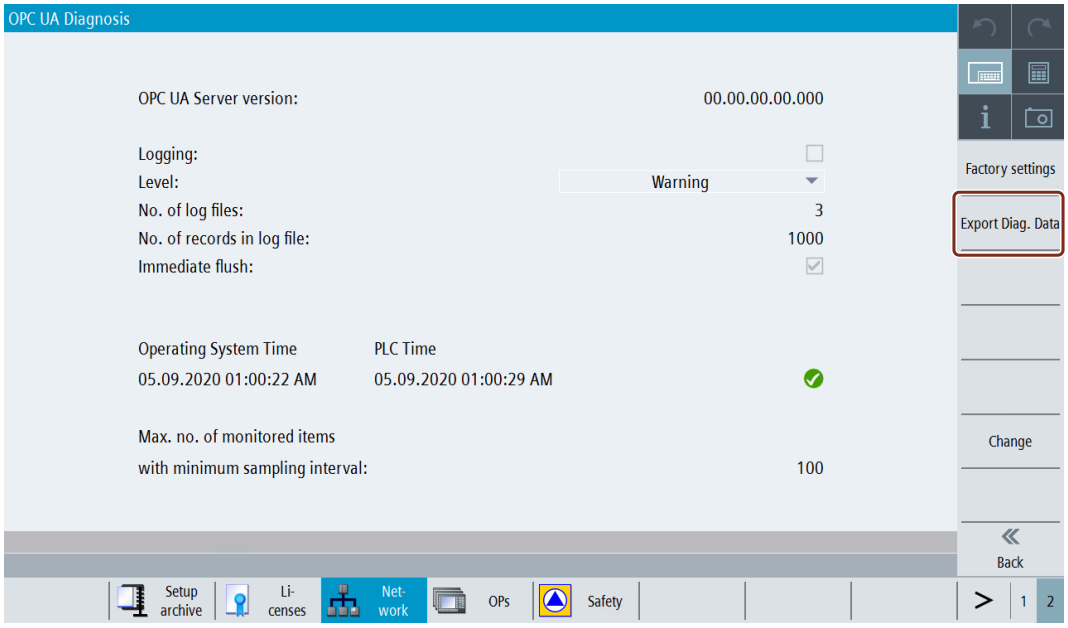


Figure 9-8 Softkey Export Diag data

5. The "Export OPC UA Diagnosis Data" popup window will appear.

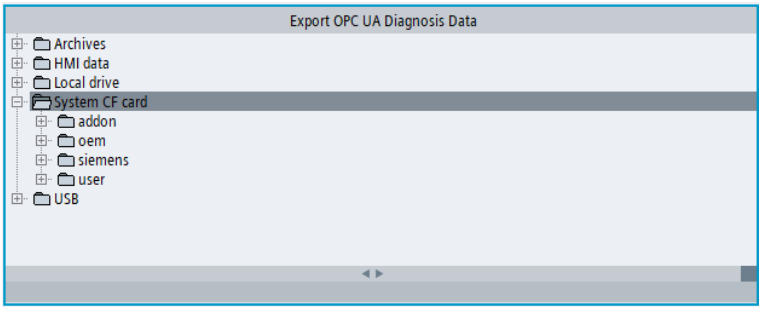


Figure 9-9 Export OPC UA Diagnosis data popup

6. Select location on SINUMERIK file system.
7. Then press "OK".
8. The diagnosis data will be saved to the folder (as selected location on SINUMERIK file system) in the format "OpcUaDiagnosisData_Year_Month_Day-Hour_Minute_Second",(for example: "OpcUaDiagnosisData_2020_06_22-11_12_56").

Settings for OPC UA Advanced logging

If you have activated the additional license for OPC UA Advanced, you can set the parameters for OPC UA Advanced logging.

1. Press the "OPC UA" softkey.
2. Press the extended softkey.
3. Press the "Diagnosis" softkey. The "OPC UA Diagnosis" dialog will appear.

4. Then press the "ADV log" softkey.

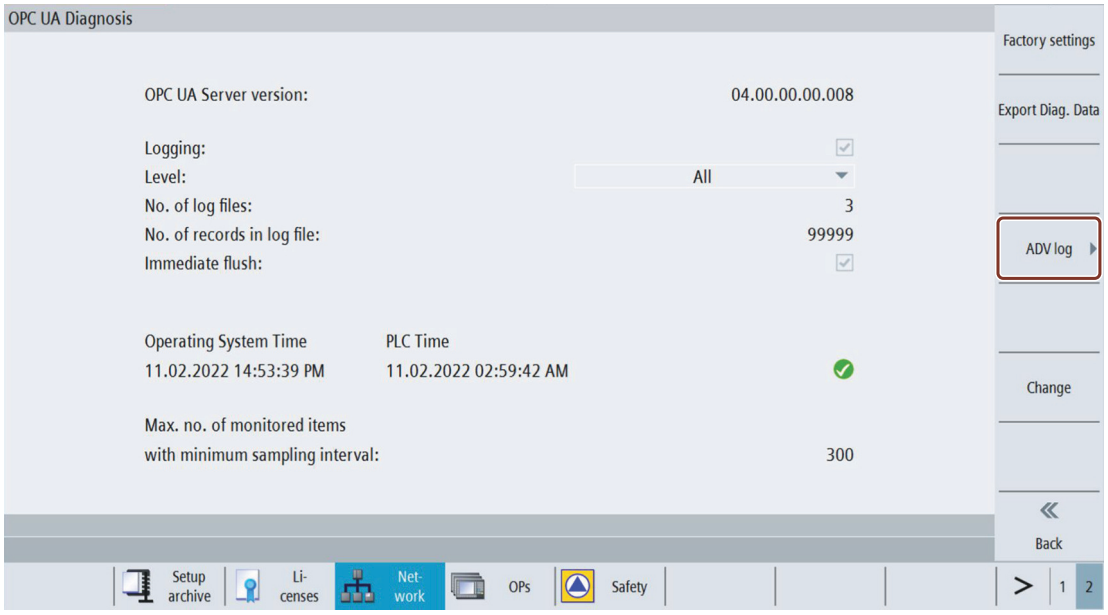


Figure 9-10 Softkey ADV log

5. The "OPC UA Adv log" dialog will appear. You can read the settings for OPC UA Advanced logging.

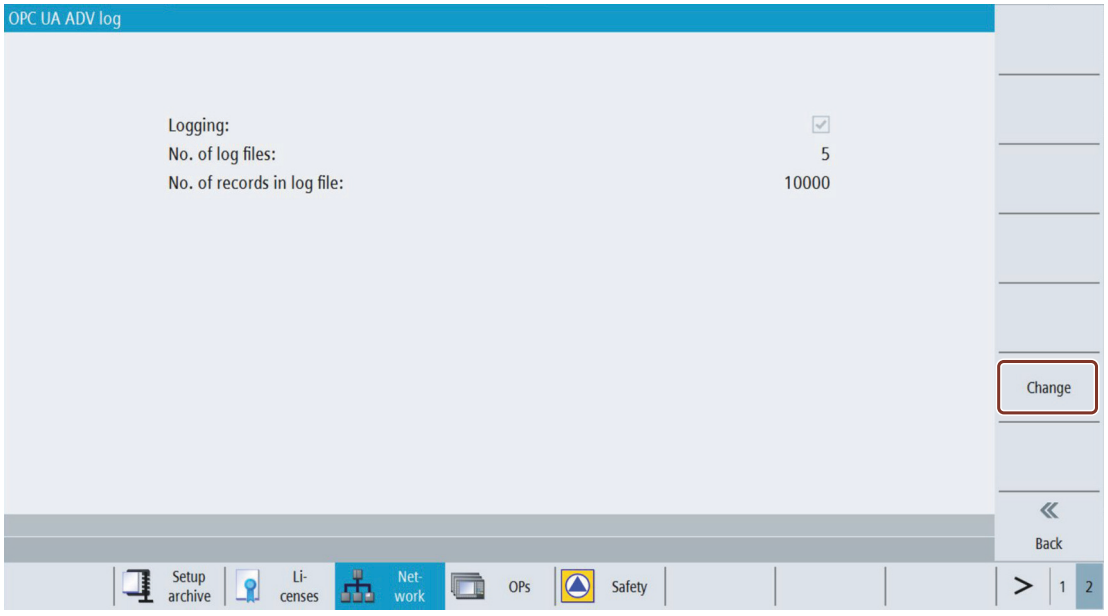


Figure 9-11 ADV log dialog

6. Press the "Change" softkey"
7. Activate the OPC UA Advanced logging with the corresponding checkbox.

8. Set the parameters for OPC UA Advanced log file.

Parameters	Value	
	Min	**Max**
Number of log files	3	10
Number of record in the log file	10,000	99,999

Note

Maximum values reached

If the maximum number of log files is reached and each file contains the maximum number of records, the oldest log file will automatically be deleted and a new log file will be created.

9. Press "OK". The settings for OPC UA Advanced logging are saved.

You can find further information on the OPC UA Advanced logging functionality in chapter OPC UA Advanced logging.

See also

OPC UA Advanced logging (Page 180)

9.4 OPC UA Archiving

Overview

Archiving feature enables the user to generate a backup of OPC UA Server so that in case of need, all settings and data can be restored. Also, user can generate a setup archive for serial commissioning and update new machines with setup archive.

From Archiving screen user can:

- Generate OPC UA Server backup archive (i.e, full archive)
- Generate OPC UA Server setup archive for serial commissioning
- Restore backup archive
- Read in setup archive for serial commissioning

Prerequisite

OPC UA archive is only part of Operate archive, if the archive service in OPC UA has been done before. Therefore create an OPC UA backup first, before taking full backup of Operate.

OPC UA Server backup archiving

The user can generate a backup archive of OPC UA Server (including configuration, user management, models, and certificates) and restore it back when required. The OPC UA back up archive can only be restored on the same controller or the same SD card, since the server certificate is part of the backup.

Generating Archive

1. Press the "OPC UA" softkey.
2. Press the extended softkey as shown in the below image.

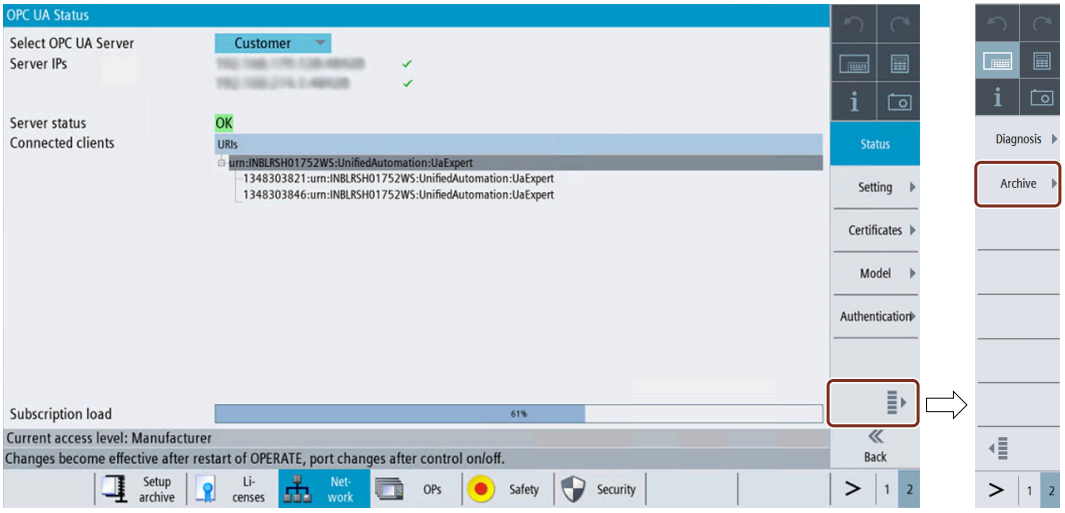


Figure 9-12 Softkey Extended

Note

If the OPC UA server license is not active, then the extended softkey is disabled.

3. Press the "Archive" softkey.

4. The "OPC UA Archives" dialog will appear. Select "All data of this OPC UA server (backup)" radio button under "Generate archive".

Note

The radio button "All data of this OPC UA server (backup)" is enabled by default only for manufacturer, service and user access levels.

Note

If additionally, you want the backup file to be copied to the SINUMERIK filesystem, then select "Export additionally to SINUMERIK files system" check box. After Pressing "OK", a popup window appears. Select the location in the SINUMERIK filesystem where you want to save the backup file.

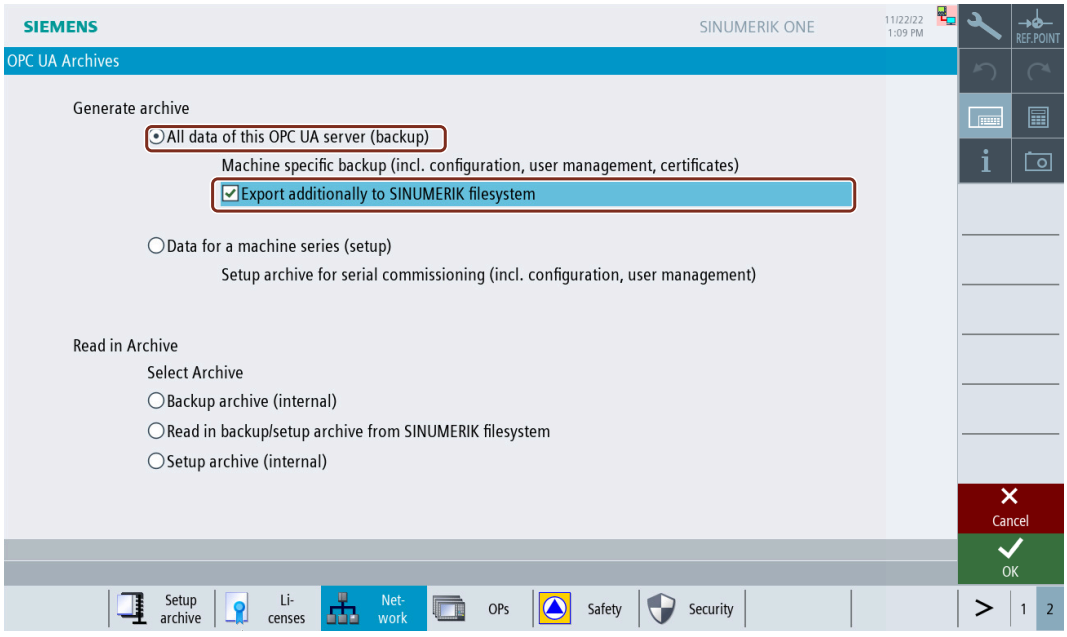


Figure 9-13 OPC UA Server generate backup archiving

5. Then press "OK". The popup window appears with message, "OPC UA archive generated successfully".

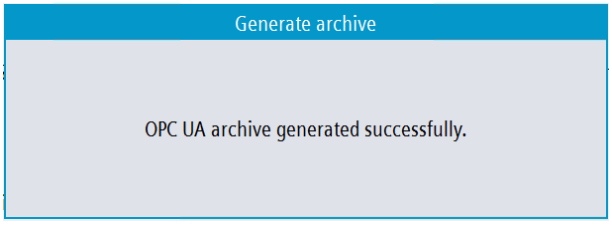


Figure 9-14 Generate archive popup

6. Press "OK". The backup file is saved automatically in the local folder (internal memory) and in the SINUMERIK file system as selected. The internal path for NCU will be "/card/user/sinumerik/hmi/cfg" and for PCU/IPC "C:\Program Files (x86)\Siemens\MotionControl/user/sinumerik/hmi/cfg".

Restoring backup archive

Note

- Backup archives can only be restored on the backed up target system.
- By restoring backup archive, all the current settings and data of the OPC UA Server will be replaced with the backup file data.

Note

Restoring the backup archive does not open the firewall port for the OPC UA Server automatically. Due to security reasons, this port must be manually opened in the firewall.

1. Select "Backup archive (internal)" radio button under "Read in Archive".

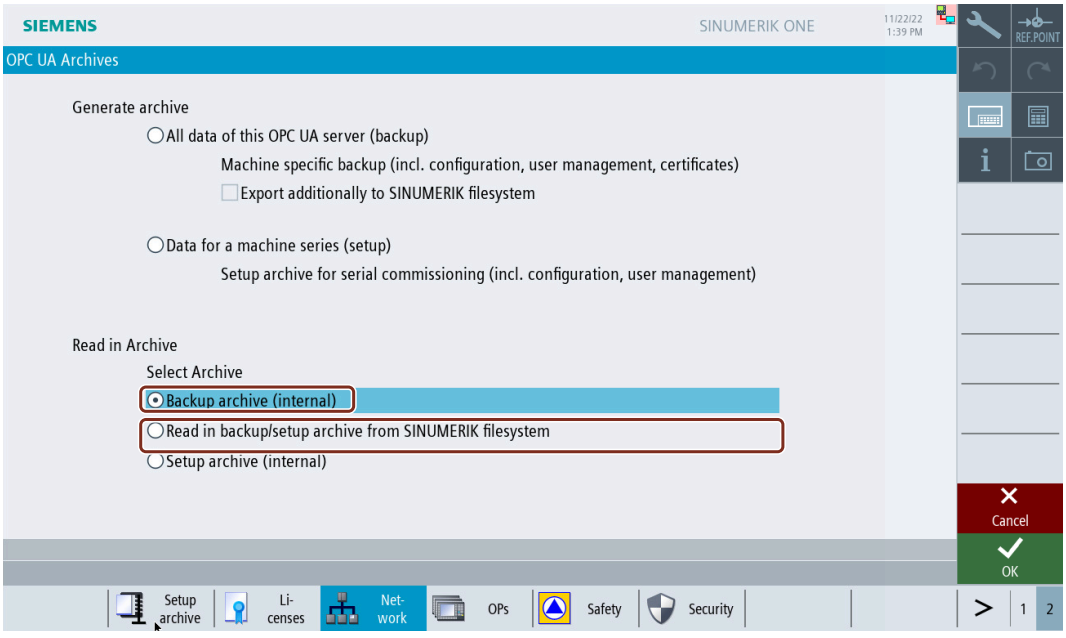


Figure 9-15 OPC UA Server restore backup archiving

In case, if you have saved and want to select the backup file from the SINUMERIK file system, then select "Read in backup/setup archive from SINUMERIK file system" radio button. After Pressing "OK", a popup window appears. Select the backup file from the SINUMERK file system.

2. Then Press "OK". With the valid input file, a popup window appears with message, "Read in completed successfully".

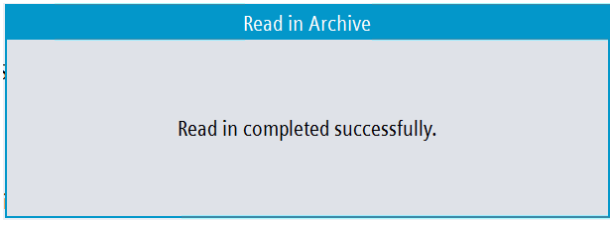


Figure 9-16 Read in archive success popup

3. Press "OK". The backup file is restored.

Note

The changes become effective only after the restart of the SINUMERIK Operate.

OPC UA Server setup archiving for serial commissioning

The user generates a setup archive of OPC UA Server for serial commissioning (including configuration, user management, and models) and update new machines with setup archive.

Generating Archive

1. Select "Data for a machine series (setup)" radio button under "Generate archive".

   Note

   This option is enabled only with manufacturer's access levels.

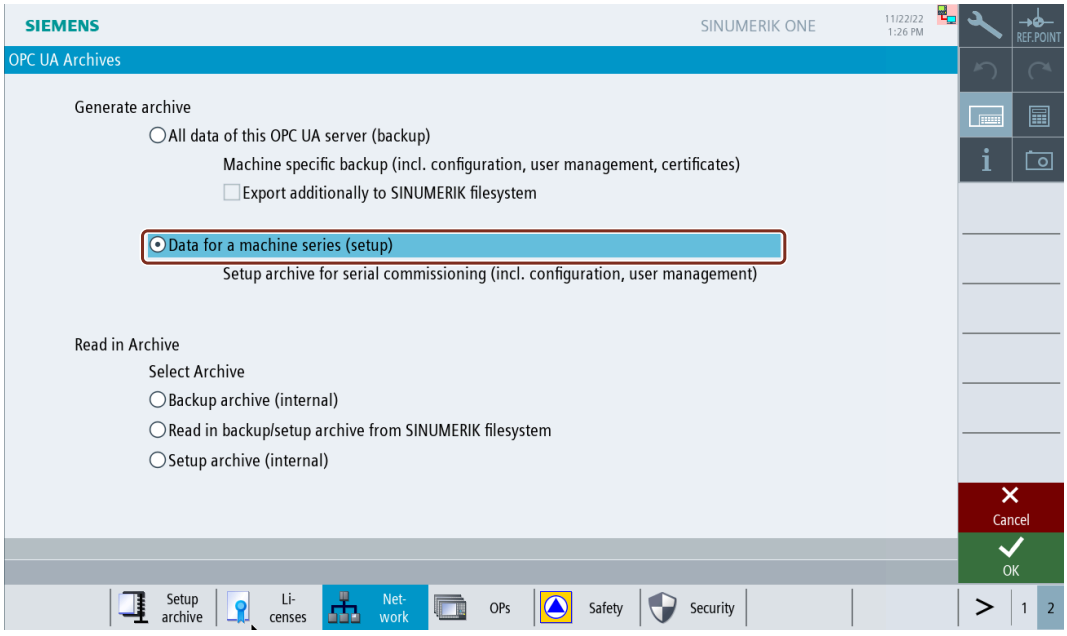


Figure 9-17 OPC UA Server generate setup archiving for serial commissioning

2. Press "OK". A popup window appears.
3. Select the location in the SINUMERIK file system where you want to save the setup file.

4. Then press "OK". The popup window appears with message, "OPC UA archive generated successfully".

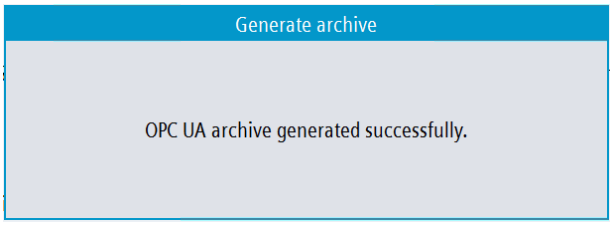


Figure 9-18 Generate archive success popup

5. Press "OK". The setup file is saved in the USB/network share as selected.

Setup archive

1. Select "Read in backup/setup archive from SINUMERIK file system" radio button under "Read in Archive".

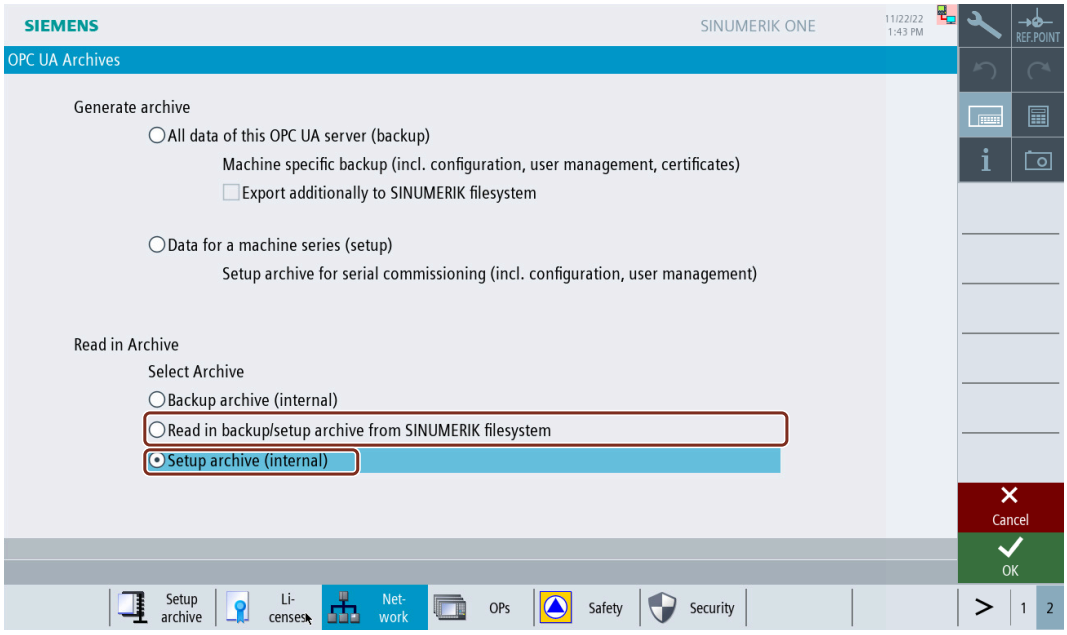


Figure 9-19 OPC UA Server setup archiving for serial commissioning

In case, if you have saved and want to select the setup file from local folder (internal memory), then select "Setup archive (internal)" radio button. The internal path for NCU will be "/card/user/sinumerik/hmi/cfg" and for PCU/IPC "C:\Program Files (x86)\Siemens\MotionControl/user/sinumerik/hmi/cfg".

