

Brownfield Connectivity BFC Gateway

Function Manual

Valid for software:
Brownfield Connectivity - Release 1.11:
- BFC Client, Version 2.15
- BFC Gateway, Version 1.11

<u>Introduction</u>	1
<u>Security instructions</u>	2
<u>Product information</u>	3
<u>Requirement</u>	4
<u>Installing the BFC gateway</u>	5
<u>Operating the BFC gateway</u>	6
<u>Using the BFC client</u>	7
<u>Using an OPC UA server</u>	8
<u>Client interface with HTTP REST protocol</u>	9
<u>BFC gateway API</u>	10
<u>File transfer</u>	11
<u>Configuring the SSA gateway</u>	12
<u>Configuring the AMP gateway</u>	13
<u>BFC apps</u>	14
<u>Appendix</u>	A

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 AG. 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.

Table of contents

1	Introduction	11
1.1	About Brownfield Connectivity - Gateway	11
1.2	About this documentation	13
1.3	Feedback on the technical documentation	15
1.4	mySupport documentation	15
1.5	Service and Support.....	16
1.6	Using OpenSSL	17
1.7	Compliance with the General Data Protection Regulation.....	17
2	Security instructions	19
2.1	Fundamental safety instructions.....	19
2.1.1	General safety instructions.....	19
2.1.2	Warranty and liability for application examples	19
2.1.3	Security information	19
2.2	Specific security instructions.....	20
2.3	Improvements to product security	22
3	Product information.....	27
3.1	Form in which the BFC client is delivered.....	27
3.2	Form in which the BFC gateway is delivered	27
3.3	BFC update	28
3.3.1	BFC Gateway update (docker-based).....	28
3.3.2	BFC Gateway update (Kubernetes-based).....	33
3.3.3	BFC client update.....	35
3.4	Contacting the hotline	36
4	Requirement	39
4.1	Specialist know-how	39
4.2	General conditions.....	39
4.3	System requirements	40
4.3.1	BFC gateway	40
4.3.2	BFC apps.....	44
4.3.3	BFC client	44
4.3.3.1	Hardware and operating software	44
4.3.3.2	Network	46
4.3.4	FANUC client.....	49
4.3.5	MTConnect client.....	51
4.3.6	Modbus client.....	52
4.3.7	OPC UA client.....	53

4.3.8	S7 client	54
4.3.9	Heidenhain client.....	56
4.3.10	Beckhoff client	59
4.3.11	Ethernet IP client.....	60
4.3.12	Omron client	61
4.3.13	HTTP script client for HP 3D printers	61
4.3.14	SFTP client	62
5	Installing the BFC gateway	63
5.1	Requirement.....	63
5.2	Installing the docker CE.....	64
5.3	Installing the BFC gateway	67
5.4	Installing BFC Gateway in a Kubernetes cluster	71
6	Operating the BFC gateway	79
6.1	Icons and buttons	79
6.2	"Landing page" area	80
6.2.1	Log in to the BFC gateway.....	80
6.2.2	Installing a user-defined certificate	81
6.2.3	Starting processing	83
6.3	"Activation" area	83
6.3.1	Show activation	84
6.3.2	Importing activation	85
6.4	"Commissioning" area	86
6.4.1	Starting the configuration	86
6.4.2	BFC Protect	87
6.4.2.1	Integrating the tunnel device	88
6.4.2.2	Configuring BFC Protect for clients.....	89
6.4.2.3	Example: Configuring BFC Protect for the Modbus client	90
6.4.3	Creating clients (Import)	96
6.4.3.1	Creating a BFC client	97
6.4.3.2	Creating a FANUC client	103
6.4.3.3	Creating an MTConnect client	111
6.4.3.4	Creating a Modbus client	117
6.4.3.5	Creating an OPC UA client	127
6.4.3.6	Creating an S7 client.....	134
6.4.3.7	Creating an HTTP REST client.....	145
6.4.3.8	Creating an HTTP script client.....	147
6.4.3.9	Creating a Heidenhain client	157
6.4.3.10	Creating a Beckhoff client.....	165
6.4.3.11	Creating an Ethernet IP client	171
6.4.3.12	Creating an Omron client	180
6.4.3.13	Creating an MQTT client.....	187
6.4.3.14	MQTT client - configuring scriptlogic	189
6.4.3.15	MQTT client - calling connection data.....	193
6.4.3.16	Creating an SFTP client.....	194
6.4.3.17	Defining a dataset configuration	198
6.4.4	Displaying/editing the client	205
6.4.5	Using configured datasets	206

6.4.5.1	Exporting datasets from a client	207
6.4.5.2	Importing data sets into a client	209
6.4.6	Creating Middlewares (Processing)	211
6.4.6.1	Creating script logic	211
6.4.7	Creating the plant hierarchy	214
6.4.8	Creating gateways (Export)	218
6.4.8.1	Gateways (Export)	218
6.4.8.2	Creating a MindSphere gateway	218
6.4.8.3	Creating the MindConnectLib asset	224
6.4.8.4	MySQL export gateway	226
6.4.8.5	Creating a MySQL export gateway	230
6.4.8.6	Creating the AMP 4.1 export gateway.....	238
6.4.8.7	Creating an MQTT export gateway	246
6.4.8.8	Elasticsearch gateway	254
6.4.8.9	Creating an Elasticsearch gateway	255
6.4.8.10	InfluxDB gateway	261
6.4.8.11	Creating an InfluxDB gateway.....	263
6.4.8.12	HTTP gateway	269
6.4.8.13	Creating an HTTP gateway	270
6.4.8.14	MS SQL gateway	277
6.4.8.15	Creating an MS SQL gateway.....	283
6.4.8.16	Kafka gateway	290
6.4.8.17	Creating a Kafka gateway	291
6.5	"System State" area	297
6.5.1	Displaying information	297
6.5.2	Area "Clients (Import)"	299
6.5.3	"Middlewares (Logic)" area	303
6.5.4	"Gateways (Export)" area	307
6.5.5	System State Report	311
6.6	"Usermanagement" area	313
6.6.1	Creating a new user	315
6.6.2	Adapting an existing user.....	316
6.6.3	Deleting an existing user.....	316
6.6.4	Using an external IAM system	317
6.6.4.1	Requirements	317
6.6.4.2	Adjusting the configuration in the user management of the BFC gateway	317
6.6.4.3	Logging on over an external IAM	318
6.7	"Custom Variable List" area	319
7	Using the BFC client.....	321
7.1	Installing a BFC client	321
7.1.1	Requirement.....	321
7.1.2	Installing the BFC client on SINUMERIK 840D sl / 828D.....	321
7.1.2.1	Preparing the installation	321
7.1.2.2	Synchronizing date and time	322
7.1.3	Installing the BFC client on HMI-Advanced.....	323
7.1.4	Installing the BFC client on SINUMERIK Operate under Windows (PCU).....	326
7.1.5	Installing the BFC client on SINUMERIK Operate under Linux.....	329
7.1.6	Configuring data buffering	337
7.1.7	BFC client update	337

7.2	Deinstalling a BFC client.....	338
7.2.1	Deinstalling the BFC client on HMI-Advanced.....	338
7.2.2	Deinstalling the BFC client on SINUMERIK Operate under Windows (PCU).....	339
7.2.3	Uninstalling the BFC client from SINUMERIK Operate under Linux	339
7.3	Configuring high-frequency data acquisition.....	340
7.3.1	Requirement.....	342
7.3.2	Configuration.....	343
7.3.2.1	Performing the configuration	343
7.3.2.2	Compiling signals.....	344
7.3.2.3	Defining a start/stop operation	345
7.3.2.4	Changing the sampling rate for signals.....	347
7.3.2.5	Saving the configuration	348
7.3.2.6	Accept configuration	349
7.3.2.7	Test configuration	350
7.4	BFC client diagnostics	351
7.4.1	Analyzing the BFC client.....	351
7.4.2	Checking the accessibility of the BFC gateway in the network	354
7.4.3	Trace	356
7.4.3.1	Activating the trace.....	356
7.4.3.2	Analyzing the trace file.....	357
8	Using an OPC UA server	361
8.1	Overview	361
8.2	Configuring a certificate	361
8.2.1	Uploading the client certificates in the ConfigUI	362
8.3	OPC UA address space.....	363
8.4	Changing the system configuration while in operation.....	367
8.5	Write access to variables.....	368
8.5.1	Configuring the release list for write accesses	368
8.5.2	Writing variables using the "WriteData" method	371
8.5.3	Call method "WriteData"	372
8.5.4	Feedback after write process	374
8.5.5	Direct write access via variables.....	376
8.6	Historical Access	376
8.7	Companion Specification.....	379
8.8	Other OPC UA server settings	389
9	Client interface with HTTP REST protocol.....	391
9.1	Overview	391
9.2	Test and documentation	391
9.3	Authorization.....	393
9.4	Format for time stamp	395
9.5	Method POST /datasets	396
9.6	Method POST /datasets/raw.....	397

9.7	Method POST /events	401
9.8	Method POST /hfdataset.....	402
9.9	GET /writes method.....	403
9.10	POST /writes/{ID}/ack method	404
10	BFC gateway API	407
10.1	SwaggerUI	407
10.2	Authentication.....	407
10.3	Client configuration	408
10.4	Middleware configuration	438
10.5	Gateway configuration	450
11	File transfer.....	465
11.1	Configuring access rights	467
11.2	Using the WebDAV interface with curl	470
11.3	Using the WebDAV interface as Windows network drive	470
11.4	Using the WebDAV interface under Linux.....	472
11.5	Using the WebDAV interface with WinSCP.....	473
11.6	Directory structure on SINUMERIK machines.....	474
12	Configuring the SSA gateway	479
12.1	Overview	479
12.2	Requirement.....	479
12.3	Check status of the middleware.....	479
12.4	Creating aspects in MindSphere	480
12.5	Creating the asset type "bfc_ssa_sinumerik".....	481
12.6	Connecting a new machine to SSA	484
12.6.1	Creating a new asset of the type "bfc_ssa_sinumerik".....	485
12.6.2	Generating connection information of the assets.....	488
12.6.3	Creating a MindSphere gateway for SSA	489
12.7	Configuring the BFC client data acquisition	495
12.8	Creating and saving a machine identity	499
13	Configuring the AMP gateway.....	503
13.1	System requirements relating to the AMP server	503
13.2	Configuring the AMP server.....	504
13.2.1	General settings in the AMC server.....	504
13.2.1.1	Setting to process alarms	504
13.2.1.2	Configuring the http interface	505
13.2.1.3	Adapting Pit.ini	505
13.2.2	Configuring the machine in the AMP server	512

13.3	Configuring the AMP gateway	515
13.3.1	Configuration parameters of an AMP gateway	515
13.3.2	Data selection and topics of the AMP gateway	517
13.3.3	Environment variables of the AMP gateway	518
13.4	Script logic to link FANUC clients to AMP.....	518
13.4.1	Creating a script configuration for a client.....	519
13.4.2	Restarting the script logic.....	520
14	BFC apps	523
14.1	IBase app.....	523
14.1.1	Using the IBase app.....	524
14.1.2	Manually sending data to IBase	525
14.1.3	Sending data semiautomatically to IBase	526
14.2	Optimization Check app	527
14.2.1	Storing licenses.....	527
14.2.2	Creating a log	529
14.2.3	Using the Optimization Check app.....	529
14.2.4	Recording a trace session	532
A	Appendix.....	535
A.1	List of abbreviations.....	535
A.2	Checking the connection to a Heidenhain control.....	536
A.3	Checking the connection to the MTConnect control	539
A.4	Check connection to FANUC control system	540
A.5	API calls of the FOCAS interface.....	543
A.6	AMP gateway.....	546
A.6.1	AMP gateway signals.....	546
A.6.2	Script logic example: AMP_SIGNAL set.....	547
A.6.3	Script logic example: AMP_PARTCOUNT set.....	548
A.6.4	Script logic example: AMP_CYCLEDATA set.....	548
A.6.5	Alarms in the AMP gateway.....	549
A.6.6	AMP gateway logging	550
A.7	BFC client data points	551
A.7.1	Overview	551
A.7.2	NC variables.....	551
A.7.3	PLC tags.....	552
A.7.4	Machine data.....	553
A.7.5	Global User Data (GUD).....	554
A.7.6	Drive parameters	557
A.8	FANUC client data points (reading).....	561
A.8.1	cnc_rdtimer	561
A.8.2	cnc_sysinfo	562
A.8.3	cnc_stainfo.....	562
A.8.4	cnc_rddynamic2	563
A.8.5	cnc_rdpdf_subdirn	564
A.8.6	cnc_rdpdf_subdir	564
A.8.7	cnc_rdpdf_alldir	564

A.8.8	cnc_rdparam	565
A.8.9	cnc_diagnoss	565
A.8.10	cnc_rdset	566
A.8.11	cnc_rdsetnum	566
A.8.12	cnc_rdaxisdata	566
A.8.13	cnc_rdaxisname	567
A.8.14	cnc_rdpdlname	567
A.8.15	cnc_alarm2	567
A.8.16	pmc_get_number_of_pmc	567
A.8.17	pmc_rdmcrng	567
A.8.18	cnc_exeprgname	568
A.8.19	tooldata	568
A.9	FANUC client data points (writing)	569
A.9.1	cnc_wrparam	569
A.9.2	cnc_wrset	569
A.9.3	pmc_wrpncrng	570
A.9.4	cnc_wrtimer	570
A.10	S7 client	571
A.10.1	Addressing in the SIMATIC PLC	571
A.10.2	Data types, accuracy and formatting for write operations	572
A.11	Heidenhain client data points (reading)	574
A.11.1	Structure of the datapoints	574
A.11.2	GetRunInfo	574
A.11.3	GetMachineParameters	577
A.11.4	DataGetValue	577
A.11.5	ReadMemory	577
A.11.6	Examples of important Heidenhain datapoints	578
A.12	Heidenhain client data points (writing)	579
A.12.1	SetMachineParameters	579
A.12.2	DataSetValue	580
A.12.3	WriteMemory	580
A.13	Data points Beckhoff client	581
A.14	EIP client	582
A.14.1	Addressing of data, general	582
A.14.2	Addressing in the PCCC message format	582
A.14.3	Interpretation of 16 and 32-bit numbers, signed or unsigned	584
A.14.4	Addressing in the Allen-Bradley CIP message format with tags	584
A.14.5	Addressing arrays	586
A.14.6	Reading and writing very large 64-bit numbers	586
A.15	Data points of the Omron client	586
A.16	MQTT export gateway	587
A.16.1	Topics	587
A.16.2	Data format: data set	588
A.16.3	Data format: alarms	590
A.16.4	Data format: alarm state changes	591
A.16.5	Data format: high frequency data	592
A.17	OPC UA	595
A.17.1	Supported OPC UA profiles	595

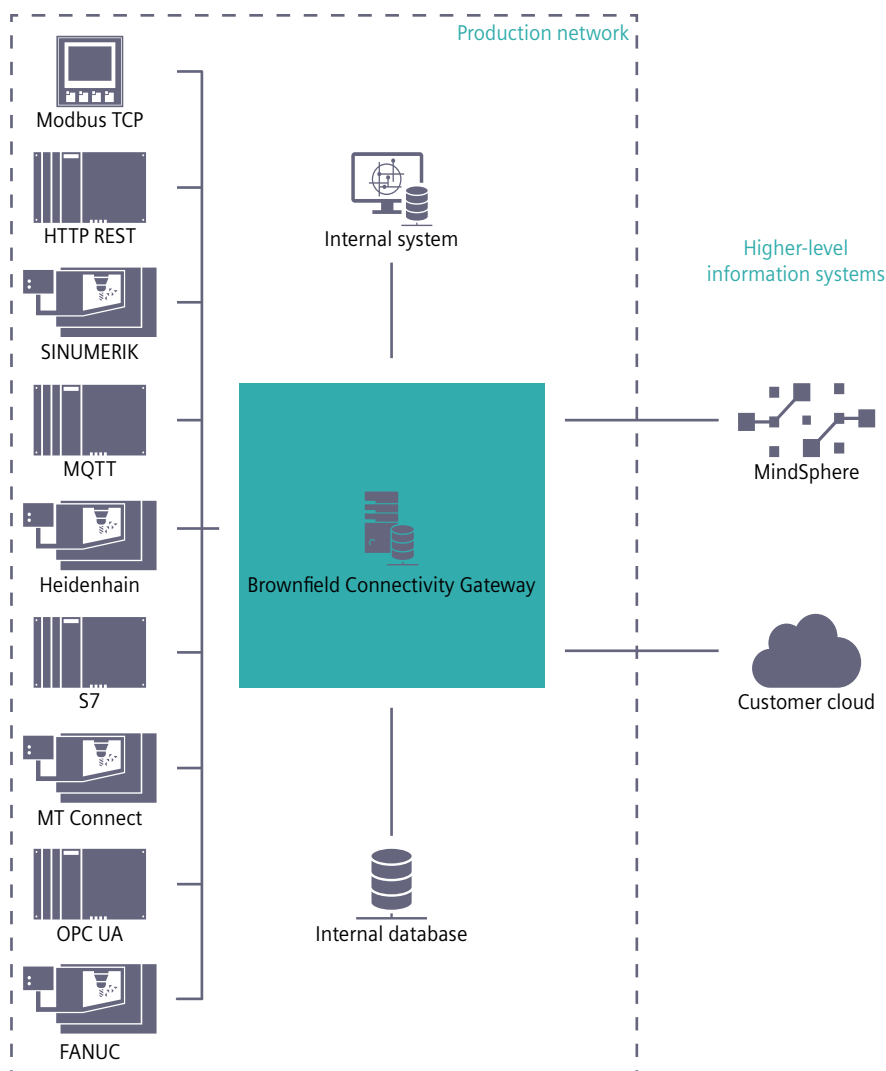
A.18	MindSphere MMM Dashboard	596
A.19	MS SQL gateway	601
A.20	HTTP script client	602
A.21	Further notes	605
A.22	Troubleshooting.....	609
A.23	Release Notes V1.11	611
	Glossary	613

Introduction

1.1 About Brownfield Connectivity - Gateway

Brownfield Connectivity (BFC) is software that can establish a connection between a heterogeneous production network and higher-level information systems. BFC integrates itself into the structural framework of an existing software and architecture concept, known as a brownfield situation.

Via a central BFC Gateway, you can connect SINUMERIK control systems, third-party controls and automation systems to higher-level systems. The system offers wide ranging options, such as reading and writing variables, preprocessing acquired data, transferring files from IT to OT and "trigger"-based reading from a central connectivity management system.



Note

High availability of the BFC Gateway

To increase the system reliability of BFC Gateway, high system availability can be implemented together with end users on a project-specific basis. If you are interested in the solution and require more detailed information, then contact your local Siemens sales partner.

Note

Performance dips in the network communication

Performance dips can occur in the network communication, for example as a result of incorrect configuration of connected devices. If a high number of data points are added, when commissioning and configuring the BFC Gateway, carefully ensure that performance dips do not occur in the network communication in the existing network infrastructure. Carry out network analysis before and after the BFC Gateway installation.

Architecture

The "BFC Gateway" software forms the architecture for this solution, which is installed on a virtual or physical system between your networked machines and office network. This software provides a connection between the various machines of the factory and a higher-level IT system.

The BFC client can be installed on SINUMERIK controls to connect them to the BFC gateway.

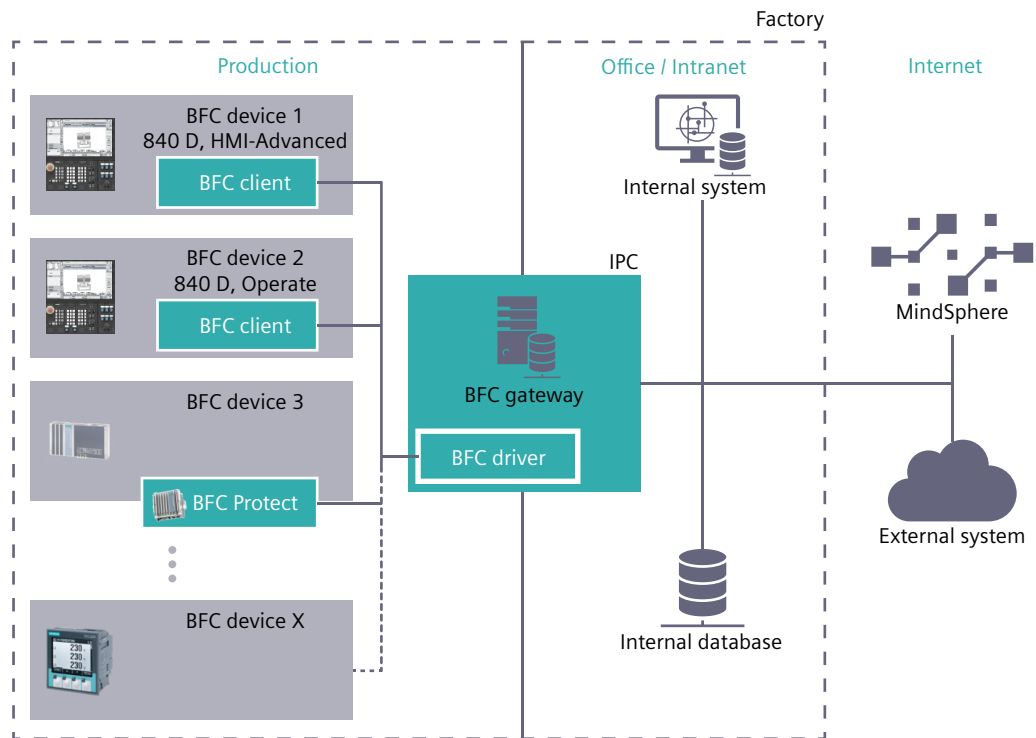
The BFC driver is a component of the BFC gateway that enables a connection to the BFC devices in the machine park of a factory.

A BFC device is a data source like a SINUMERIK control, third-party control, or another automation solution.

For devices that do not support encrypted communication, an encrypted configuration can be offered on a project-for-project basis between the device and the BFC Gateway. In the following diagram, this project-specific solution is declared as "BFC project" as example. If you are interested in the solution and require more detailed information, then contact your local Siemens sales partner.

Note

The BFC Gateway is neither a real-time nor deterministic system. As a consequence, data exchange in a precisely defined time interval cannot be guaranteed.



Different data formats

Different devices code their information in the various manufacturer-specific formats. The BFC Gateway transfers the data from the various devices and control systems unchanged. The higher-level system must be able to interpret the manufacturer-specific data.

If you wish to scale, link or preprocess the data, then you can configure this on a project-for-project basis.

If this data is to be visible in MindSphere Fleet Manager, then configuration operations are also required in MindSphere.

If you are using Apps, then carefully check the data and interface compatibility.

1.2 About this documentation

Contact your regional Siemens sales partner for training courses on the BFC Gateway.

This salesperson will provide you with a quotation for a training course that meets your individual requirements.

Note

To install and configure the BFC Gateway, you should first participate in an appropriate training course offered by Siemens AG.

FAQs

You can find answers to Frequently Asked Questions in the Service&Support pages at Product Support (<https://support.industry.siemens.com/cs/products?dtp=Faq&mf=ps&lc=en-DE>).

Target group

This documentation addresses appropriately qualified commissioning engineers.

The document provides information for commissioning engineers that they require to commission and parameterize the software.

Disclaimer

All product designations, product names, etc. may contain trademarks or other rights of Siemens AG, its subsidiaries or third parties. Unauthorized use may violate the rights of the respective owners.

Write accesses and file write accesses to connected controls and devices.

The BFC Gateway is only responsible for transferring data in the form of digital messages between sender and receiver.

The sender is responsible for the content of this message. This must be observed in particular for messages to control systems and devices that are to be executed as write accesses on the respective target device.

For file write accesses to controls and devices, too, the sender is responsible for the contents of the files transferred and, in particular, for the consequences of writing these files onto the target device.

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 Feedback on the technical documentation

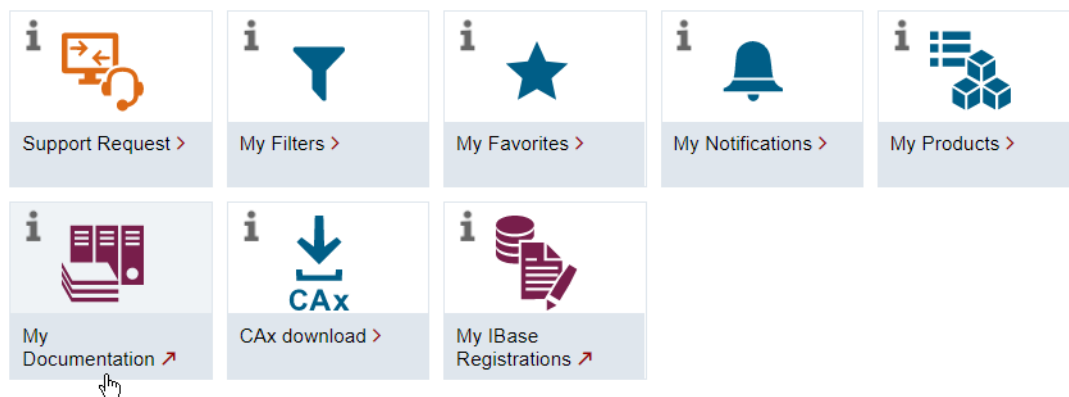
If you have any questions, suggestions or corrections regarding the technical documentation which is published in the Siemens Industry Online Support, use the link "Send feedback" link which appears at the end of the entry.

1.4 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 links and tools" (<https://support.industry.siemens.com/cs/ww/en/my>) portal page:

mySupport Links and Tools



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.

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

Data matrix code on the nameplate

The data matrix code on the nameplate contains the specific device data. This code can be read with a smartphone and technical information about the device displayed via the "Industry Online Support" mobile app.

1.6 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>)

1.7 Compliance with the General Data Protection Regulation

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

For this product, this means:

The product does not process or store any personal data, only technical function data (e.g. time stamps). If the user links this data with other data (e.g. shift plans) or if he/she stores person-related data on the same data medium (e.g. hard disk), thus personalizing this data, he/she must ensure compliance with the applicable data protection stipulations.

Security instructions

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 specific customer solutions, but are only intended to provide support for 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 Security information

Siemens provides products and solutions with industrial security 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 security 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 security measures that may be implemented, please visit

<https://www.siemens.com/industrialsecurity>.

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 Security RSS Feed under

<https://www.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 holistic, state-of-the-art industrial security concept for the installation or machine.
- Make sure that you include all installed products into the holistic industrial security concept.
- Protect files stored on exchangeable storage media from malicious software by with suitable protection measures, e.g. virus scanners.
- On completion of commissioning, check all security-related settings.

2.2 Specific security instructions

Note

Network traffic

When using the BFC gateway, network traffic is generated in the customer network. Ensure that sufficient bandwidth is available in your network depending on the use of the BFC gateway.

Note**Connection to SINUMERIK control systems**

SINUMERIK control systems are connected by default via an encrypted MQTT/TLS 1.2/TLS 1.3 connection. If a system operator explicitly wishes to use an unencrypted connection, then this can be implemented as part of the configuration process.

You can contact the hotline to obtain the necessary information to do this.

Note**Connecting SINUMERIK control systems to the network**

Connecting SINUMERIK control systems via the BFC Gateway using TLS 1.2/TLS 1.3/MQTT corresponds to current security standards.

Note**SINUMERIK control system security**

The company operating the systems is solely responsible for preventing unauthorized access to your plants, systems, SINUMERIK control systems and the network.

Systems, SINUMERIK control systems and components should only be connected to the company's network or the Internet if and to the extent necessary and with appropriate security measures in place. Security measures include the use of firewalls, whitelisting, virus scanners, network segmentation, OS patching.

NOTICE
Data misuse due to an unprotected Internet connection
An unrestricted Internet connection can result in data misuse.
Before establishing a network connection, ensure your PC is only connected to the network via a secure connection. Carefully observe the security-relevant notes and instructions.

Note**Data backup before and after commissioning the BFC client**

Before you make any changes, carefully ensure that data backups have been generated via the PCU 50 (Ghost) or the CF card of the NCU (tgz file).

Create a data backup after completing all activities.

Additional information about creating a data backup is provided under:

- Commissioning Manual SINUMERIK 840Di sl / 840D sl / 840 D, Base Software and HMI-Advanced
 - Commissioning Manual SINUMERIK 840D sl Base Software and Operating Software
 - Equipment Manual for SINUMERIK 828D PPU and components
-

Note

Backing up the commissioning PC

The necessary security measures (e.g. virus scanner, firewalls, OS patching, etc.) must be implemented on the PCs that are used to configure a BFC Gateway at the OEM or end user.

Note

Appropriate parameterization

When defining the amount of data to be transferred to the BFC Gateway and the intervals, the system load of the relevant control system and the network infrastructure involved must be carefully taken into consideration. The company operating the plant or system is responsible for carefully planning and providing the datapoints. This can avoid performance bottlenecks at the control system and in the network.

Additional information relating to IT security is provided in Chapter: Security information (Page 19).

NOTICE
Damage to the machine by writing to control variables
Writing to control variables can damage the machine. This is why, from Version 1.11, when configuring the permitted write access operations, an appropriate note must be confirmed.

Note

Writing to variables for SINUMERIK control systems

For SINUMERIK control systems, the BFC Gateway is prevented from writing to the following variables:

- Machine data, drive parameters, DB20, DB18
 - Axis DBs: DB31-DB61 (for SINUMERIK 828D: DB3800-DB3818, DB3900)
 - Basic program DBs: DB7, DB8, DB10 (for SINUMERIK 828D: DB2900)
-

2.3 Improvements to product security

Warnings are issued when configuring write access operations.

The area of application refers to writing to variables and deleting and/or changing files in the control system.

Note

The BFC client prevents the BFC Gateway from being able to change security-relevant addresses and directories.

System requirements

- The Declaration of Conformity of the machine OEM regarding machine safety (hazardous to human life and safety) is not diminished when using the BFC Gateway.
- When using BFC write functions, users must be warned about potential economic damage.

NOTICE

Economic damage to the machine as a result of incorrect use

Incorrectly using BFC write functions can result in economic damage at the machine. This cannot be completely prevented in the BFC Gateway, e.g. for a GUD variable that specifies a traversing path. If the specified traversing path is too long then the machine will be damaged.

SINUMERIK 840D PowerLine / SolutionLine / One

Data areas of the SINUMERIK control system, which have the potential of influencing "SINUMERIK Safety Integrated":

Note

Deviating areas for SINUMERIK 828D are specified in brackets.

- Machine data / Drive parameters/ DB20
- SAFE.SPF (lock the complete directory) (828D: is not used)
- Manufacturer cycles
- Safety-relevant DBs: DB18 (828D: is not used)
- Axis DBs: DB31-DB61 (828D: DB3800-DB3818, DB3900)
- Basic program DBs: DB7, DB8, DB10 (828D: DB2900)

From this, the following addresses and directories were derived to access the control system:

- "DB20.xxx", "/Plc/DataBlock/xxx[c20,xxx"
- "DB18.xxx", "/Plc/DataBlock/xxx[c18,xxx"
- "DB31.xxx" - "DB61.xx", "/Plc/DataBlock/xxx[c31,xxx" - "/Plc/DataBlock/xxx[c61,xxx"
- "DB7.xxx", "/Plc/DataBlock/xxx[c7,xxx"
- "DB8.xxx", "/Plc/DataBlock/xxx[c8,xxx"
- "DB10.xxx", "/Plc/DataBlock/xxx[c10,xxx"
- "DB3800.xxx" - "DB3818.xx", "/Plc/DataBlock/xxx[c3800,xxx" - "/Plc/DataBlock/xxx[c3818,xxx"
- "DB2900.xxx", "/Plc/DataBlock/xxx[c2900,xxx"
- "xxxTEA_ACX/xxx"
- "xxxSEA_ACX/xxx"
- "/Nck/Configuration/xxx"

2.3 Improvements to product security

- "/Nck/Settings/xxx"
- "/Nck/Drive/xxx"
- "/Nck/ProtectedArea/xxx"
- "/Channel/Settings/xxx"
- "/Channel/Drive/xxx"
- "/Channel/Configuration/xxx"
- "/Channel/ProtectedArea/xxx"
- "/Axis/Drive/xxx"
- "/Axis/Settings/xxx"
- "/DriveVsa/xxx"
- "/acx/xxx"
- Directory "/NC/CST.DIR/xxx" => Standard cycles
- Directory "/NC/CMA.DIR/xxx" => Manufacturer cycles (incl. SAFE.SPF)

This involves a superset of addresses, which can be used under HMI-Advanced (PowerLine) and Operate (SolutionLine/One/828D).

Overview

You configure write access operations to variables of the control system in the ConfigUI of the BFC client in Section "Define whitelisting configuration".

In this section, you configure which control system addresses may be written to.

The white list defined here (list of addresses) is transferred to the BFC client for each change and when the machine restarts.

The BFC client only permits write access operations to addresses that are in this list.

Attempts to access other addresses are rejected with message "Access denied".

Example:

In this example, a message is sent with two addresses to the BFC client (connectivity/machine/<MachineId>/iotclient/update/config/request/).

```
{
  "writeAccess": [
    "DB21.DBX7.1",
    "DB20.DBX1"
  ]
}
```

At this position, address "DB20.DBX1" is filtered out, as this lies in a SINUMERIK data area, which potentially has influence on "SINUMERIK Safety Integrated".

The addresses are filtered out using a regular expression.

Note

Observe uppercase/lowercase letters and space characters.

The following log message is entered:

2022-07-13T17:09:21.246498+02:00 [E] 1254 Write access to the following addresses is not allowed:
'DB20.DBX1 '

All files from directories "/NC/CST.DIR/xxx" (standard cycles) and "/NC/CMA.DIR/xxx" (manufacturer cycles incl. SAFE.SPF) are also filtered out using a regular expression in the affected functions of the BFC client.

Application in ConfigUI

ConfigUI calls a check box for "allowlist" and "filepermissions", which must be defined in order to save the configuration.

Note

The settings in ConfigUI apply to all clients.

- To be able to release an address in the "allowlist", you must set the check box as shown in the following diagram.

- If there is no entry in the "allowlist", then there is no check box for the "allowlist".
If there is no entry with skills "delete" or "write" in the file authorization list, then there is no check box for file authorization.

2.3 Improvements to product security

- The check box is displayed as soon as an entry is set for one of the skills "delete" or "write".

6 Define file permissions
Optional

?
Type file path...
filepath

☐ Read ☐ Create ☒ Write ☒ Delete +

☐ Warning regarding machine safety has been considered.

With the access rights "Write" and "Delete" it is possible to change or delete files on the controller.
This can lead to unwanted data loss and possible machine damage.

Make sure that the configured file access in your application scenario does not lead to any undesired behavior on the machine.

The manufacturer of the software (Siemens AG) is not liable for damages caused by improper file access.

Type description of the intended behavior

- Select the check box to save the skills for the appropriate path.

6 Define file permissions
Optional

?
Type file path...
filepath

☐ Read ☐ Create ☒ Write ☒ Delete +

☒ Warning regarding machine safety has been considered. Date: 11.8.2022, 14:19:04

With the access rights "Write" and "Delete" it is possible to change or delete files on the controller.
This can lead to unwanted data loss and possible machine damage.

Make sure that the configured file access in your application scenario does not lead to any undesired behavior on the machine.

The manufacturer of the software (Siemens AG) is not liable for damages caused by improper file access.

Type description of the intended behavior
Example description

You must again select the check box if you wish to edit, add or import entries.

Product information

3.1 Form in which the BFC client is delivered

Siemens provides you with the software corresponding to the order data, either on a data storage medium or per download.

Siemens includes with the software an electronic form of the software documentation.

Note**Software package**

The software package is a zip archive that contains the software and documents for commissioning a SINUMERIK 840D/840D sl/828D.

The Siemens Third-Party Software Disclosure Document (BFC_readme_oss.html) for BFC is provided in the root directory of the software provided.

3.2 Form in which the BFC gateway is delivered

Siemens provides you with the software corresponding to the order data, either on a data storage medium or per download, as well as the associated Certificate of License (CoL).

Siemens includes with the software an electronic form of the software documentation.

Siemens provides a license key, which is documented on the CoL.

Note**Installation**

Install the software package as described in this document.

More information can be found in Chapter: Installing the BFC gateway (Page 67).

The Siemens Third-Party Software Disclosure Document (BFC_readme_oss.html) for BFC is provided in the root directory of the software provided.

3.3 BFC update

3.3.1 BFC Gateway update (docker-based)

Preconditions

- Access data for your existing BFC installation
- Root rights on the BFC Gateway host system
- You can obtain the current BFC version from your regional Siemens contact person
- An SSH client, for example PuTTY
- An SFTP client, for example WinSCP

Upgrading BFC version from v1.7 to v1.10.1

1. Copy the new BFC version unzipped to the target system in a folder next to the existing installation.
2. Make the setup file `setup.linux.x64` so that it can be run.
 - `sudo chmod 744 setup.linux.x64`
 - This means that the owner of the setup is allocated the right to run this.
3. Run the setup via the Command Line Interface (CLI) of PuTTY. To do this, run the following command with user and password of the existing BFC installation in the folder of the BFC version to be installed:
`sudo ./setup.linux.x64 -u myuser -p mypass`
 - When compared to a first installation, perform the setup without parameter `-deploy`.
4. Compare file `/docker/secret.env` of the existing installation with file `/docker/secret.env` of the BFC version to be installed.
 - Lines or content are available in the new `secrets.env` file, which are not available in the existing `secrets.env` file.
5. Expand the existing `secrets.env` file to include these lines/content. Carefully ensure that you do not replace any existing "Password" or "User name" in the existing `secrets.env` file.
 - Expand line 3 to include the following: `'http://entry:9876/doghouse.crl'`
 - Insert the following in line 4: `BFC_CA_NAME='BFC CA <Custom Date>'`
 - Insert the following in line 5: `BFC_CERT_HOST_FOLDER_PATH='/etc/ssl/certs/ca-certificates.crt'`
 - Expand line 17 to include the following: `'entry'`
6. Before the update, copy the modified file into the docker directory of the new version. Replace the `secrets.env` file saved there.
7. Delete the stacks before you execute the setup using parameter `deploy`.

8. List the existing docker stacks using the following command:
`sudo docker stack ls`
9. Delete the listed stacks using the following command:
`sudo docker stack rm stack1 stack2 ... stackN`
10. To do this, run the following command with user and password of the existing BFC installation in the folder of the BFC version to be installed:
`sudo ./setup.linux.x64 -deploy -u myuser -p mypass`
11. For ALL configured clients, middleware and gateways, update the version numbers of the docker image. You can now read off the new version numbers using tile "System State".

Note

It is possible that certain lotServer services do not start as a result of queues that have not been activated.

You can check the log file of a service using the following command: `sudo docker service logs <service_name>`

You can activate the queue using the following command: `rabbitmqctl enable_feature_flags all`

Upgrading BFC version from v1.8 to v1.10.1

1. Copy the new BFC version unzipped to the target system in a folder next to the existing installation.
2. Make the setup file `setup.linux.x64` so that it can be run.
 - `sudo chmod 744 setup.linux.x64`
 - This means that the owner of the setup is allocated the right to run this.
3. Run the setup via the Command Line Interface (CLI) of PuTTY. To do this, run the following command with user and password of the existing BFC installation in the folder of the BFC version to be installed:
`sudo ./setup.linux.x64 -u myuser -p mypass`
 - When compared to a first installation, perform the setup without parameter `-deploy`.
4. Compare file `/docker/secret.env` of the existing installation with file `/docker/secret.env` of the BFC version to be installed.
 - Lines or content are available in the new `secrets.env` file, which are not available in the existing `secrets.env` file.
5. Expand the existing `secrets.env` file to include these lines/content. Carefully ensure that you do not replace any existing "Password" or "User name" in the existing `secrets.env` file.
 - Expand line 3 to include the following: `'http://entry:9876/doghouse.crl'`
 - Insert the following in line 5: `BFC_CERT_HOST_FOLDER_PATH='/etc/ssl/certs/ca-certificates.crt'`
 - Expand line 17 to include the following: `'entry'`
6. Before the update, copy the modified file into the docker directory of the new version. Replace the `secrets.env` file saved there.

7. Delete the stacks before you execute the setup using parameter `deploy`.
8. List the existing docker stacks using the following command:
`sudo docker stack ls`
9. Delete the listed stacks using the following command:
`sudo docker stack rm stack1 stack2 ... stackN`
10. To do this, run the following command with user and password of the existing BFC installation in the folder of the BFC version to be installed:
`sudo ./setup.linux.x64 -deploy -u admin -p mypass`
11. For ALL configured clients, middleware and gateways, update the version numbers of the docker image. You can now read off the new version numbers using tile "System State".

Upgrading BFC version from v1.9 to v1.10.1

1. Copy the new BFC version unzipped to the target system in a folder next to the existing installation.
2. Make the setup file `setup.linux.x64` so that it can be run.
 - `sudo chmod 744 setup.linux.x64`
 - This means that the owner of the setup is allocated the right to run this.
3. Run the setup via the Command Line Interface (CLI) of PuTTY. To do this, run the following command with user and password of the existing BFC installation in the folder of the BFC version to be installed:
`sudo ./setup.linux.x64 -u myuser -p mypass`
 - When compared to a first installation, perform the setup without parameter `-deploy`.
4. Compare file `/docker/secret.env` of the existing installation with file `/docker/secret.env` of the BFC version to be installed.
 - Lines or content are available in the new `secrets.env` file, which are not available in the existing `secrets.env` file.
5. Expand the existing `secrets.env` file to include these lines/content. Carefully ensure that you do not replace any existing "Password" or "User name" in the existing `secrets.env` file.
 - Insert the following in line 5: `BFC_CERT_HOST_FOLDER_PATH='/etc/ssl/certs/ca-certificates.crt'`
6. Before the update, copy the modified file into the docker directory of the new version. Replace the `secrets.env` file saved there.
7. Delete the stacks before you execute the setup using parameter `deploy`.
8. List the existing docker stacks using the following command:
`sudo docker stack ls`
9. Delete the listed stacks using the following command:
`sudo docker stack rm stack1 stack2 ... stackN`

10. To do this, run the following command with user and password of the existing BFC installation in the folder of the BFC version to be installed:
`sudo ./setup.linux.x64 -deploy -u myuser -p mypass`
11. For ALL configured clients, middleware and gateways, update the version numbers of the docker image. You can now read off the new version numbers using tile "System State".

Upgrading BFC version from v1.10 to v1.10.1 / BFC version from v1.10.1 to v1.11

1. Copy the new BFC version unzipped to the target system in a folder next to the existing installation.
2. Make the setup file `setup.linux.x64` so that it can be run.
 - `sudo chmod 744 setup.linux.x64`
 - This means that the owner of the setup is allocated the right to run this.
3. Run the setup via the Command Line Interface (CLI) of PuTTY. To do this, run the following command with user and password of the existing BFC installation in the folder of the BFC version to be installed:
`sudo ./setup.linux.x64 -u myuser -p mypass`
 - When compared to a first installation, perform the setup without parameter `-deploy`.
4. Compare file `/docker/secret.env` of the existing installation with file `/docker/secret.env` of the BFC version to be installed.
 - With the exception of "User name" and "Password", the new `secrets.env` file is identical with the existing `secrets.env` file. Replace the new `secrets.env` file with the existing file.
5. Delete the stacks before you execute the setup using parameter `deploy`.
6. List the existing docker stacks using the following command:
`sudo docker stack ls`
7. Delete the listed stacks using the following command:
`sudo docker stack rm stack1 stack2 ... stackN`
8. To do this, run the following command with user and password of the existing BFC installation in the folder of the BFC version to be installed:
`sudo ./setup.linux.x64 -deploy -u myuser -p mypass`
9. For ALL configured clients, middleware and gateways, update the version numbers of the docker image. You can now read off the new version numbers using tile "System State".

Installing BFC Analytics Version v1.0 on BFC Version v1.9

1. Copy the BFC Analytics version unzipped to the target system in a folder next to the existing installation.
2. Make setup file `setup_bfc-analytics.sh` so that it can be run.
 - `sudo chmod 744 setup.linux.x64`
 - This means that the owner of the setup is allocated the right to run this.
3. Execute the following command with user name and password of the existing BFC installation in the folder of the BFC Analytics version to be installed:
`sudo ./setup.linux.x64 -user myuser -password mypass -deploy -host myIP`
When doing this, ensure that you use `-user` and `-password` (instead of `-u` and `-p`).

Upgrading BFC version v1.9 with BFC Analytics v1.0 on BFC Version v1.10.1

The following describes how the BFC Gateway can be upgraded from version v1.9 to version 1.10.1 without having to reinstall the BFC Analytics tool.

Requirement:

- BFC version v1.9 is installed.
- BFC Analytics v1.0 is installed.
- BFC version v1.10.01 is to be installed without upgrading BFC Analytics.
 - The background to this is that by installing the latest BFC Analytics version, the existing version is overwritten, and therefore no longer functions.

Procedure:

1. Copy the new BFC version unzipped to the target system in a folder next to the existing installation.
2. Make the setup file `setup.linux.x64` so that it can be run.
 - `sudo chmod 744 setup.linux.x64`
 - This means that the owner of the setup is allocated the right to run this.
3. Run the setup via the Command Line Interface (CLI) of PuTTY. To do this, run the following command with user and password of the existing BFC installation in the folder of the BFC version to be installed:
`sudo ./setup.linux.x64 -u myuser -p mypass`
 - When compared to a first installation, perform the setup without parameter `-deploy`.
4. Compare file `/docker/secret.env` of the existing installation with file `/docker/secret.env` of the BFC version to be installed.
 - Lines or content are available in the new `secrets.env` file, which are not available in the existing `secrets.env` file.
5. Expand the existing `secrets.env` file to include these lines/content. Carefully ensure that you do not replace any existing "Password" or "User name" in the existing `secrets.env` file.
 - Insert the following in line 5: `BFC_CERT_HOST_FOLDER_PATH='/etc/ssl/certs/ca-certificates.crt'`

6. Before the update, copy the modified file into the docker directory of the new version.
Replace the `secrets.env` file saved there.
7. Delete the stacks before you execute the setup.
8. List the existing docker stacks using the following command:
`sudo docker stack ls`
9. Delete the listed stacks using the following command:
`sudo docker stack rm stack1 stack2 ... stackN`
10. To do this, run the following command with user name and password of the existing BFC installation in the folder of the BFC version to be installed:
`sudo ./setup.linux.x64 -deploy -u myuser -p mypass -excluded-apps bfc-analytics`
 - When doing this, ensure that the BFC Analytics App is not also installed: `-excluded-apps bfc-analytics`
11. For ALL configured clients, middleware and gateways, update the version numbers of the docker image. You can now read off the new version numbers using tile "System State".

3.3.2 BFC Gateway update (Kubernetes-based)

Installing Kubernetes

Requirement

- A Debian system is available.
- A CentOS7 system is available.
- The system has an Internet connection.

Procedure

1. Establish a connection to the system, e.g. using WinSCP/PuTTY.
2. Log in as user with root rights.
3. Enter the following command:
`sudo visudo`
4. Scroll to Defaults `secure_path`. Expand the path by `/urs/local/bin`:
Defaults `secure_path = /sbin:/bin:/urs/sbin:/urs/bin:/usr/local/bin`
5. Enter the following command:
`export KUBECONFIG=kubeconfig.yaml`
 - Map the KUBECONFIG environment variable to the path of the Kubernetes configuration file.

6. Install Kubernetes. Enter the following command:
`sudo curl -sfL https://get.k3s.io | sh -`
7. Reboot the system.
 - After logging on again, using command `sudo kubectl get nodes | grep 0/1` you can check that Kubernetes has been correctly installed.

Installing the BFC Gateway on Kubernetes

Requirement

- Kubernetes is installed on the target system.

Procedure

1. Copy the BFC version from Pridanet unzipped to the target system.
2. Make the setup file `setup.linux.x64` so that it can be run.
3. Run the setup using the following command:
`./setup.linux.x64 -u <username> -p <password> -kubernetes=true
 -kubeconfig=/etc/rancher/k3s/k3s.yaml -deploy -namespace=bfc -k3s`

Upgrading BFC version from v1.9 to v1.10.1

1. Copy the new BFC version unzipped to the target system in a folder next to the existing installation.
2. Make the setup file `setup.linux.x64` so that it can be run.
 - `sudo chmod 744 setup.linux.x64`
 - This means that the owner of the setup is allocated the right to run this.
3. Run the setup via the Command Line Interface (CLI) of PuTTY. To do this, run the following command with user name and password of the existing BFC installation in the folder of the BFC version to be installed:
`./setup.linux.x64 -u <username> -p <password> -kubernetes=true
 -kubeconfig=/etc/rancher/k3s/k3s.yaml -namespace=bfc -k3s`
 - Carefully ensure that setup is NOT executed with `-deploy`.
 - By executing this command, files `secret.env` and `configs.env` are also created.
 - From version 1.9 to version 1.10.01, the installation structure changes.
 - In version 1.9, all files that are created are saved to the following directory: `/kubernetes/builds/1.9/...`
 - In version 1.10.01, all files that are created are saved to the following directory: `/kubernetes/config/...`
4. Compare file `/kubernetes/builds/1.9/configs/secret.env` of the existing installation with file `/kubernetes/config/configs/secret.env` of the BFC version to be installed.
 - A line is available in the new `secrets.env` file which is not available in the existing `secrets.env` file.

5. Expand the existing `secrets.env` file to include this line. Carefully ensure that you do not replace any existing "Password" or "User name" in the existing `secrets.env` file.
 - Insert the following in line 5: `BFC_CERT_HOST_FOLDER_PATH='/etc/ssl/certs/ca-certificates.crt'`
6. Before the update, copy the modified file into the Kubernetes directory `kubernetes/config/configs/` of the new version. Replace the `secrets.env` file saved there.
7. Compare file `/kubernetes/builds/1.9/thirdparty/broker/config/definitions.json` of the existing installation with file `/kubernetes/config/broker/definitions.json` of the BFC version to be installed.
 - Copy the passwords of the existing `definitions.json` into the new `definitions.json` file.
 - Lines 6, 11, 16, 25, 26, 31, 89 and 112 are involved.
 - Save the modified, new `definitions.json` in directory `kubernetes/config/broker/definitions.json`.
8. Compare file `/kubernetes/builds/1.9/infrastructure/configui/config/settings.json` of the existing installation with file `/kubernetes/config/configui/settings.json` of the BFC version to be installed.
 - Copy the passwords of the existing `definitions.json` into the new `definitions.json` file.
 - Lines 37, 39, 50, 55 and 60 are involved.
 - Save the modified, new `definitions.json` in directory `kubernetes/config/broker/definitions.json`.
9. Compare file `/kubernetes/builds/1.9/infrastructure/configui/users/users.json` of the existing installation with file `/kubernetes/config/users/users.json` of the BFC version to be installed.
 - The file comprises only one line.
 - Copy the ID of "passwordHash" from the existing `users.json` into the new `users.json` file.
 - Save the modified, new `users.json` in directory `kubernetes/config/users/users.json`.
10. Run the setup using the following command:


```
sudo ./setup.linux.x64 -u admin -p admin -kubernetes=true -k3s
-kubeconfig=/etc/rancher/k3s/k3s.yaml -namespace=bfc -deploy
```

3.3.3 BFC client update

You can perform the update of a BFC client on machines that are already connected.

The configuration of the BFC client is not changed by the update.

This procedure is identical for all SINUMERIK control variants (HMI-Advanced / Operate). You do not need to distinguish between different control system types.

Procedure

1. Open the "Commissioning" area, which displays the current version of the BFC client of the affected machine.



2. Click the button  to transfer a corresponding ZIP file (BFC-Client-xx.xx.xx.zip) with the new version to the machine.
The ZIP file can be obtained through PridaNet. It is also part of the software delivery of BFC Gateway.

Name	Type	Version	Time difference	Messages last minute	State ↑	Description
SINUMERIK	BFC Client	02.10.00.00	On On Os	4	ok (online)	Version: 02.12.00.09 will be installed after SINUMERIK reboot

3. After transferring the ZIP file to the machine, you can see which version of the BFC client is currently installed on the machine in the "System State > Clients" area. The new version is displayed in the "Description" column.
Restart the machine to activate the new BFC client version on the machine. You can thus decouple the activation of a version transferred to the machine from the transfer in terms of time.

Note

In the same way, it is possible to downgrade a machine to an older BFC client version.

3.4 Contacting the hotline

Use the service request on the Internet page "Industry Online Support" to contact the hotline.

Note

Maintenance contract

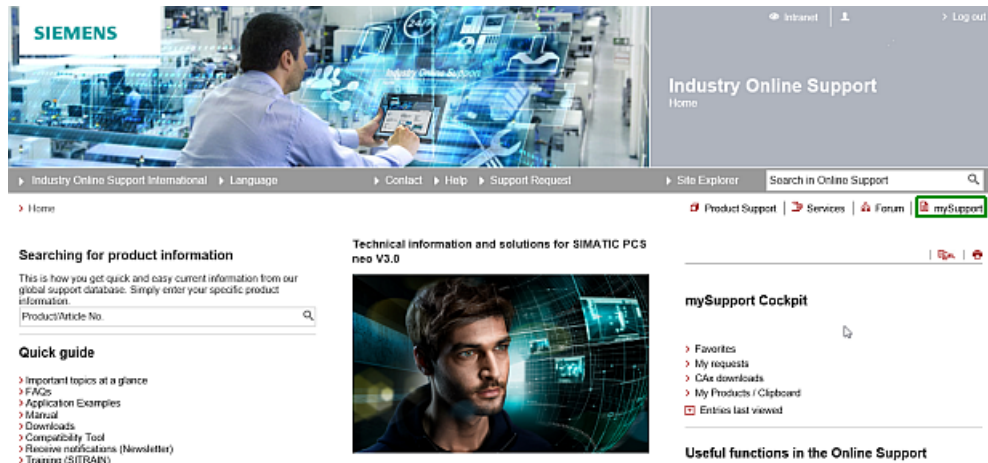
Please note that to obtain support through the hotline, it is necessary to conclude a maintenance contract (Connectivity Maintenance BF, article number: 9MC1110-1PR00-0AA5).

Requirement

You must register/log in to be able to use the "Industry Online Support" website.

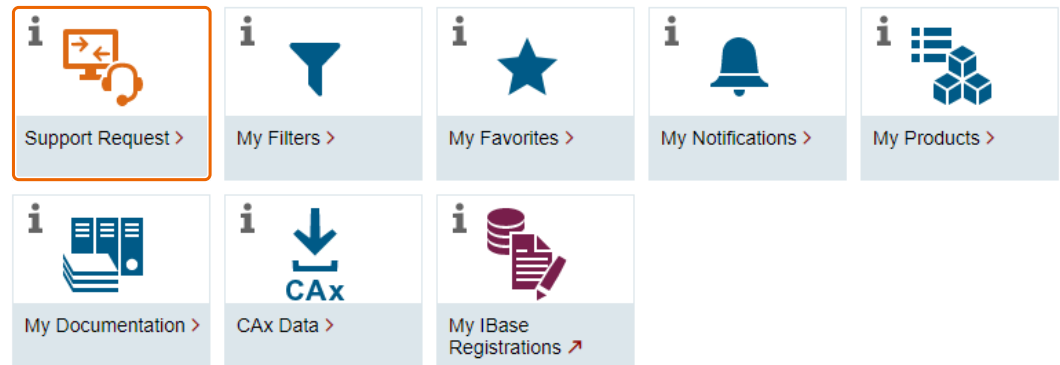
Creating a service request

1. Open this Link (<https://support.industry.siemens.com/cs/ww/en/>) to open the "Industry Online Support" website.
2. "Industry Online Support" landing page opens.
Click on "mySupport".



3. Window "mySupport Links and Tools" opens.
Select "Support Request"

mySupport Links and Tools



4. Input window "Support Request" opens.
Click on "New request".

Support request

Search for request number, product or subject

Status

Siemens staff only
Note: [Support request ID: 2100 - Industry Support Siemens](#)
Displays only those support requests which indicate you as the contact person.

3.4 Contacting the hotline

5. The "Create support request" input window opens.
 - Perform a product search using the term "BFC Brownfield Connectivity - Gateway".
 - Select the product.
 - Click on "search".

Product search

About which product do you have a technical question?

BFC Brownfield Connectivity Services Find

Select one of the following products

☐ Digital Services

☒ BFC Brownfield Connectivity Services GFC6

6. A new page opens.

Formulate your request to the hotline on the following pages.
Always specify the BFC client and BFC Gateway version.
The hotline will then immediately process your request and contact you.

Requirement

4.1 Specialist know-how

Specialist know-how is required in the following areas in order that installation and configuration can be professionally performed:

- Windows, Linux, Linux console
- Docker
- Kubernetes
- WinSCP, PuTTY
- http, https, TLS, SSH
- Handling certificates
- SINUMERIK 840D/828D/ONE: Using the service mode
- SINUMERIK 840D/828D/ONE: Creating commissioning archives
- SINUMERIK 840D/828D/ONE: Creating system backups
- MindSphere
- MindSphere API and Fleet Manager
- MindSphere application "SINUMERIK Service Assistance"

4.2 General conditions

The following constraints apply when using the BFC Gateway:

- Ensure that all of the devices, machines, PCs and higher-level customer and/or cloud systems involved are networked and ready for operation.
Network requirements: At least 100 Mbit/s full-duplex
- Provide a specification for all of the device addresses to be read for all of the required data and operating states as follows:
 - All of the required variables have been defined and documented.
 - If you need support with defining the necessary variables, you can order the Application Consulting BF (article number: 9MC1110-1PR00-0AA7).
- We strongly recommend before installing the software that a malware and/or virus check is carefully performed for every device to be networked.
- Note that the system time of all of the devices involved must be synchronized.
Synchronization can be performed by connecting to an NTP time server or by manually entering and updating, for example.
- Provide a PC with the appropriate performance with a released Linux distribution to install the BFC Gateway. When selecting the PC, take into account the installation requirements. If necessary, the extensions must be scaled using additional hardware.

4.3 System requirements

- The target system must have an Internet connection while installing and commissioning the BFC Gateway.
- For the duration of the warranty time we recommend a remote and SSH access to the gateway PC for maintenance and servicing. Carefully ensure that the machines can access ports 1883 and 8883 at the gateway PC.
If you require maintenance and care by Siemens experts after the warranty period or are unable or unwilling to provide remote access, cRSP (Common Remote Service Platform) can enable secure access to the BFC gateway. You can find out about the various remote services (<https://support.industry.siemens.com/cs/sc/2281/remote-services-for-process-automation?lc=de-DE>). The implementation process is described in the cRSP Remote Collaboration Regulations. You can also contact the DI CS SD Remote Collaboration Team about this. You will find more detailed information in the Siemens Industry Online Support (SIOS).
- At the time of commissioning, provide the appropriate access authorizations to install the BFC client on machines equipped with SINUMERIK 840D/840D sl/828D.
Access authorizations include, for example, protection levels, logins and passwords. When required, obtain this information from the machine OEMs.
- Because of the growing use of resources (hard disk space), you should integrate the lower-level hardware or virtual machine of the BFC into the customer's IT monitoring.

4.3 System requirements

4.3.1 BFC gateway

Install the software on an appropriate computer with a Linux operating system with the appropriate processing performance.

Note

Kubernetes cluster

You will find the system requirements for operating BFC in a Kubernetes cluster in Installing BFC Gateway in a Kubernetes cluster (Page 71).

To prevent data loss in the event of a power failure, we recommend the use of an uninterruptible power supply (UPS).

Perform a data backup on a regular basis.

Note

Reinstallation

If the TCP/IP address of the computer on which the BFC Gateway is installed changes, then the BFC Gateway must be reinstalled.

This is especially the case if the BFC Gateway was installed in a virtual machine, and this virtual machine is copied or shifted.

Note**Installation in a virtual machine**

Contact the hotline if you wish to install the software in a virtual machine.

Note**Hard disk memory**

Only use the SSD hard disk memory for installation and operation.

Minimum system requirements

The following minimum system requirements (valid for up to 10 devices to be connected) are applicable for the BFC Gateway.

Parameters	Value
CPU kernels	4
CPU threads	4
CPU basis frequency	1.9 GHz
RAM	8 GB
Free hard disk space (SSD)	480 GB
Free network interfaces	2 (1 Gbit/s)

Supplementary system requirements – high-frequency data acquisition

The BFC function "High-frequency data acquisition" allows you to acquire data with a high clock rate from a SINUMERIK control. You will find further information on this in: Configuring high-frequency data acquisition (Page 340).

When high-frequency data acquisition is activated, additional system resources are required on the BFC Gateway. The following additional resources are required depending on the configured number of variables for the high-frequency data acquisition.

Parameters	10 variables	20 variables
CPU performance	400 MHz	600 MHz
RAM	50 MB	50 MB
Network speed	Can be neglected for the Gigabit communication link that is required	
Hard disk speed	Can be neglected for the SSD that is required	
Memory required per hour	1 GB	2 GB

Note

The overall CPU performance is obtained from:

Number of kernels x CPU basis frequency

4.3 System requirements

Example:

If you have activated high-frequency data acquisition for five machines (for 10 variables), then the following additional resources are required:

- CPU performance: $5 \times 400 \text{ MHz} = 2 \text{ GHz}$
- RAM: $5 \times 50 \text{ MB} = 250 \text{ MB}$
- 5 GB HD space per hour = 120 GB per day

This means that the CPU must provide 2 GHz more overall CPU performance.

Selecting an industrial PC

The following tables list as example a selection of industrial PCs as target platform for installing the gateway.

Devices*	SIMATIC IPC	Processor	RAM	SSD	Article number
Up to 10	427E (box PC)	Intel® Core i5-6442EQ	8 GB	480 GB	6AG4141-5BB00-0GA0
Up to 30	427E (box PC)	Intel® Xeon E3-1505L	8 GB	480 GB	6AG4141-7AB00-0GA0
Up to 60	627E (box PC)	Intel® Core i7-8700	32 GB	960 GB	6AG4131-3GD30-8AA0
Up to 60	647E (rack PC, 19", 2HE)	Intel® Xeon E-2176G	32 GB	960 GB	6AG4112-3KR03-0XX0

* The CPU performance depends on the configured client as well as the configured gateway(s). If necessary, using the same systems, the required CPU performance must be provided through scaling.

The following CPU architectures are supported:

- AMDx64

The following non-commercial Linux distributions are supported:

- Debian 10
- Debian 9
- CentOS 7

The following chapters provide information regarding the system requirements of the components that are used.

The hotline is available if you have questions or require more information.

Supported web browsers

The following web browsers are supported for configuring the BFC Gateway:

- Mozilla Firefox Version 91 or higher
- Google Chrome Version 100 or higher

Microsoft Internet Explorer and Edge are not supported.

Network for internal communication

Within the BFC Gateway a network for internal communication is configured on the basis of Docker.

As a minimum, the Docker version should support docker-compose file format version 3.3 and use a docker engine from version 20.10.8 and higher.

The network mask 172.18.0.0/16 is the default mask for this internal network.

Please note the following:

- Avoid any collisions on the BFC Gateway between the network masks of the Docker network within BFC and the configured communication networks in the customer network.
- If the Docker network collides with another network configured on the BFC Gateway host system, you must reconfigure the Docker network.

Note

You will find more information in the Docker Online Help (<https://docs.docker.com/engine/tutorials/networkingcontainers/>).

To operate BFC in swarm mode with several physical or virtual nodes, you must configure the Docker environment according to the following instructions:

- Creating a Docker swarm (<https://docs.docker.com/engine/swarm/swarm-tutorial/create-swarm/>)
 - Adding nodes (<https://docs.docker.com/engine/swarm/swarm-tutorial/add-nodes/>)
 - Scaling a service (<https://docs.docker.com/engine/swarm/swarm-tutorial/scale-service/>)
-

Overview of system limits (quantity framework)

BFC Gateway supports up to 60 connected clients (BFC devices) by default.

Up to 50 data points are supported per BFC device, divided into "datasets".

The shortest interval for reading datasets is 200 ms.

Empirical values for tests

The following empirical values were determined when performing tests with the hardware example listed above (IPC427E with Intel® Core i5-6442EQ):

Client / gateway	Transmission rate [data points/seconds]
BFC client	250
BFC driver (all)	250
MindSphere gateway	500
AMP gateway	40
OPC UA server	250 per client

All MindSphere and AMP gateways back up data for ~90 days up to a limit of ~5 GB or ~150,000,000 data points. The computer hard disk must have the appropriate size.

4.3 System requirements

4.3.2 BFC apps

The BFC Gateway enables the installation of applications. Currently, the following applications are available for installation:

Application	Description
bfc-analytics	BFC Analytics offers a solution to visualize data and machine tool-related KPIs that give insights about production and machine conditions.

For more information on additional system requirements, refer to the documentation of the respective application. Any system requirements of applications must be added to the basic system requirements.

4.3.3 BFC client

The BFC client for SINUMERIK is the only client that is directly installed on the SINUMERIK machine control system.

- Before starting installation, ensure that the machine in your network can access the BFC gateway.
- For each machine, check the values to be read from the machine.