Note

The "Setup archive (internal)" option is enabled only with manufacturer's access levels.

2. Press "OK". A popup window appears.

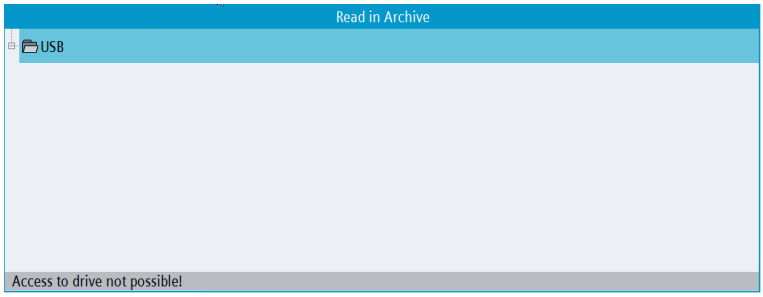


Figure 9-20 Setup archive USB_Networkshare popup

3. Select the setup file from the SINUMERIK file system. Then Press "OK".

4. With the valid input file, a popup window appears with message, "Read in completed successfully".

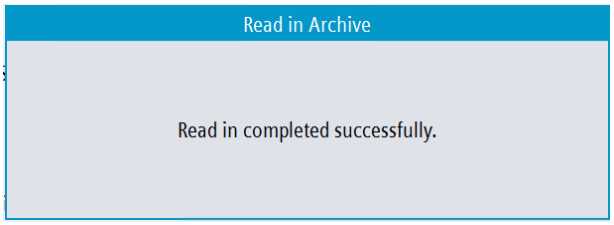


Figure 9-21 Read in archive success popup

5. Press "OK". The setup archive is completed.

Note

The changes become effective only after the restart of the SINUMERIK Operate.

Possible scenarios and error messages

Sl no.	Selection	Scenario description	Error message
1	Generate backup archive OR Generate setup archive	Generation of any archive, if completed without any issues.	OPC UA archive generated successfully.
2	Generate backup archive (internal memory)	If there is no sufficient memory available.	Insufficient memory on IPC/NCU.
3	Generate backup archive (on SINUMERIK file system) OR Generate setup archive (on SINUMERIK file system)	If there is no sufficient memory available.	Insufficient file system memory
4	Generate backup archive (on SINUMERIK file system) OR Generate setup archive (on SINUMERIK file system)	If the selected location is write protected or not accessible.	No write permission on the selected medium.
5	Generate backup archive (on SINUMERIK file system) OR Generate setup archive (on SINUMERIK file system)	If the selected location is write protected or not accessible.	Access to drive is not possible.
6	Read in Backup archive Or Read in Setup archive	No valid OPC UA archive file is found in internal memory.	No OPC UA archive is available on internal memory
7	Read in Backup archive Or Read in Setup archive	No error in reading in file.	Read in completed successfully.

8	Read in Backup archive Or Read in Setup archive	If the selected file is not a valid OPC UA Archive or the file is tampered by editing.	Read in failed. Invalid file.
9	Read in Backup archive (external memory)	If backup archive is not from the same machine	Read in failed. Backup Archive is generated on different Sinumerik device.

9.5 OPC UA server version

OPC UA server version

OPC UA server version and OPC UA dialog version information can be found in SINUMERIK Operate version screen.

1. Open SINUMERIK Operate and choose operating area "Diagnostics". Press the softkey "Version".
2. Select “System extensions" and press softkey "Details".

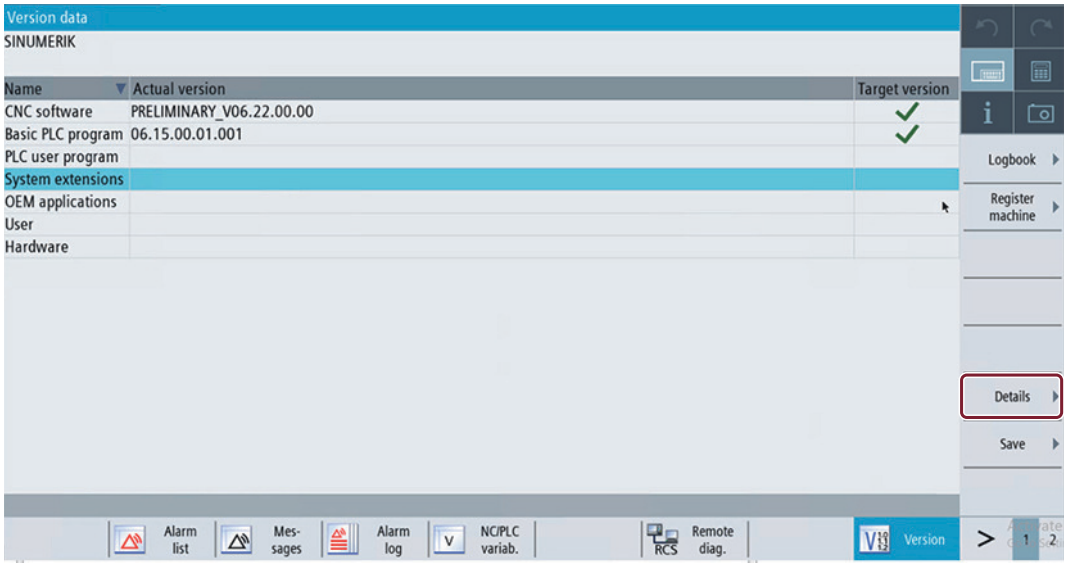


Figure 9-22 Version data

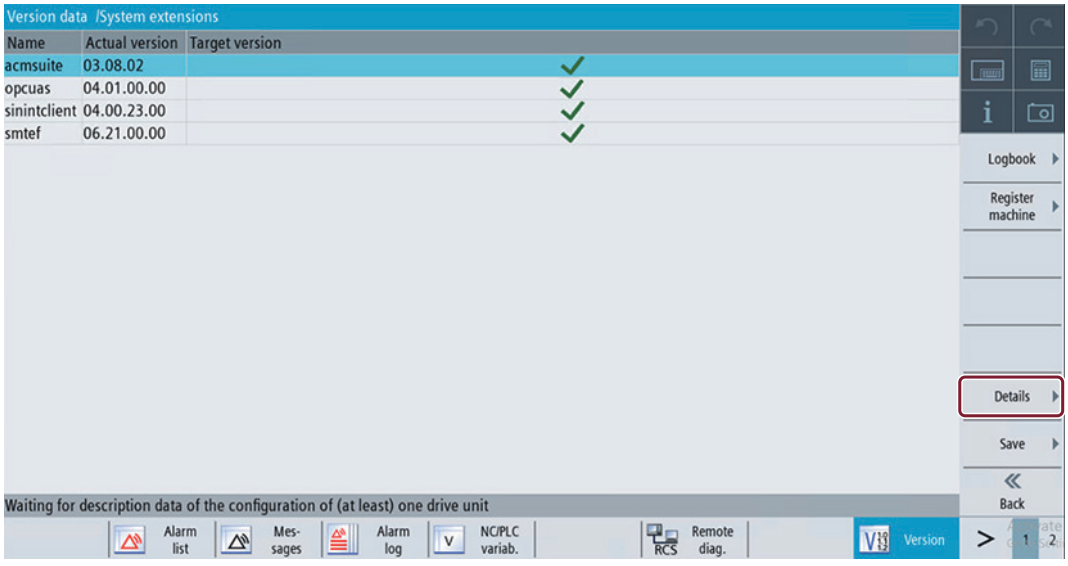


Figure 9-23 Version data / system extensions

The OPC UA entry is found.

3. Select the entry and press "Details" again to show more detailed information on OPC UA components.

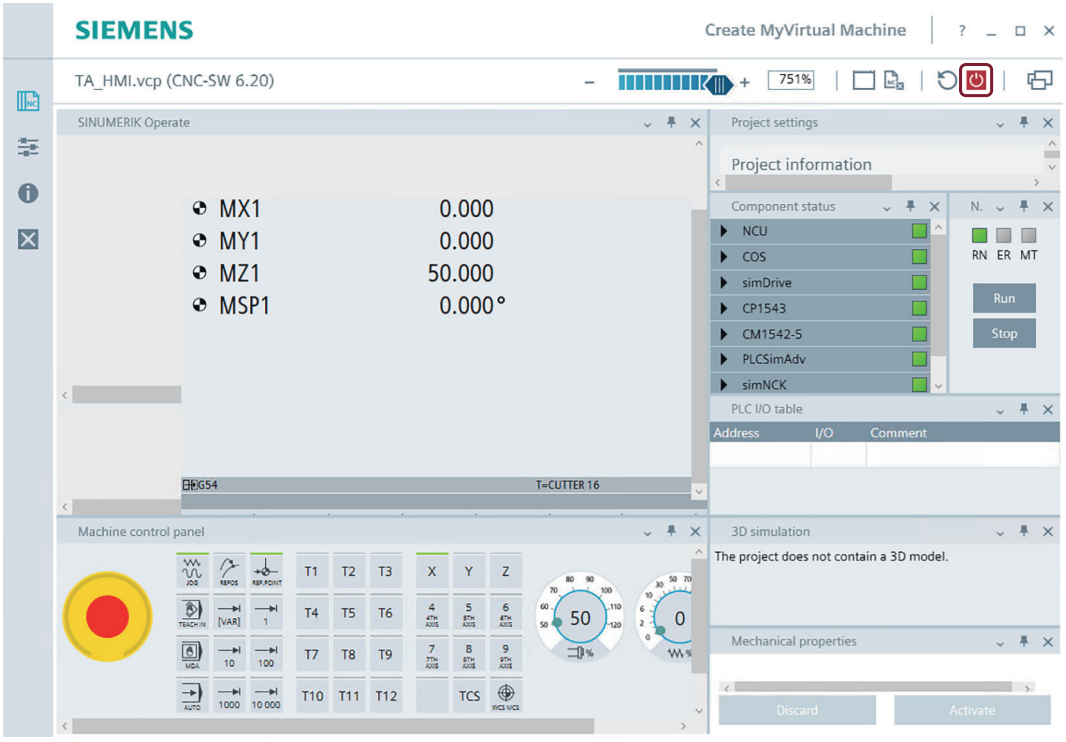
Virtual SINUMERIK ONE

10

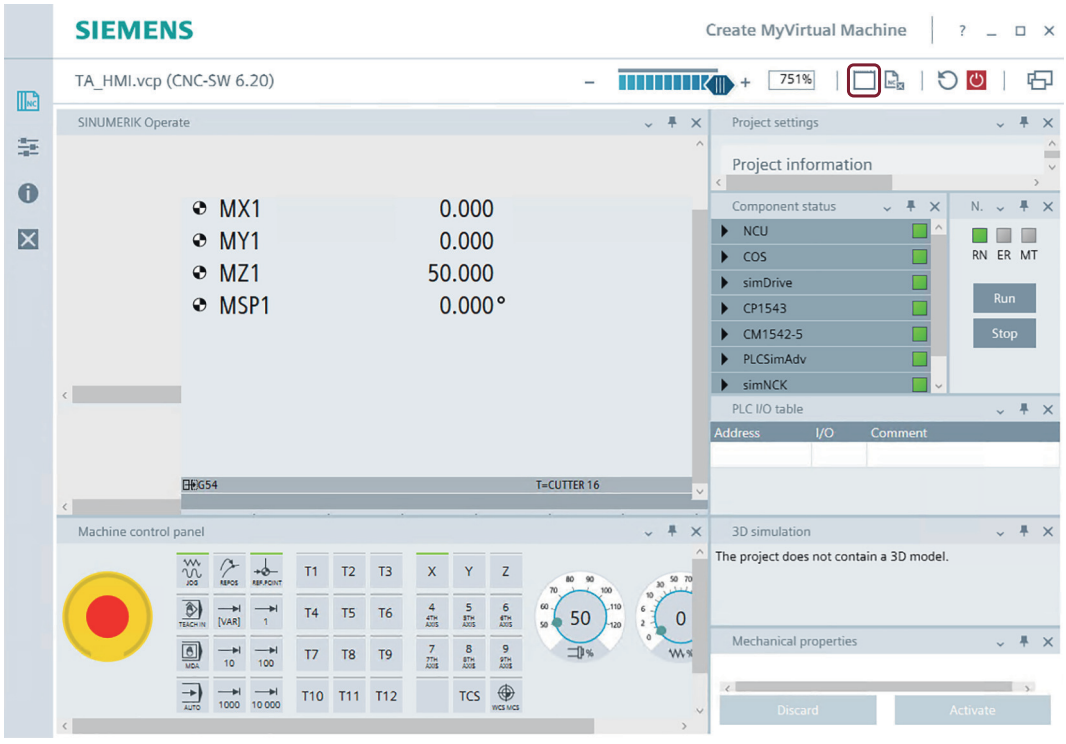
10.1 Installing OPC UA on Create MyVirtual Machine

Procedure

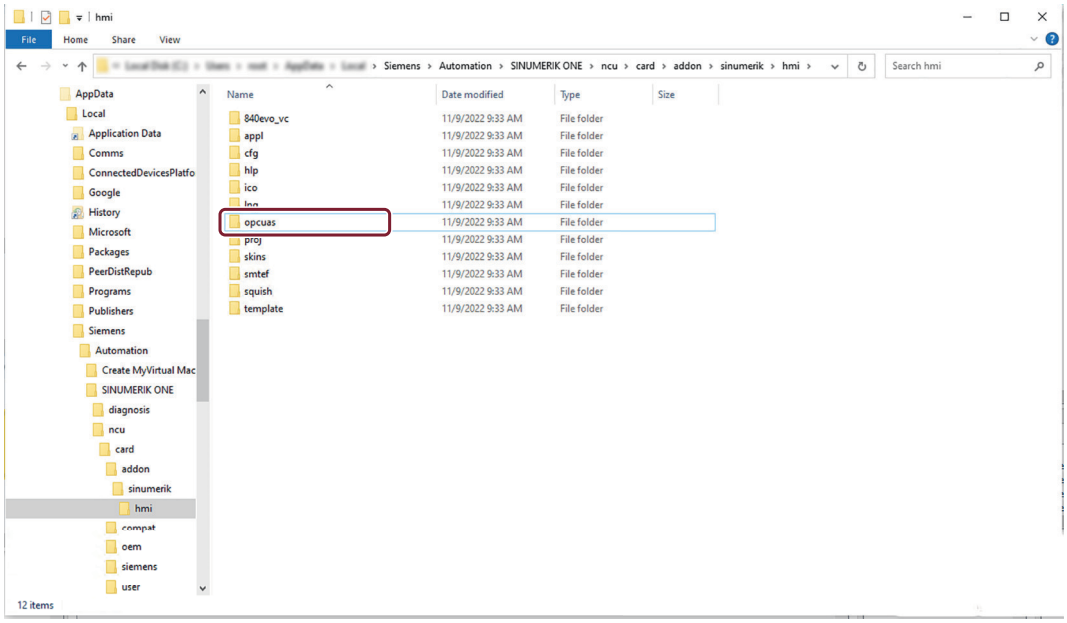
1. Open "Create MyVirtual Machine".
2. Open the project that has to be updated with OPC UA.
3. Ensure the machine is switched off before updating the OPC UA version.



4. Open the memory card in the file explorer.



5. Browse to the path "addon\sinumerik\hmi\".



6. Ensure to create a backup of the “opcuas” folder in the local location.
7. Unzip the “opcuas.zip” from software delivery.

8. Replace the ”opcuas” folder with one from the software delivery.

Note

Do not downgrade the OPC UA server version.

9. Save the project.
10. Start SINUMERIK Operate.

Note

The OPC UA version is updated only to the current project that is saved.

If the OPC UA version has to be updated for all instance, the "opcuas" folder must be updated in <CNC software installation folder>\bin\ncucontroller\cfTemplate\addon\sinumerik\hmi\opcuas.

For example: C:\Program Files (x86)\Siemens\Automation\SINUMERIK CNC-SW\SINUMERIK CNC-SW 06.21 SP00 HF00 (125)\bin\ncucontroller\cfTemplate\addon\sinumerik\hmi\opcuas

10.2 OPC UA server and virtual SINUMERIK ONE

With full functionality, the OPC UA Server is also available with the virtual SINUMERIK ONE from Run/Create MyVirtual Machine version 1.1 and from CNC software version 6.13 HF1.

By default the accessibility of the OPC UA server with the virtual SINUMERIK ONE is localhost (127.0.0.1).

For allowing further IP addresses and for accessing from external devices, please refer to settings dialog and specify additional IP addresses.

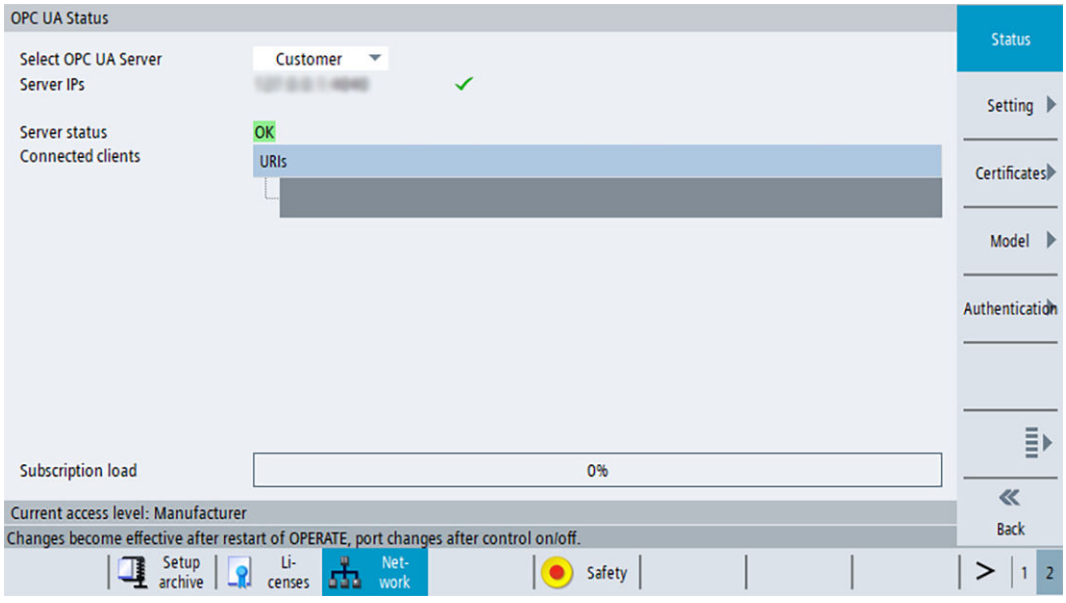


Figure 10-1 Status screen on virtual SINUMERIK ONE

Technical data

11

11.1 Technical data

Description	Value
Number of sessions [1]	10
Number of subscriptions [2]	10
Maximum number of monitored items [3]	controller specific
Min. sampling interval [4]	100 ms
Sampling intervals	{100, 250, 500, 1000, 2500, 5000} ms
Min. publishing interval	100 ms
Publishing intervals	{100, 250, 500, 1000, 2500, 5000} ms
Max. number of OPC UA users	20
Max. lifetime interval (LifeTime Count)	1 h
Session timeout	60 s
Max. monitored items queue size (Subscription Queue size) [5]	10,000
Maximum number of binary files	1 for OPC UA version <= 4.0 25 for OPC UA version > 4.1
Maximum number of CSOM namespaces	12
Maximum number of nodes in CSOM	10,000

1) Session = Connection of a client to a server

2) Subscription = Total number of subscriptions over all sessions

3) The maximum no. of monitored items depends on the sample rate of all monitored items over all sessions and a controller specific performance index.

4) Accessible communication performance / quantity structure depends on the working load of the control

5) Total queue size over all subscriptions and all monitored items

Calculating maximum no. of monitored items

Device	Max. number of monitored items	Sample rate
PPU1740/ NCU1740	100 items	100 ms
NCU1750	200 items	100 ms
NCU 1760	300 items	100 ms

Note

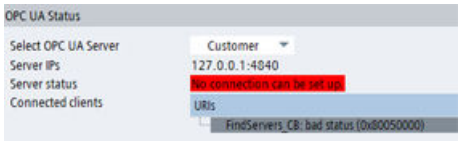
Calculation with different subscriptions and different sampling intervals:

Maximum number of monitored items = current number of monitored items * (mean value of selected sampling rates in ms / 100 ms)

Trouble shooting

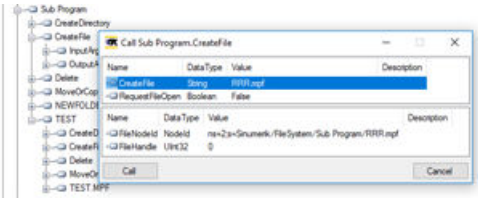
12

12.1 Frequently asked questions (FAQs)

Topic	Question	Possible solution
OPC UA client has no connection	In spite of correct commissioning my OPC UA client cannot connect. What can I do?	If no connection is possible, though you have operated the commissioning of the OPC UA server thoroughly, it is recommended to restore factory settings of the OPC UA server. Proceed as follows: • Deactivate OPC UA in the setting dialog • Switch off IPC/NCU and on again • Activate OPC UA again in the setting dialog • Switch off IPC/NCU and on again
	The server cannot be found by the client. What can I do?	• Check whether the IP address of the networking dialog is compatible to those of the OPC UA dialog. • If the IP addresses are not compatible, press "Change" in the OPC UA setting dialog. The new addresses will be directly transferred into the setting dialog. • Confirm with "Ok" and restart the SINUMERIK. The connection the server should function properly now.
	The server cannot be found by the client and the following error message is shown in the status screen:  What can I do?	• Check whether the needed port is occupied by an application (e.g. another OPC UA server) running on the system – Open the console on Windows/Linux system and type in command "netstat". – If the port for OPC UA is already in use, it will be listed as shown below:  • On IPC you can also use SIMATIC IPC DiagMonitor to check the occupied ports. • If the port is in use already, please adapt port settings for OPC UA and try again. (Per OPC UA server a unique port no. is required.)

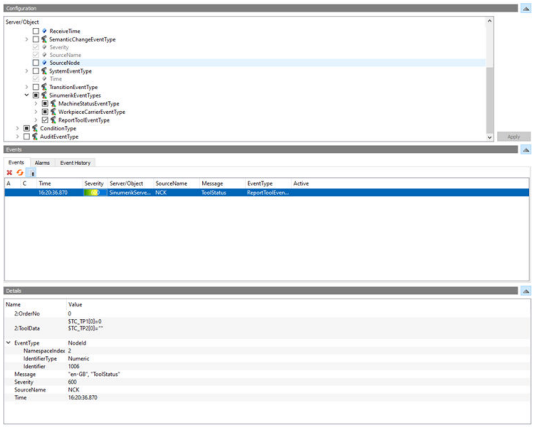
Topic	Question	Possible solution
	The OPC UA server status shows OK but the client is not able to connect. What can I do?	• Power Off/On, the control in order to activate all necessary firewall settings (e.g.: port number changed).
OPC UA export diagnosis data	How can I export OPC UA diagnosis data?	Refer to topic, "Exporting diagnosis data to an external data storage" under chapter, "Diagnosis screen (Page 187)".
OPC UA client cannot connect as certificate user	• Certificate user is created • Certificate for certificate user is trusted • OPC UA client cannot connect as certificate user	Please check date and time of the target system. In case the target system time is not within the period of validity (valid from – valid to) of the client certificate, the connection gets refused (BadIdentityTokenRejected).
OPC UA server is not accessible after update	After a server update, the server cannot be accessed anymore. What can I do?	• Check that the time on the HMI is set correctly. • Check the validity periods of the servers and client certificates. • Check whether port is open in the firewall.
AMM – SINUMERIK OPC UA Server Tools OPC UA Server Model Export	The model export does not finish. The model export does not reach 100%.	On SINUMERIK ONE it is recommended to make only relevant datablocks visible to OPC UA. • Please uncheck option “Data block accessible from OPC UA” for all non relevant datablocks in TIA portal. 
Login Issue with OPC UA client	Not able to login to OPC UA client with error message "OpcUa_BadRequestNotAllowed". Why waiting time for next attempt?	If wrong password is entered for consecutively for 3 times within a span of 5 minutes, then the user gets blocked for 20 minutes. You have to wait for 20 minutes to enter the login credentials to OPC UA client.