Additional information on the operating software is provided in:

- SINUMERIK Operate Commissioning Manual (<https://support.industry.siemens.com/cs/ww/en/view/109769186>)
- Equipment Manual for SINUMERIK 828D PPU and components (<https://support.industry.siemens.com/cs/ww/en/view/109763414>)
- Commissioning Manual SINUMERIK 840Di sl / 840D sl / 840D, Base Software and HMI-Advanced (<https://support.industry.siemens.com/cs/ww/en/view/109310641>)

4.3.3.1 Hardware and operating software

The following tables provide an overview of the hardware and operating software required for SINUMERIK control systems.

SINUMERIK 840D - HMI-Advanced

BFC client	Operating software	Hardware	Operating system
02.15.00.00	6.1	PCU 50.1	Windows NT
	6.2	PCU 50.2	Windows XP
	6.3	PCU 50.3	
	6.4	PCU 50.5	
	7.1		
	7.2		
	7.3		
	7.5		
	7.6		

SINUMERIK 840D - HMI-Advanced for retrofit

BFC client	Operating software	Hardware	Operating system
02.15.00.00	6.5 for retrofit 7.7 for retrofit	IPC427D	Windows 10

SINUMERIK 840D - MMC103*

BFC client	Operating software	Hardware	Operating system
02.15.00.00	5.3	MMC103	Windows 95

* No general release for MMC103 / Windows 95

Contact your regional Siemens sales partner for additional information or project-specific solutions.

SINUMERIK 840D - SINUMERIK Operate (PCU/TCU)*

BFC client	Operating software	Hardware	Operating system
02.15.00.00	2.6 2.7 4.5 4.7 4.8 4.9	PCU 50.3 PCU 50.5	Windows XP Windows 7 Windows 10

* No general release for operation on an IPC

Contact your regional Siemens sales partner for additional information, other versions of the operating software or project-specific solutions.

SINUMERIK 840D - SINUMERIK Operate (NCU/TCU)

BFC client	Operating software	Hardware	Operating system
02.15.00.00	2.6 2.7 4.5 4.7 4.8 4.9	NCU7x0.2 NCU7x0.3	Linux

Contact your regional Siemens sales partner for additional information, other versions of the operating software or project-specific solutions.

SINUMERIK 828D - SINUMERIK Operate (PPU)

BFC client	Operating software	Hardware	Operating system
02.15.00.00	4.5 4.7 4.8	PPU2xx3 PPU2xx4	Linux

4.3 System requirements

1:n connections

Using the BFC client on control systems equipped with several NCUs has not been released.

Contact your regional Siemens sales partner for additional information or project-specific solutions.

SINUMERIK ONE - SINUMERIK Operate (NCU/TCU)

BFC client	Operating software	Hardware	Operating system
02.15.00.00	6.13 6.20	NCU1760	Linux

DMG CELOS

BFC client	Operating software	Hardware	Operating system
02.15.00.00	4.9 preliminary CELOS V06.34.22.1759	IPC627D	Windows 7 Windows 10

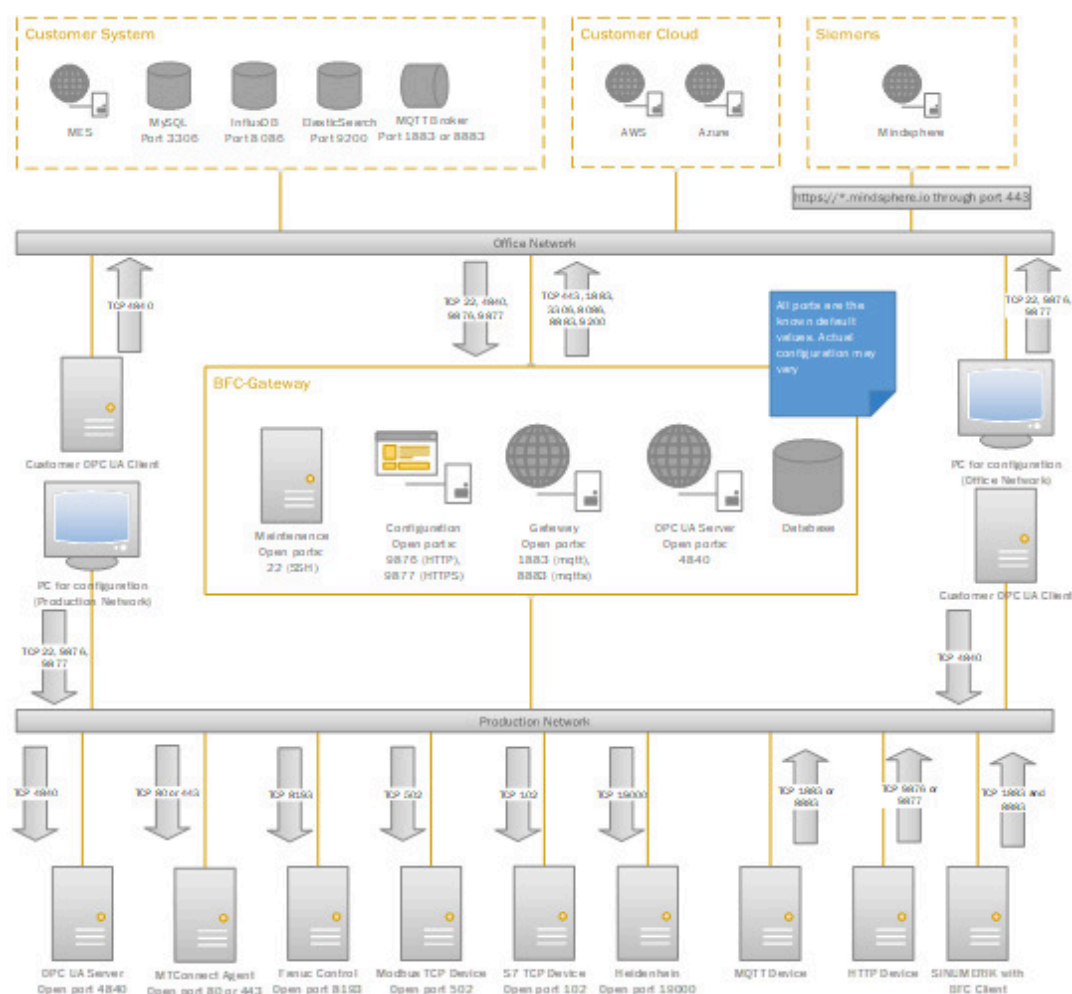
Note

You must reinstall the BFC client for each CELOS update.

4.3.3.2 Network

The system topology with the ports used is shown in the following figure.

Activate the following ports in the network. Before you do this, talk with the IT person responsible for the network.



Open BFC gateway ports for incoming communication

Port	Type	Protocol	Usage	Coded	Description
22	TCP	SSH	Commissioning/updates	Yes	SSH access for commissioning and system updates
1883	TCP	MQTT	Commissioning	No	Standard MQTT port for the commissioning of BFC clients
4840	TCP	OPC UA	Data forwarding	Yes	Standard OPC UA server port. Is used to present collected data.
8883	TCP	MQTT	Data acquisition	Yes	Standard MQTTS port for data acquisition from BFC clients
9877	TCP	HTTPS	Configuration & HTTP REST Client & WebDAV	Yes	HTTPS WebUI to configure the BFC Gateway and file functions via WebDAV

Note

Ports during operation

In operation, the BFC Gateway does not automatically open additional ports.

4.3 System requirements

Outgoing communication to the office network

Port	Type	Protocol	Usage	Coded	Description
443	TCP	HTTP	Data forwarding	Yes	Standard HTTPS port used to send data to upstream systems (e.g. MindSphere)
1883	TCP	MQTT	Data forwarding	No	Standard MQTT port for sending data to a customer MQTT broker
3306	TCP	MYSQL	Data forwarding	Yes/No	Standard MySQL server port for sending data to a customer MySQL database
3560	TCP	HTTP	Data forwarding	No	HTTP interface of the Analyze MyPerformance (AMP) server
8086	TCP	HTTP/S	Data forwarding	Yes/No	Standard InfluxDB HTTP service port to send data to a customer Influx database
8883	TCP	MQTT	Data forwarding	Yes	Standard MQTTS port for sending data to a customer MQTT broker via an encrypted connection
9200	TCP	HTTP	Data forwarding	No	Standard Elasticsearch HTTP service port to send data to a customer Elasticsearch database

Outgoing communication to the production network

Port	Type	Protocol	Usage	Coded	Description
80	TCP	HTTP	Data acquisition	No	Standard HTTP port for connection to MTConnect agents or HTTP-REST client
102	TCP	S7 comm	Data acquisition	No	Standard port ISO over TCP to SIMATIC control systems
443	TCP	HTTPS	Data acquisition	Yes	Standard HTTPS port for connection to MTConnect agents
500	TCP	TwinCAT	Data acquisition	No	Standard Beckhoff port for NC data
502	TCP	Modbus	Data acquisition	No	Standard Modbus port for connection to Modbus TCP devices
801	TCP	TwinCAT	Data acquisition	No	Standard Beckhoff port for PLC runtime system 1
811	TCP	TwinCAT	Data acquisition	No	Standard Beckhoff port for PLC runtime system 2
821	TCP	TwinCAT	Data acquisition	No	Standard Beckhoff port for PLC runtime system 3
831	TCP	TwinCAT	Data acquisition	No	Standard Beckhoff port for PLC runtime system 4
4840	TCP	OPC UA	Data acquisition	Yes/No	Standard OPC UA port for connection to OPC UA devices
8192	TCP	FOCAS	Data acquisition	No	Standard FOCAS port for connection to FANUC controllers
19000	TCP	LSV2	Data acquisition	No	Standard LSV2 communication port for connection to Heidenhain controls
44818	TCP	Ethernet Industrial Protocol	Data acquisition	No	Standard EIP communication port

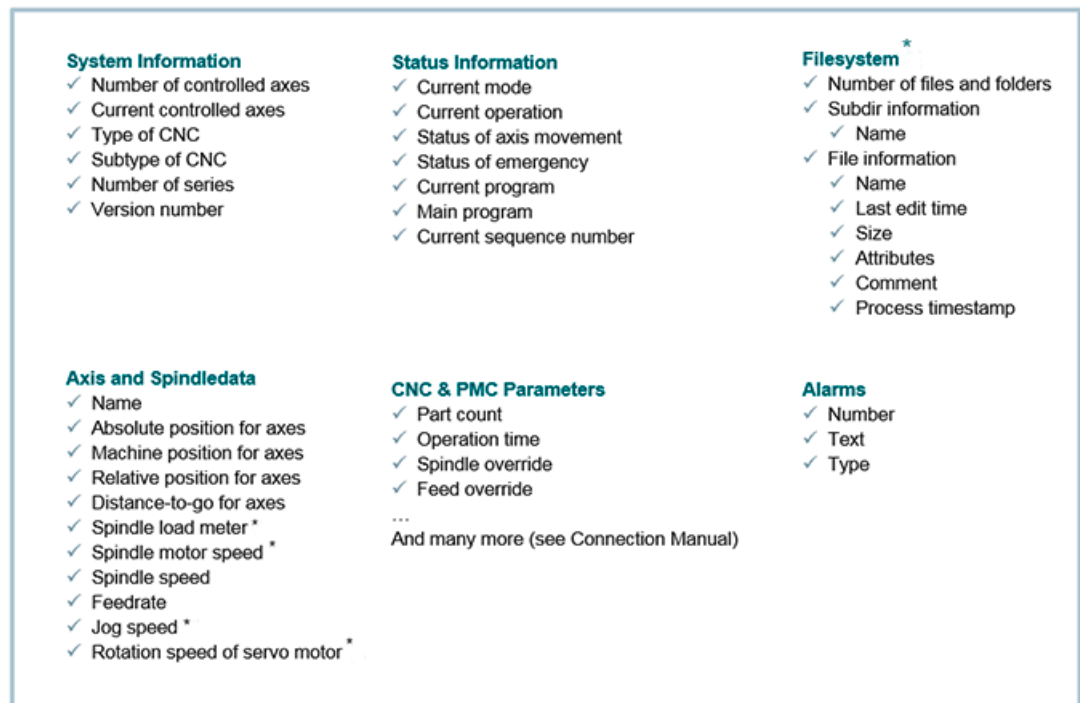
4.3.4 FANUC client

The BFC driver for machines with FANUC control supports reading and writing data, retrieving alarms, and transferring programs via the FOCAS interface.

Interface	FANUC models*
FOCAS1	0i-B/C, 15i, 16i, 18i, 21i, Mate i-D, Mate i-H
FOCAS2	0i-D/F, 30i/31i/32i-A, 30i/31i/32i/35i-B, Motion i-A

* Additional models that support FOCAS1 or FOCAS2 can possibly be released on a project-for-project basis. Contact your regional Siemens sales partner regarding verification.

The following diagram provides an overview of the data and information that can be used.



* Not available for control systems that only support the FOCAS1 interface

Requirement

- Test the connection to the FANUC control system using the "Fanuc Focas Tester" tool. More information relating to testing the connection is provided in Chapter: Check connection to FANUC control system (Page 540).
- Before starting installation, ensure that the FOCAS interface of the machine is accessible in the customer network.
- You require the following to read out data from a FANUC control system via the FOCAS interface:
 - Option "Extended driver/library function" for using the FOCAS interface must be activated
 - Ethernet connection that is ready to operate to the FANUC control system

Note

Older FANUC control systems

FANUC control systems, which only support the FOCAS1 interface, can only access a restricted functional scope in the interface.

- If you have any questions, contact your regional Siemens sales partner.
-

Note

New parameters

The application is only delivered with standard parameters.

Before you parameterize new parameters that are to be read out, we urgently recommend that you first have these parameters checked by a Siemens AG Application Center.

More information about possible NC/PMC parameters is available in the Internet:

- The "FANUC Connection Manual Function" Manual; e.g. 30i/31i/32i/35i-B, document B-64483EN-1
-

Note

API calls of the FOCAS interface

You can find information about the possible API calls of the FOCAS interface in the Appendix (Page 543).

The following must be observed when transferring programs:

- Controls that only use the FOCAS1 interface are not supported.
- On the FANUC side, depending on the control type there is a limit to the maximum size of the programs that can be transferred.
- All FANUC programs must have a certain format and contain, for example, the file name or the program number. If this format is not given, the BFC Gateway tries to emulate it so that the transferred target program is larger than the source program.
- The program file name and the name within the program data must match.
- If a FANUC program on the machine is in editing mode or is currently active, it cannot be changed with the BFC Gateway.
- The file size of all files is specified as 0 bytes in the BFC Gateway.

- The access rights to individual paths or files can be defined when configuring the FANUC client.
- The following file operations are possible:
 - Read programs
 - Write programs
 - Delete programs
 - Overwrite existing programs
 - Create directories
 - Delete directories

More information can be found in Chapter: File transfer (Page 465).

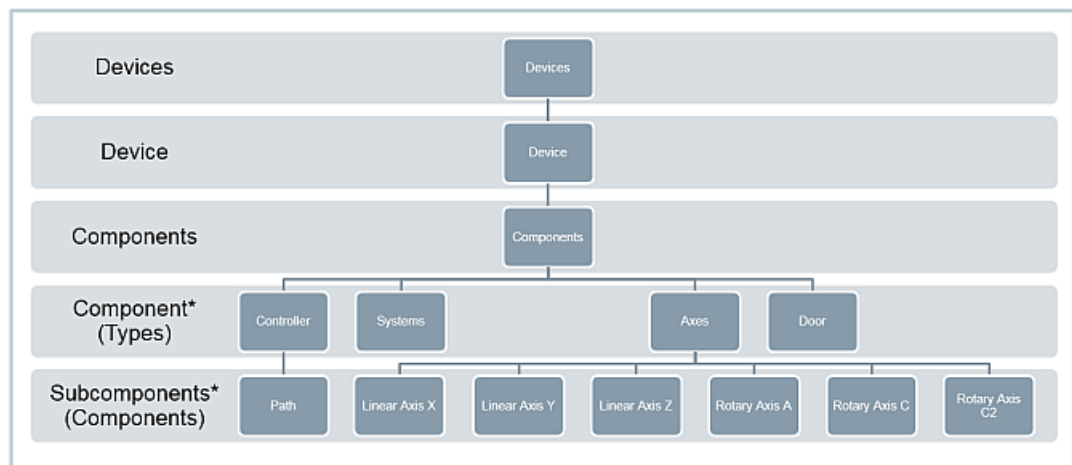
4.3.5 MTConnect client

For each MTConnect agent, from which data should be read, a BFC driver must be created for one MTConnect instance.

The BFC driver for MTConnect can only read access the MTConnect IDs of the agent.

This is a property of the MTConnect specification.

The general hierarchy of variables for MTConnect is structured as follows:



(*) The supported variables always depend on the manufacturer and the version of the MTConnect agent used by the customer.

Requirement

- Before starting the project, you require complete information relating to the following points:
 - Data regarding the versions of all MTConnect agents available locally
 - Output of the probe response/call of the locally available MTConnect Agents
`http://<MTConnect-Agent-IP>/probe`
 - Specification of the MTConnect agent available locally
The scope and format of possible control data depend on the manufacturer of the MTConnect agent and the software version of the agent.
 - Specifying the data to be read out of third-party control systems
- Before starting the installation, carefully ensure that the MTConnect interface of the machine can be accessed in the customer network.
- For each machine, determine the data made available from MTConnect agent.
To do this, from a PC in the customer network, via a web browser, call the following URL once:
 - `http://<MTConnect-Agent-IP>/probe`
 - `http://<MTConnect-Agent-IP>/current`Save the displayed result (XML) for diagnostics.
- When updating the MTConnect agent, note that the format or data made available can change.
- The function has been tested against MTConnect protocol versions 1.2 up to and including 1.8 and supports MTConnect schema version up to 1.7.0. Protocol version 1.1 is not supported.

4.3.6 Modbus client

For each device connected via Modbus TCP, you must set up an instance of the BFC driver for Modbus as a client.

Requirement

- Before starting the installation, carefully ensure that the Modbus device can be accessed in the customer network.
- For each device, check the data to be read from the device.

Specifications that are used

The Modbus client uses the following specifications:

- Connection to precisely one Modbus device
- Connection via TCP/IP
- Read and write access

- The following data types are supported:
 - Boolean
 - Integer
 - Float
- The following Modbus functions are supported:
 - (0x01) Read Coils
 - (0x02) Read Discrete Inputs
 - (0x03) Read Holding Registers
 - (0x04) Read Input Registers
 - (0x05) Write Single Coil
 - (0x06) Write Single Holding Register
 - (0x16) Write Multiple Holding Registers

4.3.7 OPC UA client

Requirement

- Before starting the installation, carefully ensure that the OPC UA server of the device can be accessed in the customer network.
- For each device, check the data to be read from the device.

Specifications that are used

The OPC UA client uses the following specifications:

- Connection to precisely one OPC UA server
- Binary OPC UA transfer protocol
- The XML-based transfer protocol is not supported.
- The connection to the OPC UA server is either encrypted or unencrypted.
- Supports the login mechanisms "Anonymous", "User name/password" and "Certificate".
- Read-only access to OPC UA nodes of the server
- The following data types are supported:
 - String
 - Float
 - Integer
 - Boolean
- Only alarms with OPC UA data types "AlarmConditionType" and "CncAlarmType" are supported.

4.3.8 S7 client

The S7 client enables the connection of the SIMATIC PLC S7 to the BFC gateway.

Requirement

- The network connection is exclusively established via PROFINET or TCP/IP networks.
For the network connection via PROFINET the PLC requires the following equipment:
 - Integrated PROFINET interface
Examples:
SIMATIC S7-300 CPU 315-2 PN/DP or SIMATIC S7-1200
SIMATIC S7-1500
 - OR -
 - Additional PROFINET communications processor
Examples:
SIMATIC CP 243/343/443
- The network addressing is done via TCP/IPv4 and uses the RFC1006 protocol.
 - The PLC must be enabled for network access.
 - Activation takes place in the configuration of the SIMATIC STEP 7 project or in the TIA Portal.
- For secure operation, you need a secure network infrastructure, since the RFC1006 protocol does not implement encrypted or signed communication. An access password set in the PLC configuration for network communication may be compromised by recording the network communication.

Addressing

- Address the CPUs of the SIMATIC controls as follows:
 - The set IP address
 - The TSAP address
The TSAP addressing is used for addressing on the backplane/backplane bus of the SIMATIC PLC and is still used compatibly in the newer PLC series with integrated PROFINET.
 - Rack and slot position
These parameters are internally converted into a TSAP address.

When addressing the PLC, you can set further optional parameters such as timeouts. However, these parameters are usually not required or their default values are sufficient in a LAN.

- Communication takes place with PDU with a pre-defined structure. The access of the S7 client takes place via a multi-variable access, which allows addressing of a maximum of 20 elements in the SIMATIC in a single access operation. This type of access leads to minimum latency and maximum performance. The maximum size of the data to be read out is limited by the fixed structure. If scalar variables such as bit, byte, word or DWord are addressed exclusively, the user data are not a limiting factor when reading. Only by addressing byte arrays, character arrays and S7 strings, can the maximum PDU user data volume be exceeded when reading.
For write jobs, the addressing and the net data (user data) are transferred to a shared PDU. As a consequence, the quantity of net data for a write job is limited. With a PDU size of 240 bytes for example (e.g. S7 300, S7 1200), a maximum of 12 values can be processed in one single write job. Details are provided in the appendix to this document. If necessary, write operations must therefore be subdivided into several substeps. You can check the PDU sizes used in the trace outputs of the S7 client. There you may also check how long the communication job was processed by the PLC.
- The addressing in the data blocks does not support symbolic addressing, but is done with byte offset addresses. For this reason, "optimized data blocks" of the SIMATIC S7-1200 and SIMATIC S7-1500 series cannot be read out.

Manual optimization of data addressing

The setup of the PDU structure means that 2 bytes are always occupied in the PDU structure, even for individual bits and bytes. If you want to read several bits of a byte or word, do not address individual bits. Address the whole word or byte and separate them in the target application or on the script level of the BFC gateway.

Examples:

- DB1.DBX0.0, DB1.DBX0.1, ... DB1.DBX0.7
Addresses the same data as DB1.DBB0, but occupies eight times the amount in the PDU. Processing in the PLC takes longer.
- DB1.DBX0.0, DB1.DBX0.1, ..., DB1.DBX1.7
Reads the same data as DB1.DBW0, but occupies sixteen times the amount in the PDU. Processing in the PLC takes longer.

To optimize the required accesses to the PLC, you may also read out several bytes as array.

Example: "E0:BYTE[8]" reads eight input bytes during one single read access.

4.3 System requirements

For addressing larger data structures in data blocks that would exceed the usable data volume of a PDU, a reading set with a single address can be read out. In this special case, the S7 client automatically detects if the data must be read out in several steps.

Example: DB1.DBB0:BYTE[256] is automatically read from the PLC by the S7 client with two separate calls when the PDU size is 240 bytes.

Performance influence on PLC cycle time

You can configure the maximum portion of the communication time of the PLC cycle in the SIMATIC STEP 7 project. An extended PLC cycle time can be observed due to the communication. If a communication job is not completed within the maximum cycle time, it is continued in the next PLC cycle.

Data consistency over several PLC cycles

Read access

The PLC communication used does not guarantee data consistency of the read values. The values that are read can originate from different PLC cycles and I/O images, since the communication does not influence the logic of the PLC user program.

Write access

It is not guaranteed that all data to be written within a PLC cycle is accepted or that the addresses at the start of the write job are written into the PLC memory before addresses at the end of the write job. If the PLC user program write accesses the same memory during a write operation, it is not guaranteed which data actually apply at the end. A coordination mechanism must be implemented for shared (common) memory areas in the PLC.

Diagnostics

After establishing the connection, diagnostic data is read out from the PLC and logged as log messages. Depending on the PLC type, only parts of the data are provided:

- Negotiated PDU size
- Information on the current access protection
- Information on the PLC, such as the MLFB no. Hardware version, serial number
- Information on the CP such as the maximum number of connections, the bandwidth of the connection, and the backplane bus
- Output of the PLC time
- Listing of the data blocks in the PLC

4.3.9 Heidenhain client

The BFC driver for machines equipped with a Heidenhain control system supports reading and writing data, retrieving alarms and transferring programs via the Ethernet. The driver supports the connection just one Heidenhain machine, but can be started several times.

The following Heidenhain machines are supported:

- Heidenhain TNC 320, TNC 360, TNC 426, TNC 430, iTNC 530, TNC 620, TNC 640
- DataPilot CP 620, DataPilot CP 640, DataPilot MP 620, DataPilot 4110, DataPilot 4290 among others.

All Heidenhain functions were tested against NC software version 340494-07 on an iTNC530. Deviations may occur with newer software versions or other control models.

Not all machines offer the same scope of functions.

If another client is already connected to the Heidenhain control system, functions that are connected with memory accesses may fail. This is because with the Heidenhain control system, access to this area is exclusive.

Preconditions

- Before starting the installation, make sure that the Heidenhain machine in the customer network is operational and can be accessed.
- To read out data from the Heidenhain control system, you may have to activate external access as follows:
 - In the machine or MOD operating mode, press the "External Access On/Off" softkey.
 - If the "External Access On/Off" softkey is not available, the entry `"REMOTE.LOCKSOFTKEYVISIBLE = YES"` must be available in the `OEM.SYS` configuration file. The entry must not be commented out by a semicolon.
 - A password may be set in the `OEM.SYS` configuration file, which is required for data access (parameters `PLCPASSWORT`, `REMOTE.PLCPASSWORTFORCED` and `REMOTE.PLCPASSWORTNEEDED`).
 - In the `OEM.SYS` configuration file, access to machine parameters can also be protected by a password (`MPPASSWORD` parameter).
 - With newer NC software versions from 340 49X-03 or higher, you can prohibit access by certain clients.
To do this, the entry `"REMOTE.PERMISSION"` must be present in the `TNC.SYS`. This entry contains a list of IP addresses or host names of clients for which remote access is permitted. Add the IP address or the host name of the BFC as shown in the following example.
Example: `REMOTE.PERMISSION = PC123;192.168.0.92`
 - A password may be set in `TNC.SYS` that is required for access to certain areas (parameters `REMOTE.TNCPASSWORD` and `REMOTE.TNCPRIVATEPATH`).
 - An SE Linux firewall is integrated in NC software versions 60642x and higher. In addition to `OEM.SYS` and `TNC.SYS`, the Linux firewall is another way to restrict access to the machine.
 - Access to some data areas is exclusive. If another application already has access to it and does not relinquish it (no logout from the area), no other applications can access the area.
 - As of software versions 34049X-03, the machine manufacturer has the possibility to restrict access to areas or variables. The machine manufacturer can block access in his PLC basic program. Without the machine manufacturer, you then have no possibility of reading data.

Specifications

The Heidenhain client uses the following specifications:

- Presently, reading and writing data is not supported.
- The following data types are supported:
 - Boolean
 - Integer
 - Float
 - String
 - Word
 - DWord

Note

Within the BFC Gateway, Word and DWord values are treated as integer values.

The following must be observed when transferring programs:

- Access rights to individual paths or files can be defined when configuring the Heidenhain client.
- It is not possible to transfer files larger than 10 MB.
- Heidenhain machines respond differently depending on the data type transferred. It cannot be ensured that all data types can be transferred.
- If a Heidenhain program on the machine is in the edit mode or is currently active, it cannot be changed via the BFC Gateway.
- The machine manufacturer has the option of inhibiting file operations.
- Files and directories can be protected on the Heidenhain machine side. In cases such as these, overwriting is not possible.
- The following file operations are possible:
 - Read programs
 - Write programs
 - Delete programs
 - Rename programs
 - Overwrite existing programs
 - Create directories
 - Delete empty directories
- It is not possible to rename directories.
- Directories that contain files or subdirectories cannot be deleted.

More information can be found in Chapter: File transfer (Page 465).

4.3.10 Beckhoff client

The BFC driver for machines with Beckhoff controls supports reading and writing of data via Ethernet.

A connection to a Beckhoff machine can only be established by the driver. However, the driver can be started multiple times and access multiple machines in parallel instances.

All Beckhoff models with TwinCAT 2 or TwinCAT 3 are supported.

Note

Writing and reading of NC programs is not supported.

Reading of alarms is not supported, but can be implemented project-specifically if necessary.

Not all control systems offer the same range of features.

Data types

The following Beckhoff data types are supported:

- BOOL
- BYTE
- WORD
- DWORD
- LWORD
- SINT
- USINT
- INT
- UINT
- DINT
- UDINT
- LINT
- ULINT
- REAL
- LREAL
- STRING
- WSTRING
- TIME
- TIME_OF_DAY
- DATE
- DATE_AND_TIME

4.3 System requirements

Note

Arrays of the respective data types are supported.

Note

Encrypted ADS is not supported.

Requirements

Before starting the installation, make sure that the Beckhoff machine in the customer network is operational and can be accessed.

4.3.11 Ethernet IP client

The Ethernet IP client (EIP client) enables the connection of "Allen Bradley", "Rockwell", and "Omron" control systems to the BFC gateway.

Requirements

The network connection is exclusively established via TCP/IP networks.

For communication, the control system requires an Ethernet port and support for the Ethernet Industrial Protocol.

The EIP client establishes a TCP connection to the control system, the default port is 44818.

An IP address must be configured on the control system and communication to the control system must be possible.

The operator is responsible for a secure network infrastructure.

Supported control system families

The following control system families can be connected to BFC with the EIP client. They use different message formats of EIP communication.

- CIP-EtherNet/IP TCP/IP
 - Rockwell/Allen-Bradley ControlLogix(tm) PLCs
 - Rockwell/Allen-Bradley CompactLogix(tm) PLCs
 - Rockwell/Allen-Bradley Micro 850/870 PLCs
 - Omron NX/NJ PLCs
- PCCC-EtherNet/IP TCP/IP
 - Rockwell/Allen-Bradley MicroLogix PLCs
 - Rockwell/Allen-Bradley SLC 500 PLCs
 - Rockwell/Allen-Bradley PLC/5 PLCs

Note**Alternative to the EIP client**

As an alternative to the EIP client, some Rockwell/Allen-Bradley control systems (e.g.: Micro 830) can be connected to BFC with the ModBus client.

Note**Supported data types**

The supported data types and data structures are documented under EIP client (Page 582).

4.3.12 Omron client

The BFC driver for Omron machines supports reading and writing of data over Ethernet.

The driver can only establish a connection to one Omron machine. However, the driver can be started multiple times and access multiple machines in parallel instances.

The driver supports all Omron models that are capable of the FINS protocol.

Note

- Writing and reading of NC programs is not supported.
 - Reading of alarms is not supported.
 - The following data types are supported:
 - BOOL
 - WORD
 - STRING
 - Not all control systems offer the same range of features.
-

Requirements

- Before starting the installation, make sure that the Omron machine in the customer network is operational and can be accessed.
- Make sure that the network, node, and device identifiers are known.

4.3.13 HTTP script client for HP 3D printers

The HTTP script client is supplied with a preconfigured JavaScript for the HP 3D API version 1.2 and higher.

4.3 System requirements

The preconfigured JavaScript supports HP Jet Fusion 3D printers of the series (<https://developers.hp.com/3d-printing-apis>):

- 5200
- 4200
- 500
- 300

Requirement

- The printer must be connected to a locally installed HP SmartStream 3D Command Center Version 3.7. Direct access to the printer is not provided. In the client, therefore, the URL of the HP SmartStream 3D Command Center and the printer name configured there are always used.
- To access the HP SmartStream 3D Command Center, you need HP's access credentials. To use https encryption securely, you need the CA certificates for the SmartStream 3D Command Center or the included device proxy. On the local network, you can also use the pre-installed self-signed certificates of the HP SmartStream 3D Command Center if you activate the "Skip verification of CA cert" option.
- Technically, the data that is available according to the documentation (<https://developers.hp.com/3d-printing-apis>) and which the printer actually provides can be read out. Write access to HP 3D printers is not supported.

4.3.14 SFTP client

The SFTP client allows files to be accessed on devices that support the SFTP protocol (SFTP server).

All file and directory operations (reading, writing, renaming, deleting) are supported.

Authorization to access individual directories can be individually defined in the BFC Gateway. More information can be found in Chapter Configuring access rights (Page 467).

You can access SFTP devices via the WebDAV interface of the BFC Gateway.

Preconditions

- Before starting the installation, ensure that the SFTP server in the customer network is operational and can be accessed.
- Ensure that the SFTP server access the data are known.

Installing the BFC gateway

This chapter describes the new installation of the BFC Gateway.

Note**Procuring the BFC Gateway**

As an end user, contact the sales representative responsible for you about procuring the BFC Gateway software.

As a Siemens employee, you can obtain the BFC Gateway via PridaNet.

Note**Installation scope of the BFC Gateway installation**

By default, when you install the BFC Gateway under Docker Swarm or Kubernetes, the BFC Analytics application is also installed. Please note the additional system requirements due to BFC Analytics. You can disable the installation of the BFC Analytics application during setup by setting the appropriate installation parameter (`-excluded-apps "bfc-analytics"`).

5.1 Requirement

Installing the BFC gateway involves the following requirements:

- The system requirements for the BFC gateway are satisfied.
Additional information is provided in Chapter: BFC gateway (Page 40).
- An interconnect connection is available.
- A training course is required to be able to perform the installation. Contact your regional Siemens sales partner for more detailed information about training courses.
- In addition, the following requirements apply to the Linux system:
 - Root account
 - Internet access
 - SSH access

Note**Create a new user**

When performing the installation, we recommend that you create a new user, which was accepted in the sudo user group.

User "apc" is used in this documentation.

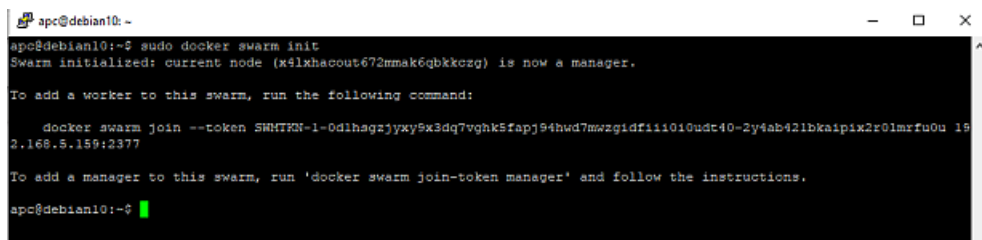
5.2 Installing the docker CE

Installation of Docker CE:

- Procedure 1 describes the installation according to the Docker installation guide.
- Procedure 2 describes the installation using the installation script on a Debian system.

Procedure 1

1. Install Docker CE according to the official Docker instructions.
More information on installing can be found here. (<https://docs.docker.com/engine/install/>)
2. Activate "Docker swarm-Mode":
"sudo docker swarm init"

A terminal window screenshot showing the execution of 'sudo docker swarm init'. The output indicates that the swarm is initialized and the current node is now a manager. It provides a token for adding workers and instructions for adding managers.

```
apc@debian10: ~  
apc@debian10:~$ sudo docker swarm init  
Swarm initialized: current node (x4lkhacout672mmak6gbkkczg) is now a manager.  
  
To add a worker to this swarm, run the following command:  
  
    docker swarm join --token SWMTKN-1-0dlhagzjyxy9x3dq7vghk5fapj94hwd7mwzgidfii0i0udr40-2y4ab42lbkaipix2r0lmrfu0u 19  
2.168.5.159:2377  
  
To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.  
apc@debian10:~$
```

Procedure 2

You can find the installation script here (<https://get.docker.com>).

1. Ensure that there is an Internet connection.
2. Log in as user "apc".

Note

Command for installation behind a proxy server

Execute the following command for installing behind a proxy server:

```
sudo curl -x '<Proxy Server Protocol>://<Proxy Server  
Address>:<Proxy Port>/' -fsSL https://get.docker.com | sh
```

3. Execute the following commands:

```
sudo curl -fsSL https://get.docker.com | sh
```

The docker software is installed.

```
apc@debian-10:~$ sudo curl -fsSL https://get.docker.com | sh
# Executing docker install script, commit: f45d7c11389049ff46a6b4d94e0dd1ffebca32c1
Warning: the "docker" command appears to already exist on this system.

If you already have Docker installed, this script can cause trouble, which is
why we're displaying this warning and provide the opportunity to cancel the
installation.

If you installed the current Docker package using this script and are using it
again to update Docker, you can safely ignore this message.

You may press Ctrl+C now to abort this script.
+ sleep 20
+ sudo -E sh -c apt-get update -qq >/dev/null
+ sudo -E sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-cert
ificates curl >/dev/null
+ sudo -E sh -c curl -fsSL "https://download.docker.com/linux/debian/gpg" | apt-key add -qq - >/d
ev/null
Warning: apt-key output should not be parsed (stdout is not a terminal)
+ sudo -E sh -c echo "deb [arch=amd64] https://download.docker.com/linux/debian buster stable" >
/etc/apt/sources.list.d/docker.list
+ sudo -E sh -c apt-get update -qq >/dev/null
+ [ -n ]
+ sudo -E sh -c apt-get install -y -qq --no-install-recommends docker-ce >/dev/null
+ sudo -E sh -c docker version
Client: Docker Engine - Community
 Version:           19.03.4
 API version:       1.40
 Go version:        go1.12.10
 Git commit:        9013bf583a
 Built:             Fri Oct 18 15:52:16 2019
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.6
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.12.16
  Git commit:       369ce74a3c
  Built:            Thu Feb 13 01:26:32 2020
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.2.10
  GitCommit:        b34a5c8af56e510852c35414db4c1f4fa6172339
 runc:
  Version:          1.0.0-rc8+dev
  GitCommit:        3e425f80a8c931f88e6d94a8c831b9d5aa481657
 docker-init:
  Version:          0.18.0
  GitCommit:        fec3683
If you would like to use Docker as a non-root user, you should now consider
adding your user to the "docker" group with something like:

    sudo usermod -aG docker apc

Remember that you will have to log out and back in for this to take effect!

WARNING: Adding a user to the "docker" group will grant the ability to run
containers which can be used to obtain root privileges on the
docker host.
Refer to https://docs.docker.com/engine/security/security/#docker-daemon-attack-surface
for more information.
apc@debian-10:~$
```

4. Add user "apc" to user group "docker" using the following command:

```
sudo usermod -aG docker apc
```

5. Test the installed docker software by executing container "hello-world":

```
sudo docker run hello-world
```

```
apc@debian10:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:9572f7cdcee8591948c2963463447a53466950b3fc15a247fcad1917ca215a2f
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

apc@debian10:~$
```

6. If the test was successful, remove container "hello-world" from the system:

```
"sudo docker system prune -a"
```

```
apc@debian10: ~
apc@debian10:~$ sudo docker system prune -a
WARNING! This will remove:
 - all stopped containers
 - all networks not used by at least one container
 - all images without at least one container associated to them
 - all build cache

Are you sure you want to continue? [y/N] y
Deleted Containers:
0d7bd79d51f6ffab7f3c61f39bce5c853af2971542cf3fe46acb0e9f7a5289a4

Deleted Images:
untagged: hello-world:latest
untagged: hello-world@sha256:9572f7cdcee8591948c2963463447a53466950b3fc15a247fcad1917ca215a2f
deleted: sha256:fce289e99eb9bca977dae136fbc2a82b6b7d4c372474c9235adc1741675f587e
deleted: sha256:af0b15c8625bb1938f1d7b17081031f649fd14e6b233688eea3c5483994a66a3

Total reclaimed space: 1.84kB
apc@debian10:~$
```

7. Activate "Docker swarm-Mode":

```
"sudo docker swarm init"
```

```
apc@debian10: ~
apc@debian10:~$ sudo docker swarm init
Swarm initialized: current node (x4lxhacout672mmak6gbkkczg) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-0dlhagzjyxy9x3dq7vghk5fapj94hwd7mwzgidfii1010udc40-2y4ab42lbkaipix2r0lmrfu0u 19
    2.168.5.159:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

apc@debian10:~$
```

5.3 Installing the BFC gateway

Install the BFC Gateway based on the following installation steps.

Parameters

The following table describes the installation command parameters:

```
sudo ./setup.linux.x64 -deploy -u admin -p mypass
```

If special characters are used, the password must be set in quotation marks.

Example: ``Strong#-Pass!$?``

```
sudo ./setup.linux.x64 -deploy -u admin -p `Strong#-Pass!$?`
```

The use of the command is described under "Procedure".

Parameters	Description
-u -p	Specification of the user data of the BFC Gateway administrator (Here, as example admin/mypass) Assign a secure password
-hosts (optional)	Using this parameter you can explicitly enter additional IP addresses or host names. If this parameter is not set, the host name and the currently configured IP addresses are used.
-addhosts (optional)	Using this parameter, additional IP addresses or host names are taken into account. If the BFC Gateway is started with this parameter, the host name of the BFC Gateway must be specified. Remark: The additional IP addresses and host names are entered, separated by a comma. Example: -addhosts 192.168.6.2,192.168.6.3
-deploy	If this parameter is specified, then the BFC Gateway is installed on the computer. If this parameter is missing, then only the configuration files are generated.
-du (optional) -dp (optional)	Using these parameters, the user data for accessing the data bus are optionally specified. If this user data is not entered, then this data is randomly generated, and displayed at the end of the installation.
-excluded-apps (optional)	This parameter optionally specifies BFC apps that should not be installed. Remark: Multiple BFC apps are specified separated by a comma. Example: "bfc-analytics"

Note

TLS certificates

During the installation, TLS certificates are generated to establish an encrypted connection. As default setting, all IP4 addresses and the host name of the Linux system are used. The host name of the BFC Gateway is also required.

- Import the TLS certificate using the browser when accessing the configuration user interface for the first time.

Procedure

1. Create a subdirectory "bfc" in the "Home" directory.
Using WinSCP, copy the content of the provide software to the created "bfc" directory.

Note

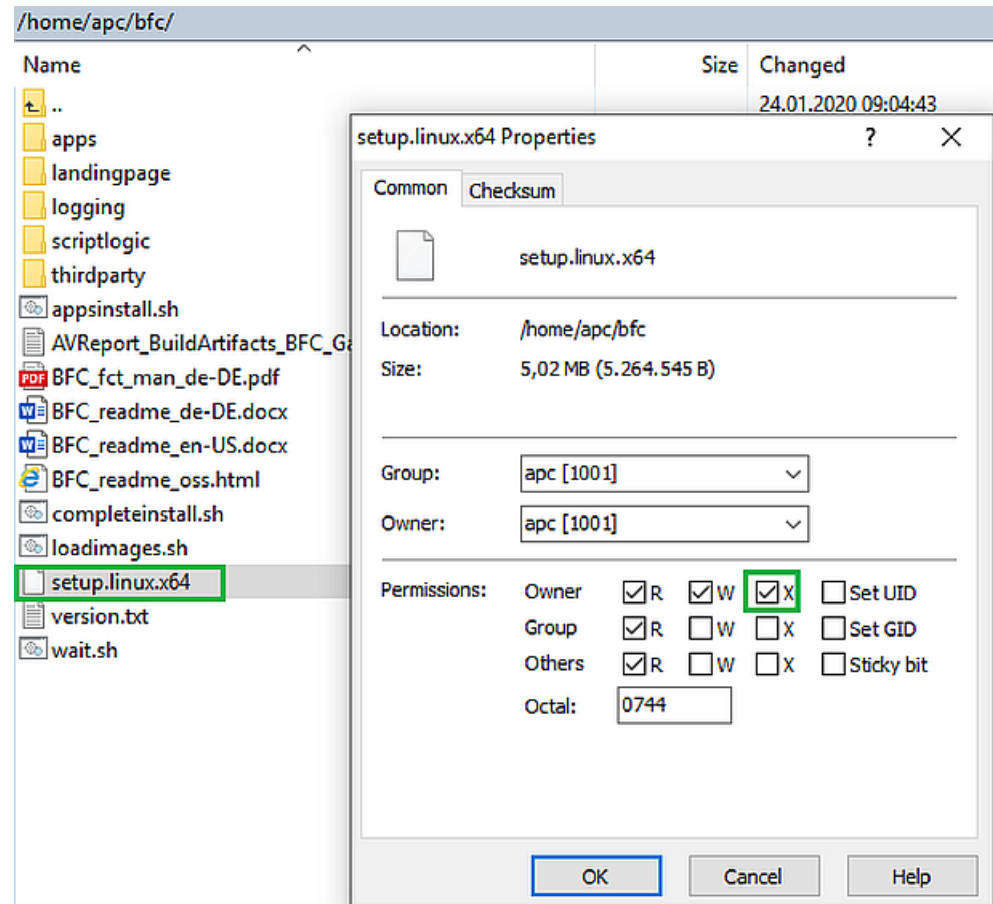
Directory name

The directory name used for the installation must not contain any space characters or special characters (examples: bfc, bfc110, bfc-gateway_v1_10, bfc.v1.10.1).

/home/apc/bfc/				
Name	Size	Changed	Rights	Owner
		24.01.2020 09:04:43	rwxf-xf-x	apc
apps		24.01.2020 09:04:59	rwxf-xf-x	apc
landingpage		24.01.2020 09:04:59	rwxf-xf-x	apc
logging		24.01.2020 09:04:59	rwxf-xf-x	apc
scriptlogic		24.01.2020 09:04:59	rwxf-xf-x	apc
thirdparty		24.01.2020 09:04:59	rwxf-xf-x	apc
appsinstall.sh	1 KB	20.01.2020 15:05:31	rw-r--r--	apc
AVReport_BuildArtifacts_BFC_Gateway_1.5.0_RC6.log	14 KB	20.01.2020 15:08:12	rw-r--r--	apc
BFC_fct_man_de-DE.pdf	2.320 KB	20.01.2020 15:05:31	rw-r--r--	apc
BFC_readme_de-DE.docx	101 KB	20.01.2020 15:05:31	rw-r--r--	apc
BFC_readme_en-US.docx	93 KB	20.01.2020 15:05:31	rw-r--r--	apc
BFC_readme_oss.html	1.113 KB	20.01.2020 15:05:31	rw-r--r--	apc
completeinstall.sh	1 KB	20.01.2020 15:05:31	rw-r--r--	apc
loadimages.sh	1 KB	20.01.2020 15:05:31	rw-r--r--	apc
setup.linux.x64	5.142 KB	20.01.2020 15:05:42	rw-r--r--	apc
version.txt	1 KB	20.01.2020 15:05:42	rw-r--r--	apc
wait.sh	1 KB	20.01.2020 15:05:31	rw-r--r--	apc

2. Activate the execution rights for file "setup.linux.x64".
One of two options can be selected for this purpose:

- Use WinSCP



- OR -

- Execute the following command:
`sudo chmod +x setup.linux.x64`

5.3 Installing the BFC gateway

3. Install the BFC Gateway using the following command:

```
sudo ./setup.linux.x64 -deploy -u admin -p mypass
```

Before calling the setup, switch to the directory in which you copied file "setup.linux.x64" (e.g. /home/apc/bfc).

Note**Docker images**

During the installation, docker images from "hub.docker.com" are loaded.

An Internet connection is required for this procedure. This procedure takes approximately 10 minutes.

The following diagram shows excerpts of the displays on the consoles during the installation.

```

apc@debian10: ~/bfc
INFO: 2020/01/24 10:30:10.895226 thirdparty/docker-compose.yml - generating...
INFO: 2020/01/24 10:30:10.896132 thirdparty/docker-compose.yml does not yet exist - generating new one...Done
INFO: 2020/01/24 10:30:10.896364 Used Thirdparty Configuration: (DefaultUsername:admin DefaultPassword:admin DBUsernam
e:iotserver DBPassword:473c1592-9c21-4026-84a5-ebb1c27b2dbf ControlBusUsername:control ControlBusPassword:a12d294-9a0
4-46ce-9ba2-0a72a7alcc2b DataBusUsername:iotserver DataBusPassword:675b4a5e-64a7-46a2-8356-59c9e4d6d7bb InternalUserna
me:Internal InternalPassword:7fb38448-6385-4a4d-875d-db32c33197f2 LobbyUsername:guest LobbyPassword:public EncryptedLo
bbyUsername:6a04e2eb-b05a-4889-8d89-ef32e51b5cc7 EncryptedLobbyPassword:37071aa9-6d63-4bec-9954-4123c368a6fa Registry:
apc-docker-registry.dlinkddns.com Version:latest Hosts:192.168.5.159 SplittedHosts:[192.168.5.159]
INFO: 2020/01/24 10:30:10.896566 logging/thirdparty/docker-compose.yml does not yet exist - generating new one...
INFO: 2020/01/24 10:30:10.896971 logging/thirdparty/docker-compose.yml does not yet exist - generating new one...Done
INFO: 2020/01/24 10:30:10.897153 Used LoggingThirdparty Configuration: (DBUsername:iotserver DBPassword:eb3cc8f-3e1f-
40b9-9d6d-188ca5b91c95)
INFO: 2020/01/24 10:30:10.897331 apps/infrastructure/docker-compose.yml does not yet exist - generating new one...
INFO: 2020/01/24 10:30:10.898220 apps/infrastructure/docker-compose.yml - generating new one...Done
INFO: 2020/01/24 10:30:10.898498 docker-compose.yml does not yet exist - generating new one...
INFO: 2020/01/24 10:30:10.900154 docker-compose.yml - generating new one...Done
INFO: 2020/01/24 10:30:10.900229 opcuaserver/docker-compose.yml does not yet exist - generating new one...
INFO: 2020/01/24 10:30:10.900675 opcuaserver/docker-compose.yml - generating new one...Done
INFO: 2020/01/24 10:30:10.900752 configui/settings.json does not yet exist - generating new one...
INFO: 2020/01/24 10:30:10.901074 configui/settings.json - generating new one...Done
INFO: 2020/01/24 10:30:10.901157 thirdparty/grafana/provisioning/datasources/datasource.yml does not yet exist - gene
rating new one...
INFO: 2020/01/24 10:30:10.901375 thirdparty/grafana/provisioning/datasources/datasource.yml - generating new one...Do
ne
INFO: 2020/01/24 10:30:10.901411 thirdparty/broker/definitions.json does not yet exist - generating new one...
INFO: 2020/01/24 10:30:10.901855 thirdparty/broker/definitions.json - generating new one...Done
INFO: 2020/01/24 10:30:10.901891 logging/docker-compose.yml does not yet exist - generating new one...
INFO: 2020/01/24 10:30:10.902409 logging/docker-compose.yml - generating new one...Done
INFO: 2020/01/24 10:30:10.902463 -----
INFO: 2020/01/24 10:30:10.902477 Please use the following username to connect remote clients:
INFO: 2020/01/24 10:30:10.902492 iotserver
INFO: 2020/01/24 10:30:10.902504 Please use the following password to connect remote clients:
INFO: 2020/01/24 10:30:10.902516 675b4a5e-64a7-46a2-8356-59c9e4d6d7bb
INFO: 2020/01/24 10:30:10.902529 -----
983d1f5df98c: Loading layer [=====>] 222.7kB/222.7kB
16a87bbde844: Loading layer [=====>] 5.703MB/5.703MB
0155e4b3fdcd: Loading layer [=====>] 224.8kB/224.8kB
005cb49b4eeef: Loading layer [=====>] 9.951MB/9.951MB
68d71a5fd6ea: Loading layer [=====>] 3.921MB/3.921MB
Loaded image: apc-docker-registry.dlinkddns.com/apc-koe/mta/iotserver/updateservice:latest
26cb4083ada3: Loading layer [=====>] 11.31MB/11.31MB
8de21d092324: Loading layer [=====>] 2.305MB/2.305MB
7846f5fdca7f: Loading layer [=====>] 690.7kB/690.7kB
01ef603571dd: Loading layer [=====>] 10.24kB/10.24kB
762aade59832: Loading layer [=====>] 5.12kB/5.12kB
e5c974205eed: Loading layer [=====>] 3.922MB/3.922MB
Loaded image: apc-docker-registry.dlinkddns.com/apc-koe/mta/iotserver/clientconfigurator:latest
81ba6388d273: Loading layer [=====>] 222.7kB/222.7kB
8e39f5a503b5: Loading layer [=====>] 6.286MB/6.286MB
68d71a5fd6ea: Loading layer [=====>] 3.921MB/3.921MB
bfcf670da2ee: Loading layer [=====>] 3.523MB/3.523MB
Loaded image: apc-docker-registry.dlinkddns.com/apc-koe/mta/cirrus/config-ui:latest
9e6f7162ac97: Loading layer [=====>] 224.8kB/224.8kB
4d1c4b61286: Loading layer [=====>] 9.658MB/9.658MB
68d71a5fd6ea: Loading layer [=====>] 3.921MB/3.921MB
Loaded image: apc-docker-registry.dlinkddns.com/apc-koe/mta/iotserver/modbus-service:latest
bcf2f3e8fe23: Loading layer [=====>] 5.792MB/5.792MB

```

4. Message "Deploy done" is output once the installation has been completed.
Calling the BFC Gateway user interface is described in Chapter: Log in to the BFC gateway (Page 80).

```

INFO: 2020/01/24 10:34:52.117826 Deploy done!
INFO: 2020/01/24 10:34:52.117867 -----
INFO: 2020/01/24 10:34:52.117880 Please use the following username to connect remote clients:
INFO: 2020/01/24 10:34:52.117905 iotserver
INFO: 2020/01/24 10:34:52.117923 Please use the following password to connect remote clients:
INFO: 2020/01/24 10:34:52.117944 675b4a5e-64a7-46a2-8356-59c9e4d6d7bb
INFO: 2020/01/24 10:34:52.117965 -----
apc@debian10:~/bfc$

```

Note**Installation on IPC227E with CentOS7**

If the system cannot be booted, make the following settings:

- Disable USB 3.0 in BIOS
 - Make the following entry in the file "/etc/default/grub":
 - GRUB_CMDLINE_LINUX="reboot=p crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet"
 - Then run "sudo grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg".
-

5.4 Installing BFC Gateway in a Kubernetes cluster

You can also install the BFC Gateway in a Kubernetes cluster.

Requirement

The following Kubernetes variants are supported:

- Azure Kubernetes Service (AKS), Kubernetes V1.22.0 or higher (<https://azure.microsoft.com/de-de/services/kubernetes-service/>)
- K3s Lightweight Kubernetes, Kubernetes V1.22.0 or higher (<https://k3s.io/>)
- Red Hat Openshift, Kubernetes V1.22.0 or higher (<https://www.redhat.com/de/technologies/cloud-computing/openshift>)

Note

For installation and configuration, it is imperative to have a technically sound knowledge of Kubernetes. In addition, a secure handling of the Kubernetes variant used is necessary.

When using Kubernetes, basic monitoring must be enabled in the orchestration software used for the BFC Gateway to monitor the operating status of the BFC Gateway.

Preparing the installation

Installation of the BFC Gateway is started from a Windows or Linux remote computer. The "kubectl" program is used for this purpose.

1. Install kubectl:

- Instructions for installing kubectl under Windows (<https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/#install-kubectl-binary-with-curl-on-windows>)
- Instructions for installing kubectl under Linux (<https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/>)

2. Configure kubectl:

In order for kubectl to communicate with the Kubernetes cluster on which the installation is to be performed, you must set the KUBECONFIG environment variable on the system from which the installation will be launched.

- Set the environment variable under Windows (Powershell) as follows:
`$Env:KUBECONFIG="kubeconfig.yaml"`
- Set the environment variable under Linux as follows:
`export KUBECONFIG=kubeconfig.yaml`

Note

kubeconfig.yaml

The `kubeconfig.yaml` file contains the connection data for the Kubernetes system.

Refer to the documentation of the Kubernetes system used to generate `kubeconfig.yaml`.

Installing the BFC Gateway (Kubernetes)

1. Transfer the installation files for the BFC Gateway to the remote computer from which the installation will be started.

Name	Änderungsdatum	Typ	Größe
docker	11.10.2021 14:18	Dateiordner	
examples	11.10.2021 14:09	Dateiordner	
kubernetes	11.10.2021 14:18	Dateiordner	
BFC_fct_man_de-DE.pdf	08.10.2021 11:07	PDF Document	14.351 KB
BFC_fct_man_en-US.pdf	08.10.2021 11:07	PDF Document	13.624 KB
BFC_readme_de-DE.pdf	08.10.2021 11:07	PDF Document	156 KB
BFC_readme_en-US.pdf	08.10.2021 11:07	PDF Document	141 KB
BFC_readme_oss.html	08.10.2021 11:07	Firefox HTML Doc...	2.589 KB
bfc-gateway.tgz	08.10.2021 11:20	TGZ-Datei	784.426 KB
setup.exe	08.10.2021 11:07	Anwendung	4.328 KB
setup.linux.x64	11.10.2021 14:09	X64-Datei	4.177 KB
version.txt	08.10.2021 11:07	Textdokument	1 KB

2. Start the installation using the following command:

- Linux:

```
./setup.linux.x64 -u admin -p admin -kubernetes=true  
-kubeconfig=/bfc/kubeconfig.yaml -deploy -namespace=bfc  
-reguser '<user>' -regpw '<pass>'
```

Note

If necessary, adjust the path to the `kubeconfig.yaml` file.

Overview of installation options:

Option	Description
-u	User name for the BFC Gateway
-p	Password for the BFC Gateway
-kubernetes	Set this option to <code>true</code> for installation in a Kubernetes cluster.
-kubeconfig	Path to the <code>kubeconfig.yaml</code> file, which contains the connection data to the Kubernetes cluster
-deploy	Keyword for installation of the BFC Gateway
-namespace	Kubernetes namespace where the BFC Gateway will be installed
-reguser	User name for the Docker Registry
-regpw	Password for the Docker Registry
-excluded-apps	BFC apps that are not installed Examples: • <code>-excluded-apps "bfc-analytics"</code> • <code>-excluded-apps "bfc-analytics, IBaseApp, OptimizationCheckApp"</code>
-k3s	Set this option to <code>true</code> to start importing the images into K3S.

Note

Namespace

If the specified namespace is already in use, you must delete it before installation with the following command:

```
kubectl delete ns bfc
```

Note

Error message during installation

If during installation the message "The ClusterRoleBinding "metricsrouter-binding-bfc" is invalid: roleRef: Invalid value: rbac.RoleRef{APIGroup:"rbac.authorization.k8s.io", Kind:"ClusterRole", Name:"metricsrouter-bfc"}: cannot change roleRef" appears, you must delete the specified roles from Kubernetes.

Then restart the installation.

This error may occur if a BFC Gateway has already been installed before.

Note

Red Hat OpenShift installation

For a Red Hat OpenShift installation, the namespace where the BFC Gateway is to be installed requires root privileges.

For example, add root privileges via the Openshift CLI:

```
oc login
oc project {name of namespace}
oc adm policy add-scc-to-user anyuid -z bfc-app-manager-service-account
```

OKD CLI download:

<https://github.com/openshift/origin/releases>

Getting Started:

https://docs.okd.io/1.5/cli_reference/get_started_cli.html

Note

BFC Gateway installation under K3S

When installing the BFC Gateway under K3S, the supplied images can be imported automatically. This eliminates the need for access data to a registry. The new installation command looks like this:

```
./setup.linux.x64 -u admin -p admin -kubernetes=true
-kubeconfig=/etc/rancher/k3s/k3s.yaml -deploy -namespace=bfc -k3s
```

Note

Installing the BFC Gateway under K3S for Debian 10

When using Debian 10, before installing the BFC Gateway under K3S, package "iptables" must be replaced by "nftables" as otherwise an excessively high CPU utilization level can occur in operation. To do this, use the following commands:

```
sudo apt remove iptables -y
sudo apt install nftables
```

Note

If the BFC OPC UA server is to be used for an Openshift or Azure Kubernetes Service (AKS) installation, a host name must be added with the keyword "-hosts", on which the OPC UA server permits client connections.

Example:

```
./setup.linux.x64 -u admin -p admin -kubernetes=true
-kubeconfig=/bfc/kubeconfig.yaml -deploy -namespace=bfc -reguser
'<username>' -regpw '<password>' -hosts bfc-bfc.apps.dev-test.bfc-
siemens.com
```

In this example, the endpoint for the BFC OPC UA server is created for this address:

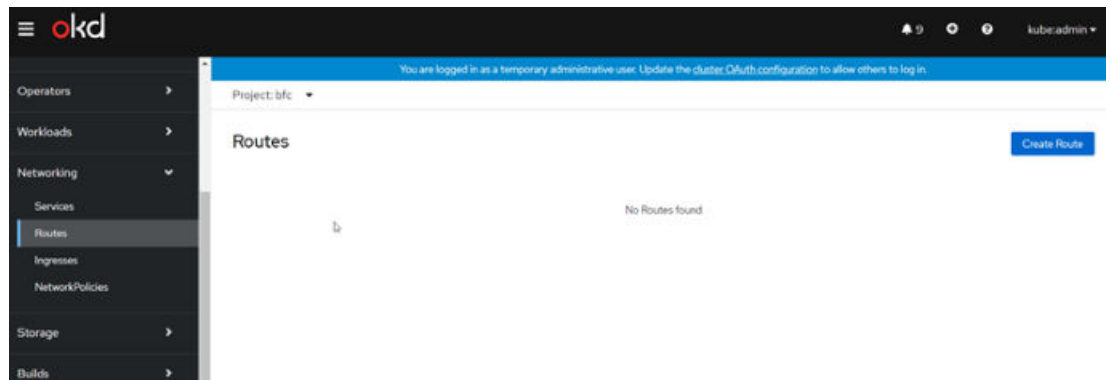
```
opc.tcp://bfc-bfc.apps.dev-test.bfcsiemens.com:4840
```

Setting up a route to BFC Gateway

For Red Hat Openshift and Azure Kubernetes Service (AKS) installations, you must set up a route to the BFC Gateway after installation so that you can call ConfigUI.

Red Hat Openshift

Configure the route to the BFC Gateway in Openshift's OKD Management Console.



1. Call up the **Networking > Routes** menu.
2. Create a new route via **Create Route**.
3. Make sure that the appropriate namespace is selected under **Project**.
4. Fill in the fields as follows:

Parameters	Value
Name	bfc
Service	entry
Target Port	9877
Security	Activate secure route
TLS Termination	Passthrough
Insecure Traffic	Redirect

Azure Kubernetes Service (AKS)

Add an annotation to the "entry" service. Azure Kubernetes Service automatically generates the routing from this.

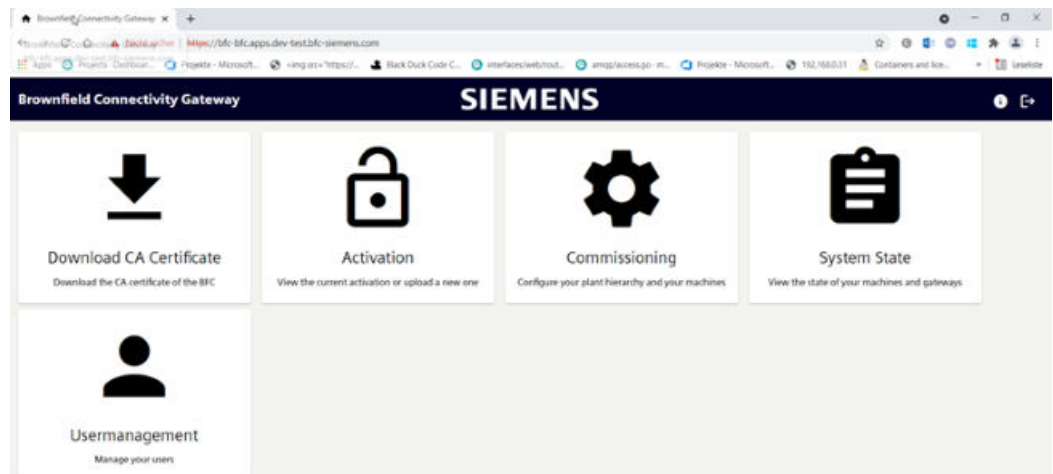
1. Open the entry service configuration with `kubectl -n bfc edit svc entry`.
2. Under `metadata` in the editor, add the annotation `service.beta.kubernetes.io/azure-dns-label-name: bfcazure`.