File System

Topic	Question	Possible solution
File System	How to use "CreateDirectory" method?	The OPC UA server can create a folder with any extension or with no extensions in the USB drive, network share, and local drive. However in the NC drive, we can create an extension with "DIR" only.
	How to use "CreateFile" method?	The "CreateFile" method is used to create a new file. The created file can be written using the "Write" method of the FileType. The OPC UA server can create file without extension in the USB drive, network share, and local drive but not in the NC memory. 
	How to use "Delete" method?	• The "Delete" method is used to delete a file/directory. The OPC UA server will not allow to delete the file if the file is opened for the file operation. You must close the file handle to delete the file. In case of directory, all the file and directory objects below the directory to delete are deleted recursively. • A file which is selected for execution or a folder which contains the selected file, are not allowed to be moved. However if the file is moved there is a possibility that an empty file will be created. • Before deleting a program, make sure it is not selected for execution. If the program is selected for execution, it is recommended to unselect it first (method "unselect").
	How to use "MoveOrCopy " method?	• The "MoveOrCopy" method is used to move or copy a file/directory to another directory or to rename a file/directory. • The OPC UA server can move any file or folder without extension in the USB drive, network share, and local drive but not in the NC memory. In the NC memory, a folder with the extension "DIR" can only be allowed to be moved to the NC memory. • The OPC UA server will not allow to move the folder if the file is opened for the file operation. • A file which is selected for execution or a folder which contains the selected file, are not allowed to be moved. However if the file is moved there is a possibility that an empty file will be created.
	How to use "Open " method?	When a client opens a file, it gets a file handle that is valid while the session is open. Clients shall use the "Close" method to release the handle when they do not need access to the file anymore. Clients can open the same file several times for read.
	How to use "Read " method?	The "Read" method is used to read a part of the file starting from the current file position. The file position is advanced by

Topic	Question	Possible solution
		the number of bytes read. The data contains the returned data of the file. If the ByteString is empty, it indicates that the end of the file is reached.
	How to use "Write " method?	The "Write" method is used to write a part of the file, starting from the current file position. The file position is advanced by the number of bytes written. When the client session is closed, all the open file handled will be closed for the respective session. In this scenario, if there is any existing file opened with write mode or append mode, the current data will be lost.
	How to use "Close " method?	The "Close" method is used to close a file represented by a FileType. When a client closes a file, the handle becomes invalid.
	How to use "GetPostion " method?	The "GetPosition" method is used to provide the current position of the file handle. If a "Read" or "Write" method is called, it starts at that position.
	How to use "SetPosition " method?	The "SetPosition" method is used to set the current position of the file handle. If a "Read" or "Write" method is called, it starts at that position. If the position is higher than the file size, the position is set to the end of the file.
	Does OPC UA standard file system support 1:N constellation?	No, only the default machine name is mapped (target system IPC only).

OPC UA Advanced

Topic	Question	Possible solution
ReportToolEvent	What does it mean if ToolData string of ReportToolEvent is empty?	PLC reported a nonexistent tool: 

See also

Technical support (https://support.industry.siemens.com/cs/sc/2090/)

12.2 Reference to OPC UA error code

You can find all relevant information on error codes at Github (https://github.com/OPCFoundation/UA-Nodeset/blob/v1.04/DotNet/Opc.Ua.StatusCodes.cs).

Technical Support

Country-specific telephone numbers for technical support are provided in the Internet at the following address (https://support.industry.siemens.com/cs/sc/2090/) in the "Contact" area.

If you have any technical questions, use the online form in the "Support Request" area.

Additional information for OPC UA Advanced

A

A.1 PLC interface information

The OPC UA advanced DB interface (OpcUaAdv) is required for communication between OPC UA Advanced and the PLC for the following cases:

- Machine status
- Tool status
- Workpiece status
- Docking station status
- Tool operation, like reading tool data
- Magazine data
- Creating and deleting a tool

Note

OPC UA Advanced relies on the correct set up of the OPC UA advanced DB, as described below. Ensure the symbolic names of PLC variables and the data types are correct.

Note

OPC UA Advanced DB interface compatible to CMI interface

As OPC UA Advanced is the successor of SINUMERIK Integrate Create MyInterface (CMI), OPC UA Advanced DB interface is fully compatible with the PLC interface (DB12) of CMI.

Specific use case related procedures

You can find a specific and use case related procedure in the chapter OPC UA Advanced use cases (Page 170).

A.1.1 Symbolic DB

PLC variable / symbolic name	Data type	OFFSET	Set and reset	Explanation	
PLCReq	Byte	0	Set by PLC Reset by OPC UA server.	Request from PLC (Page 221)	
Trigger	Byte	1	Set by PLC	Change trigger (Page 222)	
SCReq	Byte	2	N.A.	RPC SINUMERIK	This byte is not used.
MODE_PLC	Byte	3	Set by PLC	Machine mode (Page 222)	
MODE_RPC	Byte	4	Set by PLC	RPC mode (Page 223)	
DataTyp	Byte	5	Set by PLC	Identifier for tool data	This byte is not used.
MagNum	Word	6	Set by PLC	Magazine number (Page 224)	
PlaceNum	Word	8	Set by PLC	Location number (Page 224)	
TNum	Word	10	Set by PLC	T number (Page 224)	
DockPosCount	Byte	12	Set by PLC	Number of docking positions (Page 224)	
NC4WpcCount	Byte	13	Set by PLC	Number of program assignments	This byte is not used.
MachineStatus	Byte	14	Set by PLC	Machine status (Page 224)	
MachineMode	Byte	15	Set by PLC	NC operating mode (Page 225)	
Reserve1	Word	16	Set by PLC	Reserved (Page 225)	
Reserve2	Word	18	Set by PLC	Reserved (Page 225)	

PLC variable / symbolic name	Data type	OFFSET	Set and reset	Explanation
Array[0-11] of Docking stations	Struct			Docking stations
DocPosStatus	Byte	20+ (n*12)	Set by PLC	Docking position status (Page 225)
WPCStatus	Byte	21+ (n*12)	Set by PLC / Host computer	Workpiece C Status (Page 225)
WPC	Byte[6]	22--27+ (n*12)	Set by PLC	Workpiece C (Page 228)
ClampCubeSide	Word	28-29+ (n*12)	Set by PLC	Machining side (Page 228)
FB	Byte	30+ (n*12)	Set by PLC	Follow-up machining (Page 228)
Reserve1	Byte	31+ (n*12)	Set by PLC	Reserve (Page 228)
HostClientConnected	Bool	260	Set by host computer	HostClientConnected (Page 229)
PLCReportType	Byte	261	Set by PLC	(Page 229)
ToolDataForToolCreate	Struct			This structure is used to create a new tool on the NC Memory. Values like ToolArea, ForceToolNumber and ForceDuploNumber in this structure needs to be updated before setting ToolCreateRequest byte.
ToolCreateRequest	Bool		Set by PLC and reset by OPC UA server	ToolCreateRequest (Page 230)
DuploNumber	Word		Set by PLC and reset by OPC UA server	DuploNumber (Page 230)
ToolIdentifier	String[34]		Set by PLC and reset by OPC UA server	ToolIdentifier (Page 230)
ToolArea	Byte		Set by PLC and reset by OPC UA server	ToolArea (Page 230)

PLC variable / symbolic name	Data type	OFFSET	Set and reset	Explanation
ForceToolNumber	Bool		Set by PLC and reset by OPC UA server	ForceToolNumber (Page 230) For more use case related information, see chapter Hotspot-Text (Page 170).
ForceDuploNumber	Bool		Set by PLC and reset by OPC UA server	Hotspot-Text (Page 230) For more use case related information, see chapter Hotspot-Text (Page 170).
ToolCreated	Word		Set by OPC UA server and reset by PLC	Hotspot-Text (Page 231)
ToolNumberCreated	Word		Set by OPC UA server and reset by PLC	Hotspot-Text (Page 231)
DuploNumberCreated	Word		Set by OPC UA server and reset by PLC	Hotspot-Text (Page 231)
ToolCreateBusy	Bool		Set by OPC UA server and reset by OPC UA	Hotspot-Text (Page 231)
ErrorInfo	Byte		Set by OPC UA server and reset by PLC	Hotspot-Text (Page 231)
ToolDataForToolDelete	Struct			ToolDataForToolDelete
ToolDeleteRequest	Bool		Set by PLC and reset by OPC UA server	Hotspot-Text (Page 232)
ToolNumber	Word		Set by PLC and reset by OPC UA server	Hotspot-Text (Page 232) If the user sets the ToolNumber, then the DuploNumber and and ToolIdentifier needs to be set to 0.

PLC variable / symbolic name	Data type	OFFSET	Set and reset	Explanation
DuploNumber	Word		Set by PLC and reset by OPC UA server	Hotspot-Text (Page 230) If the user sets the DuploNumber and ToolIdentifier, then the ToolNumber needs to be set to 0.
ToolIdentifier	String[34]		Set by PLC and reset by OPC UA server	Hotspot-Text (Page 230) If the user sets the DuploNumber and ToolIdentifier, then the ToolNumber needs to be set to 0.
ToolArea	Byte		Set by PLC and reset by OPC UA server	Hotspot-Text (Page 230)
ToolDeleted	Bool		Set by OPC UA server and reset by PLC	Hotspot-Text (Page 232)
ToolDeleteBusy	Bool		Set by OPC UA server and reset by OPC UA	Hotspot-Text (Page 232)
ErrorInfo	Byte		Set by OPC UA server and reset by PLC	Hotspot-Text (Page 232)

See also

ToolReportType Enum value (Page 244)

A.1.2 Request from PLC (byte 0)

Bit No.	Function	Set / Reset
0	Workpiece holder status changed	Set by PLC Reset by OPC UA server
1	Report tool	Set by PLC Reset by OPC UA server
2	Status change	Set by PLC Reset by OPC UA server

The PLC indicates changes in the interface with this byte. Once the PLC has set the request byte, it must set the next bit in the trigger byte.

The PLC may only write in the request byte again when it has been set to 0 by OPC UA Advanced after processing.

Workpiece holder status changed (byte 0, bit 0)

Workpiece holder status changed is set by the PLC whenever the status of one of the workpiece holders has been changed by the PLC, see also section: OPC UA Advanced DB interface docking position information of the machine or in the case of pallet movements within the machine. OPC UA Advanced must then send the WorkpieceCarrierEventType event to the host computer.

Report tool (byte 0, bit 1)

Report tool is set by the PLC whenever a tool is to be reported to the host computer (e.g. tool breakage). Which tool is to be reported is indicated in the data elements: "Magazine number" and "Location number or T number".

Status change (byte 0, bit 2)

Status change is set by the PLC for each status change (machine mode, RPC mode, machine status, NC mode) about which the host computer is to be informed. OPC UA Advanced must then send MachineStatusEventType event to the host computer, e.g. workpiece holder arrival, NC start, NC end, mode switchovers, etc.

A.1.3 Change trigger (byte 1)

PLC sets one bit each in this byte, when changes by the PLC are present. OPC UA monitors the bytes, when there is a change in the value of the variable the event is triggered.

For each new trigger, the PLC must increment the value. When the max value is reached, it need to reset to 0.

A.1.4 Machine mode (byte 3)

Bit No.	Function	Set / Reset
1	Components switched off (after-work hours)	PLC
2	Write access to docking position data allowed	PLC

Components switched off (after work hours) (byte 3, bit 1)

Components switched off is set by the PLC when this status is reached. The request for this is made by bit 1 in the request identifiers.

Write access to docking position data permitted (byte 3, bit 2)

Write access to docking position data allowed is set by the PLC in reaction to the "Request write access to docking position data" request. These coordinated actions prevent OPC UA Advanced from writing incorrect docking positions due to a pallet movement.

A.1.5 RPC SINUMERIK mode (byte 4)

Bit No.	Operating mode	Access from
0	Autonomous host computer mode	Operator
1	Non autonomous host computer mode	Operator
2	Manual mode	Operator
3	Special mode	Host computer, Operator
4	Offline host computer 1	Host computer
6	For MachineStatusEventType, is signaled as machine mode 500	PLC
7	For MachineStatusEventType, is signaled as machine mode 600	PLC

Host computer mode autonomous/non autonomous (byte 4, bit 0 and bit 1)

In the autonomous and non autonomous host computer modes, the NC is started by the PLC, initiated by the host computer. The autonomous and non autonomous host computer modes differ in that, in autonomous production there may be different strategies for handling malfunctions than in non autonomous production.

Manual mode (byte 4, bit 2)

In manual mode, there is no automatic NC start, but material flow on the machine is still automatic.

Special mode (byte 4, bit 3)

In special mode there is no automatic NC start and no automatic material flow on the machine.

Offline host computer 1 (byte 4, bit 4)

Offline means that the client of the host is not connected to the OPC UA server. This is used to indicate to the PLC whether the host is connected or not.

This bit is only set, if the host has registered via the method RegisterAdvHost.

0= Client connected (online)

1= Client not connected (offline)

For MachineStatusEventType is signaled as `machine mode 500` (byte 4, bit 6)

For bit 6 = 1, for MachineStatusEventType is signaled as `machine mode 500` to the host computer.

For MachineStatusEventType, is signaled as `machine mode 600` (byte 4, bit 7)

For bit 7 = 1, for MachineStatusEventType is signaled as `machine mode 600` to the host computer.

A.1.6 Magazine number, location number, T number (word 6 - 7, 8 - 9, 10 - 11)

The tool to be reported is specified by magazine number and location number or alternatively by T number. When the T number is specified, magazine number and location are set to 0 and vice versa.

With the request from the PLC "Report tool", OPC UA Advanced is prompted to read the following data elements and to transfer the tool data to the host computer.

- Magazine number (word 6 - 7)
- Location number (word 8 - 9)
- T number (word 10 - 11)

OPC UA Advanced must then clear these data elements (by filling them with zeros).

Note

Tool reports

Tool reports that occur within the scope of tool loading and unloading are not initiated by the PLC user program.

A.1.7 Number of docking positions (byte 12)

The number of docking positions of the machine (maximum 12) is statically stored when the machine is commissioned. It corresponds to the number of docking position data blocks in the interface.

A.1.8 Machine status (byte 14)

The machine status is reported to the host computer. Byte 14 must be set by the OEM on the PLC side.

Bit No.	Function	Access from
0	Machine is active	PLC
1	Machine fault	PLC
2	Restart of the machine	PLC

Machine status

Byte 14 must be supplied by the OEM on the PLC side.

When the machine status is read out from OpcUaAdv.MachineStatus, OPC UA Advanced responds as follows:

Bit 1 of OpcUaAdv. MODE_PLC	Bit 0 of OpcUaAdv. MachineStatus	Bit 1 of OpcUaAdv. MachineStatus	Bit 2 of OpcUaAdv. MachineStatus	Status	Meaning
1	X	X	X	4	Components switched off
0	1	0	0	2	Active
0	0	1	0	3	Fault
0	0	0	1	0	Restart
0	0	0	0	1	Inactive

A.1.9 NC operating mode (byte 15)

Byte 15 must be supplied by the OEM on the PLC side.

The NC operating mode is reported to the host computer. In the host computer you can evaluate.

Table A-1 NC operating mode

Bit No.	Function	Access from
0	AUTO	PLC
1	MDA	PLC
2	JOG	PLC
3	TEACH IN	PLC

A.1.10 Reserve 1 (word 16 - 17) and reserve 2 (word 18 -19)

The machine manufacturer can use these variables as required by way of the PLC. The values are reported to the host computer.

These values can then be evaluated by the host computer and processed further.

A.1.11 Docking position information of the machine

Introduction

The docking position data describes the respective machine location, machining location, In/Out location.

The number of machine locations is stored in OPC UA Advanced DB "Number of docking positions" of the global data.

The following docking position data is available:

Table A-2 Docking position data

Data element	Short name	Data type	Set / Reset	Offset
Docking position status	DockPosStatus	Byte	PLC	20 + (n * 12)
Workpiece holder status	WPCStatus	Byte	PLC/Host computer	21 + (n * 12)
Workpiece holder	WPC	Byte[6]	PLC	22 - 27 + (n * 12)
Machining side	Clamp-CubeSide	Word	PLC	28 - 29 + (n * 12)
Follow-up machining	FB	Byte	PLC	30 + (n * 12)
Reserve	Reserve1	Byte	PLC	31 + (n * 12)

Docking position status (byte 20 + [n * 12])

For every docking position, a block of 12 bytes must be attached in OPC UA advanced DB, from byte 20.

n = number of the docking position

1st docking position = 0

2nd docking position = 1 .... etc.

A maximum of 12 docking positions can be used.

The following states are available:

- Bit 0 = Fault
- Bit 1 = Disabled

The bit field describes the current status of the docking position. It is set by the PLC. If no bit is set, the docking position is enabled.

The "Fault" bit is set or cleared on the basis of I/O signals. The cause of the fault is communicated to the host computer using the "Reports" function ReportWpcErrorEventType.

The PLC does not execute any pallet transports between docking positions with "Fault" status.

If the docking position has the "Disabled" status, it may not be approached by the transport system.

Workpiece holder status (byte 21 + [n * 12])

Table A-3 Workpiece holder states

Bit No.	Function	Status	Set / Reset
0	Newly delivered (without program assignment)	1	PLC / Host computer
1	Machining is planned (with program assignment)	2	PLC / Host computer
2	Prepare program selection	4	PLC / Host computer
3	Program selection complete	8	PLC / Host computer

Bit No.	Function	Status	Set / Reset
4	In progress	16	PLC / Host computer
5	Machining finished	32	PLC / Host computer
6	Machining aborted	64	PLC / Host computer
7	Machining is not planned (only for buffering)	128	PLC / Host computer

Newly delivered (byte 21, bit 0)

The "Newly delivered" status (status 1) is assigned by the PLC for a newly delivered workpiece holder.

Exception: Machining is not planned.

This status causes the host computer to perform the program assignment. When the program has been assigned, the "Machining is planned" status (status 2) is set by the host computer.

Machining is planned (byte 21, bit 1)

For a workpiece holder with the "Machining is planned" status (status 2), the PLC sets the "Prepare program selection" status as soon as the active machining has finished (status 4).

Prepare program selection (byte 21, bit 2)

The precondition for "Prepare program selection" (status 4) is that the program assigned for this pallet by the host computer has been loaded into the NC and selected for machining.

Program selection complete (byte 21, bit 3)

When the program has been selected, the host computer sets the "Program selection complete" status (status 8) for the associated workpiece holder.

The PLC can now initiate the NC start. However, the machine manufacturer is responsible for ensuring the PLC will only start the NC when all safety-related requirements are met, e.g. protective door closed, etc.

Machining started (in progress) (byte 21, bit 4)

Once machining has been started, the PLC sets the "In progress" status (status 16). When machining has finished, the PLC then sets the "Machining finished" status (status 32) for the appropriate workpiece holder. Workpiece holders with the "Machining finished" status (status 32) are transported independently to an unloading location by the PLC.

Machining finished (byte 21, bit 5)

When the "Follow-up machining" identifier is set, the workpiece holder remains at the machining location. As response to the "Machining finished" status (status 32), the host computer sets the "Machining is planned" status (status 2) again. As subsequent response, the PLC uses the "Prepare program selection" status (status 4) to prompt the host computer to select the program for the next machining steps. The remaining sequence of operations is the

same as for initial machining.

Machining aborted (byte 21, bit 6)

The "Machining aborted" status (status 64) is set when a workpiece holder will no longer be machined after a malfunction. This identifier might occur, in particular, in unmanned

production. A workpiece holder with this identifier may not be brought to another machine for a further operation. It may only be brought to a storage location.

Machining is not planned (byte 21, bit 7)

A workpiece holder that was delivered for buffering by the TPS is assigned the "Machining is not planned" status (status 128) and not the "Newly delivered" status (status 1). The TPS transfers this information to the PLC. For workpiece holders with this status, there is no program assignment by the host computer.

A.1.12 Workpiece carrier (byte 22 - 27)

Identifier of the workpiece holder that is currently located at the docking position, e.g. "WST01". This information is entered by the PLC. This requires that this information can be transferred from the transport system or directly from the workpiece holder. If no workpiece holder is in the docking position, the field must be filled with binary 0. The identifier must be terminated with '\0' after the last character because the host computer expects a string. This means only identifiers up to 5 bytes long are permissible.

A.1.13 Machining side (byte 28 - 29)

The host computer sets this information simultaneously with the "Program selection complete" machining status (status 8). The PLC uses this value to set a corresponding side of a clamp cube for machining, or passes this value on to the NC.

The computer link specifies the order in which the sides of clamp cubes will be machined based on the program assignment. If the PLC is to be able to influence the machining order, an entry can be optionally made in the configuration file of the host computer that causes the program assignment data to be mirrored in a separate data block of the PLC. The PLC then has read access to this data.

The host computer is notified of the machining side selected by the PLC in the "Machining side" field. This takes place simultaneously with the setting of the "Prepare program selection" machining status (status 4). The host computer selects the program for the side specified by the PLC. The rest of the sequence remains unchanged.

A.1.14 Follow-up machining (byte 30)

The host computer sets this identifier simultaneously with the "Program selection complete" machining status (status 8). It informs the PLC as to whether - in addition to the current machining - follow-on machining is planned. The PLC uses this information to control the workpiece holder transport within the machine.

A.1.15 Reserve (byte 31)

Free for the OEM.

A.1.16 HostClientConnected (byte 260)

Bit No.	Function	Set / Reset
0	Host client is not connected. RPC SINUMERIK mode (byte 4) is switched to 1.	Host computer, Operator
1	Host client is connected. RPC SINUMERIK mode (byte 4) is switched to 0.	Host computer, Operator

There are two possibilities to see whether the host client is connected:

- Byte 4 for compatibility reason with "Create MyInterface"
  Value of variable MODE_RPC (Byte 4):
  - Bit 4 = 0: Client connected
  - Bit 4 = 1: Client disconnected
- The variable "HostClientConnected" for a comfortable way to use
  Please note that this works only if the client has been registered. (Page 182)
  HostClientConnectedted
  - 0= Client disconnected
  - 1= Client Connected

A.1.17 PLCReportType (byte 261)

Value	Function	Set / Reset
0-6	If PLCReportType is set to 0-6, the ReportToolEvent will deliver a ToolReportType 1 (PLCReport).	Set by PLC Reset by OPC UA server
7	If PLCReportType is set to 7, the ReportToolEvent will deliver a ToolReportType 7 (PlcLoadReport).	Set by PLC Reset by OPC UA server
8	If PLCReportType is set to 8, the ReportToolEvent will deliver a ToolReportType 8 (PlcUnLoadReport).	Set by PLC Reset by OPC UA server
9	If PLCReportType is set to 9, the ReportToolEvent will deliver a ToolReportType 9 (PlcMoveReport).	Set by PLC Reset by OPC UA server
>9	If PLCReportType is set greater than 9, the ReportToolEvent will deliver a ToolReportType with the given value.	Set by PLC Reset by OPC UA server

A.1.18 ToolDataForToolCreate

A.1.18.1 ToolCreateRequest

ToolCreateRequest bit 0 (Bool 164.0) is used to trigger new tool creation procedure. At the end of every tool creation, this bit will be reset by OPC UA server.

Before triggering the "ToolCreateRequest" oem/user must set the below bytes:

- Duplo number and ToolIdentifier
- or ToolNumber
- ToolArea
- ForceDuploNumber (optional)
- ForceToolNumber (optional)

A.1.18.2 DuploNumber

DuploNumber (Word 166) indicates the sister tool number.

A.1.18.3 ToolIdentifier

ToolIdentifier (String 34) of upto 32 characters. Unique name of the tool.

A.1.18.4 ToolArea

ToolArea (Byte 204) indicates the tool area number. Maximum value can be 10.

A.1.18.5 ForceToolNumber

ForceToolNumber (Bool 205.0) is used to create the tool only with a specified tool number in tooldata.

Bit description

- 1 - if the tool with same tool number exists, then the tool won't be created and an error will be reported.
- 0 - if the tool with same tool number exists, then the next available tool number will be used.

A.1.18.6 ForceDuploNumber

ForceDuploNumber (Bool 205.1) is used to create the tool only with a specified duplo number in tooldata.

Bit description

- 1 - if the tool with same duplo number exists, then the tool won't be created and an error will be reported.
- 0 - if the tool with same duplo number exists, then the next available duplo number will be used.

A.1.18.7 ToolCreated

ToolCreated (Bool 205.2) indicates if the tool is created or not.

Bit description

- 1 - if tool is created successfully.
- 0 - if tool is not created.

A.1.18.8 ToolNumberCreated

ToolCreated (word 206) is the tool number of the created tool.

A.1.18.9 DuploNumberCreated

DuploNumberCreated (word 208) is the duplo number of the created tool.

A.1.18.10 ToolCreateBusy

ToolCreateBusy (bool 210.0) indicates if there is already a tool creation procedure in progress.

Bit description

1 - Tool creation process in progress. Cannot start a new tool creation.

0 - Tool creation process can be started.

A.1.18.11 ErrorInfo

ErrorInfo (Word) indicates the error number in case the tool creation fails.

These error numbers correspond to different error scenarios while tool is created. This value is reset to 0 when a ToolCreateRequest is set by PLC.

A.1.19 ToolDataForToolDelete

A.1.19.1 ToolDeleteRequest

ToolDeleteRequest (Bool) is used to trigger tool delete procedure.

At the end of every tool delete, this bit will be reset by OPC UA server.

A.1.19.2 ToolNumber

ToolNumber (Word) indicates a unique number of the tool to be deleted.

A.1.19.3 DuploNumber

DuploNumber (Word 166) indicates the sister tool number.

A.1.19.4 ToolIdentifier

ToolIdentifier (String 34) of upto 32 characters. Unique name of the tool.

A.1.19.5 ToolArea

ToolArea (Byte 204) indicates the tool area number. Maximum value can be 10.

A.1.19.6 ToolDeleted

ToolDeleted (Bool) indicates if the tool is deleted or not.

Bit description

- 1 - If tool is deleted successfully.
- 0 - if tool deletion has failed.

A.1.19.7 ToolDeleteBusy

ToolDeleteBusy (Bool) indicates if there is already a tool deletion procedure is in progress.

Bit description

- 1 - Tool deletion process in progress. Cannot start a new tool delete.
- 0 - Tool deletion process can be started.

A.1.19.8 ErrorInfo

ErrorInfo (Word) indicates the error number in case the tool deletion fails.

These error numbers correspond to different error scenarios while tool is created. This value is reset to 0 when a ToolCreateRequest is set by PLC.

A.2 Detailed description of OPC UA events, methods and variables

A.2.1 OPC UA Advanced events

A.2.1.1 Overview

Note

Target group

The following descriptions are intended for the target group of system integrators. For more information on PLC programming and communication interaction with OPC UA Advanced, you can find further information in chapter PLC interface information (Page 217).

The functionality of OPC UA Advanced relies on the PLC programmer to interact with the interface DB correctly as described in the chapter PLC interface information (Page 217). Otherwise, the functionality of OPC UA Advanced will not work correctly.

When using the OPC UA Advanced functionalities you need to subscribe to "ADV" node or child nodes of "ADV". Or additionally the status can be requested by calling the corresponding method. The following is a list of each event's description.

Events

- Machine (Page 236)
- ToolManagement (Page 242)
- WorkpieceCarrier (Page 255)

A.2.1.2 Subscribing to OPC UA Advanced events

In order to get the events, the events must be subscribed. They can either be subscribed individually at the nodes "Machine", "ToolManagement" or "WorkpieceCarrier" or "Adv" node for all events .

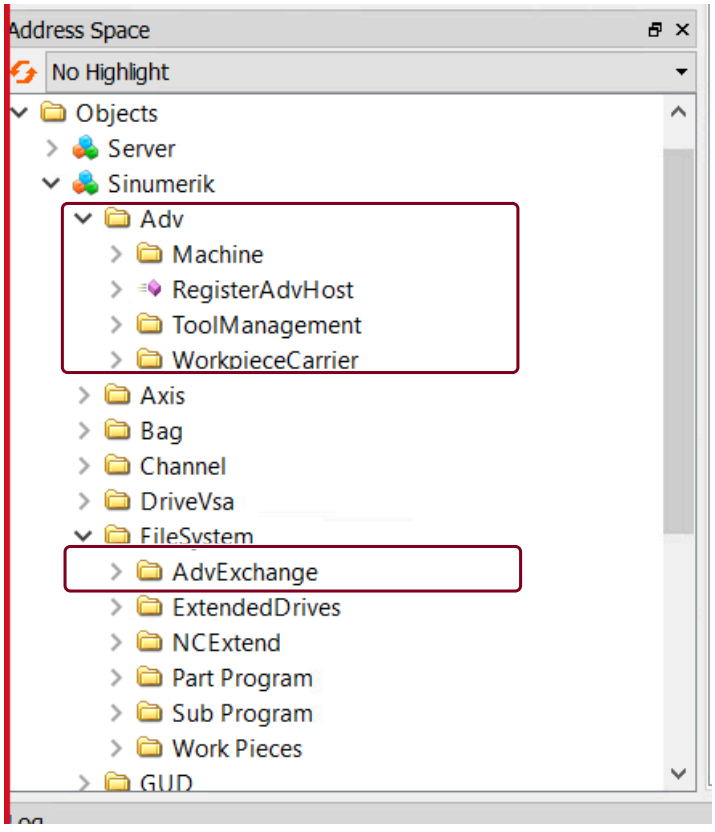


Figure A-1 Subscribing to OPC UA Advanced events

1. The MachineStatusEventType can be subscribed at the "Machine" node.
2. The ReportToolEventType can be subscribed at the "ToolManagement" node.
3. The WorkpieceCarrierEventType can be subscribed at the "WorkpieceCarrier" node.

A.2.1.3 Attributes of BaseEventType

The attributes of BaseEventType forms the basis for all the event type attributes, the description for each BaseEventType attribute is described.

Attributes of BaseEventType

Attribute	Data type	Mapping with respect to SINUMERIK	Description
EventId	String	Unique node id generated from SINUMERIK system.	EventId is generated by the server to uniquely identify a particular event notification. The EventId shall always be returned as value and the server is not allowed to return a StatusCode for the EventId indicating an error.
EventType	NodeId	It is always set to 'EventType'.	The EventType shall always be returned as value and the server is not allowed to return a StatusCode for the EventType indicating an error.
SourceNode	NodeId	Event source identifier provided by SINUMERIK system.	SourceNode identifies the node that the event originated from. If the event is not specific to a node, the NodeId is set to null.
SourceName	String	Supported event source names are always CNC.	SourceName provides a description of the source of the event. This could be the string-part of the DisplayName of the event source using the default locale of the server.
Time	UtcTime	Event time stamp	Time provides the time of the event occurred. Once set, intermediate OPC UA servers shall not alter the value.
ReceiveTime	UtcTime	Event time stamp of the server.	ReceiveTime provides the time the OPC UA server received the event from the underlying device of another server.
Message	Localized Text	Message will be displayed as based on the events	Event message provides a human readable and localizable text description of the event.
Severity	UInt16	Reading attributes via (SLAE_EV_ATTR_SEVERITY)	Severity of the event message. Event severity is always set to 600.
LocalTime	TimeZoneDataType	Offset and the DaylightSavingInOffset flag	LocalTime is a structure containing the Offset and the DaylightSavingInOffset flag. The Offset specifies the time difference (in minutes) between the Time Property and the time at the location in which the event was issued. If DaylightSavingInOffset is - TRUE: Standard/Daylight savings time (DST) at the originating location is in effect and Offset includes the DST correction. FALSE: The Offset does not include DST correction and DST may or may not have been in effect.

A.2.1.4 Machine

MachineStatusEventTypes

This event is raised when the PLC reports a new machine status via OPC UA Advanced DB interface. You have to subscribe to the Machine node to get notified by the MachineStatusEventType.

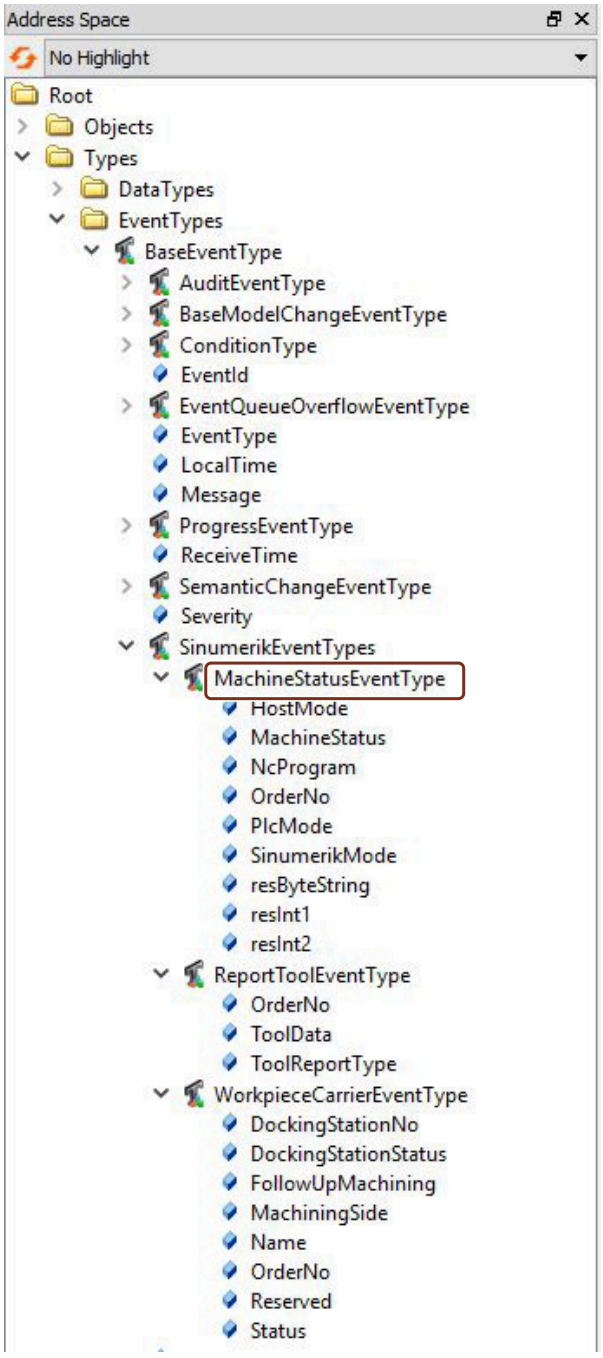


Figure A-2 MachineStatusEventType

Since the MachineStatusType is derived from the BaseEventType (Page 234), it does contain also all the attributes which are inherited from this object.

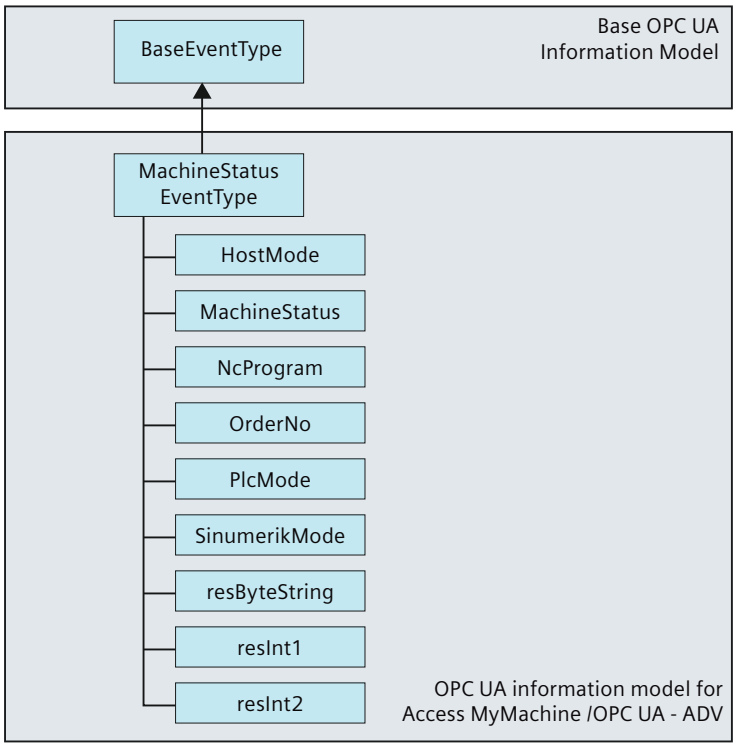


Figure A-3 OpcUa Information Model for MachineStatusEventType

Additional attributes of the MachineStatusEventType

Attribute	Data type	Mapping with respect to SINUMERIK	PLC source	Description
orderNo	OpcUa_UInt32	-	-	Order number of request. If the event is triggered by the host via method, the order number provided in the method is returned. If the event is triggered by the PLC the orderNo == 0.
sinumerik-Mode	SinumerikMode	bytes (long int), source PLC interface (AdvOpcUa Symbolic DB)	Machine-Mode	Operating mode sinumerikMode enum value is shown here. For details and explanation see chapter SinumerikMode Enum value (Page 239) • 0 - None • 1 - Automatic • 2 - MDA • 4 - JOG • 8 - TeachIn

Attribute	Data type	Mapping with respect to SINUMERIK	PLC source	Description
hostMode	HostMode	4 bytes (long int), source PLC interface (AdvOpcUa Symbolic DB)	MODE_RPC	Operating mode hostMode enum value is shown here. For details and explanation see chapter HostMode Enum value (Page 240) • 0 - None • 1 - UnManned • 2 - Manned • 3 - Manual • 4 - Special • 5 - HostMode500 • 6 - HostMode600
plcMode	PlcMode	4 bytes (long int), source PLC interface (AdvOpcUa Symbolic DB)	MODE_PLC	PlcMode enum value is shown here. For details and explanation see chapter PlcMode enum value (Page 240) • 0 - NoDB • 1 - DBExists
machineStatus	MachineStatus	4 bytes (long int), source PLC interface (AdvOpcUa Symbolic DB)	MachineStatus	Machine status enum value is shown here. See chapter MachineStatus Enum value (Page 241) • 0 - Restart • 1 - Inactive • 2 - Active • 3 - Error • 4 - ComponentsOff
ncProgram	OpcUa_String	-	-	Path to part program name of channel 1. OPC UA folder syntax, e.g.: Sinumerik/FileSystem/Part Program/HELIX.MPF
resInt1 [1)]	OpcUa_UInt32	4 bytes (long int), source PLC interface (AdvOpcUa Symbolic DB)	Reserve1	Reserve 1
resInt2 [1)]	OpcUa_UInt32	4 bytes (long int), source PLC interface (AdvOpcUa Symbolic DB)	Reserve2	Reserve 2
resByteString	OpcUa_ByteString	8 bytes (string),	-	Reserve 3

1) Resint1 and 2 are placed on the DB interface of the PLC, values are entered there by the PLC, they are transferred to the host computer.

Note

Further information

You can find more detailed description of PLC interaction in chapter PLC interface information (Page 217).

Machine Enumerations

The Machine Status Type enumerations are as follows:

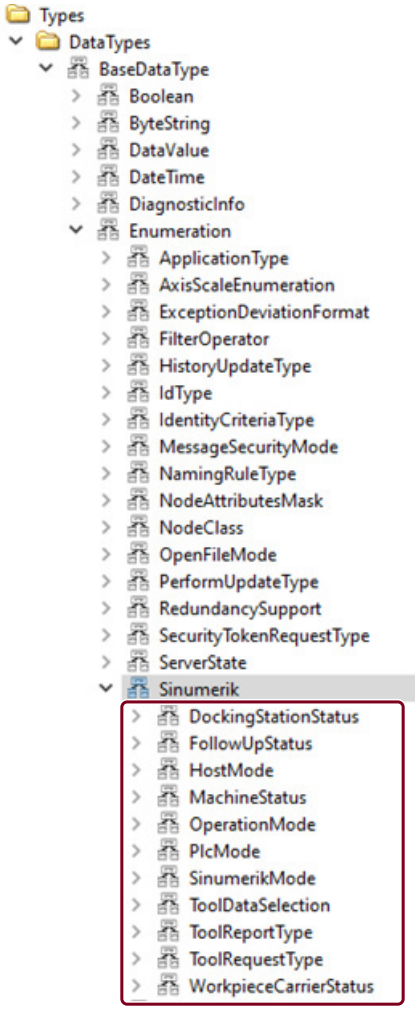


Figure A-4 MachineEnum

- SinumerikMode Enum (Page 239)
- HostMode Enum (Page 240)
- PlcMode Enum (Page 240)
- MachineStatus Enum (Page 241)

SinumerikMode Enum value

The SInumerikMode Enum is defined as follows:

The enumeration is defined in the following table:

Name	Value	Description
None	0	No mode selected in the control
Automatic	1	Automatic mode is selected in the control
MDA	2	MDA mode is selected in the control
JOG	4	JOG mode is selected in the control
TeachIn	8	TeachIn mode is selected in the control

Further information

You can find more detailed description of PLC interaction in chapter Symbolic DB (Page 218).

HostMode Enum value

The HostMode Enum is defined as follows:

The enumeration is defined in the following table:

Name	Value	Description
None	0	No host computer mode selected
UnManned	1	If a malfunction occurs, it is possible to terminate the machining of the malfunctioning workpiece holder and continue with another workpiece holder.
Manned	2	If a malfunction occurs, the workpiece holder remains in the machine. Operator must eliminate the malfunction before the machining can be continued.
Manual	3	The host computer specifies, selects and displays the NC program which the operator must start.
Special	4	Operator controls the transport of workpiece holders, transfer of program assignments and starting the NC program.
HostMode500	5	For bit 6 = 1, for event is signaled as machine mode 500 to the host computer.
HostMode600	6	For bit 7 = 1, for event is signaled as machine mode 600 to the host computer.

Further information

You can find more detailed description of PLC interaction in chapter Symbolic DB (Page 218).

PlcMode Enum value

The PlcMode Enum is defined as follows:

The enumeration is defined in the following table:

Name	Value	Description
NoDB	0	PLC DB not configured
DBExists	1	PLC DB configured

Note

For OPC UA Advanced mode operation, OPC UA advanced DB interface settings have to be set as stated in the Annex, chapter Symbolic DB (Page 218).

MachineStatus Enum value

The MachineStatus Enum is defined as follows:

The enumeration is defined in the following table:

Name	Value	Description
Restart	0	Restart of the control.
Inactive	1	No NC command is active in the NC channel. For example, channel is idle.
Active	2	There is an active command being executed by the NC channel.
Error	3	The NC execution is interrupted. Execution of a program in the channel can be re-started.
ComponentsOff	4	Components switched off.

Further information

You can find more detailed description of PLC interaction in chapter Symbolic DB (Page 218).

A.2.1.5 ToolManagement

ReportToolEventType

This event is raised for events indicated by PLC OPC UA advanced DB interface. You have to subscribe to the ToolManagement node to get notified by the ReportToolEventType.

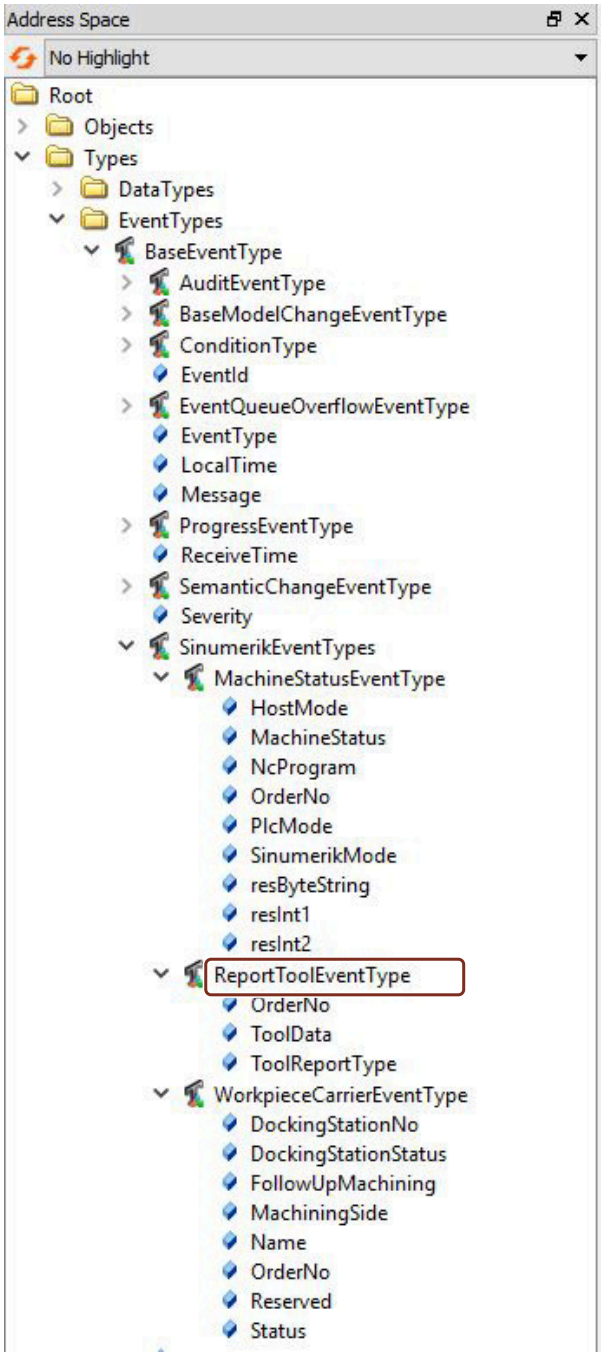


Figure A-5 ReportToolEventType

Since the ReportToolEventType is derived from the BaseEventType (Page 234), it does contain also all the attributes which are inherited from this object.

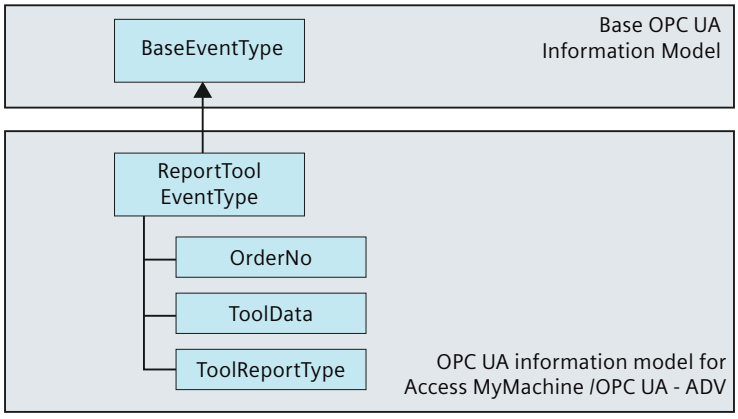


Figure A-6 OpcUa Information Model for ReportToolEventType

Additional attributes of the ReportToolEventType

Attribute	Data type	Mapping with respect to SINUMERIK	PLC source	Description
`orderNo`	OpcUa_UInt32	-	-	Order number of request. If the event is triggered by the host via method, the order number provided in the method is returned. If the event is triggered by the PLC the orderNo is 0. For tool load, tool unload, tool create and tool delete the order number is the increment number. This order number is the same for request, response and report events.
toolReport-Type	ToolReportType	-	-	Type of report For details and explanation see chapter ToolReportType Enum Value (Page 244)
toolData	OpcUa_String		-	tool data (String holding .ini file syntax). For details see following examples.

Note

Further information

You can find more detailed description of PLC interaction in chapter PLC interface information (Page 217).

Example for toolData with a configured tool

In the following example you can see the result of ReportToolEvent with a string for a configured tool: Example for tool data string (Page 265)

ToolReportType Enum value

The ToolReportType enum defines the reason why the ReportToolEvent event was triggered.

This tool report event is initiated by on the of the following sources:

- PLCReport
- PlcLoadReport
- PlcUnLoadReport
- PlcMoveReport
- LoadReport
- UnloadReport
- HostReport
- CreateReport
- DeleteReport

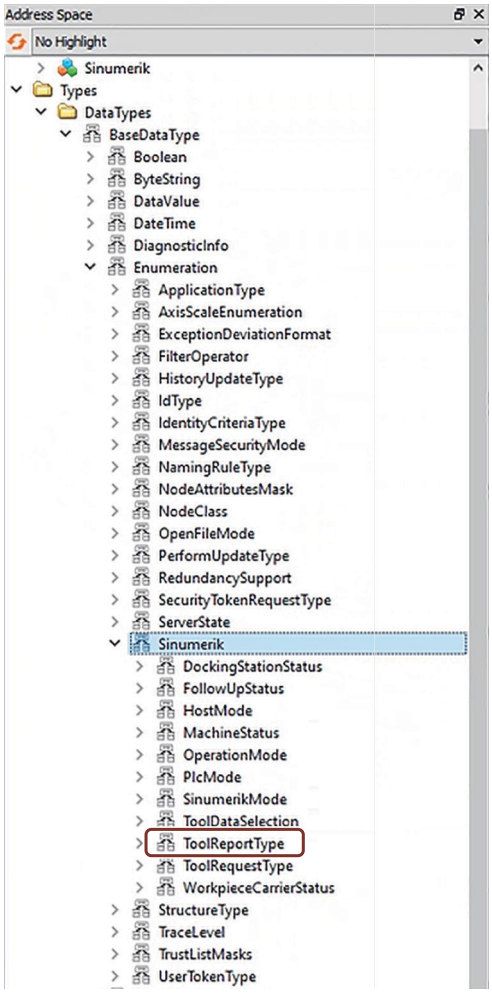


Figure A-7 ToolReportType Enum

The enumeration is defined in the following table:

Name	Value	Description
None	0	No report is given
PlcReport	1	Report by the PLC
LoadReport	2	Report that a tool has been loaded (TDI request)
UnloadReport	3	Report that a tool has been unloaded (TDI request)
HostReport	4	Host has requested
CreateReport	5	Report on tool create request
DeleteReport	6	Report on tool delete request
PlcLoadReport	7	Report by the PLC that a tool has been loaded
PlcUnloadReport	8	Report by the PLC that a tool has been unloaded
PlcMoveReport	9	Report by the PLC that the tool has been moved

Further information

You can find more detailed description of PLC interaction in chapter Symbolic DB (Page 218).