Example:

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving the
# file, the editor will be reopened with the relevant failures.
#
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"service.beta.kubernetes.io/azure-dns-label-name: bfcazure
creationTimestamp: "2021-10-01T14:02:51Z"
finalizers:
- service.kubernetes.io/load-balancer-cleanup
labels:
  com.siemens.bfc.base.service: entry
  v1.3: "true"
  v1.4: "true"
```


Checking the installation of the BFC Gateway

1. Once installation has been completed, use the following command to verify that all containers/pods are running:
`kubectl -n bfc get pods`
After 10 to 15 minutes at the latest, all containers/pods should have the status "running".
2. Once all containers/pods are running and the routes have been set up, invoke ConfigUI.
Example of an OpenShift installation:



Operating the BFC gateway

To facilitate data transfer between the machines and the different target systems - for example, MindSphere, local databases, clouds - you must configure the BFC Gateway for your specific system.

The user interface is divided into the following areas:

- "Activation" area (Page 83)
Here you see the currently activated functions and can import a new activation.
- "Commissioning" area (Page 86)
The system-specific configuration is realized here.
- "System State" area (Page 297)
You obtain an overview of the system here. Further, you have the option of generating a final report.
- "Usermanagement" area (Page 313)
This is where the user and rights management takes place.

Note

Automatic logout

BFC ConfigUI automatically logs out after two hours. Configuration adaptations not saved up until then are lost.

6.1 Icons and buttons

The following icons and buttons are available for operation and navigation:

Note

Tooltip

The icons and buttons have a tooltip.

Icons/buttons	Explanation
	Landing page Remark: Back to the landing page
	Information, e.g.: • Version • Details
	Refresh

6.2 "Landing page" area

Icons/buttons	Explanation
	Print
	Add ...
	Delete ...
	Move ...
	Save ...
	Show credentials
	Edit
	Show live data
	Show logfile
	Start Client
	Stop Client
	Running
	Stopped
	"OK" signal
	"Error" signal
	"Warning" message

6.2 "Landing page" area

6.2.1 Log in to the BFC gateway

In order to utilize the functions of the BFC gateway, you must log in on the BFC gateway user interface.

You must log on to each device on which you open the BFC "Landing page".

Procedure

1. Start your web browser.
2. Enter the URL of the BFC gateway into the address line of the web browser:
<https://<BFC gateway host>:9877/>

3. Install the certificate of the BFC gateway.
 - Install the self-signed certificate provided by the BFC.
The certificate generated during installation is self signed. To remove warning messages from the browser, download the certificate and add to the Trusted Root Certification Authorities of the operating system.
 - OR –
 - Install a user-specific certificate.
Additional information on using user-specific certificates (e.g. CA certificates) is provided in Chapter: Installing a user-defined certificate (Page 81).
4. The login window opens.
 - Enter the user name.
 - Enter the password.
 - Click on "Sign in".

6.2.2 Installing a user-defined certificate

You have the option of installing a user-defined certificate to securely access the BFC "Landing page". The user-defined certificate replaces a self-signed certificate that is created during the first installation.

The requirements and installation instructions for a docker-based and a Kubernetes-based installation of the BFC Gateway are described in the following.

Requirement

An existing BFC installation is required in order to install a user-defined certificate.

Before installing the user-defined certificate, ensure that the appropriate host names and IPs as well as the host name "entry" are referenced by the new certificate.

Files:

- Certificate
- Private key
- Password file (optional, if the private key is unencrypted)
The password file is normally called "pass.pwd", and contains the password in plain text.

Procedure for a docker-based installation

1. Using an SCP-capable client (WinSCP), log on to the Linux system on which the BFC Gateway runs.
2. Open a shell. You require root rights (sudo).
3. Remove the web ingress layer containers by running the following command:
`sudo docker service rm thirdparty_entry thirdparty_entryinit`
4. In the BFC master directory, navigate to folder "/docker/thirdparty/certs".

6.2 "Landing page" area

5. Delete all existing "cert.crt" and "cert.key". Do **not** delete "pass.pwd" if you wish to use an unencrypted private key.
 - Optional: Delete file "pass.pwd" if you wish to use an unencrypted private key.
6. Copy the new server certificate as "cert.crt" and the new private key as "cert.key" into directory "docker/thirdparty/certs".
 - Optional: Copy the new password file as "pass.pwd" into directory "docker/thirdparty/certs".
7. Navigate from the BFC master directory back to "/docker".
8. Run `sudo ./completeinstall.sh`. Set the run rights if the file cannot be run.

Note

Invalid certificate

If the certificate generated by the BFC Gateway is invalid, then you must reinstall the user-defined certificate.

Procedure for a Kubernetes-based installation

1. Using an SCP-capable client (WinSCP), log on to the Linux system on which the BFC Gateway runs.
2. Open a shell. You require root rights (sudo).
3. Copy the new server certificate as "cert.crt" and the new private key as "cert.key" into any directory.
4. If you are using an unencrypted private key, then create an empty "pass.pwd" in any directory.
 - OR -
 - If you are using an encrypted private key with password file, then copy the new password file as "pass.pwd" into any directory.
5. If you are using an unencrypted private key, then run the following command:

```
sudo kubectl apply configmap entry-cert-config --from-file=cert.crt=<path to cert.crt> --from-file=cert.key=<path to cert.key> --from-file=pass.pwd=<path to empty pass.pwd>
```

 - OR -
 - If you are using an encrypted private key with password file, run the following command:

```
sudo kubectl apply configmap entry-cert-config --from-file=cert.crt=<path to cert.crt> --from-file=cert.key=<path to cert.key> --from-file=pass.pwd=<path to pass.pwd>
```
6. Run the following command:

```
sudo kubectl rollout restart deployment entry-deployment -n bfc
```

Note

Invalid certificate

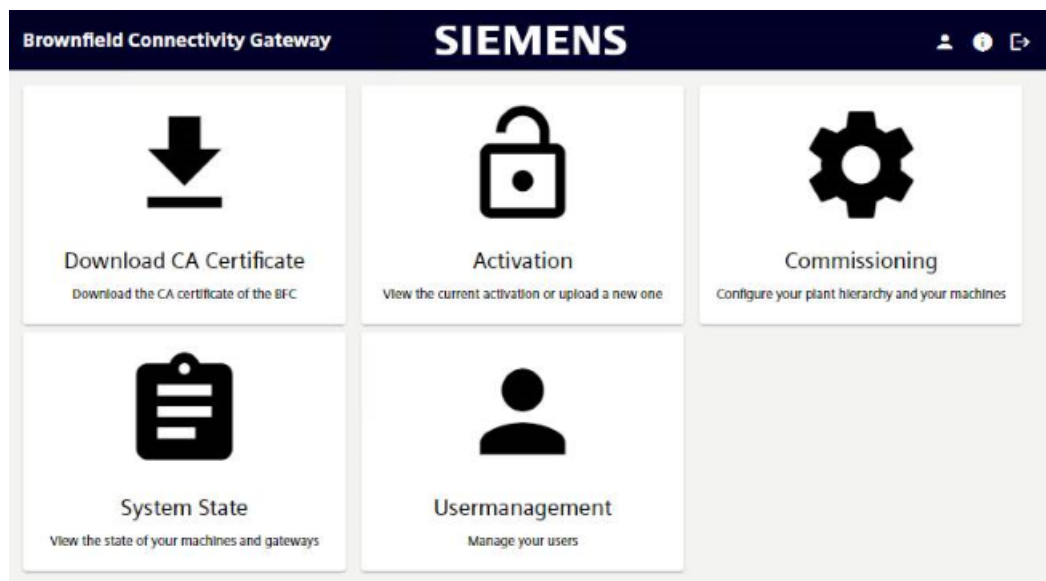
If the certificate generated by the BFC Gateway is invalid, then you must reinstall the user-defined certificate.

6.2.3 Starting processing

The landing page opens once you have successfully logged in.

Procedure

1. Select the "Activation" area to view the current activation or to import a new activation.
For more information, see Chapter: "Activation" area (Page 83).
- OR -
2. Select the "Commissioning" area to configure your system and machines or to make changes to the existing configuration.
For more information, see Chapter: "Commissioning" area (Page 86).
- OR -
3. Select the "System State" area to obtain an overview of the BFC Gateway.
For more information, see Chapter: "System State" area (Page 297).
- OR -
4. Select the "Usermanagement" area to configure users or make changes to existing users.
For more information, see Chapter: "Usermanagement" area (Page 313)



6.3 "Activation" area

An activation is required for the BFC gateway.

The default activation after the initial installation runs for 180 days. During this time, the number of clients, middleware and gateways is not restricted.

6.3 "Activation" area

Procure a license key for Brownfield Connectivity from the Siemens Industry Mall (<https://mall.industry.siemens.com/goos/WelcomePage.aspx?regionUrl=/oms&language=en>). Follow the instructions provided there.

- Article number for a Brownfield Connectivity Pro license: 9MC1110-1EG00-0AD2

Note

With a Pro license, all supported clients can be activated.

- Article number for a Brownfield Connectivity Basic license: 9MC1110-1EG00-0AD3

Note

With a Basic license, the following clients can be activated: WebDAV, Beckhoff TwinCAT (ADS), MQTT, HTTP REST, S7 (RFC1006), and Modbus TCP.

6.3.1 Show activation

Procedure

1. Start the BFC configuration interface from your web browser.
2. Open the "Activation" area.

- Under the "Current Activation Information" tile you can see the currently active clients, middlewares, and gateways. It also shows how many activations are still available. Basically, only clients need to be activated.
- The "Client Information" table shows which client belongs to which client type (Pro/Basic).

Activation **SIEMENS**

BFC Gateway Information
BFC Gateway ID: 1693c2ee-6b56-4b25-b756-f37f88aa2b24 Copy ID

Current Activation Information Import new activation

Clients (Import)		Middlewares (Logic)		Gateways (Export)	
Pro Client:	8/∞ - available: ∞	Pro Middleware:	0/∞ - available: ∞	Pro Gateway:	0/∞ - available: ∞
MTConnect Client:	4	Basic Middleware:	3/∞ - available: ∞	Basic Gateway:	3/∞ - available: ∞
FANUC Client:	1	Scriptlogic:	3	InfluxDB Gateway:	1
BFC Client:	2			MQTT Gateway:	1
OPC UA Client:	1			OPC UA Server:	1
Basic Client:	2/∞ - available: ∞				
MQTT Client:	1				
S7 Client:	1				

Apps

BFC Analytics Demo Package: 0/4

This activation expires in 101 days 6 hours 23 minutes 36 seconds!

Client information

Pro	Basic
BFC Client	Modbus Client
MTConnect Client	S7 Client
OPC UA Client	HTTP Client
FANUC Client	MQTT Client
Heidenhain Client	Beckhoff Client

Middleware information

Pro	Basic
	Scriptlogic

Gateway information

Pro	Basic
	MindSphere Gateway
	InfluxDB Gateway
	MQTT Gateway
	AMP4.1 Gateway
	MySQL Gateway
	Elasticsearch Gateway
	HTTP Gateway
	OPC UA Server

6.3.2 Importing activation

After you have received a new activation license, you can import it into the system.

Requirement

You have received the email from the address: bfc-support.di.de@siemens.com (<mailto:bfc-support.di.de@siemens.com>).

Procedure

- Start the BFC configuration interface from your web browser.
- Open the "Activation" area.
- Click the "Import new activation" button.

6.4 "Commissioning" area

4. The input window opens.
Select the received file "bfc<ID>.activation" from the attachment of the e-mail you received.
5. The new activation is imported and displayed on the user interface.

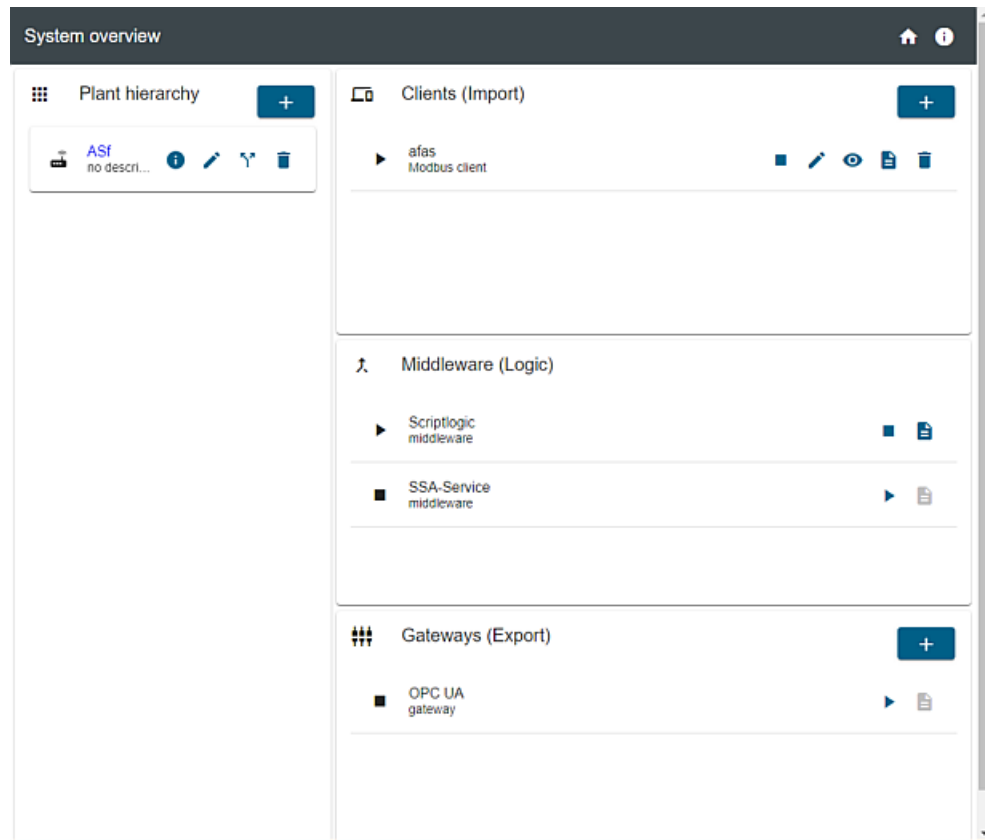
6.4 "Commissioning" area

6.4.1 Starting the configuration

The "Commissioning" area has 4 sub-areas:

- Plant hierarchy
This area is used to logically sort and permits the creation of "Plants", "Lines", "Cells" and "Machines" in any combination.
- Client (Import)
In this area, the clients are configured for the selected level in the "Plant hierarchy" area.
- Middlewares
In this area, the middleware for the selected level is displayed in the "Plant hierarchy" area.
- Gateways (Export)
In this area, the gateways are configured for the selected level in the "Plant hierarchy" area.

Structure



6.4.2 BFC Protect

Protocols not capable of encryption can be secured by additional hardware (tunnel device).

Using the integrated "BFC Protect" procedure, you can connect the following BFC drivers over a secure SSH tunnel:

- Modbus TCP
- S7
- FANUC
- Heidenhain
- Beckhoff
- EIP
- Omron
- MTConnect

Requirement

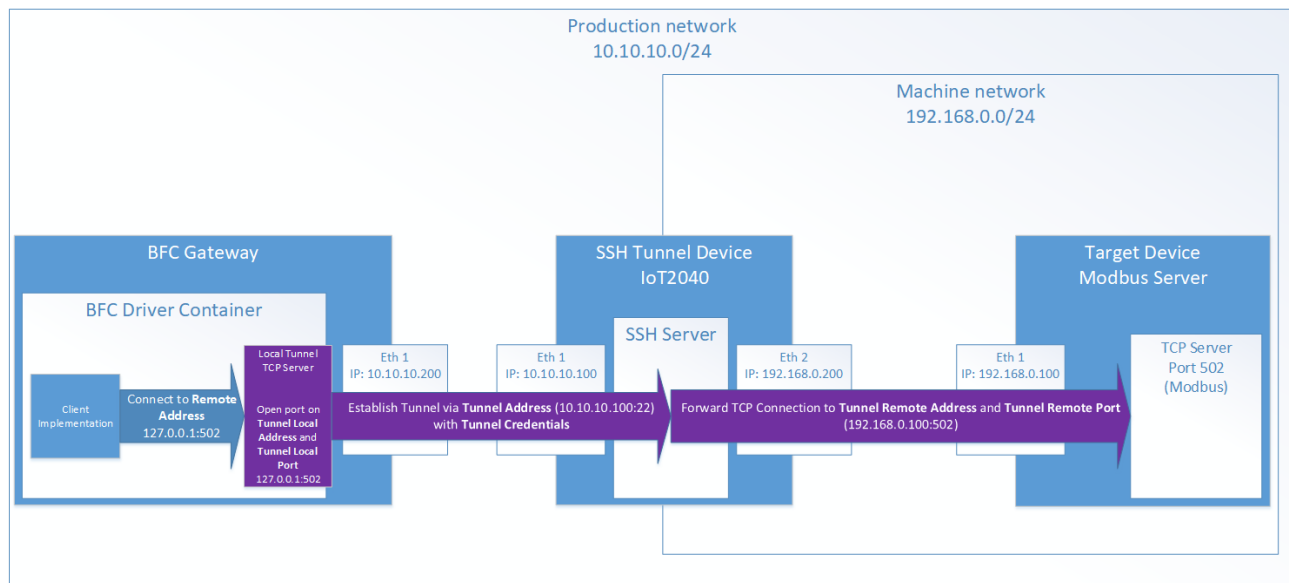
- To guarantee the physical separation of the networks you require the following:
 - Additional hardware with an SSH server
 - 2 network adapters

Example of Siemens hardware

The following additional hardware serves as a tunnel device and securely separates the "production network" and "machine network" from one another.

- IOT2040
- IOT2050
- IPC 127E

The following diagram shows a schematic configuration example for secure communication to a Modbus TCP server.



6.4.2.1 Integrating the tunnel device

When integrating the tunnel device, note the following:

- Integrate the tunnel device into both networks.

To successfully integrate the tunnel device into the networks, please consult the operator of the networks.
- Configure the network adapters as follows:
 - Network adapter 1: For access to the production network
 - Network adapter 2: For access to the machine network

For the configuration of the network adapters, consult the documentation of the installed operating system.

- Secure the access from the production network to the SSH server.
- When the tunnel device has been successfully integrated into both networks, configure the BFC driver in the "Commissioning" area of the BFC gateway user interface.
More information on configuring the BFC driver can be found in Chapter: Configuring BFC Protect for clients (Page 89).

6.4.2.2 Configuring BFC Protect for clients

You can use the "BFC Protect" function for both existing clients and those to be created.

Requirement

- The client must support the "BFC Protect" function.
- Further information on the requirements can be found in Chapter: BFC Protect (Page 87).

Parameters

Parameter	Description
Remote address	Destination IP address to which the driver should connect itself In the case of a tunneled connection this address is always "127.0.0.1".
Remote port	Destination port to which the driver should connect itself
Tunnel local address -	Local TCP server address via which the tunnel can be set up. Set this address to the value "127.0.0.1".
Tunnel local port	Local TCP server port via which the tunnel can be set up. The value must correspond to the value of the "remote port".
Tunnel remote address	Destination IP address to which a tunnel connection is to be set up. The value must correspond to the IP address of the destination device in the machine network.
Tunnel remote port	Destination TCP port to which a tunnel connection is to be set up. The value must correspond to the device port. (e.g. 102 for S7, 502 for Modbus TCP...)
Tunnel address	IP address of the tunnel device that is to act as a communication gateway between the two networks
Tunnel port	TCP port of the SSH server on the tunnel device
Tunnel username	Name of a user who may authenticate himself on the SSH server
Tunnel password	Password of the SSH user

Procedure

In order to be able to use the "BFC Protect" function, proceed as follows:

- Existing clients
 - Navigate into the "Commissioning" area.
 - In the "Client (Import)" area, click on the "Edit" icon to change the configuration of an existing client.
- New clients
 - Navigate into the "Commissioning" area.
 - In area "Client (Import)" click on the "+" button to add a new client.

6.4.2.3 Example: Configuring BFC Protect for the Modbus client

The necessary settings for the "BFC Protect" function are explained using the Modbus client as an example.

The Modbus client supports the "BFC Protect" function.

In order to configure the "BFC Protect" function for a Modbus client, carry out the following steps:

- "Define basic configuration"
- "Define advanced configuration"

You can apply this procedure to every client that supports the "BFC Protect" function.

Requirement

The requirements in Chapter: BFC Protect (Page 87) must be met.

Parameters

Parameter	Description
Define basic configuration	
Remote address	Destination IP address to which the driver should connect itself In the case of a tunneled connection this address is always "127.0.0.1".
Remote port	Destination port to which the driver should connect itself In the example: 502
Define advanced configuration / Optional	
Type a new environment variable...	New environment variable for the image
Template: IOTCLIENT_OVERRIDE={"remoteConfig":{"tunnel":{"localAddress":"127.0.0.1","localPort":[" Tunnel Local Port "],"remoteAddress":[" Tunnel Remote Address "],"remotePort":[" Tunnel Remote Port "],"tunnelAddress":[" Tunnel Address "],"tunnelPort":[" Tunnel Port "],"tunnelUsername":[" Tunnel Username "],"tunnelPassword":[" Tunnel Password "]}}}	

Parameter	Description
Tunnel local port	Local TCP server port via which the tunnel can be set up. The value must correspond to the value of the "remote port".
Tunnel remote address	Destination IP address to which a tunnel connection is to be set up. The value must correspond to the IP address of the destination device in the machine network.
Tunnel remote port	Destination TCP port to which a tunnel connection is to be set up. The value must correspond to the device port. (e.g. 102 for S7, 502 for Modbus TCP...)
Tunnel address	IP address of the tunnel device that is to act as a communication gateway between the two networks
Tunnel port	TCP port of the SSH server on the tunnel device
Tunnel username	Name of a user who may authenticate himself on the SSH server
Tunnel password	Password of the SSH user

Procedure

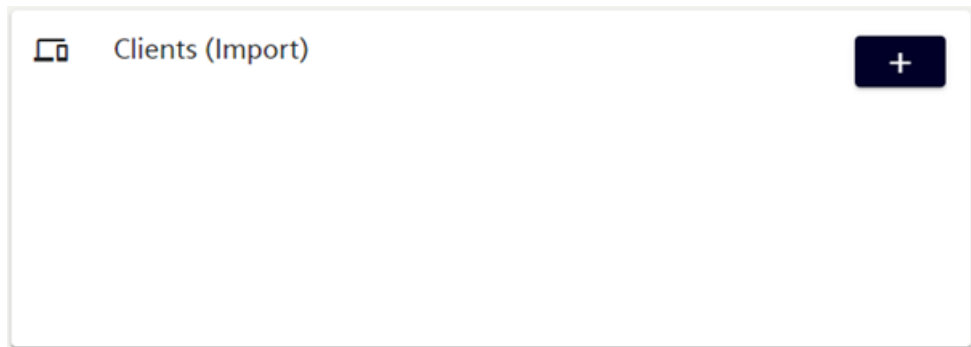
1. Navigate into the "Commissioning" area.
 - In area "Client (Import)" click on the "+" button to add a new client.

Note

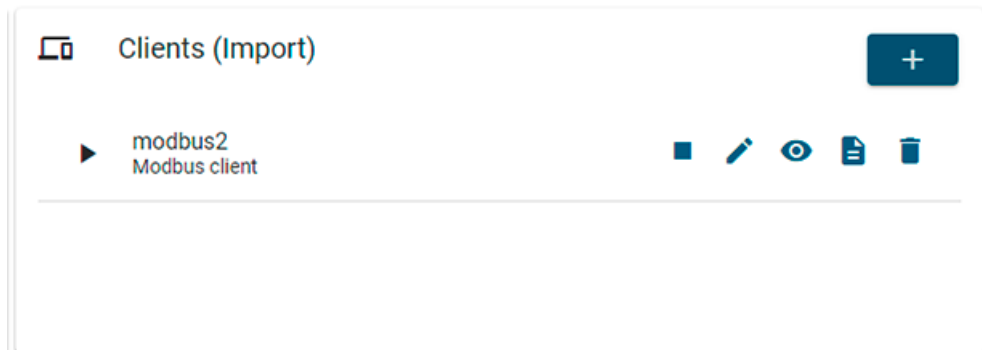
Parameters

The following descriptions refer only to the "BFC Protect" function.

All other settings for creating a new Modbus client can be found in Chapter: Creating a Modbus client (Page 117).



- - OR -
- In the "Client (Import)" area, click on the "Edit" icon to change the configuration of an existing client.



2. Navigate to the step: "Define basic configuration"
 - To perform the basic configuration, fill out the fields for "BFC Protect"
 - Click on "Next".

Edit client "testmodbustunnel"

1 Define basic configuration

Type client ID...

testmodbustunnel

Supported input: lowercase letters, numbers, - (must start with letter) 16 / 50

Type client description...

this client connection will be tunneled via SSH

47 / 100

Type Unit ID... *

1

Range: 0 or greater

Type Modbus server URL... *

127.0.0.1:502

Next

2 Define dataset configuration

3 Define advanced configuration Optional



3. Navigate to the step: "Advanced configuration" / Optional
 - Configure the tunnel parameters.
To do this, insert an additional environment variable according to the template stated in the table.
Replace the parameters quoted in the square brackets according to your network and SSH server configuration.
We recommend that you set the values for "Remote Port" and "Tunnel Local Port" to the value of "Tunnel Remote Port" to avoid configuration errors.
 - Click on "Save" to save the Modbus client.

Edit client "testmodbustunnel"

1

Define basic configuration

2

Define dataset configuration

3

Define advanced configuration
Optional

Type image path... *

bfc-registry.com/apc-koe/mts/iotserver/modbus-service:v1.5.0

Type username to access the image path...

Type password to access the image path...

Environment variable 1

IOTCLIENT_OVERRIDE={"remoteConfig":{"tunnel": {"localAddr

Type new environment variable... +

Save

Cancel

4. The logs of the Modbus client now show messages relating to the tunnel set-up.

Note**Log messages**

Until the SSH tunnel has been successfully established, any attempts to connect the client will fail and be displayed as log messages.

View logs of testmodbustunnel

```
INFO: 2020/09/22 13:12:11.287522 Module[Databus-BaseClient] - Creating SSH-Tunnel...
INFO: 2020/09/22 13:12:11.287834 Module[Databus-BaseClient] - SSH-Tunnel is starting...
ERROR: 2020/09/22 13:12:11.288082 Module[Modbus-Master] - Could not connect to configured address:dial tcp
127.0.0.1:502: connect: connection refused - {Address:127.0.0.1:502 UnitID:1 Decoupled:false}
INFO: 2020/09/22 13:12:11.288211 Module[Databus-BaseClient] - No CoordinatorCfg.URL provided. Using default:
http://coordinator:8080
INFO: 2020/09/22 13:12:11.288282 Module[Databus-BaseClient] - No ManagerConfig.Skills provided. Using default of
every skill for a client: [com.siemens.bfc.skills.basic.publish.reading.sets com.siemens.bfc.skills.basic.publish.alarm.lists
com.siemens.bfc.skills.advanced.rpc.receive.write.data com.siemens.bfc.skills.diagnosis.publish.metrics]
INFO: 2020/09/22 13:12:11.288315 Module[Databus-BaseClient] - SSH-Tunnel is started...
INFO: 2020/09/22 13:12:11.363662 Module[Databus-BaseClient] - We acquired an activation. Woop, Woop...
INFO: 2020/09/22 13:12:11.363766 Module[Databus-BaseClient] - SubscribeConnectionStateChange:
{URL:127.0.0.1:502 Data:map[unitID:1] Tunnel:0xc0002048c0}
INFO: 2020/09/22 13:12:11.363780 Module[Databus-BaseClient] - SubscribeConnectionStateChange...Done
INFO: 2020/09/22 13:12:11.363799 Module[Databus-BaseClient] - Starting ReadingJob for testset: {ReadingMode:
{Mode:interval Interval:500 Debounce:500} Mappings:[{Name:testtp1 Address:h:1,1 Type:int}}]
INFO: 2020/09/22 13:12:11.363824 Module[Databus-BaseClient] - Starting ReadingJob for testset...Done
INFO: 2020/09/22 13:12:11.363831 Module[Databus-BaseClient] - Starting CoordinationWorker...
INFO: 2020/09/22 13:12:11.363834 Module[Databus-BaseClient] - Read ENV - using default value - [PORT]=8080
INFO: 2020/09/22 13:12:11.363848 Module[Databus-BaseClient] - Read ENV - using default value -
[APPNAME]=baseclient
```