ReportToolErrorEventType

This event is raised while tool status, tool create, tool delete, tool load and tool unload failure conditions. You have to subscribe to the ToolManagement node to get notified by the ReportToolErrorEventType.

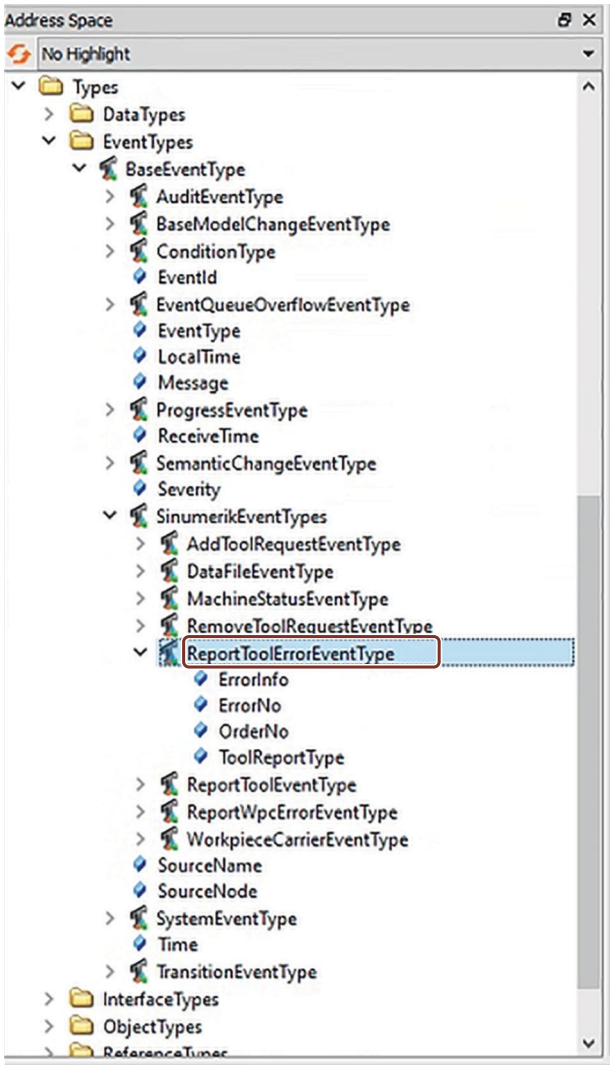


Figure A-8 ReportToolErrorEvent

Since the ReportToolErrorEvent Type is derived from the BaseEventType (Page 234), it does contain also all the attributes which are inherited from this object.

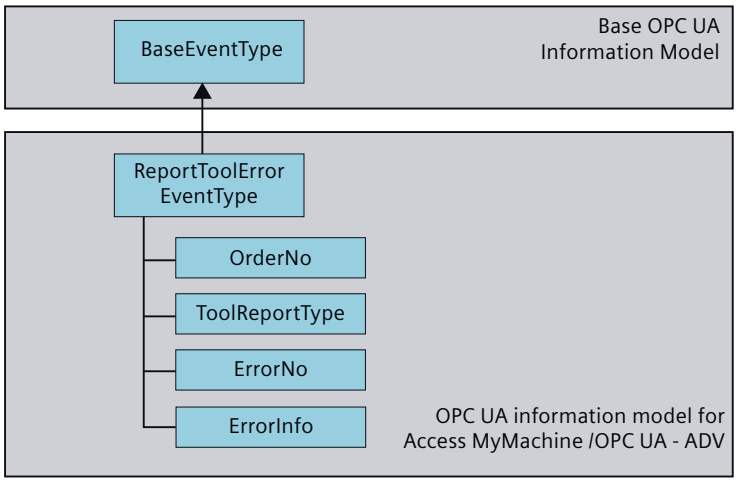


Figure A-9 OpcUa Information Model for ReportToolErrorEventType

Additional attributes of the ReportToolErrorEvent Type

Attribute	Data type	Mapping with respect to SINUMERIK	PLC source	Description
orderNo	OpcUa_UInt32	-	-	Order number of request. If the event is triggered by the host via method, the order number provided in the method is returned. If the event is triggered by the PLC the orderNo is 0. For tool load, tool unload, tool create and tool delete the order number is the increment number. This order number is the same for request, response and report events.
toolReportType	ToolReportType	-	-	Type of report For details and explanation see chapter ToolReportType Enum value (Page 244)
errorNo	OpcUa_UInt16	-	-	For description of error codes, see chapter Error numbers (Page 262)
errorInfo	OpcUa_String	-	-	

Note

Further information

You can find more detailed description of PLC interaction in chapter PLC interface information (Page 217).

DataFileEventType

This event is raised when the requested data file is created. The data file will be available in "AdvExchange" node.

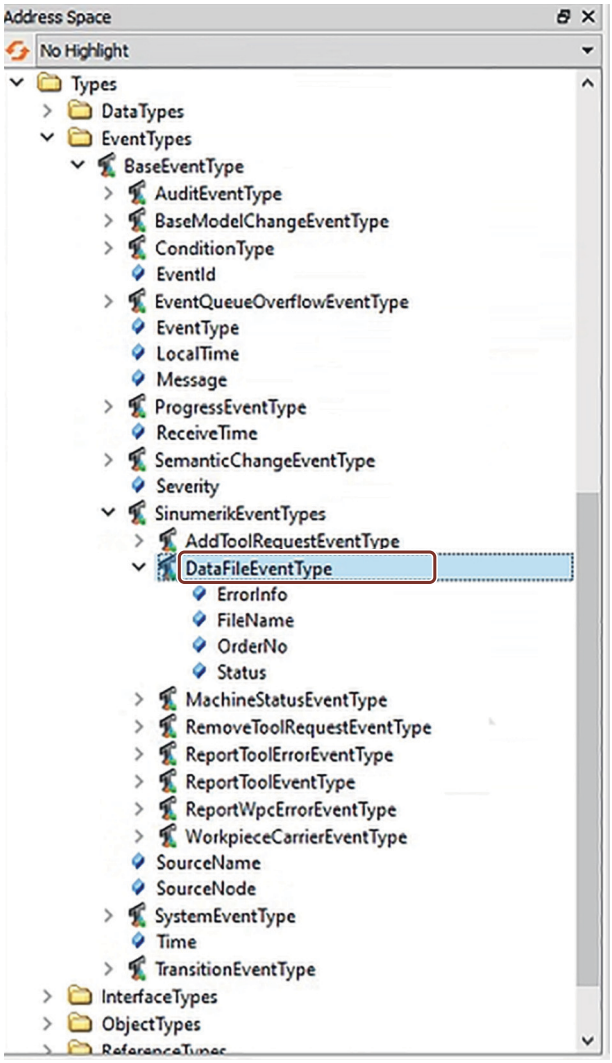


Figure A-10 DataFileEvent

Description of DataFileEvent

Since the DataFileEvent Type is derived from the BaseEventType (Page 234), it does contain also all the attributes which are inherited from this object.

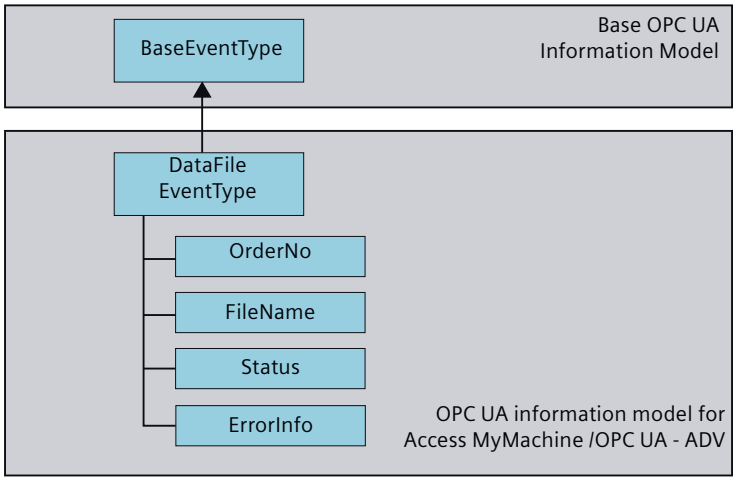


Figure A-11 OpcUa Information Model for DataFileEventType

Additional attributes of the DataFileEvent Type

Attribute	Data type	Mapping with respect to SINUMERIK	PLC source	Description
`orderNo`	OpcUa_UInt32	-	-	Order number provided when requesting for the DataFileEvent will be returned in the event triggered.
filename	OpcUa_String		-	The created filename including the path will be returned in the triggered event. The format is as follows: `<order number>_<ToolData/MagazineData>.ini` The folder "AdvExchange" will be created automatically. You can find further information in chapter File transfer with standard methods (Page 146).
status	OpcUa_StatusCode	-	-	The status of the method operation.
errorinfo	OpcUa_String	-	-	For more information, check chapter error numbers (Page 262).

Suppress machine data alarm 22070

Note

If tools are in the buffer while data backup has been started, a machine data alarm 22070 will be triggered each time. You can suppress this alarm with the following settings in SINUMERIK Operate:

- Go to the machine data **MD11410 $MN_SUPPRESS_ALARM_MASK**
- Activate bit 7 to suppress the described alarm.

AddToolRequestEventType

This event is raised when there is a notification requesting for a tool load or when the PLC triggers a create tool request. You have to subscribe to the ToolManagement node to get notified by the AddToolRequestEventType.

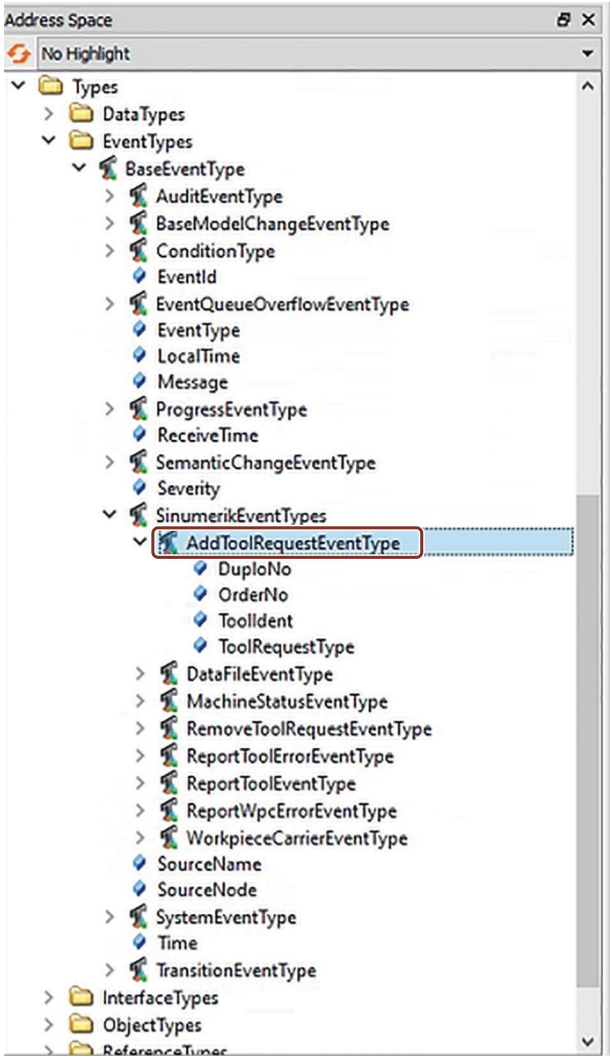


Figure A-12 AddToolRequestEvent

Description of AddToolRequestEvent

Since the AddToolRequestEvent Type is derived from the BaseEventType (Page 234), it does contain also all the attributes which are inherited from this object.

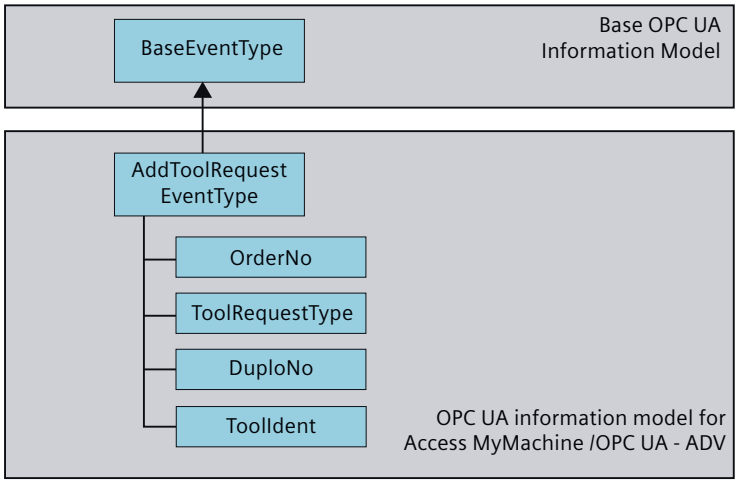


Figure A-13 OpcUa Information Model for AddToolRequestEventType

Additional attributes of the AddToolRequestEvent Type

Attribute	Data type	Mapping with respect to SINUMERIK	PLC source	Description
orderNo	OpcUa_UInt32	-	-	The order number will be generated by the OPC UA server. This number will be incremented for each tool operation.
toolRequestType	OpcUa_UInt32	-	-	For details and explanation see chapter ToolRequestType Enum value (Page 252) • 1 - LoadRequest • 2 - CreateRequest
duploNo	OpcUa_UInt32		Duplo-Number	Duplo number
toolIdent	OpcUa_String	-	ToolIdentifier	Tool ident as a string

Note

Request number will be held internal to distinguish what was the trigger.

ToolRequestType Enum value

The ToolRequestType Enum serves as an input parameter for the following events:

Event	Input parameter
AddToolRequestEventType (Page 250)	1 - LoadRequest 2 - CreateRequest
RemoveToolRequestEvent (Page 253)	3 - UnloadRequest 4 - DeleteRequest

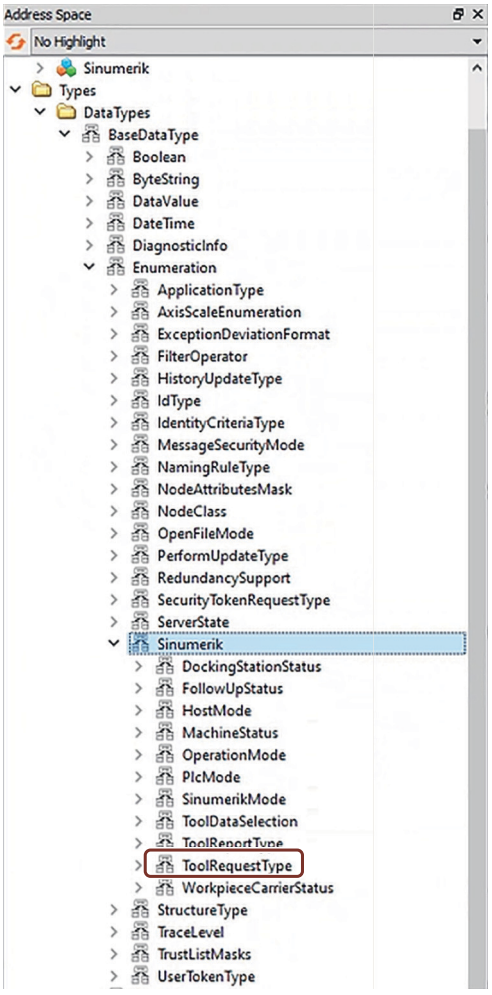


Figure A-14 ToolRequestType Enum

The enumeration is defined in the following table:

Name	Value	Description
LoadRequest	1	Request to initiate tool loading
CreateRequest	2	Request to initiate tool creating
UnloadRequest	3	Request to initiate tool unloading
DeleteRequest	4	Request to initiate tool deleting

Further information

You can find more detailed description of PLC interaction in chapter Symbolic DB (Page 218).

RemoveToolRequestEventType

This event is raised when there is a TDI notification requesting for a tool unload or when the PLC triggers a Delete request. You have to subscribe to the ToolManagement node to get notified by the RemoveToolRequestEventType.

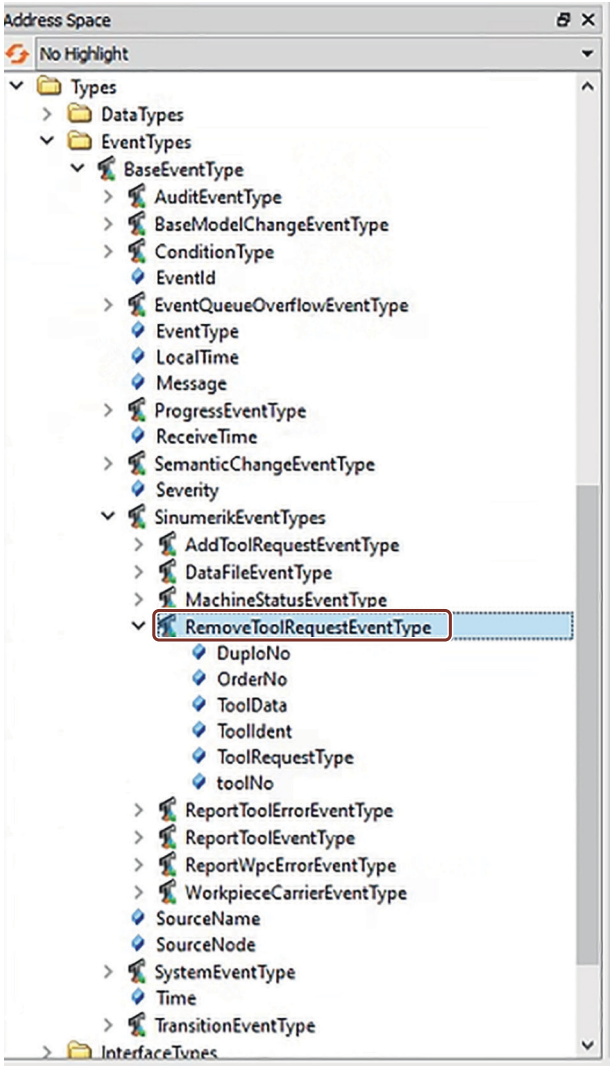


Figure A-15 RemoveToolRequestEvent

Description of RemoveToolRequestEvent

Since the RemoveToolRequestEvent Type is derived from the BaseEventType (Page 234), it does contain also all the attributes which are inherited from this object.

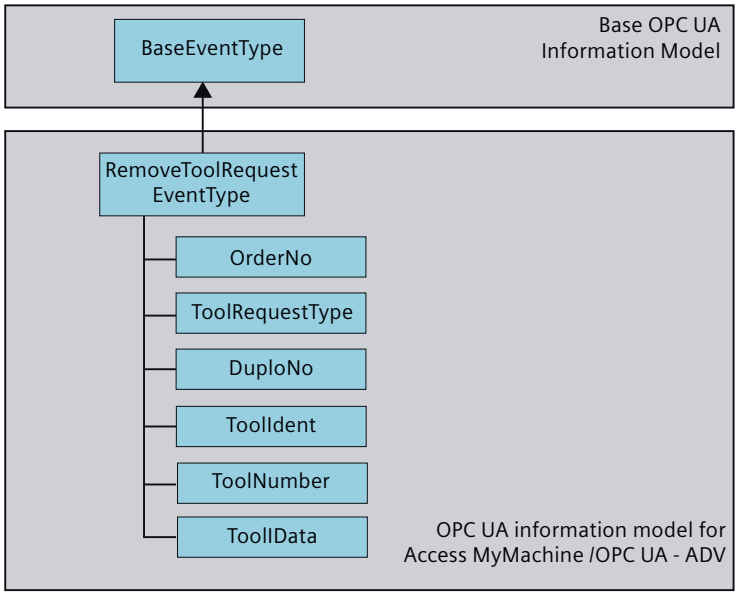


Figure A-16 OpcUa Information Model for RemoveToolRequestEventType

Additional attributes of the RemoveToolRequestEvent Type

Attribute	Data type	Mapping with respect to SINUMERIK	PLC source	Description
`orderNo`	OpcUa_UInt32	-	-	Order number provided when requesting for the DataFileEvent will be returned in the event triggered.
toolRequestType	OpcUa_UInt32	-	-	For details and explanation see chapter ToolRequestType Enum value (Page 252) • 3 - UnLoadRequest • 4 - DeleteRequest
duploNo	OpcUa_UInt32		Duplo-Number	Duplo number
toolIdent	OpcUa_String	-	ToolIdentifier	Tool ident as a string
toolNumber	OpcUa_UInt32		ToolNumber	Tool Number
toolData				tool data (String holding .ini file syntax) , in case of unload and Delete a valid tooldata will be provided.

A.2.1.6 WorkpieceCarrier

WorkpieceCarrierEventType

This event is raised when the PLC needs to report a new workpiece carrier status. You have to subscribe to the WorkpieceCarrier node to get notified by the WorkpieceCarrierEventType.

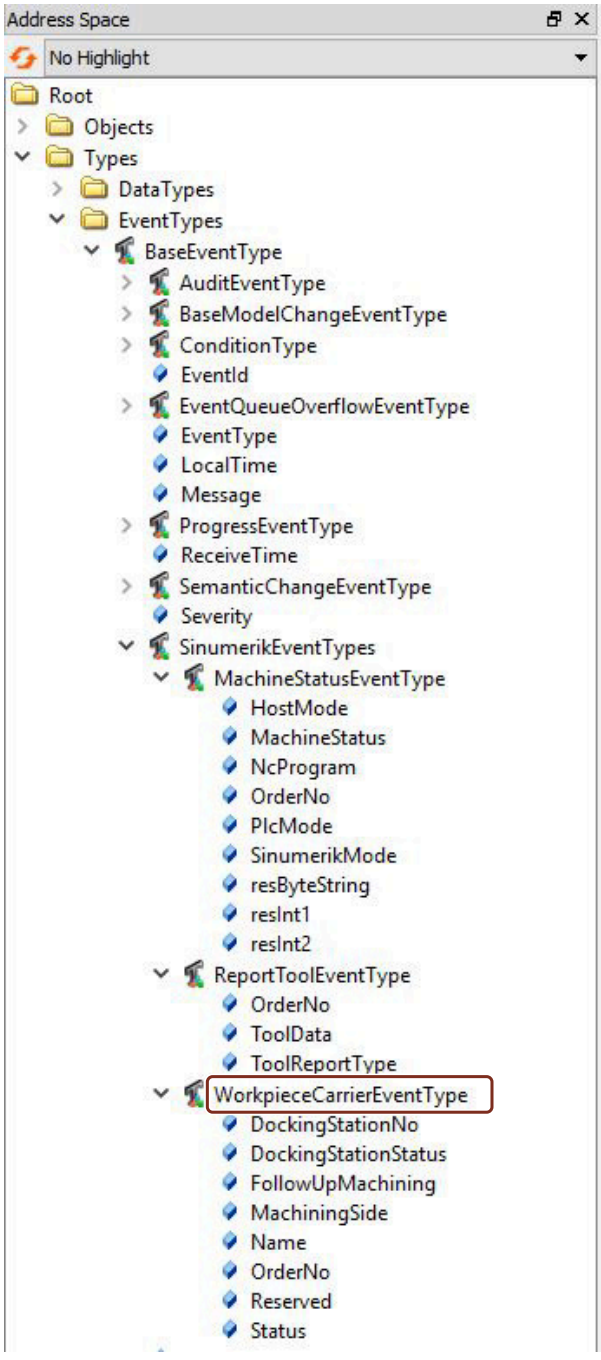


Figure A-17 WorkpieceCarrierEventType

Since the WorkpieceCarrierEventType is derived from the BaseEventType (Page 234), it does contain also all the attributes which are inherited from this object.