```
INFO: 2020/09/22 13:12:11.363860 Module[Databus-BaseClient] - Create Server...port: 8080 - staticpath: root
INFO: 2020/09/22 13:12:11.363871 Module[Databus-BaseClient] - Create Server...Done - port: 8080 - staticpath: root
INFO: 2020/09/22 13:12:11.363886 Module[Modbus-Master] - Using SampleInterval:500ms
INFO: 2020/09/22 13:12:11.592809 Module[Modbus-Master] - Connected to Modbus-Slave
```



6.4.3 Creating clients (Import)

A client reads data from a machine or a server, which provides data. The data to be read is organized in data sets and data points.

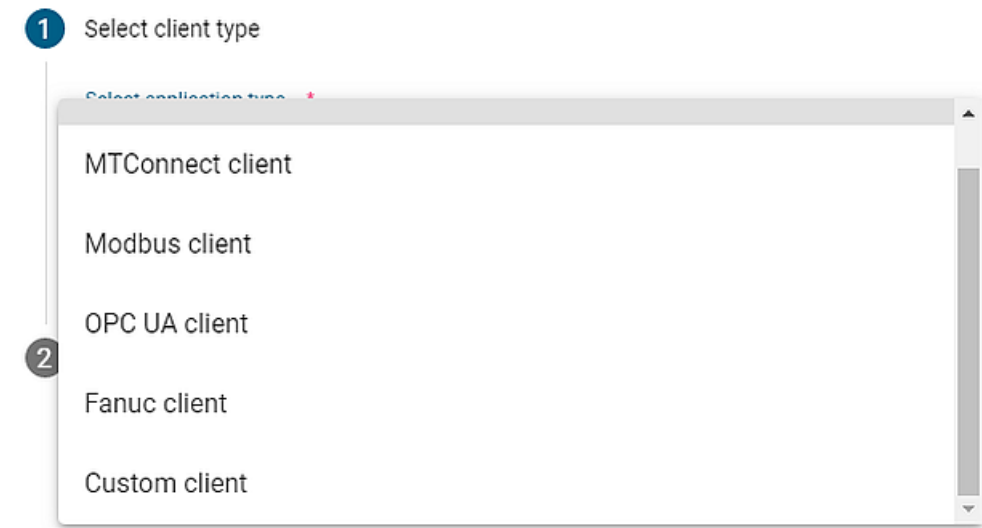
- A data set comprises one or several data points.
- A data point corresponds to a variable in a machine control, for example.

A client can contain several data sets.

You can create the following clients in the "Commissioning" area:

- MTConnect client
- Modbus client
- OPC UA client
- FANUC client
- BFC client
- S7 client
- HTTP REST client
- Heidenhain client
- Beckhoff client
- EIP client
- Omron client
- MQTT client
- Custom-client (placeholder for project-specific extensions)

Add new client



Additional information on creating the individual clients is provided in the subsequent chapters.

6.4.3.1 Creating a BFC client

A BFC client can read data from a SINUMERIK 840D, SINUMERIK 840D sl and SINUMERIK 828D.

Perform the following steps to create a BFC client:

Step 1: "Select client type"

Step 2: "Define basic configuration"

Step 3: "Define dataset configuration"

Requirement

- The SINUMERIK control system must be online in order for the BFC client to be created and edited.
- The BFC client has been successfully installed on the machine.
- The "Commissioning" area is open.

Parameters

Parameters	Description
① Select client type	
Select application type...*	Select BFC client
② Define basic configuration	
Search in list of pending clients...	Select the machine that is to be connected to the BFC gateway.
Type a new machine name (this has to be unique)...*	Assign a unique machine name
③ Define dataset configuration	
→ ① New Dataset	
Type dataset name...*	Dataset name
Select reading mode...*	Selecting the reading mode: • Interval The set time is called "Interval time" The configured variable is read cyclically according to the interval defined in the BFC gateway, regardless of whether the value of the variable has changed. • On Change The set time is called "Debounce time" The configured variable is read if the value changes.
Set interval / debounce time in milliseconds...*	Specify the interval and/or debounce time
→ ② New Datapoint	
Type datapoint name...*	Name of the datapoint

6.4 "Commissioning" area

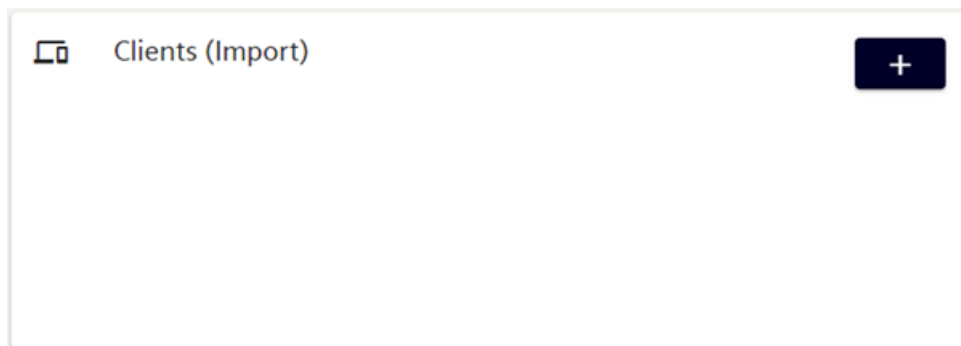
Parameters	Description
Type datapoint address... *	Address of the datapoint Remark: If an invalid address is used when configuring the BFC client, error messages are output in the client's log file during runtime. To diagnose these incorrect entries, check the log file of the BFC client after every configuration change using the "Show logfile" function.
Select data type... *	Selecting the data type: • Integer • Float • String • Bool

*: Obligatory data

Pattern: The character sequence to be used for the various entries is provided in the input windows.

Procedure

1. In the "Client (Import)" area, click on "+" to add a BFC client.



2. Step 1: "Select client type"
 - From the drop-down list, select entry "BFC client".
 - Click on "Next".

Add new client

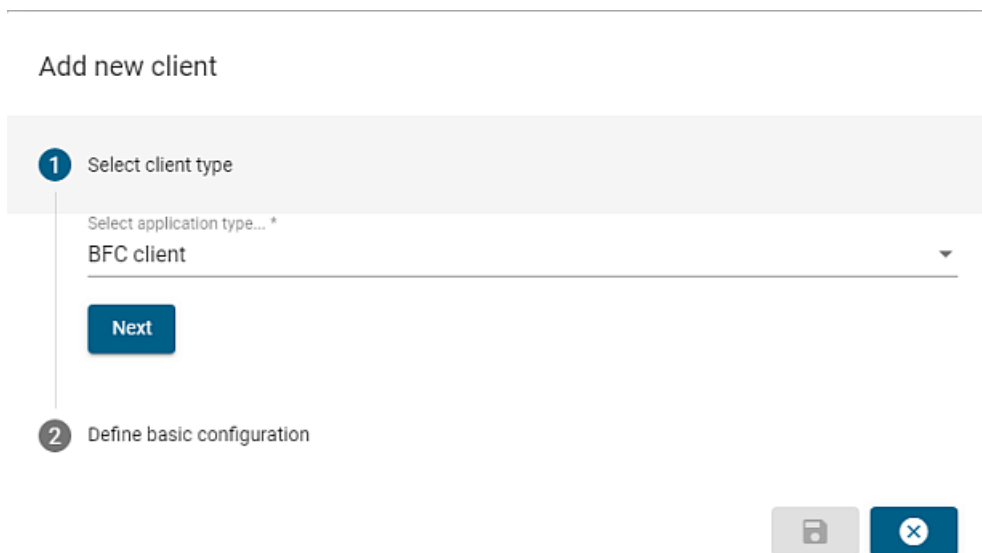
1 Select client type

Select application type... *

BFC client

Next

2 Define basic configuration

A screenshot of a wizard titled "Add new client". It shows a progress bar with two steps: "1 Select client type" (active) and "2 Define basic configuration". Under step 1, there is a label "Select application type... *" and a dropdown menu with "BFC client" selected. Below the dropdown is a blue "Next" button. At the bottom right of the wizard, there are two buttons: a grey "Save" button and a blue "Close" button with a white 'X' icon.

6.4 "Commissioning" area

3. Step 2: "Define basic configuration"

- Select a machine that should supply data to the client.
- Assign a name to the machine.
- Click on "Save" to save your selection and the machine name.

Add new client "Lola"

1

Select client type

2

Define basic configuration

Type client ID... *
Lola

Supported input: letters, numbers, -, _ (must start with letter or number) 4 / 50

Search in list of pending clients...
Lola

☒ Machine description: LoLaMaschine - ClientID: lmkgxzzgxn - HMI: Operate - Host: LolaLinux - IP: 192.168.4.143

Save

Cancel

4. The BFC client was successfully created, and is shown in the overview in the "Clients (Import)" area.

Click on the "Edit" icon to edit the BFC client.

Clients (Import)

+

Lola

BFC Client 02.14.00.16

↑

✎

⋮

5. Step 3: "Define dataset configuration" / ① New Dataset

- Populate the fields under "New Dataset" to configure the dataset.
- Click on "Configure datapoints" to define the datapoints belonging to the dataset.

Edit client "Lola"

1 New Dataset: Betriebsdaten

Type dataset name... *

Betriebsdaten

Select reading mode... *

On Change

Set debounce time in milliseconds... *

500

Range: 200-86400000

Configure datapoints

2 New Datapoint:

Save Cancel

6. Step 3: "Define dataset configuration" / ② New Datapoint

- Populate the fields under "New Datapoint" to define the datapoint belonging to the dataset.
- Click on "Add current datapoint" to add an additional datapoint.

Edit client "Lola"

1 New Dataset: Betriebsdaten

2 New Datapoint: Programmstatus

Type datapoint name... *

Programmstatus

Type datapoint address... *

/Channel/State/progStatus[u1]

Select data type... *

Float

Add current datapoint

Save Cancel

7. Step 3: "Define dataset configuration" / ② New Datapoint

- Click on the "Delete" icon to delete a datapoint.
- Click on "Add current dataset" to add the datapoints to the dataset.

Edit client "Lola"

Type datapoint address... *

Select data type... *

Integer

Add current datapoint

1 datapoint(s) configured - do not forget to add dataset!

Datapoint name	Address	Type
Programmstatus	/Channel/State/progStatus[u1]	float

Add current dataset

8. Step 3: "Define dataset configuration" / ① New Dataset

Note**Configured datasets**

The "Export" and "Import" functions allow you to export and import configured datasets within the clients.

More information can be found in Chapter: Using configured datasets (Page 206).

Click on "Save" to save the dataset.

More information can be found in Chapter Defining a dataset configuration (Page 198).

Edit client "Lola"

6 dataset(s) configured Export Import

Dataset name	Mode (time)	Amount of Datapoints	
CH1_BasicConfig	On Change (2000)	This dataset has 8 datapoints	
SINUMERIK_CSRAW	Interval (60000)	This dataset has 4 datapoints	
CSM_General_Info	Interval (30000)	This dataset has 4 datapoints	
CSM_AX01	Interval (30000)	This dataset has 5 datapoints	
SINUMERIK_CSALARMREACTION	On Change (200)	This dataset has 1 datapoints	
Betriebsdaten	On Change (500)	This dataset has 1 datapoints	

9. The BFC client is created.

6.4.3.2 Creating a FANUC client

A FANUC client can read and write data from/to a FANUC control system (FOCAS 1 and FOCAS 2).

Perform the following steps to create a FANUC client:

- Step 1: "Select client type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define alarm configuration"
- Step 5: "Define whitelisting configuration"
- Step 6: "Define advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select client type	
Select application type... *	Selection of the client
② Define basic configuration	
Type client ID... *	ID of the FANUC client
Type client description...	Description of the machine
Type Fanuc control URL... *	URL or IP address of the FANUC control system
Type Fanuc control port... *	Port number Remark: "8193" is the default port for FANUC
③ Define dataset configuration	
→ ① New Dataset: main	
Type dataset name... *	Dataset name
Select reading mode... *	Selecting the reading mode: - Interval The set time is called "Interval time" - On Change The set time is called "Debounce time"
Set interval / debounce time in milliseconds... *	Specify the interval and/or debounce time
→ ② New Datapoint	
Type datapoint name... *	Name of the datapoint
Type datapoint address... *	Address of the datapoint
Select data type... *	Selecting the data type: - Integer - Float - String - Bool
④ Define alarm configuration / Optional	
Set alarm sample time in milliseconds...	Alarm sampling time in milliseconds
⑤ Define whitelisting configuration / optional	
Type whitelisted address...	Address of the datapoint for which writing is permitted.
Select whitelisted type...	Selecting the data type: - Integer - Float - String - Bool
⑥ Define advanced configuration / optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path... *	Path for FANUC docker image
Type username to access the image path...	User name to access the image path

Parameter	Description
Type password to access the image path...	Password to access the image path
Set request timeout in seconds...*	Maximum wait time to access the data
Type new environment variable...	New environment variable for the image
Type file system address...	Internal address for program transmission As a rule, no changes are required.

*: Obligatory data

Pattern: The character sequence to be used for the various entries is provided in the input windows.

For information on configuring access rights for file operations, see Chapter: Configuring access rights (Page 467).

Datapoints

The datapoint of the datasets must have valid addresses.

Address positions	Example	Explanation
Start	focas	Address for the FOCAS-API
1st position	cnc	API call type Remark: For FOCAS-API there are cnc and pmc calls.
2nd position	rddynamic2	Function name
Additional positions	pos	Specification as to which variables should be called
	absolute	
End	[x]	The input parameters required for the particular function are in square brackets.

More information regarding the supported addresses is provided in Appendix: FANUC client data points (reading) (Page 561).

The grouping is realized after the API call. For each address, the data type to be parameterized is defined.

Note

API call structure

Siemens defined the addresses used here.

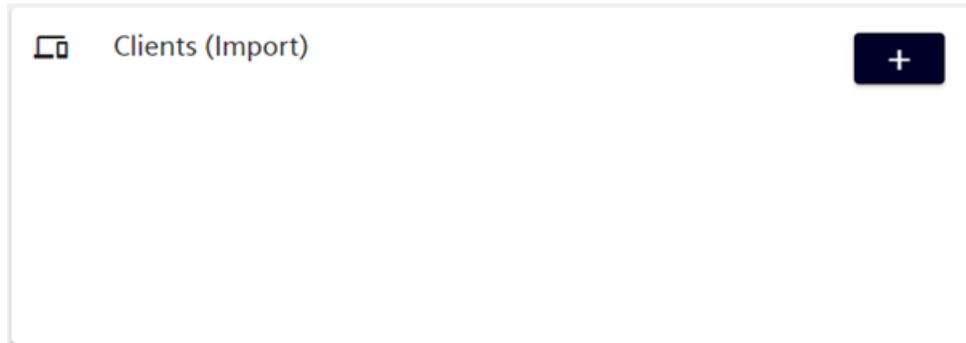
More information about the API call structure is available in the Internet (<https://www.inventcom.net/fanuc-focas-library/general/fwlib32>).

Example

```
/focas/cnc/rddynamic2/pos/absolute[1]
```

Procedure

1. In the "Client (Import)" area, click on "+" to add a new client.



2. Step 1: "Select client type"
 - From the drop-down list, select entry "Fanuc client".
 - Click on "Next".

Add new client

1 Select client type

Select application type... *

Fanuc client

Next

3. Step 2: "Define basic configuration"
 - Populate the fields to perform the basic configuration.
 - Click on "Next".

2 Define basic configuration

Type client ID... *

fanuc02

Type client description... Supported input: lowercase letters, numbers, - (must start with letter) 7 / 50

Fanuc machine 1

Type Fanuc control URL... * 15 / 100

192.168.6.26

Type Fanuc control port... *

8193

Range: 0 or greater

Next

4. Step 3: "Define dataset configuration" / ① New Dataset

- Populate the fields under "New Dataset" to configure the dataset.

Note**Configured datasets**

You can use the "Import" function to import configured datasets.

More information can be found in Chapter: Using configured datasets (Page 206).

- Click on "Configure datapoints" to define the datapoints belonging to the dataset.

The screenshot displays the 'Define dataset configuration' interface. At the top, a blue circle with the number '3' is followed by the text 'Define dataset configuration'. Below this, a vertical line separates the 'New Dataset: main' section from the 'New Datapoint:' section. The 'New Dataset: main' section contains three input fields: 'Type dataset name... *' with the value 'main', 'Select reading mode... *' with the value 'On Change', and 'Set debounce time in milliseconds... *' with the value '500'. To the right of the 'Select reading mode...' field, there is a note: 'Supported input: letters, numbers, - (must start with letter) 4 / 50'. Below the 'Set debounce time...' field, there is a note: 'Range: 200-86400000'. A blue button labeled 'Configure datapoints' is positioned below the 'Set debounce time...' field. The 'New Datapoint:' section is currently empty. At the bottom of the interface, there is a blue button labeled 'Import' and a grey button labeled 'Next'.

5. Step 3: "Define dataset configuration" / ② New Datapoint

- Populate the fields under "New Datapoint" to configure the datapoints belonging to the dataset.
- Click on "Add current datapoint" to add the data to the datapoint.

3 Define dataset configuration

1 New Dataset: main

2 New Datapoint: Operation-Mode

Type datapoint name... *

Operation-Mode

Supported input: letters, numbers, - (must start with letter) 14 / 50

Type datapoint address... *

/focas/cnc/statinfo/aut

Select data type... *

Integer

Add current datapoint

No datapoints configured yet

No datasets configured yet

Import

Next

6. Step 3: "Define dataset configuration"

- Populate the fields under "New Datapoint" to configure additional datapoints belonging to the dataset.
- Click on "Add current dataset" to add the complete dataset.

More information can be found in Chapter Defining a dataset configuration (Page 198).

2 New Datapoint:

Type datapoint name... *

Supported input: letters, numbers, - (must start with letter) 0 / 50

Type datapoint address... *

Select data type... *

Integer

Add current datapoint

1 datapoint(s) configured - do not forget to add dataset!

Datapoint name	Address	Type
Operation-Mode	/focas/cnc/statinfo/aut	int

Add current dataset

No datasets configured yet Import

Next

Save Cancel

7. Step 4: "Define alarm configuration"

- Configure the alarm sample time.
- OR -
- Keep the "Default value".
- Click on "Next".

4 Define alarm configuration

Optional

Set alarm sample time in milliseconds...

10000

Range: 5000-86400000

Next

6.4 "Commissioning" area

8. Step 5: "Define whitelisting configuration" / optional

- Populate the fields under "Whitelisted address" with the addresses that can be written to, and select the appropriate data type.

5

Define whitelisting configuration
Optional

Type whitelisted address...

/focas/cnc/wrparam/dword[0,6712]

Select whitelisted type...

Integer

+

9. Step 6: "Define advanced configuration" / Optional

- As a rule, no entries/changes are required.

Note**Advanced configuration**

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click on "Save" to save the FANUC client.

6 Define advanced configuration
Optional

Type image path... *

bfc-registry.com/apc-koe/mts/iotserver/fanuc-linux-service:v1.4.0

Type username to access the image path...

Type password to access the image path...

Set request timeout in seconds... *

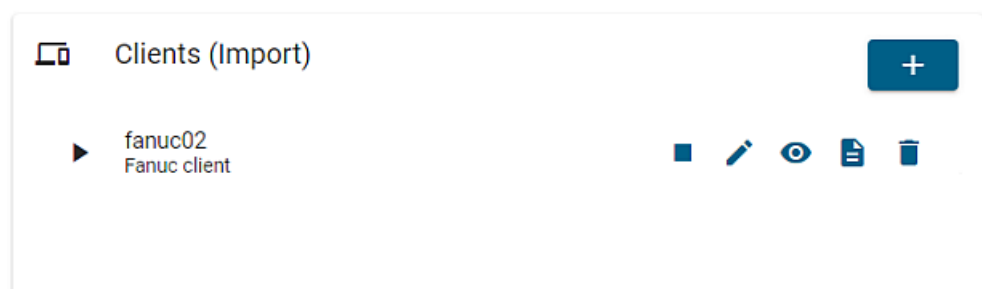
30

Range: 30-95400

Type new environment variable... +

10. The FANUC client was successfully created and is shown in the overview in the "Clients (Import)" area.



6.4.3.3 Creating an MTConnect client

An MTConnect client can read data from an MTConnect agent.

Perform the following steps to create an MTConnect client:

- Step 1: "Select client type"
- Step 2: "Define basic configuration"

6.4 "Commissioning" area

- Step 3: "Define dataset configuration"
- Step 4: "Define alarm configuration"
- Step 5: "Define advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select client type	
Select application type... *	Selecting the MTConnect client
② Define basic configuration	
Type client ID... *	ID of the MTConnect client
Type client description...	Description of the machine
Type MTConnect agent URL... *	Address of the agent
Type MTConnect agent username...	User name of the MTConnect client
Type MTConnect agent password...	Password
③ Define dataset configuration	
→ ① New Dataset	
Type dataset name... *	Dataset name
Select reading mode... *	Selecting the reading mode: • Interval The set time is called "Interval time" • On Change The set time is called "Debounce time"
Set interval / debounce time in milliseconds... *	Specify the interval and/or debounce time
→ ② New Datapoint	
Type datapoint name... *	Name of the datapoint
Type datapoint address... *	Address of the datapoint
Select data type... *	Selecting the data type: • Integer • Float • String • Bool
④ Define alarm configuration / Optional	
Type new MTConnect condition ...	ID for MTConnect alarm condition
⑤ Define advanced configuration / optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path... *	Path for MTConnect docker image
Type username to access the image path...	User name to access the image path

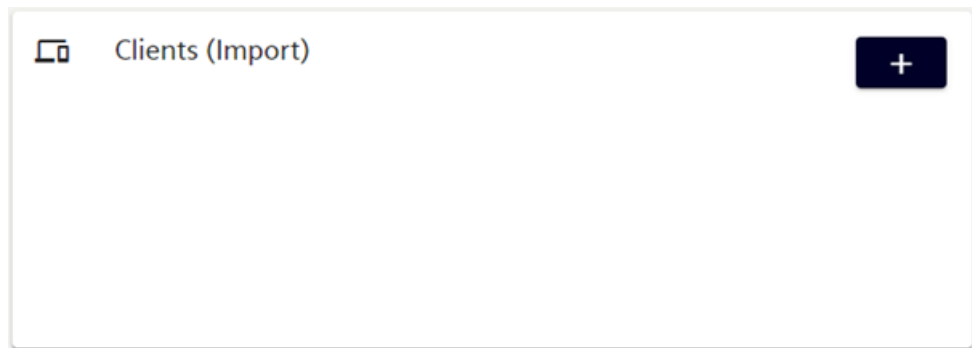
Parameter	Description
Type password to access the image path...	Password to access the image path
Set sampling time in milliseconds... *	MTConnect agent sampling time
Set request timeout in milliseconds... *	Maximum wait time to access the data
Type new environment variable...	New environment variable for the image

*: Obligatory data

Pattern: The character sequence to be used for the various entries is provided in the input windows.

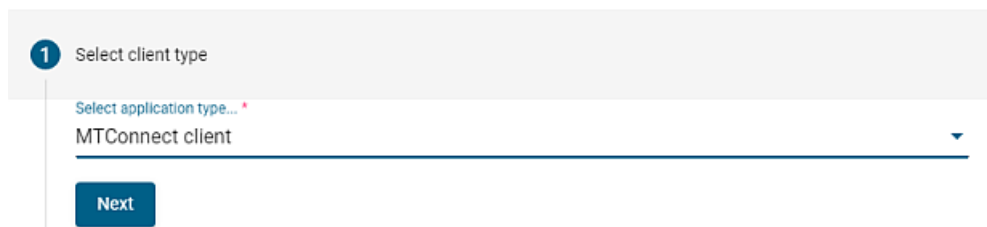
Procedure

1. In the "Client (Import)" area, click on "+" to add a new client.



2. Step 1: "Select client type"
 - From the drop-down list, select entry "MTConnect client".
 - Click on "Next".

Add new client



3. Step 2: "Define basic configuration"
 - Populate the fields to perform the basic configuration.
 - Click on "Next".

The screenshot shows the 'Define basic configuration' step, which is the second step in the process. It contains several input fields for configuring the MTConnect agent:

- Type client ID... ***: The field contains 'mtconnect02'.
- Type client description...**: The field contains 'mtconnecttest'. A note above the field states 'Supported input: lowercase letters, numbers, - (must start with letter)' and '11 / 50'.
- Type MTConnect agent URL... ***: The field contains '192.168.0.90'. A note to the right indicates '13 / 100'.
- Type MTConnect agent username...**: This field is empty.
- Type MTConnect agent password...**: This field is empty.

At the bottom left of the form is a blue 'Next' button. At the bottom right are two buttons: a grey 'Save' button and a blue 'Close' button with an 'X' icon.

4. Step 3: "Define dataset configuration" / ① New Dataset
 - Populate the fields under "New Dataset" to configure the dataset.
 - Click on "Configure datapoints" to configure the datapoints belonging to the dataset.

The screenshot shows the 'Define dataset configuration' step, which is the third step in the process. It is divided into two sections:

- ① New Dataset: mtconnecttestset**
 - Type dataset name... ***: The field contains 'mtconnecttestset'.
 - Select reading mode... ***: A dropdown menu is set to 'On Change'. A note above the field states 'Supported input: letters, numbers, - (must start with letter)' and '16 / 50'.
 - Set debounce time in milliseconds... ***: The field contains '5000'. A note to the right indicates 'Range: 200-86400000'.
- ② New Datapoint:**

A blue 'Configure datapoints' button is located below the first section. At the bottom right are two buttons: a grey 'Save' button and a blue 'Close' button with an 'X' icon.

5. Step 3: "Define dataset configuration" / ② New Datapoint

- Populate the fields under "New Datapoint" to configure the datapoints belonging to the dataset.

Note**Configured datasets**

You can use the "Import" function to import configured datasets.

More information can be found in Chapter: Using configured datasets (Page 206).

- Click on "Add current datapoint" to add the current datapoint to the dataset.

The screenshot displays the 'Define dataset configuration' window, which is divided into two main sections. The top section, labeled '1 New Dataset: mtconnecttestset', contains a sub-section '2 New Datapoint: Operation-Mode'. This sub-section includes three input fields: 'Type datapoint name...' with the value 'Operation-Mode', 'Type datapoint address...' with the value 'z21', and 'Select data type...' with the value 'Integer'. A blue button labeled 'Add current datapoint' is positioned below these fields. The bottom section of the window, labeled 'No datapoints configured yet', contains a blue button labeled 'Import'. At the very bottom of the window, there is a 'Next' button and a close button (X). The interface is clean and modern, with a light gray background and blue accents for buttons and text.

6. Step 3: "Define dataset configuration"

- Populate the fields under "New Datapoint" to configure additional datapoints belonging to the dataset.
- Click on "Add current dataset" to add the current dataset.
- Click on "Next".

More information can be found in Chapter Defining a dataset configuration (Page 198).

The screenshot shows the 'Define dataset configuration' step (Step 3) in the BFC Gateway interface. It features a progress bar with two steps: '1 New Dataset: mtconnecttestset' and '2 New Datapoint:'. The 'New Datapoint' section includes input fields for 'Type datapoint name...' (with a character count of 0/50), 'Type datapoint address...', and a dropdown for 'Select data type...' currently set to 'Integer'. Below these is an 'Add current datapoint' button. A table shows one configured datapoint: 'Operation-Mode' at address 'z21' of type 'int'. At the bottom, there are buttons for 'Add current dataset', 'No datasets configured yet', and 'Import'. A vertical scrollbar is on the right, and a close button is at the bottom right.

1 New Dataset: mtconnecttestset

2 New Datapoint:

Type datapoint name... *

Supported input: letters, numbers, - (must start with letter) 0 / 50

Type datapoint address... *

Select data type... *

Integer

Add current datapoint

1 datapoint(s) configured - do not forget to add dataset!

Datapoint name	Address	Type
Operation-Mode	z21	int

Add current dataset

No datasets configured yet Import

7. Step 4: "Define alarm configuration"

- Configure the additional conditions for the MTConnect client.
- Click on "Next".

The screenshot shows the 'Define alarm configuration' step (Step 4) in the BFC Gateway interface. It features a progress bar with four steps, with the fourth step '4 Define alarm configuration' (Optional) being the active one. Below the progress bar is a text input field for 'Type new MTConnect condition...' containing the text 'cloudc'. A blue '+' icon is to the right of the input field. At the bottom, there is a 'Next' button.

4 Define alarm configuration
Optional

Type new MTConnect condition...

cloudc

Next

8. Step 5: "Advanced configuration" / Optional

- As a rule, no entries/changes are required.

Note**Advanced configuration**

Entries and changes are only made in the case of service, and only performed by the hotline.

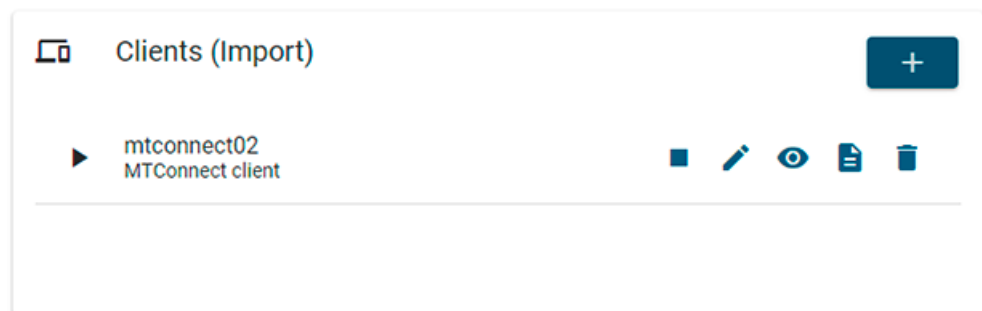
- Click on "Save" to save the MTConnect client.

The screenshot shows a web form titled "5 Define advanced configuration" with a sub-header "Optional". The form contains several input fields and a list of environment variables:

- Type image path... ***: `apc-docker-registry.dlinkddns.com/apc-koe/mts/iotserver/go-mtconnect:latest`
- Type username to access the image path...**: (empty)
- Type password to access the image path...**: (empty)
- Set sampling time in milliseconds... ***: `200` (Range: 200-86400000)
- Set request timeout in milliseconds... ***: `30000` (Range: 5000-86400000)
- Type new environment variable...**: (empty) with a "+" icon to add more.

At the bottom right, there are two buttons: a blue "Save" button with a floppy disk icon and a blue "Cancel" button with an "X" icon.

9. The MTConnect client was successfully created, and is shown in the overview in the "Clients (Import)" area.



6.4.3.4 Creating a Modbus client

A Modbus client can exchange data (reading and writing) with a Modbus TCP server.

Perform the following steps to create a Modbus client:

- Step 1: "Select client type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define whitelisting configuration"
- Step 5: "Define advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select client type	
Select application type... *	Selection of the client
② Define basic configuration	
Type client ID... *	ID of the Modbus client
Type client description...	Description of the machine
Type Unit ID... *	ID of the module
Type Modbus server URL... *	URL or IP address of the Modbus server
③ Define dataset configuration	
→ ① New Dataset	
Type dataset name... *	Dataset name
Select reading mode... *	Selecting the reading mode: • Interval The set time is called "Interval time" • On Change The set time is called "Debounce time"
Set interval / debounce time in milliseconds... *	Specify the interval and/or debounce time
→ ② New Datapoint	
Type datapoint name... *	Name of the datapoint
Type datapoint address... *	Address of the datapoint
Select data type... *	Selecting the data type: • Integer • Float • Bool
④ Define whitelisting configuration	
Type whitelisted address... *	Address of the Modbus register
Select whitelisted type	Data type of the address
⑤ Define advanced configuration / optional	
Remark:	
These fields do not require any entries, and are only populated by the hotline in the case of service.	

Parameter	Description
Type image path...*	Path for Modbus docker image
Type username to access the image path...	User name to access the image path
Type password to access the image path...	Password to access the image path
Type a new environment variable...	New environment variable for the image

*: Obligatory data

Pattern: The character sequence to be used for the various entries is provided in the input windows.

Modbus functions

The Modbus client supports the following reading Modbus functions:

- (0x01) Read Coils
- (0x02) Read Discrete Inputs
- (0x03) Read Holding Registers
- (0x04) Read Input Registers

Using these functions, numerical and Boolean values can be read from the Modbus device.

Addressing is based on the following schematic:

{function code}:{register number},{register count}

The following function codes can be used:

- c - (0x01) Read Coils
- d - (0x02) Read Discrete Inputs
- h - (0x03) Read Holding Registers
- i - (0x04) Read Input Registers

The Modbus client supports the following writing Modbus functions:

- (0x05) Write Single Coil
- (0x06) Write Single Holding Register
- (0x16) Write Multiple Holding Registers

Using these functions, numerical and Boolean values can be written to the Modbus device.

Addressing is based on the following schematic:

{function code}:{register number},{register count}

The following function codes can be used:

- c - (0x05) Write Single Coil
- h - (0x06) Write Single Holding Register oder (0x16) Write Multiple Holding Registers

Data types

The following data types are supported:

- **32-bit float**
Addressing example:
Read or write a 32-bit float variable in the 42nd Holding Register
h:42.2
- **64-bit float**
Addressing example:
Read or write a 64-bit float variable in the 42nd Holding Register
h:42.4
- **16-bit integer**
Addressing example:
Read or write a 16-bit integer variable in the 42nd Holding Register
h:42.1
- **32-bit integer**
Addressing example:
Read or write a 32-bit integer variable in the 42nd Holding Register
h:42.2
- **64-bit integer**
Addressing example:
Read or write a 64-bit integer variable in the 42nd Holding Register
h:42.4
- **bool**
Addressing example:
Read or write a Boolean value in the 42nd Coil
c:42.1
Read or write a Boolean value in the 42nd Discrete Input
d:42.1

Note

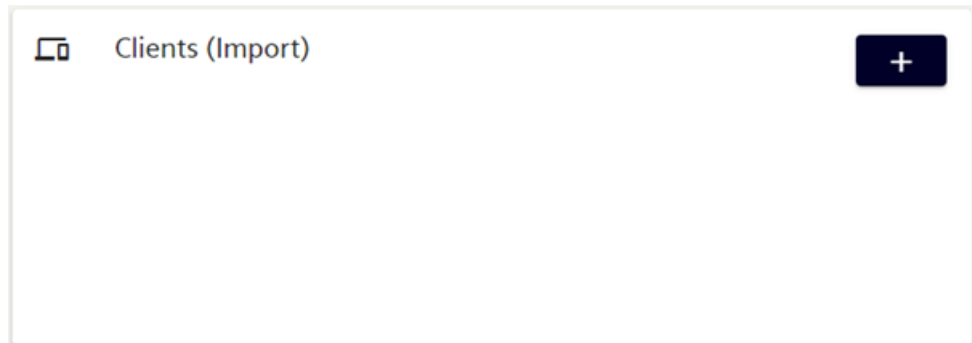
Read out Boolean value from the holding register of a Modbus device that does not support the Modbus functions "Read Coils" and "Read Discrete Inputs"

- Populate the register with "0" or "1", and read out the values as integer.
- The value can be converted into a Boolean value using the scriptlogic.

Contact the hotline if you have any questions or require more information.

Procedure

1. In the "Client (Import)" area, click on "+" to add a new client.



2. Step 1: "Select client type"
 - From the drop-down list select entry "Modbus client".
 - Click on "Next".

Add new client

1 Select client type

Select application type... *

Modbus client

Next

3. Step 2: "Define basic configuration"
 - Populate the fields to perform the basic configuration.
 - Click on "Next".

2 Define basic configuration

Type client ID... *

modbus2

Supported input: lowercase letters, numbers, - (must start with letter) 7 / 50

Type client description...

Type Unit ID... *

1

0 / 100

Type Modbus server URL... *

192.168.0.90.502

Range: 0 or greater

Next

4. Step 3: "Define dataset configuration" / ① New Dataset

- Populate the fields under "New Dataset" to configure the dataset.

Note

Configured datasets

You can use the "Import" function to import configured datasets.

More information can be found in Chapter: Using configured datasets (Page 206).

- Click on "Configure datapoints" to define the datapoints belonging to the dataset.

3 Define dataset configuration

1 New Dataset: main

Type dataset name... *
main

Select reading mode... *
On Change

Set debounce time in milliseconds... *
500

Supported input: letters, numbers, - (must start with letter) 4 / 50

Range: 200-86400000

Configure datapoints

2 New Datapoint:

No datasets configured yet **Import**

Next

4 Define advanced configuration
Optional

Save **Close**

5. Step 3: "Define dataset configuration" / ② New Datapoint

- Populate the fields under "New Datapoint" to configure a datapoint belonging to the dataset.
- Click on "Add current datapoint" to add the data to the datapoint.

3 Define dataset configuration

1 New Dataset: main

2 New Datapoint: Voltage

Type datapoint name... *

Voltage

Type datapoint address... * Supported input: letters, numbers, - (must start with letter) 7 / 50

h:9,1

Select data type... *

Float

Add current datapoint

No datapoints configured yet

No datasets configured yet Import

Next

Save Close

6. Step 3: "Define dataset configuration"

- Populate the fields under "New Datapoint" to configure additional datapoints belonging to the dataset.
- Click on "Add current dataset" to add the complete dataset.
- Click on "Next".

More information can be found in Chapter Defining a dataset configuration (Page 198).

1 New Dataset: main

2 New Datapoint:

Type datapoint name... *

Supported input: letters, numbers, - (must start with letter) 0 / 50

Type datapoint address... *

Select data type... *

Integer

Add current datapoint

1 datapoint(s) configured - do not forget to add dataset!

Datapoint name	Address	Type
Voltage	h:9,1	float

Add current dataset

No datasets configured yet

Import

7. Step 4: "Define whitelisting configuration"

- Populate field "Type whitelisted address" to define the address of the Modbus register that may be written to using the client.
The address format corresponds to the format for the read configuration.

Add new client

The screenshot shows a vertical sequence of five steps in a configuration wizard:

- 1 Select client type
- 2 Define basic configuration
- 3 Define dataset configuration
- 4 Define whitelisting configuration
Optional
- 5 Define advanced configuration
Optional

Step 4 is the active step. It contains a text input field labeled "Type whitelisted address..." and a dropdown menu labeled "Select whitelisted type..." with "Integer" selected. A blue "+" button is next to the dropdown. At the bottom right, there are two buttons: a grey "Save" button and a blue "Cancel" button with a white "X" icon.

- From the drop-down list "Select whitelisted type" select the address of the data type.

- Press the "+" button at the end of the newly created line to release additional addresses.

Add new client

1

Select client type

2

Define basic configuration

3

Define dataset configuration

4

Define whitelisting configuration

Optional

Whitelisted address 1	h:9,2	Select whitelisted type...	Float	
Whitelisted address 2	h:20,1	Select whitelisted type...	Integer	
Type whitelisted address...	c:1,1	Select whitelisted type...	Bool	+

5

Define advanced configuration

Optional

Save

Cancel

8. Step 5: "Define advanced configuration" / Optional

- As a rule, no entries/changes are required.

Note**Advanced configuration**

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click on "Save" to save the Modbus client.

4 Define advanced configuration
Optional

Type image path... *

apc-docker-registry.dlinkddns.com/apc-koe/mts/iotserver/modbus-service:latest

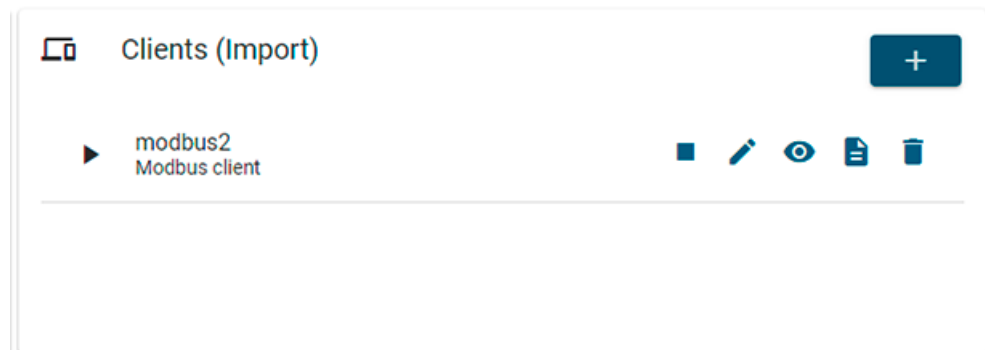
Type username to access the image path...

Type password to access the image path...

Type new environment variable... +

9. The Modbus client was successfully created, and shown in the overview in the "Clients (Import)" area.



6.4.3.5 Creating an OPC UA client

An OPC UA client can read data from a OPC UA server.

Perform the following steps to create an OPC UA client:

- Step 1: "Select client type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define whitelisting configuration"

6.4 "Commissioning" area

- Step 5: "Define alarm configuration"
- Step 6: "Define advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select client type	
Select application type...*	Selection of the client
② Define basic configuration	
Type client ID...*	ID of the OPC UA client
Type client description...	Description of the machine
Type OPC UA server URL...*	URL or IP address of the OPC UA server
Select OPC UA server security mode...	Selection of the security mode of the OPC UA server: • None • Sign • SignAndEncrypt
Select OPC UA server security policy...	Selection of the encryption of the OPC UA server: • None • Basic256 • Basic256Sha256 • Basic128Rsa15 • Aes256_Sha256_RsaPss • Aes128_Sha256_RsaOaep
Type OPC UA security cert...	OPC UA client X509 certificate PEM string
Type OPC UA security key...	OPC UA client X509 certificate key PEM string
Select OPC UA server user identification...	User identification for the OPC UA server: • Username / Password: user name and password • Anonymous: anonymous access • Certificate: certificate
Type OPC UA server username	User name for the OPC UA server
Type OPC UA server password	Password for the OPC UA server
Type OPC UA server ident cert...	X509 certificate as PEM string for user identification
Type OPC UA server ident key....	X509 certificate key as PEM string for user identification
③ Define dataset configuration	
→ ① New Dataset:	
Type dataset name...*	Dataset name

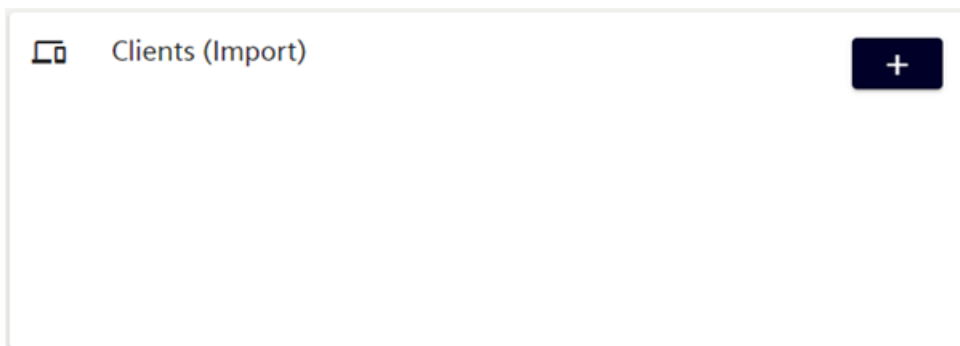
Parameter	Description
Select reading mode... *	Selecting the reading mode: - Interval The set time is called "Interval time" - On Change The set time is called "Debounce time"
Set interval / debounce time in milliseconds... *	Specify the interval and/or debounce time
→ ② New Datapoint:	
Type datapoint name... *	Name of the datapoint
Type datapoint address... *	Address of the datapoint
Select data type... *	Selecting the data type: - Integer - Float - String - Bool
④ Define alarm configuration / Optional	
Type default alarm number...	OPC UA standard alarm number
Type alarm property name...	OPC UA alarm node name
Type alarm property address...	OPC UA alarm node address
Type new OPC UA alarm address...	OPC UA alarm address
⑤ Define whitelisting configuration / optional	
Type whitelisted address...	List of OPC UA addresses that may be written to
⑥ Define advanced configuration / optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path... *	Path for OPC UA client docker image
Type username to access the image path...	User name to access the image path
Type password to access the image path...	Password to access the image path
Type new environment variable...	New environment variable for the image

*: Obligatory data

Pattern: The character sequence to be used for the various entries is provided in the input windows.

Procedure

1. In the "Client (Import)" area, click on "+" to add a new client.



2. Step 1: "Select client type"
 - From the drop-down list select entry "OPC UA client".
 - Click on "Next".



3. Step 2: "Define basic configuration"

- Populate the fields to perform the basic configuration.
- Click on "Next".

The screenshot shows the 'Define basic configuration' step of the commissioning process. The form includes the following fields and options:

- Type client ID... ***: `opcuaclient` (Supported input: letters, numbers, -, _ (must start with letter or number) 11 / 50)
- Type client description...**: (Empty field, 0 / 100)
- Type OPC UA server URL... ***: `opc.tcp://192.168.0.90:4840`
- Select OPC UA server security mode...**: `None` (dropdown menu)
- Select OPC UA server security policy...**: `None` (dropdown menu)
- Select OPC UA server user identification...**: `Anonymous` (dropdown menu)

A 'Next' button is located at the bottom left of the form. At the bottom right, there are two icons: a save icon (floppy disk) and a close icon (X).

Note

If the parameter "Username/Password" is selected in the field "Select OPC UA server user identification", a valid user name and password must be entered in the fields "Type OPC UA server username" and "Type OPC UA server password".

If the parameter "Certificate" is selected in the field "Select OPC UA server user identification", a valid X509 certificate must be stored as PEM string in the field "Type OPC UA server ident cert" and a valid key in the field "Type OPC UA server ident key".

If a value other than "None" is selected in the field "Select OPC UA server security mode" or "Select OPC UA server security policy", a valid X509 certificate must be stored as PEM string in the field "Type OPC UA security cert" and a valid key in the field "Type OPC UA security key".

Note**Creating an X509 certificate**

The OPC UA client uses X509 certificates. These certificates can be generated with the "openssl" tool, for example:

1. `openssl genrsa -out test.key -des 1024`
2. `openssl req -new -x509 -key test.key -out openssl_cert.pem -outform pem`
3. `openssl x509 -in openssl_cert.pem -inform pem -out openssl_cert.der -outform der`

4. Step 3: "Define dataset configuration" / ① New Dataset

- Populate the fields under "New Dataset" to configure the dataset.

Note

Configured datasets

You can use the "Import" function to import configured datasets.

More information can be found in Chapter: Using configured datasets (Page 206).

- Click on "Configure datapoints" to configure the datapoints belonging to the dataset.

Define dataset configuration

1 New Dataset: opcudata

Type dataset name... *

opcudata

Select reading mode... * Supported input: letters, numbers, -, _ (must start with letter) 9 / 50

On Change

Set debounce time in milliseconds... * 500

Configure datapoints

Range: 200-86400000

2 New Datapoint: Operation-Mode

5. Step 3: "Define dataset configuration" / ② New Datapoint

- Populate the fields under "New Datapoint" to configure the datapoint belonging to the dataset.
- Click on "Add current datapoint" to add the current datapoint to the dataset.

Define dataset configuration

1 New Dataset: testset

2 New Datapoint: Operation-Mode

Type datapoint name... *

Operation-Mode

Type datapoint address... * Supported input: letters, numbers, -, _ (must start with letter) 14 / 50

ns=2;i=5486

Select data type... * Integer

Add current datapoint

6. Step 3: "Define dataset configuration"

- Populate the fields under "New Datapoint" to configure additional datapoints belonging to the dataset.
- Click on "Add current dataset" to add the current dataset.
- Click on "Next" to save the dataset.

More information can be found in Chapter Defining a dataset configuration (Page 198).

opcuaData On Change (500) This dataset has 1 datapoints

Type dataset name...
opcuaData

Select reading mode... *
On Change

Set debounce time in milliseconds... *
500

Range: 200-86400000

1 datapoint(s) configured

Datapoint name	Address	Type
Operation-Mode	ns=2; i=1234	int

Type datapoint name... *

Type datapoint address... *

Supported input: letters, numbers, -, _ (must start with letter) 0 / 50

Save Cancel

7. Step 4: "Define alarm configuration"

- Fill in the fields.
- Click on "Next".

4 Define alarm configuration
Optional

Type default alarm number...
0

Range: 0 or greater

Type alarm property name...
Alarm

Type alarm property address...
ns=2;s=AlarmNumber

Type new OPC UA alarm address...
ns=2;s=AlarmSource

Next

Save Cancel

8. Step 5: "Define whitelisting configuration"

- Fill in the fields.

9. Step 6: "Define advanced configuration" / Optional

- As a rule, no entries/changes are required.

Note

Advanced configuration

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click on "Save" to save the OPC UA client.

10. The OPC UA client was successfully created, and is shown in the overview in the "Clients (Import)" area.

6.4.3.6 Creating an S7 client

The S7 client enables the reading of inputs, outputs, memories and data blocks.

Perform the following steps to create an S7 client:

- Step 1: "Select client type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select client type	
Select application type...*	Selection of the client
② Define basic configuration	
Type client ID...*	ID of the S7 client
Type client description...	Description of the machine
Type S7 device IP...*	IP v4 address of the PLC Remark: According to the configuration in the Step 7 project
Type number of rack...*	Rack and slot:
Type number of slot...*	The slot of the addressed CPU at the back panel/ backplane bus Default values: • S7 300: Rack 0, Slot 2 • S7 1200: Rack 0, Slot 0 • S7 1500: Rack 0, Slot 0 Remark: These parameters are ignored if the RemoteT-SAP is specified directly.
Type S7 device port...*	TCP port number A port that deviates from the standard port TCP 102 for RFC1006 can be configured. This is only necessary in special network infrastructure cases with firewall or NAT/port forwarding.

Parameter	Description
Type S7 device password...	Password - Up to 8 characters - ASCII characters Remark: The password must at least allow read access. This functionality is not supported by all SIMATIC PLCs for the S7 basic protocol used, as it only provides weak protection.
③ Define dataset configuration	
→ ① New Dataset:	
Type dataset name... *	Dataset name
Select reading mode... *	Selecting the reading mode: - Interval The set time is called "Interval time" - On Change The set time is called "Debounce time"
Set interval / debounce time in milliseconds... *	Specify the interval and/or debounce time
→ ② New Datapoint	
Type datapoint name... *	Name of the datapoint
Type datapoint address... *	Address of the datapoint
Select data type... *	Selecting the data type: - Integer - Float - String - Bool The numerical values may be returned deviating from the addressed data type. Addressed character strings, byte arrays and S7 strings must be configured as "String". Remark: If you configure a number as data type "Bool", the return value is "true" if the number read is not equal to 0. This allows you to determine, for example, whether at least one bit of a byte, word or DWord is set.
④ Define advanced configuration / Optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path... *	Path for S7 UA client docker image
Type username to access the image path...	User name to access the image path
Type password to access the image path...	Password to access the image path

Parameter	Description
Type connection type...*	Specification of the device type Default: 0 (The predefined default value is used.) Remark: The parameter specifies the device type with which the S7 client reports to the PLC. Values: 1: PG (standard) Connection takes place as PG 2: OP/HMI Connection takes place as operator panel 3-16: S7 Basic
Type connection timeout in milliseconds...*	Maximum time for establishment of a connection Default: 0 (The predefined default value is used.) The maximum time for a connection establishment of the S7 client to the PLC in milliseconds. The range covers 10 ... 5000 ms Remark: If the PLC is offline, the value influences the cycle of the connection establishment. If no connection is established, the S7 client always waits 5 seconds until the next connection attempt. With a maximum timeout of 5000 ms, a new connection establishment is started every 10 seconds. The predefined default value of 750 ms is optimized for access to the PLC via a LAN connection.
Type send timeout in milliseconds...*	Maximum time for sending the request PDU to the PLC Default: 0 (The predefined default value is used.) Remark: The predefined default value of 10 ms is optimized for access to the PLC via a LAN connection.
Type receive timeout in milliseconds...*	Maximum time to receive the response PDU from the PLC Default: 0 (The predefined default value is used.) Remark: The predefined default value of 3000 ms is optimized for access to the PLC via a LAN connection.

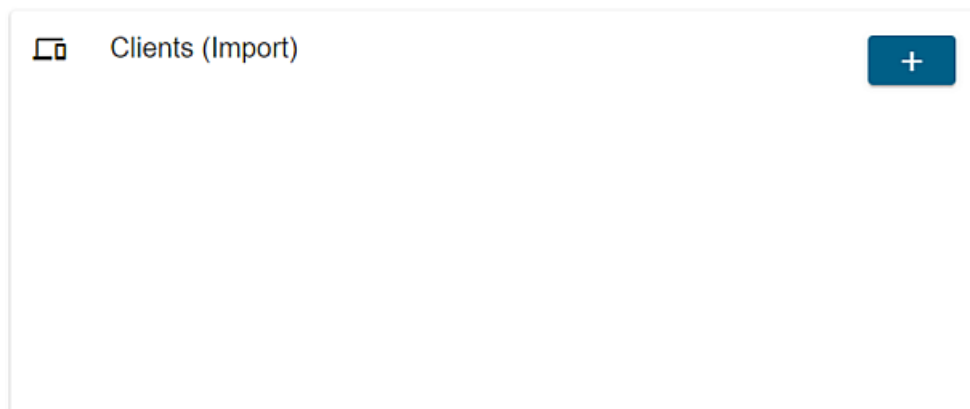
Parameter	Description
Type PDU request size...*	Default value for PDU negotiation with the PLC. Default: 0 (The predefined default value is used.) You can specify for the S7 client what maximum PDU size it will use for negotiation. Value range: 0, 240, 480 or 960 bytes If there are problems with large PDU sizes in the network or in the S7 client, you can specify a smaller value for the connection setup. The default value of the S7 client is 480 bytes. You can increase this value for the PLC SIMATIC S7-1500. A PDU size of 240 limits the maximum usable addresses in a reading set to 19 items.
Type number of remoteTSAP...*	Alternatively to the rack+slot configuration you can directly configure the TSAP of the addressed CPU. Use the decimal notation: • 0x100 = 256 or • 0x102 = 258 Default values: • S7-300: RemoteTSAP: 258, LocalTSAP: 256 • S7-1200: RemoteTSAP: 256, LocalTSAP: 256 • S7-1500: RemoteTSAP: 256, LocalTSAP: 256 The LocalTSAP may differ from the specified values. If the RemoteTSAP is configured with a value greater than 0, the Rack and Slot parameters are ignored.
Type number of localTSAP...*	
Type new environment variable...	New environment variable for the image With the environment variable TRACE=TRUE you can activate the extended diagnostic output of the S7 client. A very large number of messages is generated. Therefore, deactivate the mode after commissioning.

*: Obligatory data

Pattern: The character sequence to be used for the various entries is provided in the input windows.

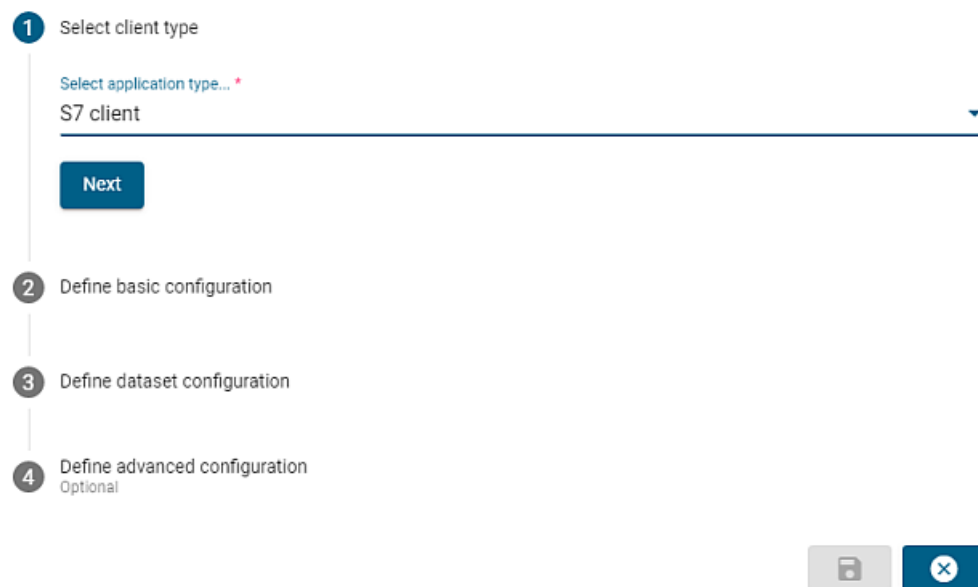
Procedure

1. In the "Client (Import)" area, click on "+" to add a new client.



2. Step 1: "Select client type"
 - Select the entry "S7 client" from the drop-down list.
 - Click on "Next".

Add new client



1 Select client type

Select application type... *

S7 client

Next

2 Define basic configuration

3 Define dataset configuration

4 Define advanced configuration
Optional

Save Close

3. Step 2: "Define basic configuration"
 - Populate the fields to perform the basic configuration.
 - Click on "Next".

Add new client

1 Select client type

2 Define basic configuration

Type client ID... *

Supported input: lowercase letters, numbers, - (must start with letter) 4 / 50

Type client description...

0 / 100

Type S7 device IP... *

192.168.1.1

Type number of rack... *

0

Range: 0 or greater

Type number of slot... *

1

Range: 0 or greater

Type S7 device port... *

102

Range: 0 or greater

Type S7 device password...

4 / 8

Next

4. Step 3: "Define dataset configuration" / ① New Dataset
 - Populate the fields under "New Dataset" to configure the dataset.
 - Click on "Configure datapoints" to configure the datapoints belonging to the dataset.

Add new client

The screenshot shows the 'Define dataset configuration' step, which is the third step in a three-step process. The steps are: 1. Select client type, 2. Define basic configuration, and 3. Define dataset configuration. The third step is currently active and contains two sub-steps: 1. New Dataset: s7-data and 2. New Datapoint:.

1 New Dataset: s7-data

Type dataset name... *
s7-data

Select reading mode... *
On Change

Supported input: letters, numbers, - (must start with letter) 7 / 50

Set debounce time in milliseconds... *
500

Range: 200-86400000

Configure datapoints

2 New Datapoint:

No datasets configured yet **Import**

At the bottom right, there are two buttons: a save icon and a close icon (X).

5. Step 3: "Define dataset configuration" / ② New Datapoint

- Populate the fields under "New Datapoint" to configure the datapoints belonging to the dataset.
- Click on "Add current datapoint" to add the current datapoint to the dataset.

Add new client

3 Define dataset configuration

1 New Dataset: s7-data

2 New Datapoint: db-byte

Type datapoint name... *

db-byte

Type datapoint address... *

DB19.DBB1

Select data type... *

Integer

Add current datapoint

No datapoints configured yet

No datasets configured yet Import

6. Step 3: "Define dataset configuration"

- Populate the fields under "New Datapoint" to configure additional datapoints belonging to the dataset.
- Click on "Add current dataset" to save the dataset.

More information can be found in Chapter Defining a dataset configuration (Page 198).

Add new client

3

Define dataset configuration

1

New Dataset: s7-data

2

New Datapoint:

Type datapoint name... *

Supported input: letters, numbers, - (must start with letter) 0 / 50

Type datapoint address... *

Select data type... *

Integer

Add current datapoint

1 datapoint(s) configured - do not forget to add dataset!

Datapoint name	Address	Type
db-byte	DB19.DBB1	int

Add current dataset

7. Step 4: "Define advanced configuration" / Optional

- As a rule, no entries/changes are required.

Note

Advanced configuration

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click on "Save" to save the S7 client.

Add new client

4

Define advanced configuration

Optional

Type image path... *

apc-docker-registry.dlinkddns.com/apc-koe/mts/iotserver/s7-service:latest

Type username to access the image path...

Type password to access the image path...

Type connection type... *

0

Range: 0-16

Set connection timeout in milliseconds... *

0

Range: 0-5000

Type send timeout in milliseconds... *

0

Range: 0-5000

Type receive timeout in milliseconds... *

0

Range: 0-5000

Type PDU request size... *

0

Range: 0, 240, 480 or 960

Type number of remoteTSAP... *

0

Range: 0 or greater

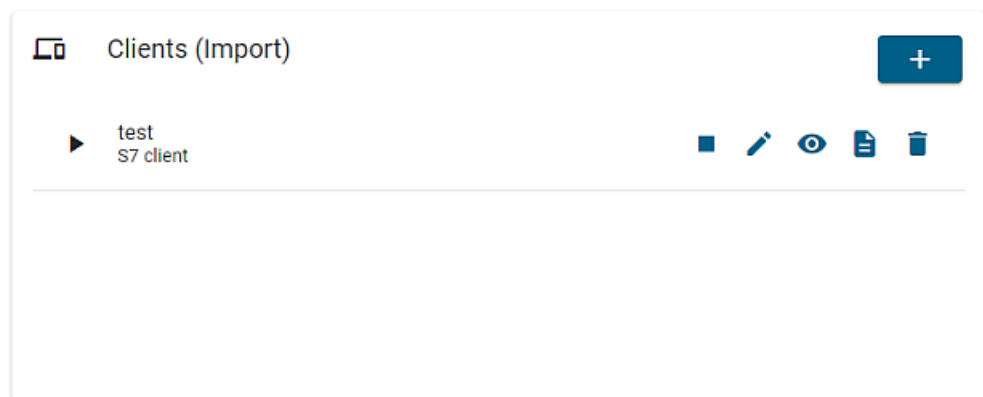
Type number of localTSAP... *

0

Save

Cancel

8. The S7 client has been successfully created and is shown in the overview in the "Clients (Import)" area.



6.4.3.7 Creating an HTTP REST client

The HTTP REST client enables the reading of inputs, outputs, memories and data blocks.

To create an HTTP REST client, perform the following steps:

- Step 1: "Select client type"
- Step 2: "Define basic configuration"

Requirement

The "Commissioning" area is open.

Parameter

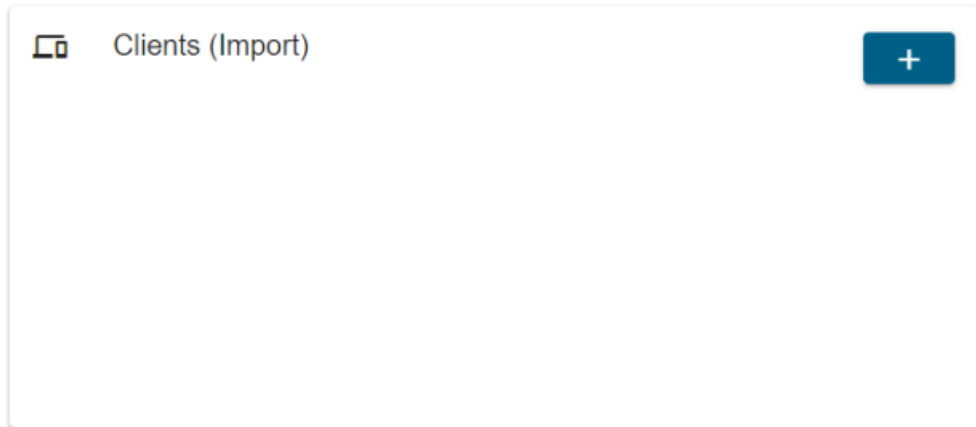
Parameter	Description
① Select client type	
Select application type... *	Selection of the client
② Define basic configuration	
Type client ID... *	ID of the HTTP REST client
Skills... *	Selection of the skills of the HTTP REST client Possible skills: • com.siemens.bfc.skills.basic.publish.reading.sets • com.siemens.bfc.skills.basic.publish.alarm.lists • com.siemens.bfc.skills.advanced.publish.hfdata.sets Select at least one skill.

*: Obligatory data

Pattern: The character sequence to be used for the various entries is provided in the input windows.

Procedure

1. In area "Client (Import)" click on "+" to add a new client.



2. Step 1: "Select client type"
 - From the drop-down list, select the "HTTP REST client" entry
 - Click on "Next".

Add new client



3. Step 2: Define basic configuration
 - Populate the fields to perform the basic configuration.
 - Click on "Save" to save the HTTP REST client.

Add new client "http-rest"

1 Select client type

2 Define basic configuration

Type client ID... *

http-rest

Supported input: letters, numbers, -, _ 9 / 50

- ☒ com.siemens.bfc.skills.basic.publish.reading.sets
- ☒ com.siemens.bfc.skills.basic.publish.alarm.lists
- ☒ com.siemens.bfc.skills.advanced.publish.hfdata.sets

Save Cancel

4. The HTTP REST client has been successfully created and is shown in the "Clients (Import)" area of the overview.
 - Click on the "Show credentials" icon to read out the access data.

Clients (Import)

http-rest
HTTP REST client

Help Key Edit Eye Delete

6.4.3.8 Creating an HTTP script client

The HTTP script client can address HTTP-based services, prepare the data with JavaScript functions and provide them in the BFC Gateway as alarms and datasets.

A predefined JavaScript for HP 3D printer API version 1.2 is provided to acquire alarm messages, the printer status, and the active print job status.

Perform the following steps to create an HTTP script client:

- Step 1: "Select client type"
- Step 2: "Define basic configuration"

- Step 3: "Define dataset configuration"
- Step 4: "Define script configuration"
- Step 5: "Define advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select client type	
Select target store..*	Selection of the client
② Define basic configuration	
Type client ID...*	Identifier for the HTTP script client
Type client description...	Description of the client
Type HTTP URL...*	URL of the HTTP server Is read in JavaScript with the function getRemoteUrl() The address of the HP 3D Control Center system
Type HTTP login name...	User name for the HTTP server Is read in JavaScript with the function getRemoteData("login")
Type HTTP login password...	Password for the HTTP server Is read in JavaScript with the function getRemoteData("password")
Type name of device...	If multiple devices are made available via an HTTP server, this name can be used to uniquely identify the device. Is read in JavaScript with the function getRemoteData("name") Specify the printer name as it is configured in the HP 3D Control Center.
Upload or type CA cert of HTTP Server...	CA certificate for establishing an encrypted connection to the target system Is read in JavaScript with the function getRemoteData("ca-certs")
③ Define dataset configuration	
→ ① New Dataset:	
Type dataset name...*	Dataset name This configuration can be read with the getReadingSetConfigs() function in JavaScript. Use the "MachineInfo", "JobInfo" or "Alarms" values to read out the predefined datasets for HP 3D printers.

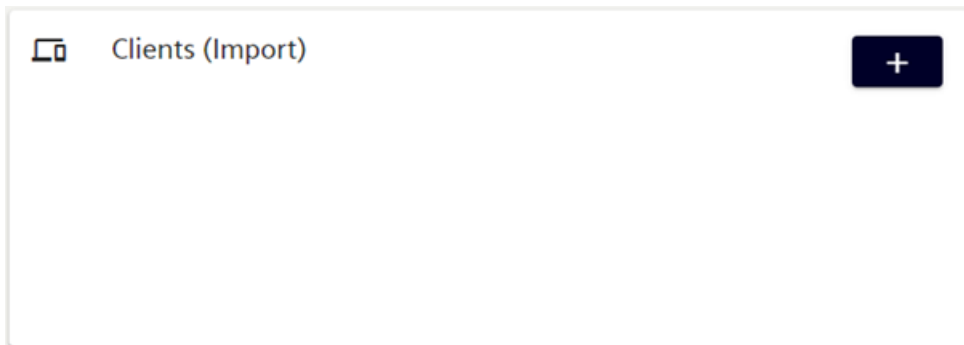
Parameter	Description
Select reading mode...*	Selecting the reading mode: - Interval Each read value is processed on the internal data bus in the specified interval. - On Change The data is read at the configured interval, but is only processed further if at least one value in the dataset has changed.
Set interval / debounce time in milliseconds...*	Specify the interval and/or debounce time Deviations of the actual reading time are caused by reading intervals that are too short, the network, and by the operating system.
→ ② New Datapoint:	
Type datapoint name...*	Name of the datapoint When using the predefined datasets, these parameters are ignored. Specify any values.
Type datapoint address...	Address of the datapoint
Select data type...*	Selecting the data type: - Integer - Float - String - Bool
④ Define script configuration	
Type script...	Script with which the client is started Note: A script for HP 3D printers is used as default value.
⑤ Define advanced configuration / optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path...*	Path to the software image
Type username to access the image path...	User name to access the image path
Type password to access the image path...	Password to access the image path
Type language of HTTP results...	For language-specific texts, the output language can be specified here, e.g.: 'en-US', 'de-DE'. For the addressed HP 3D printer, the available languages are output in the log.
Skip verification of CA cert...*	If you select "true", any server certificate will be accepted. Remark: Do not use in production if possible.
Type logic url...	The logic is stored in an environment variable by default. If the logic is external, it can be included here.

Parameter	Description
Type logic config...	Any JSON structure can be specified here for further parameterization of the JavaScript code. The data can be read using the getConfig() function in the JavaScript code. Example: <pre>{ "httpTimeout":15000 }</pre>
Type new environment variable	New environment variable for the image (e.g.: IOT_WILD-BROKERURI)

*: Obligatory data

Procedure

1. In the "Client (Import)" area, click on "+" to add a new client.



2. Step 1: "Select client type"
 - Select the "HTTP script client" entry from the drop-down list.
 - Click on "Next".



3. Step 2: "Define basic configuration"

- Populate the fields to perform the basic configuration.
- Click on "Next".

2 Define basic configuration

Type client ID... *
testHP

Type client description... Supported input: letters, numbers, -, _ (must start with letter or number) 6 / 50
test HP desc

Type HTTP URL... * 12 / 100
http://my-test:1234

Type HTTP login name...
admin

Type HTTP login password...

Type name of device...
my-machine

Upload CA cert of HTTP Server

Upload or type CA cert of HTTP Server...
-----BEGIN CERTIFICATE-----
MIIFXzCCA0egAwIBAgIBATANBgkqhkiG9w0BAQsFADBQMQswCQYDVQQGEwJERTET
MBEGA1UEChMKU2llbWVucyBBRzEQMA4GA1UECzMHVBDIE1UUzEaMBGGA1UEAxMR
QkZDIENBIDlwMjEtMDYtMTcwIENBcNMjEwNjE3MTIzNzI5WhgPMjkzMzEyMTcxMjM3
MjlaMFaxCzAJBgNVBAYTAkRFRMRMwEQYDVQQKEwpTaWVtZW50IEFHMRAwDgYDVQQL
EwdBUeMgTVRTMR0wGAYDVQQDExFCRkMgQ0EgMjAyMS0wNi0xNzCCAILwDQYJKoZI
hvcNAQEBBQADggIPADCCAggIBAO4J8uh7kp7TWEOPD90yOj9HIVOdgd04jde
WNm8J+IDg2txtGQ4dD4mQTPug5BBcc7WTAmb9CYXevl0ZqopWpKjZGaTCUVk+rx8

Next

4. Step 3: "Define dataset configuration" / ① New Dataset
 - Populate the fields under "New Dataset" to configure the dataset.
 - Click on "Configure datapoints" to configure the datapoints belonging to the dataset.

③ Define dataset configuration

① New Dataset: my-set

Type dataset name... *

my-set

Select reading mode... *

Interval

Supported input: letters, numbers, -, _ . (must start with letter) 6 / 50

Set interval time in milliseconds... *

500

Range: 200-86400000

Configure datapoints

② New Datapoint:

No datasets configured yet Import

Next

5. Step 3: "Define dataset configuration" / ② New Datapoint

- Populate the fields under "New Datapoint" to configure the datapoints belonging to the dataset.

Note

Configured datasets: You can use the "Import" function to import configured datasets. More information can be found in Chapter: Using configured datasets (Page 206).

- Click on "Add current datapoint" to add the current datapoint to the dataset.

3 Define dataset configuration

1 New Dataset: my-set

2 New Datapoint: my-point

Type datapoint name... *

my-point

Type datapoint address... *

address1

Supported input: letters, numbers, -, _ (must start with letter) 8 / 50

Select data type... *

Float

Add current datapoint

No datapoints configured yet

No datasets configured yet **Import**

Next

6. Step 3: "Define dataset configuration"

- Populate the fields under "New Datapoint" to configure additional datapoints belonging to the dataset.
- Click on "Add current dataset" to add the current dataset.
- Click on "Next".

More information can be found in Chapter Defining a dataset configuration (Page 198).

3 Define dataset configuration

1 New Dataset: my-set

2 New Datapoint:

Type datapoint name... *

Supported input: letters, numbers, -, _ (must start with letter) 0 / 50

Type datapoint address... *

Select data type... *

Float

Add current datapoint

1 datapoint(s) configured - do not forget to add dataset!

Datapoint name	Address	Type	
my-point	address1	float	

Add current dataset

No datasets configured yet

Import

Next

7. Step 4: "Define script configuration"

- Configure the script for the HTTP script client.
- Click on "Next".

4 Define script configuration