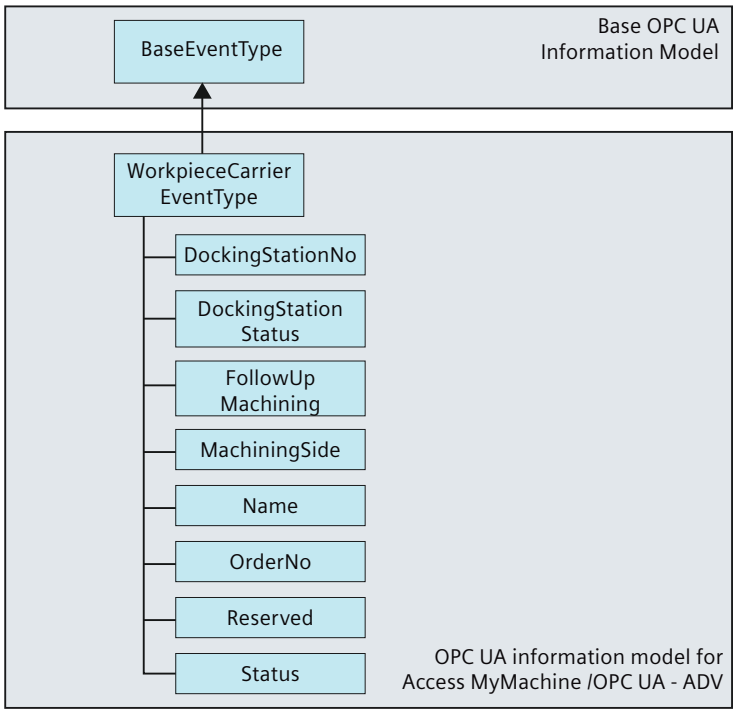


Figure A-18 OpcUa Information Model for WorkpieceCarrierEventType

Additional attributes of the WorkpieceCarrierEventType

Attribute	Data type	Mapping with respect to SINUMERIK	PLC source	Description
orderNo	OpcUa_UInt32	-	-	Order number provided by the method. If the event is triggered by the PLC the orderNo is 0.
dockingStation-No	OpcUa_UInt16	-	DockPo-sCount	Number of docking positions. (maximum 12)
dockingStation-Status	DockingStation-Status	-	DocPosSta-tus	DockingStationStatus enum value is shown here. See chapter DockingStationStatus Enum value (Page 260) 0 - Enabled 1 - Error 2 - Disabled

Attribute	Data type	Mapping with respect to SINUMERIK	PLC source	Description
status	WorkpieceCarrierStatus	-	WPCStatus	WorkpieceCarrierStatus enum value is shown here. See chapter WorkpieceCarrierStatus Enum value (Page 261) 0 - Reset 1 - NewWpc 2 - MachiningPlanned 4 - PrepareProgramSelection 8 - ProgramSelectionComplete 16 - MachiningInProgress 32 - MachiningFinished 64 - MachiningAborted 128 - BufferingOnly
name	OpcUa_String	-	WPC	Workpiece carrier name
MachiningSide	OpcUa_UInt16	-	ClampCubeSide	The host computer sets this information simultaneously with the "ProgramSelectionComplete" status (status 8). The PLC uses this value to set a corresponding side of a clamp cube for machining or passes this value on to the NC. The computer link specifies the order in which the sides of clamp cubes will be machined based on the program assignment. If the PLC is to be able to influence the machining order, an entry can be optionally made in the configuration file of the host computer that causes the program assignment data to be mirrored in a separate data block of the PLC. The PLC then has read access to this data. The host computer is notified of the machining side selected by the PLC in the "MachiningSide" field. This takes place simultaneously with the setting of the "PrepareProgramSelection" machining status (status 4). The host computer selects the program for the side specified by the PLC. The rest of the sequence remains unchanged.
followUpMachining	FollowUpStatus	-	FB	FollowupStatus enum value is shown here. See chapter FollowUpStatus Enum value (Page 262) 0 - NoFollowUp 1 - FollowUp 9 - Buffering
reserved	OpcUa_Byte	-	Reserve1	Reserved byte

Note

Further information

You can find more detailed description of PLC interaction in chapter PLC interface information (Page 217).

ReportWpcErrorEventType

This event is raised when the PLC needs to report workpiece carrier error status. You have to subscribe to the WorkpieceCarrier node to get notified by the ReportWpcErrorEventType.

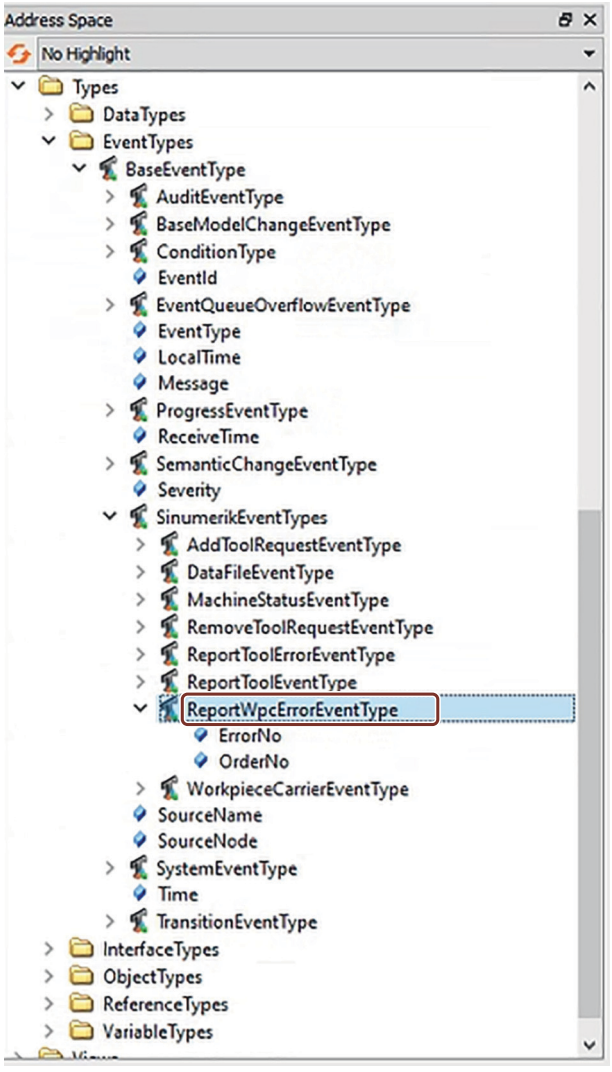


Figure A-19 ReportWpcErrorEvent

Description of the ReportWpcErrorEventType

Since the ReportWpcErrorEventType is derived from the BaseEventType (Page 234), it does contain also all the attributes which are inherited from this object.

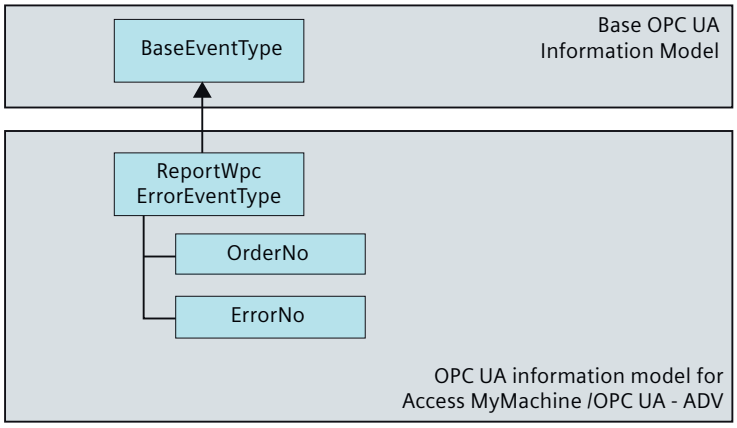


Figure A-20 OpcUa Information Model for WPCErrorEventType

Additional attributes of the ReportWpcErrorEventType

Attribute	Data type	Mapping with respect to SINUMERIK	PLC source	Description
orderNo	OpcUa_UInt32	-	-	Order number provided by the method. If the event is triggered by the PLC the orderNo == 0.
ErrorNo	OpcUa_Int16	-	-	For description of error codes, refer chapter Error numbers (Page 262).

Note

Further information

You can find more detailed description of PLC interaction in chapter PLC interface information (Page 217).

WorkpieceCarrier Enumerations

The WorkpieceCarrier Enumerations are as follows:

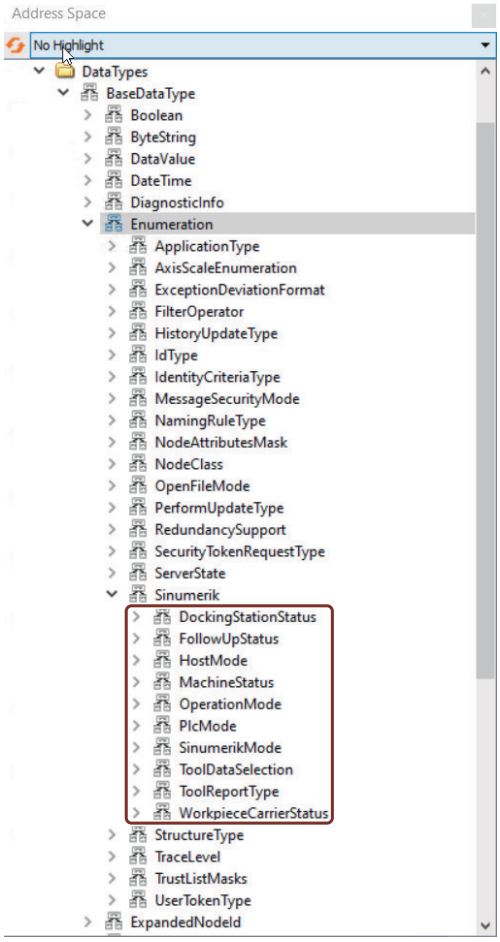


Figure A-21 WorkpieceCarrierEnum

- DockingStationStatus Enum (Page 260)
- WorkpieceCarrierStatus Enum (Page 261)
- FollowUpStatus Enum (Page 262)

DockingStationStatus Enum value

This enumeration shows which states can be assigned for a docking station by the PLC.

The enumeration is defined in the following table:

Name	Value	Description
Enabled	0	Docking station is enabled
Error	1	The "Error" bit is set or cleared based on I/O signals. The PLC does not execute any pallet transports between docking station status with "Error" status.
Disabled	2	If the docking station has the "Disabled" status, it may not be approached by the transport system.

Further information

You can find more detailed description of PLC interaction in chapter Symbolic DB (Page 218).

WorkpieceCarrierStatus Enum value

The WorkpieceCarrierStatus Enum is defined as follows:

The enumeration is defined in the following table:

Name	Value	Description
Reset	0	WorkpieceCarrierStatus can be initialized by the host computer by setting the status to zero.
NewWpc	1	The "NewWpc" status (value 1) is assigned by the PLC for a newly delivered workpiece carrier. Exception: Machining is not planned. This status causes the host computer to perform the program assignment. When the program has been assigned, the "Machining is planned" status 2 is set by the host computer.
MachiningPlanned	2	For a workpiece carrier with the status MachiningPlanned (value 2), the PLC sets the PrepareProgramSelection status as soon as the active machining has finished (value 4).
PrepareProgramSelection	4	The precondition for "PrepareProgramSelection" (value 4) is that the program assigned for this pallet by the host computer has been loaded into the NC and selected for machining.
ProgramSelectionComplete	8	When the program has been selected, the host computer sets the "ProgramSelectionComplete" status (value 8) for the associated workpiece carrier. The PLC can now initiate the NC start. However, the machine manufacturer is responsible for ensuring the PLC will only start the NC when all safety-related requirements are met, e.g. protective door closed, etc.
MachiningInProgress	16	Once machining has been started, the PLC sets the "MachiningInProgress" status (value 16). When machining has finished, the PLC then sets the "MachiningFinished" status (value 32) for the appropriate workpiece carrier. Workpiece carriers with the "MachiningFinished" status (value 32) are transported independently to an unloading location by the PLC.
MachiningFinished	32	When the "followUpMachining" identifier is set, the workpiece holder remains at the machining location. As response to the "MachiningFinished" status (value 32), the host computer sets the "MachiningPlanned" status (value 2) again. As subsequent response, the PLC uses the "PrepareProgramSelection" status (value 4) to prompt the host computer to select the program for the next machining steps. The remaining sequence of operations is the same as for initial machining.

Name	Value	Description
MachiningAborted	64	The "MachiningAborted" status (value 64) is set when a workpiece holder will no longer be machined after a malfunction. This identifier might particularly occur in unmanned production. A workpiece holder with this identifier may not be brought to another machine for a further operation. It may only be brought to a storage location.
BufferingOnly	128	A workpiece carrier that was delivered for buffering by the TPS is assigned the "BufferingOnly" status (value 128) and not the "NewWpc" status (status 1). The TPS transfers this information to the PLC. For workpiece carriers with this status, there is no program assignment by the host computer.

Further information

You can find more detailed description of PLC interaction in chapter Symbolic DB (Page 218).

FollowUpStatus Enum value

The FollowUpStatus Enum is defined as follows:.

The enumeration is defined in the following table:

Name	Value	Description
NoFollowUp	0	No follow-up machining is planned.
FollowUp	1	Follow-up machining is planned. If the follow-up operation flag is set for a side, the workpiece carrier remains at the machining station after the end of the NC program.
Buffering	9	Only for buffering

Further information

You can find more detailed description of PLC interaction in chapter Symbolic DB (Page 218).

A.2.1.7 Error numbers

Overview

In the following table you can find the valid error codes of OPC UA Advanced and their corresponding description.

Note

TDI specific errors

If the error code is not part of this OPC UA Advanced error code table and is of load / unload functionality, refer to Function Manual SINUMERIK ONE Tool Management (https://support.industry.siemens.com/cs/document/109777604/sinumerik-one-tool-management?dti=0&lc=en-AT), chapter "Code carrier connection - Tool Ident Connection (option), section "Error codes".

Error code / ErrorNo	Description / ErrorInfo	Use cases applicable
900	Access denied to write docking position.	
901	Set Workpiece Status has failed.	
902	Invalid WPC name.	
903	Invalid docking station number.	
904	PLC DB not configured.	
905	File <XX_TOOLDATA.INI> is created and oldest file <XX_TOOLDATA.INI> is deleted as file share count has reached max limit of 20 files.	
911	Invalid User Input	Create/Delete tool
912	Importing to INI file failed.	Create/Delete tool
913	Selected tool not found.	Create/Delete tool
914	Delete tool failed due to internal error	Create/Delete tool
916	Invalid order number.	Create/Delete tool
920	Create tool failed due to internal error.	Create tool
922	Create tool failed with timeout.	Create/Delete tool
924	Delete tool failed with timeout.	Delete tool
932	Tool Area is not ready.	All
933	Invalid Tool Area.	All
934	Tool operation already In progress.	All
940	Reading of tool data failed.	Create/Delete tool
941	Invalid Tool Ident or Duplo number.	Create/Delete tool
942	Invalid Magazine Location or Tool number.	Create/Delete tool
943	Invalid tool ident number	Create/Delete tool
950	Insufficient memory.	All
951	Internal error. For additional information related to tool load / unload refer to Function Manual SINUMERIK ONE Tool Management (https://support.industry.siemens.com/cs/document/109777604/sinumerik-one-tool-management?dti=0&lc=en-AT) with the TDI error code.	All
952	Plc Variable could not be read or written.	All
953	Host response timeout.	Load/Unload Tool
954	Request rejected by host computer.	Load/Unload Tool
1 to 5	Internal error	Create/Delete tool
-1 to -9	Internal error	Create/Delete tool
-10	Magazine is disabled and time is up	Create/Delete tool
-11	Loading point is not free	Create/Delete tool
-12	Function may not work with a given parameter	Create/Delete tool
-13	Device is not existing in SITm datastore	Create/Delete tool
-14	An error occurred by accessing SITm datastore	Create/Delete tool
-15	Destination place doesn't fit or doesn't exist	Create/Delete tool
-16	No loading point in container or none connected to magazine	Create/Delete tool
-17	Please check actual place	Create/Delete tool
-18	Tool is not available	Create/Delete tool

Error code / ErrorNo	Description / ErrorInfo	Use cases applicable
-19	Magazine is not available	Create/Delete tool
-20	Last function not ready	Create/Delete tool
-21	There's a problem with refreshing changed data	Create/Delete tool
-22	Container is not active (a power on reset may be in progress)	Create/Delete tool
-23	There's an error in configuration of device	Create/Delete tool
-24	A power on reset has started	Create/Delete tool
-25	Internal Error	Create/Delete tool
-26	Unspecified error in SlTmService	Create/Delete tool
-27	Device not ready for working	Create/Delete tool
-28	There's a warning concerning configuration of device	Create/Delete tool
-29	Multitool to be deleted is not empty but deleteIfNotEmpty is false	Create/Delete tool
-30	Specified tool is not in multitool place	Create/Delete tool
-31	Multitool is not available	Create/Delete tool
-32	D-numbers are not unique	Create/Delete tool
-33	Wear group is not existing	Create/Delete tool
-34	Edge respectively d-number is not existing	Create/Delete tool
-35	There's no active wear group (in magazine)	Create/Delete tool
-40	No more multitools may be created or maximal number of multitools is reached	Create/Delete tool
-41	No more multitool locations may be defined or maximal number of multitool locations is reached	Create/Delete tool
-42	Tool with ident is not a tool inside multitool, name is no ident of a tool	Create/Delete tool
-43	Duplonumber doesn't match with a toolident inside multitool, duplonumber doesn't match with toolident	Create/Delete tool
-44	Number is not number of a toolholder toolholder doesn't exist	Create/Delete tool
-45	Miscellaneous error of Pis function ends with an error	Create/Delete tool
-46	Option not activated	Create/Delete tool
-47	Toolmanagement is not active (in container)	Create/Delete tool
-49	Tool doesn't exist (wrong toolnumber)	Create/Delete tool
-50	Invalid argument in PI call	Create/Delete tool
-51	Maximal number of tools is reached no more tools may be created	Create/Delete tool
-52	Maximal number of edges is reached, no more edges may be created	Create/Delete tool
-53	Magazine not existing	Create/Delete tool
-54	There's already a tool with toolident/ duplo	Create/Delete tool
-55	Specified monitoring not active for tool	Create/Delete tool
-57	There's no active wear compound in specified magazine	Create/Delete tool
-58	Ident is no multitool ident, multitool ident is already existing	Create/Delete tool
-59	mTno is no multitool number mTno is not valid or already existing	Create/Delete tool
-60	Too many multitool locations or multitool location number doesn't match with multitool locations	Create/Delete tool
-61	Multitool is loaded and may not be deleted	Create/Delete tool
-62	There's an active tool inside multitool	Create/Delete tool
-63	Multitool contains a tool with an active compensation, it may not be deleted	Create/Delete tool
-65	Previous move is already active	Create/Delete tool

Error code / ErrorNo	Description / ErrorInfo	Use cases applicable
-66	Magazine location not existing	Create/Delete tool
-67	Tool is loaded and may not be deleted	Create/Delete tool
-68	Magazine is locked or not free for (un)loading	Create/Delete tool
-69	TO-unit not existing or not assigned to a channel	Create/Delete tool
-70	Internal Error	Create/Delete tool
-71	Internal Error	Create/Delete tool
-72	Wrong internal magazine number	Create/Delete tool
-73	Wrong number of magazin location	Create/Delete tool
-81	Access level not sufficient	Create/Delete tool
-82	Checked value out of admissible range	Create/Delete tool
-83	Actual value out of admissible range	Create/Delete tool
-84	Adapter number already exists or adapter number not valid	Create/Delete tool
-85	Function adapter is not active	Create/Delete tool
-86	Adapter is still in use	Create/Delete tool
-87	Maximum count of adapters reached	Create/Delete tool

A.2.1.8 Example for tool data string

The following example shows an example for a tool data string (text is selectable):

```
$TC_TP1[11]=1
$TC_TP2[11]="E45KT_D50_45GR"
$TC_TP3[11]=1
$TC_TP4[11]=1
$TC_TP5[11]=1
$TC_TP6[11]=1
$TC_TP7[11]=1
$TC_TP8[11]=131
$TC_TP9[11]=0
$TC_TP10[11]=0
$TC_TP11[11]=0
;$A_TOOLMN[11]=1
;$A_TOOLMLN[11]=5
;$P_TOOLND[11]=1
;$A_MYMN[11]=1
;$A_MYMLN[11]=5
$TC_TP_PROTA[11]=""
$TC_TP_MAX_VELO[11]=0.000
$TC_TP_MAX_ACC[11]=0.000
;$A_MYMTN[11]=0
;$A_MYMTLN[11]=0
;$A_TOOLMTADAPT[11]=0
$TC_DP1[11,1]=140
$TC_DP2[11,1]=9.000
$TC_DP3[11,1]=91.706
$TC_DP4[11,1]=0.000
$TC_DP5[11,1]=0.000
```

$TC_DP6[11,1]=25.000
$TC_DP7[11,1]=30.000
$TC_DP8[11,1]=0.000
$TC_DP9[11,1]=0.000
$TC_DP10[11,1]=0.000
$TC_DP11[11,1]=45.000
$TC_DP12[11,1]=0.000
$TC_DP13[11,1]=0.000
$TC_DP14[11,1]=0.000
$TC_DP15[11,1]=0.000
$TC_DP16[11,1]=0.000
$TC_DP17[11,1]=0.000
$TC_DP18[11,1]=0.000
$TC_DP19[11,1]=0.000
$TC_DP20[11,1]=0.000
$TC_DP21[11,1]=0.000
$TC_DP22[11,1]=0.000
$TC_DP23[11,1]=0.000
$TC_DP24[11,1]=0.000
$TC_DP25[11,1]=256.000
$TC_DPH[11,1]=0
$TC_DPV[11,1]=0
$TC_DPV3[11,1]=0.000
$TC_DPV4[11,1]=0.000
$TC_DPV5[11,1]=0.000
$TC_DPVN3[11,1]=0.000
$TC_DPVN4[11,1]=0.000
$TC_DPVN5[11,1]=0.000
$TC_DPNT[11,1]=4
$TC_DPROT[11,1]=0.000
$TC_MOP1[11,1]=0.000
$TC_MOP2[11,1]=0.000
$TC_MOP3[11,1]=0.000
$TC_MOP4[11,1]=0.000
$TC_MOP5[11,1]=0.000
$TC_MOP6[11,1]=0.000
$TC_MOP11[11,1]=0.000
$TC_MOP13[11,1]=0.000
$TC_MOP15[11,1]=0.000
$TC_MPP6[1,5]=11

A.2.2 OPC UA Advanced methods and variables

A.2.2.1 Overview

Note

Target group

The following descriptions are intended for the target group of system integrators. If you are more interested in PLC programming and communication interaction with OPC UA Advanced, you can find further information in chapter PLC interface information (Page 217).

When using the OPC UA Advanced functionalities it is either possible to subscribe to certain EventTypes or you can call a method to request a status event directly. In the further course of the chapter all methods are explained in detail with their parameters and results.

A.2.2.2 Machine

RequestMachineStatus method

The RequestMachineStatus method is provided under "Sinumerik > Adv > Machine" in the address space, which allows the explicit request of a machine status event. You can call this method by providing an order number stated by the host.

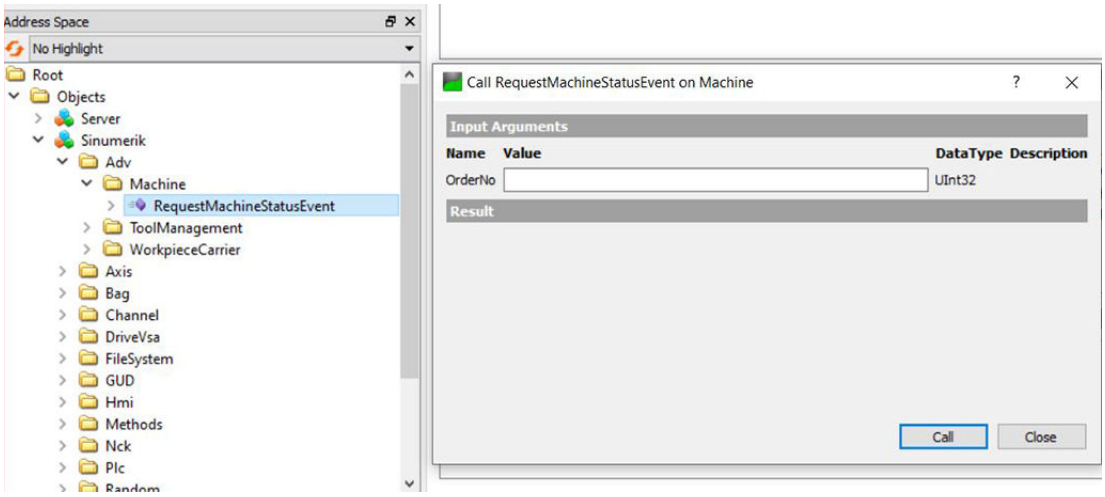


Figure A-22 Calling RequestMachineStatusEvent method

Prerequisite

To explicitly request the machine status, the OPC UA administrator has to provide the user with the access right "ADV_Read".

Structure and arguments

Signature of the method RequestMachineStatus is as follows:

```
OpcUa_StatusCode RequestMachineStatus(OpcUa_UInt32 orderNo)
```

Argument	Description
orderNo	Order no. stated by host request will be reported in event

As an output argument, the OPC UA Status code will be provided.

After successful request of the method, the MachineStatusEvent will be provided.

OPC UA Status code

The following table gives details on values and description of the OPC UA method call status:

Result	Description
Succeeded	Method is executed with success/failure.
OpcUa_BadInvalidArgument	Invalid inputs are provided.
OpcUa_BadUserAccessDenied	User does not have permission to invoke the method.

SelectedPartProgram method

The SelectedPartProgram method is provided under "Sinumerik > Adv > Machine" in the address space, which allows to read a selected part program per channel. You can call this method by providing the channel number.

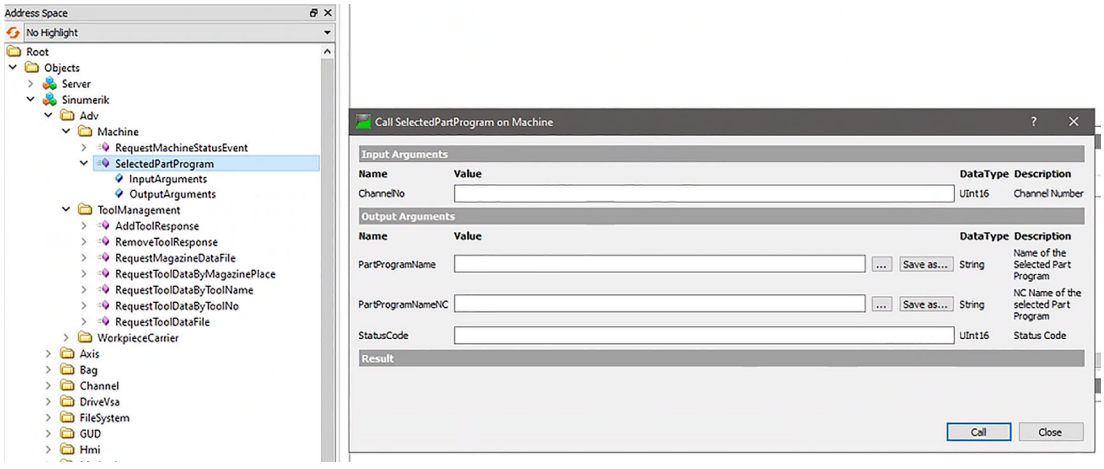


Figure A-23 Calling SelectedPartProgram method

Prerequisite

To read a selected part program per channel, the OPC UA administrator has to provide the user with the access right "ADV_Read".

Structure and arguments

Signature of the method SelectedPartProgram is as follows:

```
OpcUa_StatusCode SelectedPartProgram(OpcUa_UInt16 channelNo,
OpcUa_String partProgramName, OpcUa_String PartProgramNameNC,
OpcUa_UInt16 statusCode)
```

Argument	Description
channelNo [in]	Number of the channel
partProgramName [out]	Path and name of selected partprogram, e. g.: Sinumerik/FileSystem/Part Program/HELIX.MPF
partProgramNameNC [out]	Path and name of the selected partprogram NC-style, e. g: /_N_MPF_DIR/_N_HELIX_MPF

As an additional output argument, the OPC UA Status code will be provided.

OPC UA Status code

The following table gives details on values and description of the OPC UA method call status:

Result	Description
0	good
1	Channel does not exist
2	Part program cannot be found

Examples for OPC UA status codes

0 = Good

```
[in]  channelNo -          e.g.: 1-10
[out] partProgramName-     e.g.: Sinumerik/FileSystem/Part Program/
HELIX.MPF
[out] partProgramNameNC-   e.g.: /_N_MPF_DIR/_N_HELIX_MPF
[out] statusCode -         0
```

1 = No part program selected in channel

```
[in]  channelNo -         e.g.: 1-10
[out] partProgramName-    empty
[out] partProgramNameNC-  /_N_MPF0
[out] statusCode -         1
```

2 = Channel not available

```
[out] partProgramNameNC-  empty
[out] partProgramNodeId-  empty
[out] statusCode -         2
```

PlcClock variable

The PlcClock variable is provided under "Sinumerik > Adv > Machine" in the address space, which allows the reading and writing of the PLC clock.

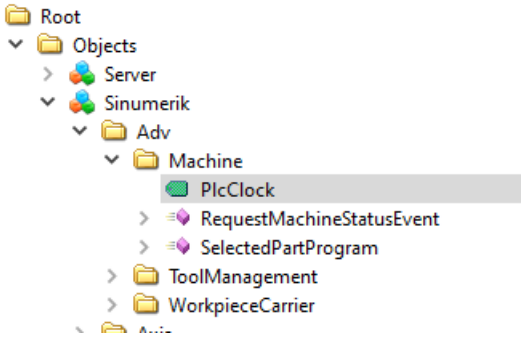


Figure A-24 Browse tree: variable PlcClock

Prerequisite

To read the PLC clock, the OPC UA administrator has to provide the user with the access right "ADV_Read".

To write the PLC clock the OPC UA administrator has to provide the user with the access right "ADV_Write".

Acess of variable

For a general explanation of PLC variable access refer to chapter: Variable paths for PLC access operations (Page 118)

Read and write PlcClock	Description
OPC UA Read („/Plc/clock“)	Variable has the value of the current PLC clock in the following format: YYYY-MM-DDThh:mm:ss.nnnZ Eg.: 2022-10-28T16:02:33.914Z
OPC UA Write („/Plc/clock“, 2021-03-15T08:10:55)	To set the variable to a new date and time, specify the value in the following format: YYYY-MM-DDThh:mm:ss.nnnZ Eg.: 2021-03-15T08:10:55

A.2.2.3 RegisterAdvHost

The RegisterAdvHost method is provided under "Sinumerik > Adv" in the address space, which allows the actual user to register as the host computer. The function has to be called by the OPC UA user that acts as the host (only 1 allowed). Whenever this user is connected, the PLC gets the information that a connection to the host is established.

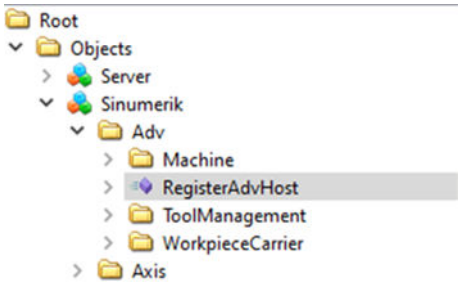


Figure A-25 Calling RegisterAdvHost_method

Prerequisites

- To change the online/offline detection bit the the OPC UA administrator has to provide the user with the access right "ADV_Write".

Structure

Signature of the method RegisterAdvHost is as follows:
`OpcUa_StatusCode RegisterAdvHost()`

The method is called without arguments/parameters.

After successful request of the method, the defined user in the "advhostsettings.ini" file (Page 183)is saved and registered. The online/offline bits are automatically set/reset each time you log in/out.

- If you want to register another user, you must run the RegisterAdvHost method again with this new user.
- If you want to delete the user, then you have to delete the entry in the ini file.

A.2.2.4 ToolManagement

RequestToolDataByMagazinePlace method

The RequestToolDataByMagazinePlace method is provided under "Sinumerik > Adv > ToolManagement" in the address space, which allows to provide tool data for the tool located in the specified magazine place.

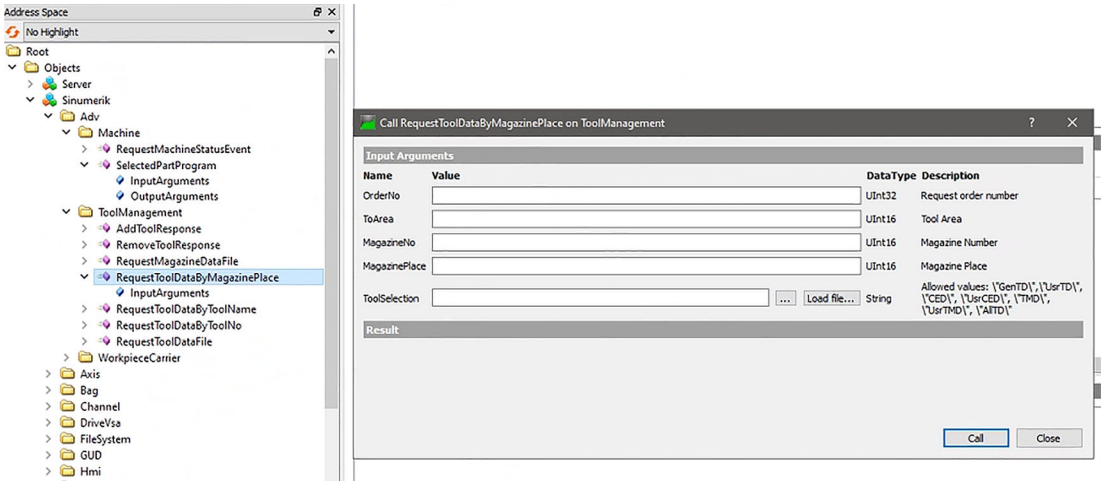


Figure A-26 Calling RequestToolDataByMagazinePlace method

Prerequisite

To provide tool data for the tool located in specified magazine place, the OPC UA administrator has to provide the user with the access right "ADV_Read".

Structure and arguments

Signature of the method RequestToolDataByMagazinePlace is as follows:

```
OpcUa_StatusCode RequestToolDataByMagazinePlace(OpcUa_UInt32
orderNo, OpcUa_UInt16 toArea, OpcUa_UInt16 magazineNo, OpcUa_UInt16
magazinePlace, OpcUa_string toolSelection)
```

Argument	Description
orderNo	Order no. stated by host request will be reported in fired event
toArea	Tool area: tool areas are assigned to NC channels toAreaNo == 0, reports data of all tool areas
magazineNo	Magazine number
magazinePlace	Magazine place number

Argument	Description	
toolSelection	Possible realm strings for toolSelection. You can choose one or multiple realms.	
	Description	**Realm name**
	General tool data	GenTD
	User tool data	UsrTD
	Cutting edge data	CED
	User cutting edge data	UsrCED
	Tool monitor data	TMD
	User Tool monitor data	UsrTMD
	All Tool Data	AllTD

As an output argument, the OpcUaStatusCode will be provided.

After successful request of the method a ReportToolEvent (Page 242) will be sent to the host.

In case of unsuccessful load process, a ReportToolErrorEventType (Page 246) will be send to the host.

OpcUaStatusCode

The following table gives details on values and description of the OPC UA method call status:

Result	Description
Succeeded	Method is executed with success/failure.
OpcUa_BadInvalidArgument	Invalid inputs are provided.
OpcUa_BadUserAccessDenied	User does not have permission to invoke the method.

RequestToolDataByToolName method

The RequestToolDataByToolName method is provided under "Sinumerik > Adv > ToolManagement" in the address space, which allows to provide tool data for the tool specified in tool ident and duplo number.

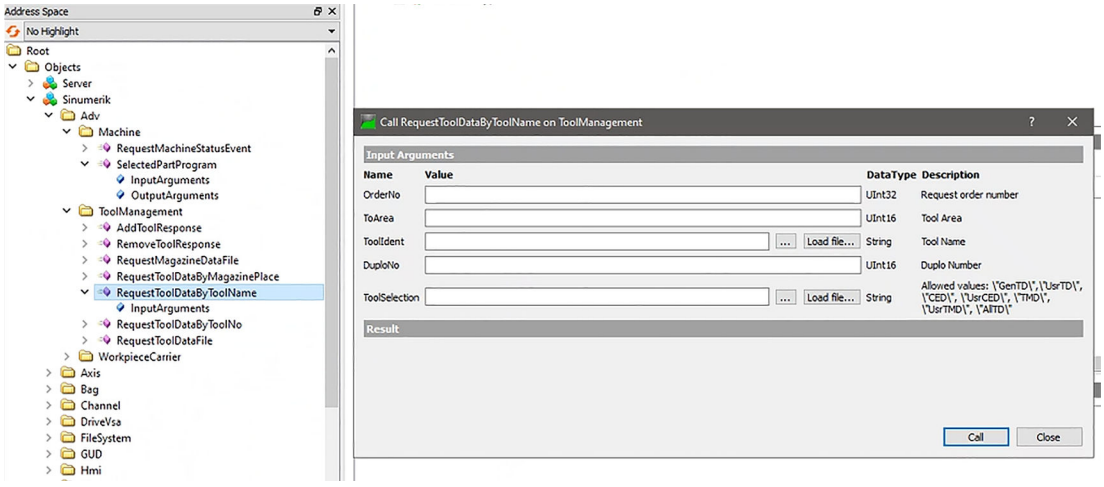


Figure A-27 Calling RequestToolDataByToolName method

Prerequisite

To provide tool data for the tool specified in tool ident and duplo number, the OPC UA administrator has to provide the user with the access right "ADV_Read".

Structure and arguments

Signature of the method RequestToolDataByToolName is as follows:

```
OpcUa_StatusCode RequestToolDataByToolName(OpcUa_UInt32 orderNo,
OpcUa_UInt16 toArea, OpcUa_String toolIdent, OpcUa_UInt16 duploNo,
OpcUa_string toolSelection)
```

Argument	Description	
orderNo	Order no. stated by host request will be reported in fired event	
toArea	Tool area: tool areas are assigned to NC channels	
toolIdent	Tool name	
duploNo	Duplo number	
toolSelection	Possible realm strings for toolSelection. You can choose one or multiple realms.	
	Description	**Realm name**
	General tool data	GenTD
	User tool data	UsrTD
	Cutting edge data	CED
	User cutting edge data	UsrCED
	Tool monitor data	TMD
	User tool monitor data	UsrTMD
	All tool data	AllTD

As an output argument, the OpcUaStatusCode will be provided.

After successful request of the method a ReportToolEvent (Page 242) will be sent to the host.

In case of unsuccessful load process, a ReportToolErrorEventType (Page 246) will be send to the host.

OpcUaStatusCode

The following table gives details on values and description of the OPC UA method call status:

Result	Description
Succeeded	Method is executed with success/failure.
OpcUa_BadInvalidArgument	Invalid inputs are provided.
OpcUa_BadUserAccessDenied	User does not have permission to invoke the method.

RequestToolDataByToolNo method

The RequestToolDataByToolNo method is provided under "Sinumerik > Adv > ToolManagement" in the address space, which allows to provide tool data for the tool specified in tool number.

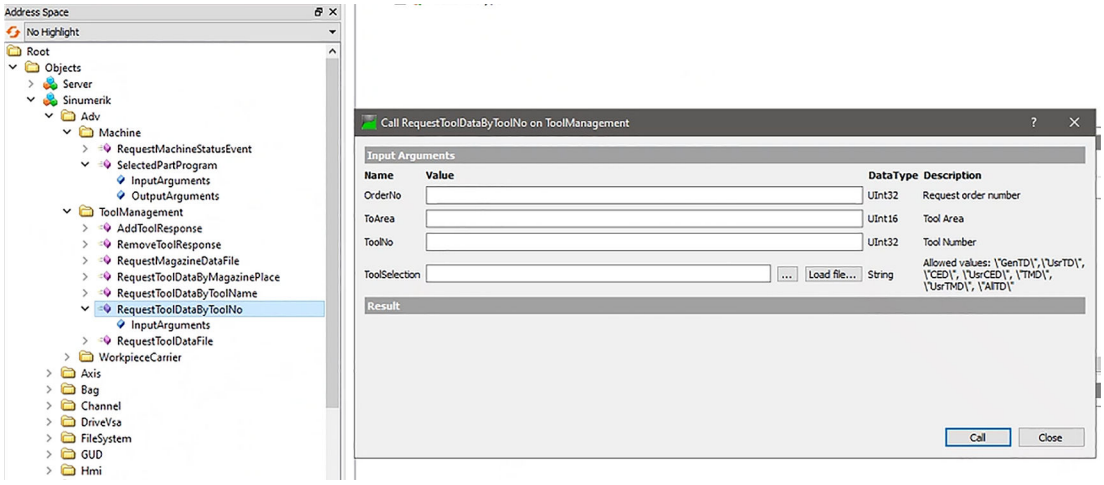


Figure A-28 Calling RequestToolDataByToolNo method

Prerequisite

To provide tool data for the tool specified in tool number, the OPC UA administrator has to provide the user with the access right "ADV_Read".

Structure and arguments

Signature of the method RequestToolDataByToolNo is as follows:

```
OpcUa_StatusCode RequestToolDataByToolNo(OpcUa_UInt32 orderNo,
OpcUa_UInt16 toArea, OpcUa_UInt32 toolNo, OpcUa_string
toolSelection)
```

Argument	Description	
orderNo	Order no. stated by host request will be reported in fired event	
toArea	Tool area: tool areas are assigned to NC channels	
toolNo	Tool number	
toolSelection	Possible realm strings for toolSelection. You can choose one or multiple realms.	
	Description	**Realm name**
	General tool data	GenTD
	User tool data	UsrTD
	Cutting edge data	CED
	User cutting edge data	UsrCED
	Tool monitor data	TMD
	User tool monitor data	UsrTMD
	All tool data	AllTD

As an output argument, the OpcUaStatusCode will be provided.

After successful request of the method a ReportToolEvent (Page 242) will be sent to the host.

In case of unsuccessful load process a ReportToolErrorEventType (Page 246) will be send to the host.

OpcUaStatusCode

The following table gives details on values and description of the OPC UA method call status:

Result	Description
Succeeded	Method is executed with success/failure.
OpcUa_BadInvalidArgument	Invalid inputs are provided.
OpcUa_BadUserAccessDenied	User does not have permission to invoke the method.

RequestMagazineDataFile method

The RequestMagazineDataFile method is provided under "Sinumerik > Adv > ToolManagement" in the address space, which allows to provide magazine data for all magazines.

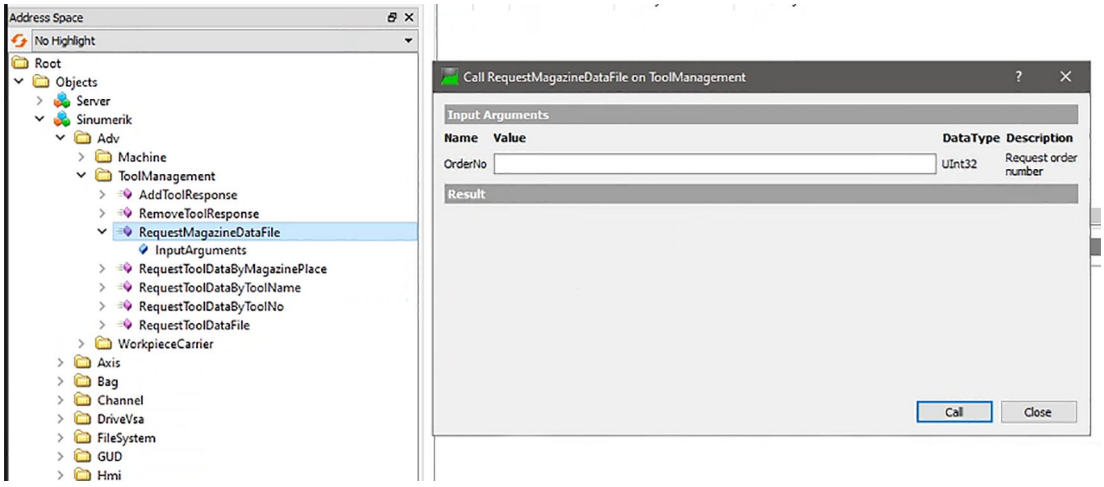


Figure A-29 Calling RequestMagazineDataFile method

Prerequisite

To provide magazine data for the specified magazine, the OPC UA administrator has to provide the user with the access right "ADV_Read".

Structure and arguments

Signature of the method RequestMagazineDataFile is as follows:

```
OpcUa_StatusCode RequestMagazineDataFile(OpcUa_UInt32 orderNo)
```

Argument	Description
orderNo	Order no. stated by host request will be reported in fired event

As an output argument, the OpcUaStatusCode will be provided.

After successful request of the method a DataFileEventType (Page 248) will be sent to the host, containing the order number. Subsequently the file is accessible via file methods under the "AdvExchange" folder.

OpcUaStatusCode

The following table gives details on values and description of the OPC UA method call status:

Result	Description
Succeeded	Method is executed with success.
OpcUa_BadInvalidArgument	Invalid inputs are provided.
OpcUa_BadUserAccessDenied	User does not have permission to invoke the method.

RequestToolDataFile method

The RequestToolDataFile method is provided under "Sinumerik > Adv > ToolManagement" in the address space, which allows to provide complete tool data.

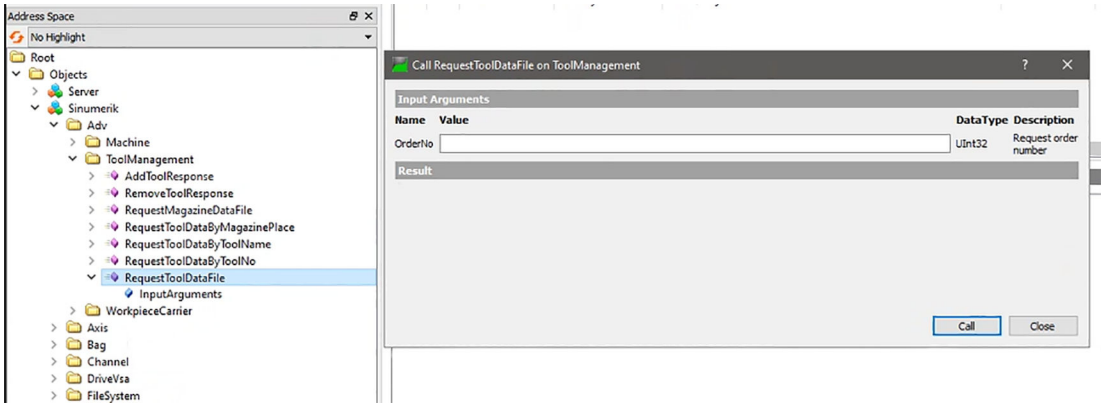


Figure A-30 Calling RequestToolDataFile method

Prerequisite

To provide complete tool data, the OPC UA administrator has to provide the user with the access right "ADV_Read".

Structure and arguments

Signature of the method RequestToolDataFile is as follows:

```
OpcUa_StatusCode RequestToolDataFile(OpcUa_UInt32 orderNo)
```

Argument	Description
orderNo	Order no. stated by host request will be reported in fired event

As an output argument, the OpcUaStatusCode will be provided.

After successful/unsuccsessful request of the method a DataFileEventType (Page 248) will be sent to the host, containing the order number. Subsequently the file is accessible via file methods under the "AdvExchange" folder.

OpcUaStatusCode

The following table gives details on values and description of the OPC UA method call status:

Result	Description
Succeeded	Method is executed with success/failure.
OpcUa_BadInvalidArgument	Invalid inputs are provided.
OpcUa_BadUserAccessDenied	User does not have permission to invoke the method.

AddToolResponse method

The AddToolResponse method is provided under "Sinumerik > Adv > ToolManagement" in the address space, which allows to provide tool data for tool loading or tool creation. This method is called by the host on the event AddToolRequestEventType (Page 250) providing an order number and tool data. If the user does not want load or create the tool, then he has to set the error number and the error info. The error number can be from 0 to 255.

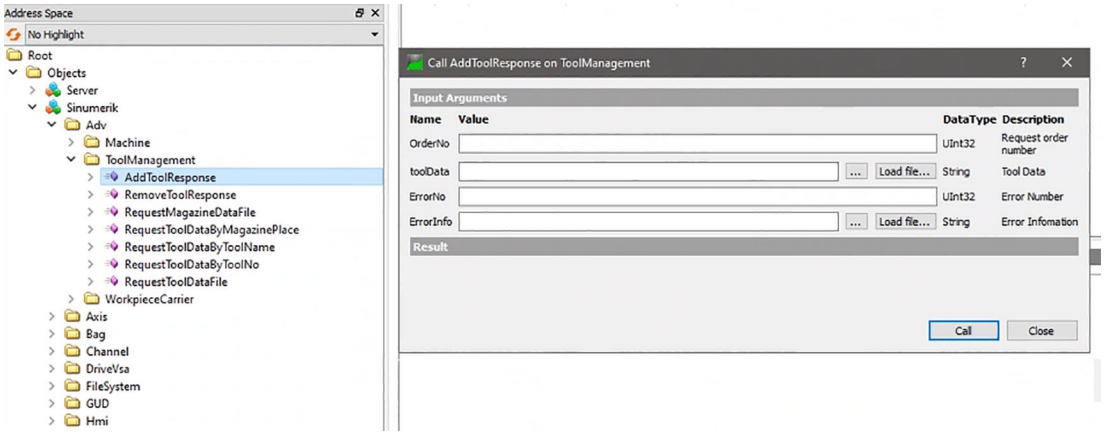


Figure A-31 Calling AddToolResponse method

Prerequisite

To initiate a tool response for loading or creating a tool, the OPC UA administrator has to provide the user with the access right "ADV_Write".

Structure and arguments

Signature of the method AddToolResponse is as follows:

```
OpcUa_StatusCode AddToolResponse(OpcUa_UInt32 orderNo, OpcUa_string
toolData, OpcUa_UInt32 errorNo, OpcUa_string errorInfo)
```

Argument	Description
orderNo	Order number, as received in the AddToolRequestEventType event.
toolData	Tool data (String holding .ini file syntax). For details see example below.
errorNo	If you do not want load or create the tool, then the error number (Page 262)has to be provided. If errorNo is set to 0, the tool will be loaded or created.
errorInfo	If you do not want load or create the tool, then the error info (Page 262)has to be provided. If errorInfo is empty, the tool will be loaded or created.

As an output argument, the OpcUaStatusCode will be provided. And the OPC UA server initiates the tool loading or creating process and writes tool data to SINUMERIK Tool Management.

After successful tool load or tool creation a ReportToolEvent (Page 242) will be sent to the host. In case of unsuccessful load or creation process a ReportToolErrorEvent (Page 246) will be sent.

OpcUaStatusCode

The following table gives details on values and description of the OPC UA method call status:

Result	Description
Succeeded	Method is executed with success.
OpcUa_BadInvalidArgument	Invalid inputs are provided.
OpcUa_BadUserAccessDenied	User does not have permission to invoke the method.

Example for toolData

Example for tool data string (Page 265)

RemoveToolResponse method

The RemoveToolResponse method is provided under "Sinumerik > Adv > ToolManagement" in the address space, which allows to unload or delete a tool. This method is called by the host on the event RemoveToolRequestEvent (Page 253) providing an order number. If the user does not want unload or delete the tool, then he has to set the error number and the error info.

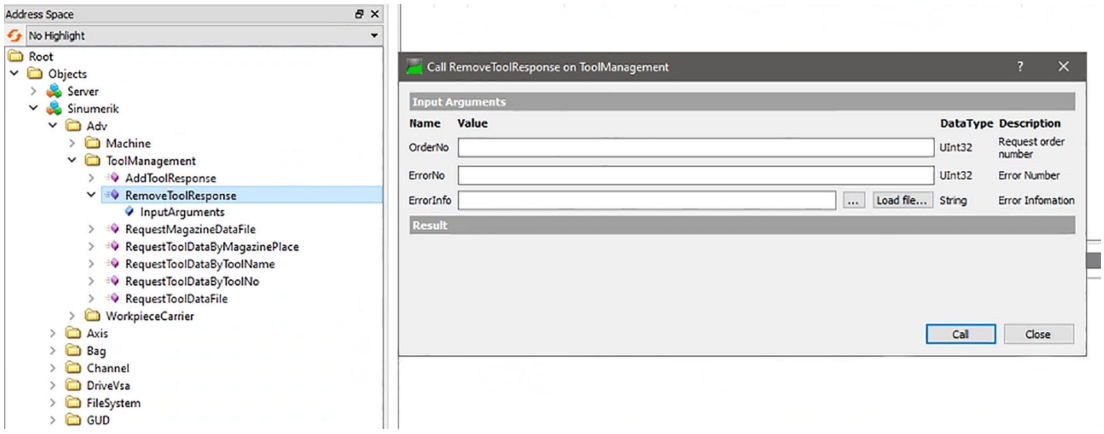


Figure A-32 Calling RemoveToolResponse method

Prerequisite

To initiate a tool response for unloading or deleting a tool, the OPC UA administrator has to provide the user with the access right "ADV_Write".

Structure and arguments

Signature of the method RemoveToolResponse is as follows:

```
OpcUa_StatusCode RemoveToolResponse(OpcUa_UInt32 orderNo,
OpcUa_UInt32 errorNo, OpcUa_string errorInfo)
```

Argument	Description
orderNo	Order number, as received in the RemoveToolRequestEventType event.
errorNo	If you do not want unload or delete the tool, then the error number (Page 262)has to be provided. If errorNo is set to 0, the tool will be unloaded or deleted.
errorInfo	If you do not want unload or delete the tool, then the error info (Page 262)has to be provided. If errorInfo is empty, the tool will be unloaded or deleted.

As an output argument, the OpcUaStatusCode will be provided. And the OPC UA server initiates the tool unloading or deleting process.

After successful tool unload or tool deleting a ReportToolEvent (Page 242) will be sent to the host. In case of unsuccessful unload or deleting process a ReportToolErrorEvent (Page 246) will be sent.

OpcUaStatusCode

The following table gives details on values and description of the OPC UA method call status:

Result	Description
Succeeded	Method is executed with success/failure.
OpcUa_BadInvalidArgument	Invalid inputs are provided.
OpcUa_BadUserAccessDenied	User does not have permission to invoke the method.

A.2.2.5 Workpiece Carrier

RequestWpcStatusByDockingIndex method

The RequestWpcStatusByDockingIndex method is provided under "Sinumerik > Adv > WorkpieceCarrier" the address space, which allows the explicit request of a workpiece carrier status. You can call this method by providing an order number stated by the host and a docking station index.

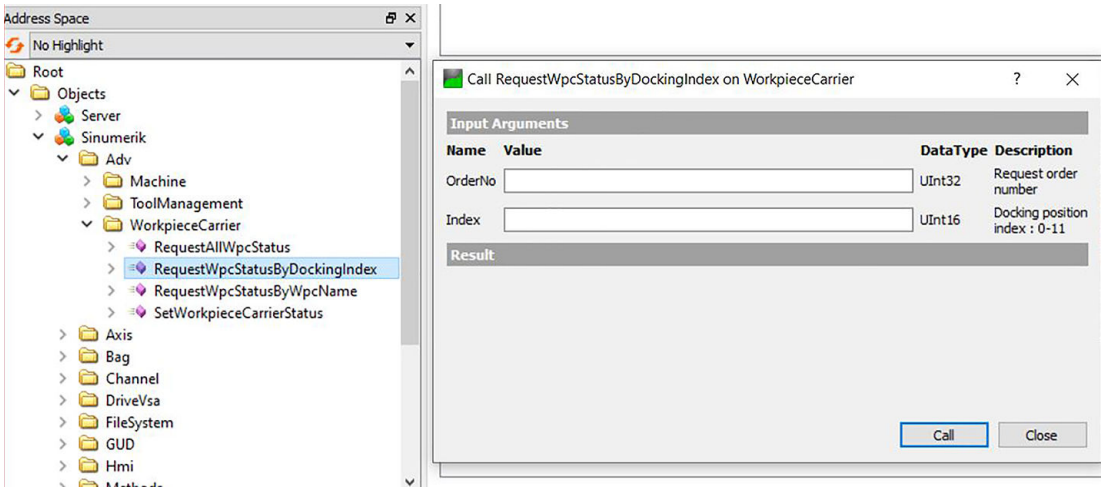


Figure A-33 Calling RequestWpcStatusByDockingIndex method

Prerequisite

To request the workpiece carrier status by the docking station index, the OPC UA administrator has to provide the user with the access right "ADV_Read".

Structure and arguments

Signature of the method RequestWpcStatusByDockingIndex is as follows:

```
OpcUa_StatusCode RequestWpcStatusByDockingIndex(OpcUa_UInt32
orderNo, OpcUa_UInt16 index)
```

Argument	Description
orderNo	Order no. stated by host request will be reported in fired event
index	Docking station index (PLC DB line up)

As an output argument, the OpcUaStatusCode will be provided and the WorkpieceCarrierEventType (Page 255) will be raised. The order number oderNo stated by the host request will be reported in the fired event.

OpcUaStatusCode

The following table gives details on values and description of the OPC UA method call status:

Result	Description
Succeeded	Method is executed with success.
OpcUa_BadInvalidArgument	Invalid inputs are provided.
OpcUa_BadUserAccessDenied	User does not have permission to invoke the method.

RequestWpcStatusByWpcName method

The RequestWpcStatusByWpcName method is provided under "Sinumerik > Adv > WorkpieceCarrier" in the address space, which allows the explicit request of a workpiece carrier status. You can call this method by providing an order number stated by the host and workpiece carrier name.

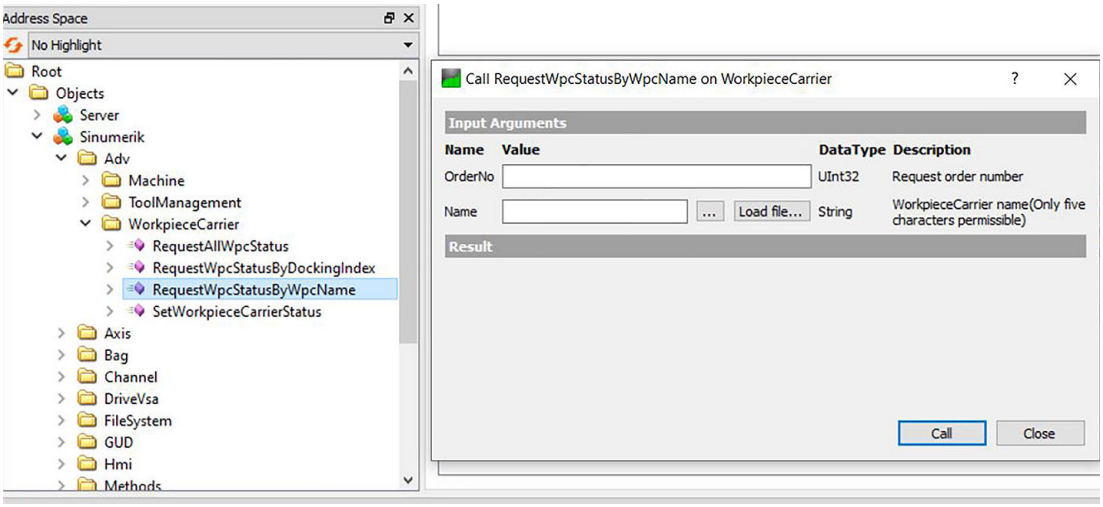


Figure A-34 Calling RequestWpcStatusByWpcName method

Prerequisite

To request the workpiece carrier status by the workpiece carrier name, the OPC UA administrator has to provide the user with the access right "ADV_Read".

Structure and arguments

Signature of the method RequestWpcStatusByWpcName is as follows:

```
OpcUa_StatusCode RequestWpcStatusByWpcName(OpcUa_UInt32 orderNo,
OpcUa_string name)
```

Argument	Description
orderNo	Order no. stated by host request will be reported in event
name	Workpiece carrier name

As an output argument, the WorkpieceCarrierEventType (Page 255) will be provided. The order number oderNo stated by the host request will be reported in the event.

OpcUaStatusCode

The following table gives details on values and description of the OPC UA method call status:

Result	Description
Succeeded	Method is executed with success.
OpcUa_BadInvalidArgument	Invalid inputs are provided.
OpcUa_BadUserAccessDenied	User does not have permission to invoke the method.

RequestAllWpcStatus method

The RequestAllWpcStatus method is provided under "Sinumerik > Adv > WorkpieceCarrier" in the address space, which allows the explicit request of all workpiece carrier status events. You can call this method by providing an order number stated by the host.

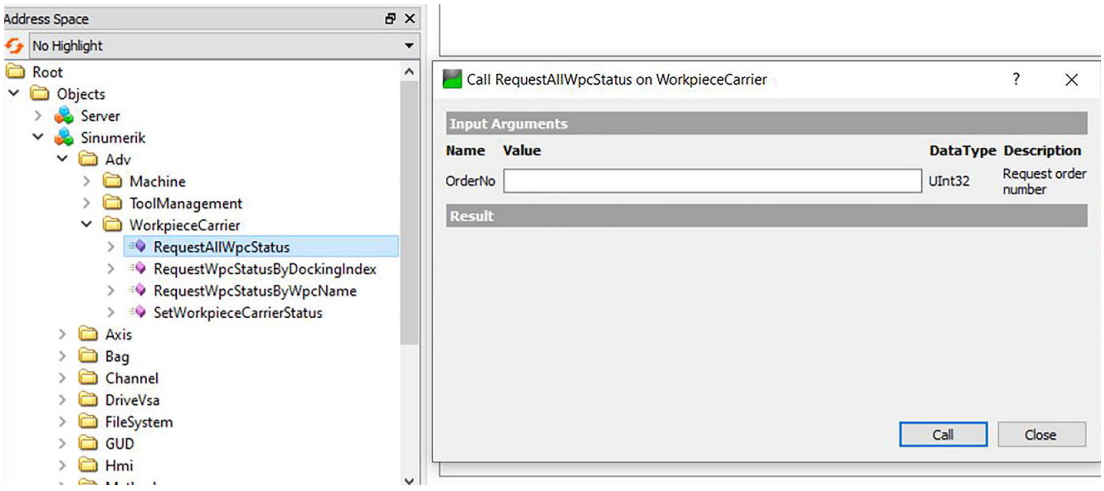


Figure A-35 Calling RequestAllWpcStatus method

Prerequisite

To request all workpiece carrier status, the OPC UA administrator has to provide the user with the access right "ADV_Read".

Structure and arguments

Signature of the method RequestAllWpcStatus is as follows:

```
OpcUa_StatusCode RequestAllWpcStatus(OpcUa_UInt32 orderNo)
```

Argument	Description
orderNo	Order no. stated by host request will be reported in event

As an output argument, the OpcUaStatusCode will be provided and the WorkpieceCarrierEventType (Page 255) (up to 12) will be raised. The order number stated by the host request will be reported in the events.

OpcUaStatusCode

The following table gives details on values and description of the OPC UA method call status:

Result	Description
Succeeded	Method is executed with success/failure.
OpcUa_BadInvalidArgument	Invalid inputs are provided.
OpcUa_BadUserAccessDenied	User does not have permission to invoke the method.

SetWorkpieceCarrierStatus method

The SetWorkpieceCarrierStatus method is provided under "Sinumerik > Adv > WorkpieceCarrier" in the address space, which allows the explicit setting of a workpiece carrier status. You can call this method by providing an order workpiece carrier name and workpiece carrier status to set.

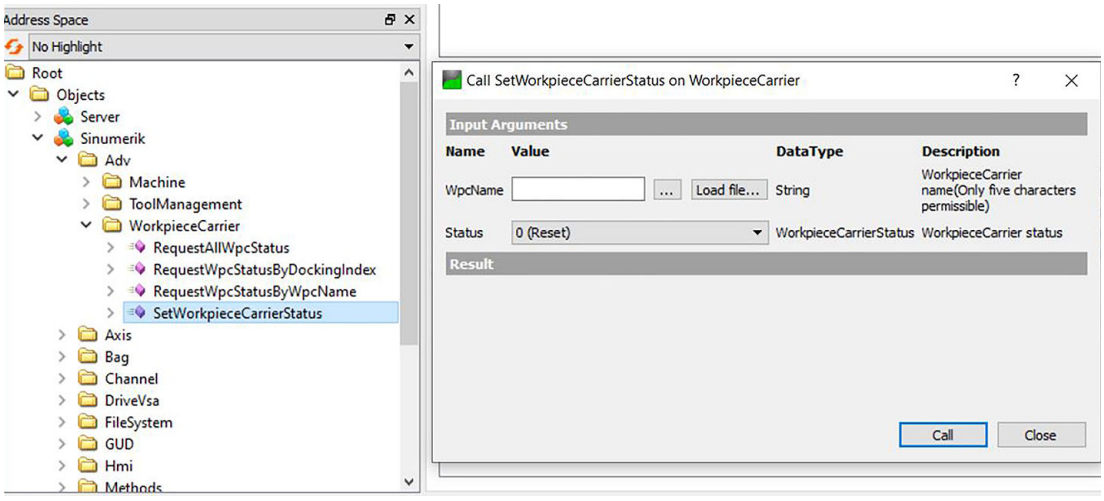


Figure A-36 Calling SetWorkpieceCarrierStatus method

Prerequisite

To set the workpiece carrier status, the OPC UA administrator has to provide the user with the access right "ADV_Write".

Note

The request for write access to the docking station data is done implicitly in this method call.

Structure and arguments

Signature of the method SetWorkpieceCarrierStatus is as follows:

```
OpcUa_StatusCode SetWorkpieceCarrierStatus(OpcUa_string name,
WorkpieceCarrierStatus status)
```

Argument	Description
name	Workpiece carrier name
status	Status (Page 261) to set

As an output argument, the OpcUaStatusCode will be provided.

OpcUaStatusCode

The following table gives details on values and description of the OPC UA method call status:

Result	Description
Succeeded	Method is executed with success.
OpcUa_BadInvalidArgument	Invalid inputs are provided.
OpcUa_BadUserAccessDenied	User does not have permission to invoke the method.

Index

A

Access MyMachine /OPC UA - ADV, 168
Accessibility, 23
AddCertificateUser, 102
Adding file system node
    SiOME, 70
Adding instance for SINUMERIK alarm
    SiOME, 68
Adding license
    SiOME, 76
Adding method node
    SiOME, 64
Address space, 113
    GUD, 113
    machine data, 113
    PLC blocks, 113
    setting data, 113
AddToolRequestEvent","Attribute, 251
AddToolResponse
    method, 278
AddTooolRequestEvent, 250
AddUser, 102
Adv node, 170
AdvExchange, 146, 249
AdvExchange node, 170
Alarms
    CnCAlarmType, 134
    Event types, 133
    sequence, 132
    subscribe, 130
    unsubscribe, 131
Application scenario, 23

B

Browsing, 113

C

ChangeMyPassword, 102
Checking the time, 30
ClampCubeSide, 219
Client, 21
Close
    method, 147
CNC software version 6.13 HF1, 207
Compatibility
    Virtual SINUMERIK ONE, 207
CopyFileFromServer
    method, 152
CopyFileToServer
    method, 152
Create tool, 159
    Method Result Codes, 159
    Parameters, 159
    Status code, 159
CreateCutting Edge, 162
    Method Result Codes, 162
    Parameters, 162
    Status code, 162
CreateDirectory
    method, 146
CreateFile
    method, 146
Creating new namespace
    SiOME, 60
Creating new object
    SiOME, 61
CSOM, 53
    Exporting, 54
    Importing, 89, 91
    Modeling with SiOME, 57
    process, 54
    workflow, 54
Customer Specific Object Model, 53

D

Data types, 119
DataFileEvent, 248
DataFileEvent","Attribute, 249
Delete
    method, 146
DeleteCuttingEdge, 163
    Method Result Codes, 163
    Parameters, 163
    Status code, 163
DeleteTool, 160
    Method Result Codes, 160
    Parameters, 160
    Status code, 160
DeleteUser, 102
DeleteUserAccess, 103

Deleting
rejected certificates, 43
trusted certificates, 42
development kit, 21
DNS based addressing, 34
DockingStationStatus, 260
DockPosCount, 218
DocPosStatus, 219
DuploNumber, 219, 221, 230, 232
DuploNumberCreated, 220, 231

E

Encryption, 22
ErrorInfo, 220, 221, 231, 232, 262
ErrorNo, 262
Exporting
server certificates, 41
ExtendedDrives, 146

F

FB, 219
File access rights, 144
FolderType, 146
FollowUpStatus, 262
ForceDuploNumber, 220, 230
ForceToolNumber, 220, 230
Functionalities, 21
Functionality
OPC UA Advanced, 167

G

GetMyAccessRights, 103
GetPosition
method, 147
GetUserAccessRights, 103
GetUserList, 102
GiveUserAccess, 107

H

Host name, 34
HostMode, 240

I

Importing
trusted certificates, 42
Industry 4.0, 21
Information flow
OPC UA Advanced, 169
Interaction mechanism
OPC UA Advanced, 169
IP address OPC UA server
Virtual SINUMERIK ONE, 207

L

Language identifier, 138
License, 30
LocalizedText, 138
LTime, 83

M

Machine node, 170
MachineMode, 218
MachineStatus, 218, 241
MachineStatusEvent, 236
MachineStatusEventType, (Attributes)
MagNum, 218
Mapping data types, 82
Max. lifetime interval (LifeTime Count), 209
Max. monitored items queue size (Subscription Queue size), 209
Max. number of users, 209
Maximum number monitored items, 209
Method
AddToolResponse, 278
Close, 147
CopyFileFromServer, 152
CopyFileToServer, 152
CreateDirectory, 146
CreateFile, 146
Delete, 146
GetPosition, 147
MoveOrCopy, 146
OpenCount, 147
Read, 147
RemoveToolResponse, 280
RequestAllWpcStatus, 283
RequestMachineStatus, 267
RequestMagazineDataFile, 276
RequestToolDataByMagazinePlace, 272
RequestToolDataByToolName, 273
RequestToolDataByToolNo, 275
RequestToolDataFile, 277
RequestWpcStatusByDockingIndex, 281
RequestWpcStatusByWpcName, 282

SelectedPartProgram, 268
SetPosition, 147
SetWorkpieceCarrierStatus, 284
Size, 147
Unselect, 157
UserWritable, 147
Writable, 147
Write, 147
Min. publishing interval, 209
Min. sampling interval, 209
MODE_PLC, 218
MODE_RPC, 218
Modeling rules
SiOME, 88
monitored items, 209, (maximum number)
Monitored items, 127
MoveOrCopy
method, 146

N

NCExtend, 146
Number of sessions, 209
Number of subscriptions, 209

O

OPC UA - Advanced, 167
OPC UA Advanced
functionality, 167
Information flow, 169
interaction mechanism, 169
OPC UA Advanced events
subscribing, 234
OPC UA Advanced license, 168
OPC UA data type, 82
OPC UA standard, 21
Open
method, 147
OpenCount
method, 147

P

Part Programs, 146
PCU / IPC, 23
PlaceNum, 218
PLC access
Absolute access, 118
Symbolic access, 118
PLC data types, 83
PlcClock
variable, 270
PlcMode, 240
PlcReq, 218
Publish interval, 127
Publishing intervals, 209

R

Read
method, 147
Rejected certificates
deleting, 43
trusting, 44
RemoveToolRequestEvent","Attribute, 254
RemoveToolResponse
method, 280
RemoveTooolRequestEvent, 253
Renewing
Server certificates, 40
ReportToolErrorEvent, 246
ReportToolErrorEvent","Attributes, 247
ReportToolEvent, 242
ReportToolEventType
Attributes, 243
ReportWpcErrorEvent, 258
ReportWpcErrorEventType
Attributes, 259
RequestAllWpcStatus
method, 283
RequestMachineStatus
method, 267
RequestMagazineDataFile
method, 276
RequestToolDataByMagazinePlace
method, 272
RequestToolDataByToolName
method, 273
RequestToolDataByToolNo
method, 275
RequestToolDataFile
method, 277
RequestWpcStatusByDockingIndex
method, 281
RequestWpcStatusByWpcName
method, 282
Reserve1, 218, 219
Reserve2, 218
Run/Create MyVirtual Machine version 1.1, 207

S

Sampling interval, 127
Sampling intervals, 209
Security settings, 22
Select
  Status code, 155
SelectedPartProgram
  method, 268
Server certificates
  exporting, 41
  renewing, 40
Session timeout, 209
SetPosition
  method, 147
Setting licenses for CSOM namespace
  SiOME, 80
SetWorkpieceCarrierStatus
  method, 284
SIMATIC data types, 83
SINUMERIK
  supported languages, 138
SINUMERIK Access MyMachine/ P2P, 54
SINUMERIK data type, 82
SINUMERIK namespace, 87
SINUMERIK ONE, 23
SINUMERIK OPC UA server, 21
SINUMERIK systems, 23
SinumerikMode, 239, 240
SiOME, 53
  Adding file system node, 70
  Adding instance for SINUMERIK alarm, 68
  Adding license, 76
  Adding method node, 64
  Creating new namespace, 60
  Creating new object, 61
  download, 53
  Modeling rules, 88
  Setting licenses for CSOM namespace, 80
Size
  method, 147
Sub Programs, 146
Subscribing
  OPC UA Advanced events, 234
Supported data types, 87
Supported Languages
  SINUMERIK, 138

T

Thin Client, 23
TNum, 218
ToolArea, 219, 221, 230, 232
ToolCreateBusy, 220, 231
ToolCreated, 220, 231
ToolCreateRequest, 219, 230
toolData
  Example, 244, 279
ToolDeleteBusy, 221, 232
ToolDeleted, 221, 232
ToolDeleteRequest, 220
ToolIdentifier, 219, 221, 230, 232
ToolManagement node, 170
ToolNumber, 220, 232
ToolNumberCreated, 220
ToolReportType, 244
ToolRequestType, 252
Trigger, 218
Trusted certificates
  deleting, 42
  importing, 42
Trusting
  rejected certificates, 44

U

UaExpert client, 140
Unselect
  method, 157
User administration, 22
UserWritable
  method, 147

V

Variable
  PlcClock, 270
Variable paths, 116
Virtual SINUMERIK ONE
  compatibility, 207
  IP address OPC UA server, 207

W

Work Pieces, 146
WorkpieceCarrier node, 170
WorkpieceCarrierEvent, 255

WorkpieceCarrierEventType
Attributes, 256
WorkpieceCarrierStatus, 261
WPC, 219
WPCStatus, 219
Writable
method, 147
Write
method, 147