```
1 // used API functions: uncomment while editing to reduce jslint warnings
2 // function startIntervalWorker(){}
3 // function doHTTP(){}
4 // function publishConnectionState(){}
5 // function parseXML(){}
6 // function publishAlarmList(){}
7 // function publishReadingSet(){}
8 // function getRemoteUrl(){}
9 // function getRemoteData(){}
10 // function getReadingSetConfigs(){}
11 // function getConfig(){}
12
13
14 // standard helper functions
15 // add a header element to a header object (eg.: var h={});
16 function addHeader(h,o,v){h[a] = v;return h;}
17 // add a basic authentication header element to a header object
18 function addBasicAuthHeader(h,u,p){return addHeader(h, "Authorization", "Basic " + btoa(u + ':' + p));}
19 // add a bearer authentication header element to a header object
20 function addBearerAuthHeader(h,t){return addHeader(h, "Authorization", "Bearer " + t);}
21 // add a jwt-token authentication header element to a header object
22 function addTokenAuthHeader(h,t){return addHeader(h, "Authorization", t);}
23
24 // get the first element node from an xmlDocument, the result of a successfull parseXML()
25 // a well formatted XML document starts with a declaration tag, may have empty string elements etc.
26 function getXMLRootElement(xmlDoc){
27     try
28     {
29
30
```

Next



8. Step 5: "Define advanced configuration" / Optional

- As a rule, no entries/changes are required.

Note

Advanced configuration

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click the "Save" button to save the HTTP script client.

9. The HTTP script client was successfully created and is shown in the overview in the "Clients (Import)" area.

6.4.3.9 Creating a Heidenhain client

A Heidenhain client can read and write data from/to a Heidenhain control system.

Perform the following steps to create a Heidenhain client:

- Step 1: "Select client type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define alarm configuration"
- Step 5: "Define advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select client type	
Select application type... *	Selection of the client
② Define basic configuration	
Type client ID... *	ID of the Heidenhain client
Type client description...	Description of the machine
Type Heidenhain control URL... *	URL or IP address of the Heidenhain control system (with port, standard: 19000)
③ Define dataset configuration	
→ ① New Dataset: main	
Type dataset name... *	Dataset name
Select reading mode... *	Selecting the reading mode: • Interval The set time is called "Interval time" • On Change The set time is called "Debounce time"
Set interval / debounce time in milliseconds... *	Specify the interval and/or debounce time
→ ② New Datapoint:	
Type datapoint name... *	Name of the datapoint
Type datapoint address... *	Address of the datapoint
Select data type... *	Selecting the data type: • Integer • Float • String • Bool
④ Define alarm configuration / Optional	
Set alarm sample time in milliseconds...	Alarm sampling time in milliseconds

Parameter	Description
⑤ Define whitelisting configuration / optional	
Type whitelisted address...	Address of the datapoint for which writing is permitted.
Select whitelisted type...	Selecting the data type: • Integer • Float • String • Bool
⑥ Define advanced configuration / optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path...*	Path for Heidenhain Docker image
Type username to access the image path...	User name to access the image path
Type password to access the image path...	User name to access the image path
Set request timeout in seconds...*	Maximum wait time to access the data
Do not swap bytes...*	As standard, the bytes are swapped with Word and DWord data (setting: false). To prevent this, select "true".
Do logout after every PLC-Area operation...*	As standard, no logout is performed during operations in the PLC area; setting: "false". If the logout is to be performed, select "true" here.
Specify custom configuration for Heidenhain connection...	Standard connection parameters are used for the connection to the Heidenhain machine; setting: empty. For parameter changes, set the required parameters in this field. Separate the parameters with a semicolon.
Type new environment variable...	New environment variable for the image
Type file system address...	Internal address for program transfer As a rule, no changes are required.

*: Obligatory data

Pattern: The character sequence to be used for the various entries is provided in the input windows.

For information on configuring access rights for file operations, see Chapter: Configuring access rights (Page 467).

Datapoints

The datapoint of the datasets must have valid addresses.

Address positions	Example	Explanation
Start	hh	Address for the Heidenhain API
1st position	getruninfo	Function name

Address positions	Example	Explanation
Additional positions	Execution_mode	Specification as to which variables should be called
	...	
End	[x]	The input parameters required for the particular function are in square brackets.

More information regarding the supported addresses is provided in Appendix: Heidenhain client data points (reading) (Page 574).

The grouping is realized after the API call. For each address, the data type to be parameterized is defined.

Note**API call structure**

Siemens defined the addresses used here.

You can find more information on the Heidenhain-specific addresses in the official Heidenhain documentation.

Example

/hh/getruninfo/execution_mode

Procedure

1. In the "Client (Import)" area, click on "+" to add a new client.



2. Step 1: "Select client type"
 - Select the entry "Heidenhain client" from the drop-down list.
 - Click on "Next".

Add new client

1 Select client type

Select application type... *

Heidenhain client

Next

3. Step 2: "Define basic configuration"
 - Populate the fields to perform the basic configuration.
 - Click on "Next".

2 Define basic configuration

Type client ID... *

heidenhain1

Type client description...

heidenhain machine 1

Type Heidenhain control URL... *

192.168.0.2:19000

Next

4. Step 3: "Define dataset configuration" / ① New Dataset

- Populate the fields under "New Dataset" to configure the dataset.

Note**Configured datasets**

You can use the "Import" function to import configured datasets.

More information can be found in Chapter: Using configured datasets (Page 206).

- Click on "Configure datapoints" to define the datapoints belonging to the dataset.

3 Define dataset configuration

1 New Dataset: main

Type dataset name... *

main

Select reading mode... *

On Change

Set debounce time in milliseconds... *

500

Supported input: letters, numbers, - (must start with letter) 4 / 50

Range: 200-86400000

Configure datapoints

2 New Datapoint:

No datasets configured yet Import

Next

5. Step 3: "Define dataset configuration" / ② New Datapoint

- Populate the fields under "New Datapoint" to configure the datapoints belonging to the dataset.
- Click on "Add current datapoint" to add the data to the datapoint.

3 Define dataset configuration

1 New Dataset: main

2 New Datapoint: executionmode

Type datapoint name... *

executionmode

Type datapoint address... * Supported input: letters, numbers, - (must start with letter) 13 / 50

/hh/getruninfo/execution_mode

Select data type... *

String

Add current datapoint

No datapoints configured yet

No datasets configured yet **Import**

6. Step 3: "Define dataset configuration"

- Populate the fields under "New Datapoint" to configure additional datapoints belonging to the dataset.
- Click on "Add current dataset" to add the complete dataset.

More information can be found in Chapter Defining a dataset configuration (Page 198).

2 New Datapoint:

Type datapoint name... *

Supported input: letters, numbers, - (must start with letter) 0 / 50

Type datapoint address... *

Select data type... *

Integer

Add current datapoint

1 datapoint(s) configured - do not forget to add dataset!

Datapoint name	Address	Type
executionmode	/hh/getruninfo/execution_mode	string

Add current dataset

No datasets configured yet

Import



7. Step 4: "Define alarm configuration"

- Configure the alarm sample time.
- OR - Keep the "Default value".
- Click on "Next".

4 Define alarm configuration

Optional

Set alarm sample time in milliseconds...

10000

Range: 5000-86400000

Next

6.4 "Commissioning" area

8. Step 5: "Define whitelisting configuration" / optional

- Populate the fields under "Whitelisted address" with the addresses that can be written to, and select the appropriate data type.

5 Define whitelisting configuration
Optional

Type whitelisted address...	Select whitelisted data type
<u>/hh/setmachineparameters/MP3515</u>	Integer

9. Step 6: "Define advanced configuration" / Optional

- As a rule, no entries/changes are required.

Note**Advanced configuration**

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click on "Save" to save the Heidenhain client.

6

Define advanced configuration
Optional

Type image path... *

bfc-registry.com/apc-koe/mts/iotserver/hh-service:v1.1.0

Type username to access the image path...

Type password to access the image path...

Set request timeout in milliseconds... *

15000

Range: 5000-86400

Do not swap bytes... *

false

Do logout after every PLC-Area operation... *

false

Specify custom configuration for Heidenhain connection...

Type new environment variable...



10. The Heidenhain client has been successfully created and is shown in the "Clients (Import)" area of the overview.



heidenhain1

bfc-registry.com/apc-koe/mts/iotserver/hh-service:v1.1.0



6.4.3.10 Creating a Beckhoff client

A Beckhoff client can read and write data from/to a Beckhoff control system.

Perform the following steps to create a Beckhoff client:

- Step 1: "Select client type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select client type	
Select application type... *	Selection of the client
② Define basic configuration	
Type client ID... *	ID of the Beckhoff client
Type Beckhoff control URL... *	URL or IP address of the Beckhoff control system
Type local route to BFC Gateway... *	Default routing URL or IP address to the BFC Gateway
③ Define dataset configuration	
→ ① New Dataset: main	
Type dataset name... *	Dataset name
Select reading mode... *	Selecting the reading mode: • Interval The set time is called "Interval time" • On Change The set time is called "Debounce time"
Set interval / debounce time in milliseconds... *	Specify the interval and/or debounce time
→ ② New Datapoint:	
Type datapoint name... *	Name of the datapoint
Type datapoint address... *	Address of the datapoint
Select data type... *	Selecting the data type
④ Define whitelisting configuration / optional	
Type whitelisted address...	Address of the datapoint for which writing is permitted.

Parameter	Description
Select whitelisted type...	Selecting the data type: • Integer • Float • String • Bool
⑤ Define advanced configuration / optional Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	

*: Obligatory data

Procedure

1. In the "Client (Import)" area, click on "+" to add a new client.



2. Step 1: "Select client type"
 - Select the entry "Beckhoff client" from the drop-down list.
 - Click on "Next".



3. Step 2: "Define basic configuration"
 - Populate the fields to perform the basic configuration.
 - Click on "Next".

The screenshot shows the 'Define basic configuration' step, which is the second step in the process. It contains four input fields:

- 'Type client ID ... *' with the value 'beckhoff' and a character count of 8 / 50. A note below indicates 'Supported input: letters, numbers, -, _ (must start with letter or number)'.
- 'Type client description ...' with a character count of 0 / 100.
- 'Type Beckhoff control URL ... *' with the value '192.168.3.175'.
- 'Type local route to BFC Gateway ... *' with the value '192.168.3.89.1.1'.

 A blue 'Next' button is located at the bottom left of the form.

4. Step 3: "Define dataset configuration" / ① New Dataset
 - Populate the fields under "New Dataset" to configure the dataset.

Note

Configured datasets

You can use the "Import" function to import configured datasets.

More information can be found in Chapter: Using configured datasets (Page 206).

- Click on "Configure datapoints" to define the datapoints belonging to the dataset.

The screenshot shows the 'Define dataset configuration' step, which is the third step in the process. It is divided into two main sections:

- 1 New Dataset: main**: This section contains three input fields:
 - 'Type dataset name... *' with the value 'main' and a character count of 4 / 50. A note below indicates 'Supported input: letters, numbers, - (must start with letter)'.
 - 'Select reading mode... *' with the value 'On Change' and a dropdown arrow.
 - 'Set debounce time in milliseconds... *' with the value '500' and a range note 'Range: 200-86400000'.
 A blue 'Configure datapoints' button is located below these fields.
- 2 New Datapoint:**: This section is currently empty.

 At the bottom of the form, there is a status message 'No datasets configured yet' and a blue 'Import' button. A grey 'Next' button is located at the very bottom left.

5. Step 3: "Define dataset configuration" / ② New Datapoint

- Populate the fields under "New Datapoint" to configure the datapoints belonging to the dataset.
- Click on "Add current datapoint" to add the data to the datapoint.

1 New Dataset: main

2 New Datapoint: Counter

Type datapoint name... *

Counter

Type datapoint address... *

AMSPORT_R0_PLC_TC3.MAIN Counter4

Select data type... *

Integer

Add current datapoint

No datapoints configured yet

No datasets configured yet Import

6. Step 3: "Define dataset configuration"

- Populate the fields under "New Datapoint" to configure additional datapoints belonging to the dataset.
- Click on "Add current dataset" to add the complete dataset.

More information can be found in Chapter Defining a dataset configuration (Page 198).

2 New Datapoint:

Type datapoint name... *

Type datapoint address... *

Select data type... *

Integer

Add current datapoint

1 datapoint(s) configured - do not forget to add dataset!

Datapoint name	Address	Type
Counter	AMSPORT_R0_PLC_TC3.MAIN Counter4 int	int

Add current dataset

No datasets configured yet Import

7. Step 4: "Define whitelisting configuration" / optional

- Populate the fields under "Whitelisted address" with the addresses that can be written to, and select the appropriate data type.

8. Step 5: "Define advanced configuration" / Optional

- As a rule, no entries/changes are required.

Note

Advanced configuration

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click on the "Save" button to save the Beckhoff client.

9. The Beckhoff client has been successfully created and is shown in the overview in the "Clients (Import)" area.



Defining data point addresses

The data points of the data sets must have valid addresses.

Addressing via index and offset as well as via symbolic names is supported.

Data points

Address positions	Example	Explanation
Start	{port}	The used port address range of the Beckhoff control system (name or number)
Separation	:	Separator
End	{index}:{offset}	Addressing via index and offset (separated by ":")
	{symbol}	Addressing via a symbolic name

More information regarding the supported addresses can be found under Data points Beckhoff client (Page 581).

Note

Structure of the addresses

You can find information on Beckhoff-specific addresses in the official Beckhoff documentation.

Examples

```
AMSPORT_R0_PLC:21312:10
801:21312:10
AMSPORT_R0_PLC:MAIN.Counter4
```

6.4.3.11 Creating an Ethernet IP client

The EIP client permits reading and writing of system and program tags or data in memory areas (PCCC). Addressing of the data depends on the control system family.

Perform the following steps to create an EIP client:

- Step 1: "Select client type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define whitelisting configuration"
- Step 5: "Define advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select client type	
Select application type...*	Selection of the client
② Define basic configuration	
Type client ID...*	Identifier for the Ethernet IP client
Type client description...	Description of the client
Type Ethernet IP URL...*	TCP/IP address of the EIP device Optionally, the IP address and port can be configured (e.g.: 192.168.1.15:44818) If no port is specified, then the default value 44818 is used.
Select device family...	Selection of device family or compatibility value (e.g.: specify the value "controllogix" for CompactLogix, ControlLogix)
Type path...	This value is only required for the "controllogix" device family and specifies the CPU slot (e.g.: "1,0").
③ Define dataset configuration	
→ ① New Dataset:	
Type dataset name...*	Dataset name
Select reading mode...*	Selecting the reading mode: - Interval Each read value is processed on the internal data bus in the specified interval. - On Change The data is read at the configured interval, but is only processed further if at least one value in the dataset has changed.
Set interval / debounce time in milliseconds...*	Specify the interval and/or debounce time Deviations of the actual reading time are caused by reading intervals that are too short, the network, and by the operating system.
→ ② New Datapoint:	
Type datapoint name...*	Name of the datapoint
Type datapoint address...*	Address of the datapoint
Select data type...*	Selecting the data type: - Integer - Float - String - Bool
④ Define whitelisting configuration	
Type whitelisted address...	The address that is released for write operations (see EIP client (Page 582))
Select whitelisted type...	Data type of the address

Parameter	Description
⑤ Define advanced configuration / optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path... *	Path to the software image
Type username to access the image path...	User name to access the image path
Type password to access the image path...	Password to access the image path
Type Protocol type...	A default value "ab-eip" can be configured.
Set connect timeout in milliseconds...	Maximum time of connection establishment to the device (default value: 300 ms) After a connection is lost, an additional five seconds is waited before the next connection attempt.
Set send timeout in milliseconds...	Permissible time to send data to the control system synchronously (default value: 1000 ms).
Set receive timeout in milliseconds...	Permissible time to read data from the control system synchronously (default value: 3000 ms)
Type a new environment variable...	New environment variable for the image
Special environment parameters to reduce warning messages:	
NOIDENTITY	For devices that support the reading of the CIP Identity object, the device identifier is read once when the connection is established and logged (example: Device Info: Rockwell Automation/ Allen-Bradley 1763-L16AWA B/16.0). To disable the read attempt of the Identity object, you set the environment variable: NOIDENTITY=TRUE
FORCEPOLLING	The EIP client independently updates the data from the control system during the read interval. This permits optimized read access. If the device should have problems with this reading mode, the client can be set to strict polling. Then the values are successively read from the device. To activate strict polling, set the environment variable: FORCEPOLLING=TRUE

*: Obligatory data

** : By setting the environment variables, the configured read interval may not be adhered to, because no read optimization (simultaneous reading of several variables) can be performed.

Functions

The EIP client can communicate with Allen Bradley and compatible control systems via the Ethernet Industrial Protocol.

The client supports cyclic reading of data via "Dataset". In addition, direct reading and writing of data in the control system is possible.

The data can be read as single values and arrays. Write operations are only permitted for single values.

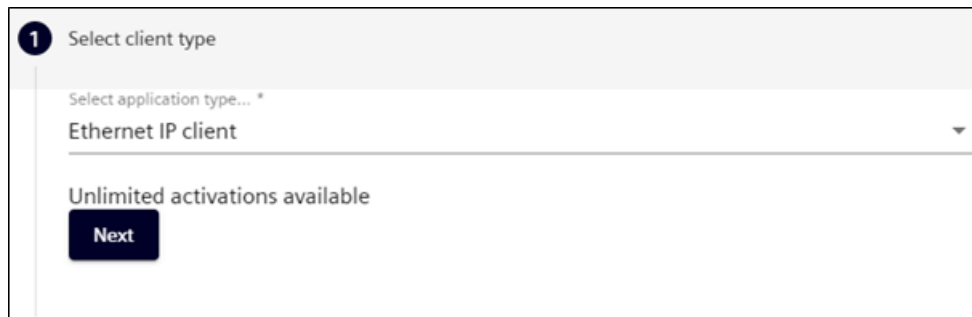
The addressable data and the structure of the addresses depend on the device. Details can be found at EIP client (Page 582).

Procedure

1. In the "Client (Import)" area, click on "+" to add a new client.



2. Step 1: "Select client type"
 - Select the entry "Ethernet IP client" from the drop-down list.
 - Click on "Next".



3. Step 2: "Define basic configuration"
 - Populate the fields to perform the basic configuration.
 - Click on "Next".

2

Define basic configuration

Type client ID... *

micrologix

Supported input: letters, numbers, -, _ (must start with letter or number) 10 / 50

Type client description...

Allen Bradley Micrologix 1100

29 / 100

Type Ethernet IP URL... *

192.168.5.164

Select device family...

micrologix

▼

Type path...

Next

4. Step 3: "Define dataset configuration" / ① New Dataset
 - Populate the fields under "New Dataset" to configure the dataset.

Note

Configured datasets

You can use the "Import" function to import configured datasets.

More information can be found in Chapter: Using configured datasets (Page 206).

- Click on "Configure datapoints" to define the datapoints belonging to the dataset.

1 New Dataset: numbers

Type dataset name... *
numbers

Select reading mode... * Supported input: letters, numbers, -, _ . (must start with letter) 7 / 50
On Change

Set debounce time in milliseconds... *
500

Range: 200-86400000

Configure datapoints

2 New Datapoint:

No datasets configured yet **Import**

Next

5. Step 3: "Define dataset configuration" / ② New Datapoint

- Populate the fields under "New Datapoint" to configure the datapoints belonging to the dataset.
- Click on "Add current datapoint" to add the data to the datapoint.

The screenshot displays the 'Define dataset configuration' interface. At the top, a progress bar shows three steps: '1 New Dataset: numbers', '2 New Datapoint: MyIntegerNumber' (the current step), and '3 Define dataset configuration'. The 'New Datapoint' section contains three input fields: 'Type datapoint name...' with the value 'MyIntegerNumber', 'Type datapoint address...' with the value 'N7:0', and 'Select data type...' with the value 'Integer'. A blue button labeled 'Add current datapoint' is positioned below these fields. At the bottom, the text 'No datapoints configured yet' is visible.

3 Define dataset configuration

1 New Dataset: numbers

2 New Datapoint: MyIntegerNumber

Type datapoint name... *

MyIntegerNumber

Type datapoint address... *

N7:0

Select data type... *

Integer

Add current datapoint

No datapoints configured yet

6. Step 3: "Define dataset configuration"

- Populate the fields under "New Datapoint" to configure additional datapoints belonging to the dataset.
- Click on "Add current dataset" to add the complete dataset.
- Click on "Next".

More information can be found in Chapter Defining a dataset configuration (Page 198).

Add current datapoint

2 datapoint(s) configured - do not forget to add dataset!

Datapoint name	Address	Type	
MyIntegerNumber	N7:0	int	
MyFloatNumber	F8:0	float	

Add current dataset

No datasets configured yet

7. Step 4: "Define whitelisting configuration"

- Fill in the "Type whitelisted address" field to specify the address that may be written using the client.

The address format corresponds to the format for the read configuration.

4 Define whitelisting configuration
Optional

Whitelisted address 1
N7:1
Select whitelisted ty... Integer

Whitelisted address 2
F8:1
Select whitelisted ty... Float

Type whitelisted address... Integer +

- Select the data type of the address in the "Select whitelisted type" selection box.
- Press the "+" button at the end of the newly created line to release additional addresses.

8. Step 6: "Define advanced configuration" / Optional

- As a rule, no entries/changes are required.

Note**Advanced configuration**

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click on the "Save" button to save the client configuration.

Note

Empty fields as well as zero values for the timeouts lead to the use of the default values.

9. The Ethernet IP client was successfully created and is shown in the overview in the "Clients (Import)" area.

6.4.3.12 Creating an Omron client

An Omron client can read and write data from/to an Omron control system.

Perform the following steps to create an Omron client:

- Step 1: "Select client type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define whitelisting configuration"
- Step 5: "Define advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select client type	
Select application type... *	Selection of the client
② Define basic configuration	
Type client ID... *	ID of the Omron client
Type client description...	Description of the machine
Type Omron control URL... *	URL or IP address of the Omron control (with port, standard: 9600)
Type source network... *	Network ID of the local system
Type source node... *	Node ID of the local system
Type source unit... *	Unit ID of the local system
Type destination network... *	Network ID of Omron control
Type destination node.... *	Node ID of Omron control
Type destination unit... *	Unit ID of Omron control
③ Define dataset configuration	
→ ① New Dataset: main	
Type dataset name... *	Dataset name
Select reading mode... *	Selecting the reading mode: • Interval The set time is called "Interval time" • On Change The set time is called "Debounce time"
Set interval / debounce time in milliseconds... *	Specify the interval and/or debounce time
→ ② New Datapoint:	
Type datapoint name... *	Name of the datapoint
Type datapoint address... *	Address of the datapoint

Parameter	Description
Select data type...*	Selecting the data type: • Integer • Float • String • Bool
④ Define whitelisting configuration / optional	
Type whitelisted address...	Address of the datapoint for which writing is permitted.
Select whitelisted type...	Selecting the data type: • Integer • Float • String • Bool
⑤ Define advanced configuration / optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path...*	Path for Omron Docker Image
Type username to access the image path...	User name to access the image path
Type password to access the image path...	Password to access the image path
Set request timeout in seconds...*	Maximum wait time to access the data
Type new environment variable...	New environment variable for the image

*: Obligatory data

Pattern: The character sequence to be used for the various entries is provided in the input windows.

Procedure

1. In the "Client (Import)" area, click on "+" to add a new client.



2. Step 1: "Select client type"
 - Choose the item "Omron client" from the drop-down list.
 - Click on "Next".



3. Step 2: "Define basic configuration"

- Populate the fields to perform the basic configuration.
- Click on "Next".

2 Define basic configuration

Type client ID... *
omron-test
Supported input: letters, numbers, -, _ (must start with letter or number) 10 / 50

Type client description...
0 / 100

Type Omron control URL... *
192.168.1.100

Type source network... *
0
Range: 0-255

Type source node... *
1
Range: 0-255

Type source unit... *
0
Range: 0-255

Type destination network... *
0
Range: 0-255

Type destination node... *
0
Range: 0-255

Type destination unit... *
0
Range: 0-255

Next

4. Step 3: "Define dataset configuration" / ① New Dataset

- Populate the fields under "New Dataset" to configure the dataset.

Note

Configured datasets

You can use the "Import" function to import configured datasets.

More information can be found in Chapter: Using configured datasets (Page 206).

- Click on "Configure datapoints" to define the datapoints belonging to the dataset.

3 Define dataset configuration

1 New Dataset: main

Type dataset name... *

main

Select reading mode... *

On Change

Set debounce time in milliseconds... *

500

Range: 200-86400000

Configure datapoints

2 New Datapoint:

No datasets configured yet

Import

Next

5. Step 3: "Define dataset configuration" / ② New Datapoint

- Populate the fields under "New Datapoint" to configure the datapoints belonging to the dataset.
- Click on "Add current datapoint" to add the data to the datapoint.

1 New Dataset: test

2 New Datapoint: test

Type datapoint name... *

test

Type datapoint address... *

cioword:160:1

Select data type... *

Integer

Add current datapoint

No datapoints configured yet

No datasets configured yet Import

6. Step 3: "Define dataset configuration"

- Populate the fields under "New Datapoint" to configure additional datapoints belonging to the dataset.
- Click on "Add current dataset" to add the complete dataset.

More information can be found in Chapter Defining a dataset configuration (Page 198).

2 New Datapoint:

Type datapoint name... *

Type datapoint address... *

Select data type... *

Integer

Add current datapoint

1 datapoint(s) configured - do not forget to add dataset!

Datapoint name	Address	Type
test	cioword:160:1	int

Add current dataset

No datasets configured yet Import

7. Step 4: "Define whitelisting configuration" / optional

- Populate the fields under "Whitelisted address" with the addresses that can be written to, and select the appropriate data type.

8. Step 5: "Define advanced configuration" / Optional

- As a rule, no entries/changes are required.

Note

Advanced configuration

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click the "Save" button to save the Omron client.

9. The Omron client was successfully created and is shown in the overview in the "Clients (Import)" area.

 **omron-test**
bfc-registry.com/apc-koe/mts/iotserver/omron-service:v1.0.2

Defining data point addresses

The data points of the datasets must have valid addresses.

Addressing according to the pattern described below is supported.

Data points

Address positions	Example	Explanation
Beginning	{address}	The address range used of the Omron control (name)
Separation	:	Separation
End	{index}	Addressing via the address index
	{index,bit-number}	Addressing via the address index and bit number
	{index:length}	Addressing via the address index and length (separated by ":")

You will find more information about the supported addresses under Data points of the Omron client (Page 586).

Examples

- `dmword:1`
- `ciobit:200,0`
- `cioword:130:10`

6.4.3.13 Creating an MQTT client

An MQTT client is used for logging on external MQTT clients.

Perform the following steps to create an MQTT client:

- Step 1: "Select client type"
- Step 2: "Define basic configuration"

Requirement

The "Commissioning" area is open.

Parameter

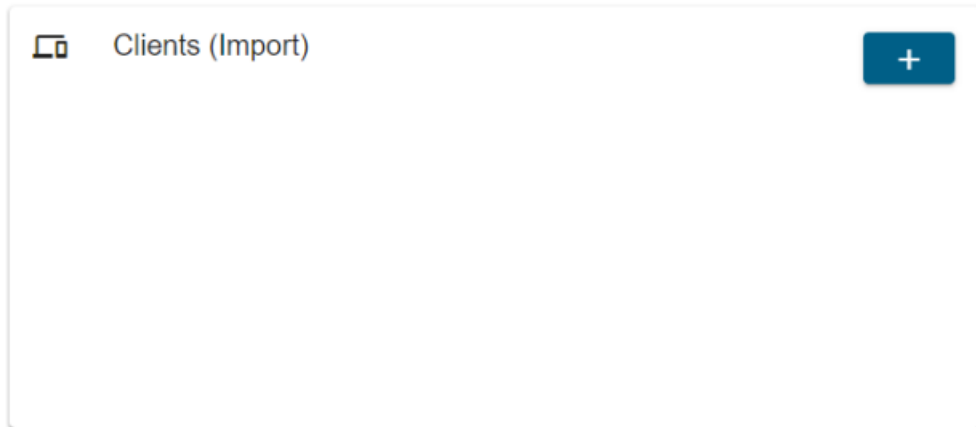
Parameter	Description
① Select client type	
Select application type...*	Selection of the client
② Define basic configuration	
Type client ID...*	ID of the MQTT client

*: Obligatory data

Pattern: The character sequence to be used for the various entries is provided in the input windows.

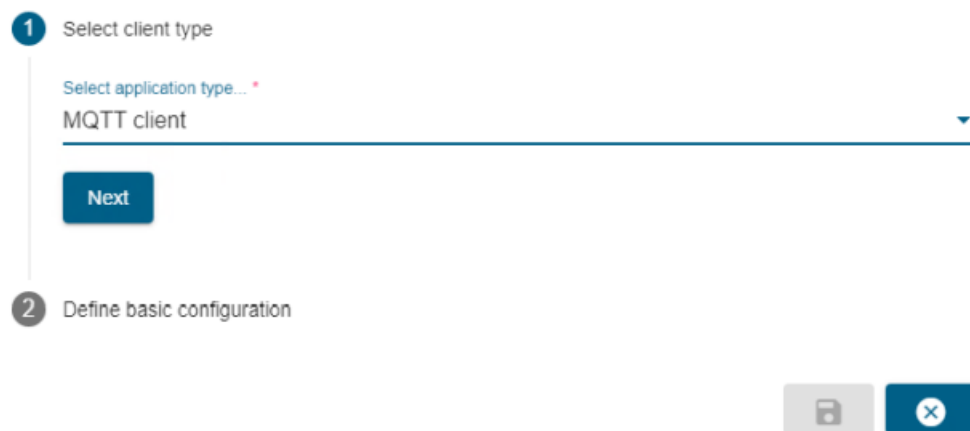
Procedure

1. In area "Client (Import)" click on "+" to add a new client.



2. Step 1: "Select client type"
 - Select the "MQTT client" entry from the drop-down list.
 - Click on "Next".

Add new client



3. Step 2: Define basic configuration
 - Populate the fields to perform the basic configuration.
 - Click on "Save" to save the MQTT client.

Add new client "machine9-voltage"

1 Select client type

2 Define basic configuration

Type client ID... *

machine9-voltage

Supported input: letters, numbers, -, _ 16 / 50

Save Cancel

4. The MQTT client has been successfully created and is shown in the "Clients (Import)" area of the overview.



6.4.3.14 MQTT client - configuring scriptlogic

Before you send data to the MQTT client and can edit it in the system, you must make this data available internally.

To do this, you must configure a script logic.

Note

To be able to process additional write jobs, you must configure an additional script logic or modify the existing one. An example of this script logic is provided in the scope of delivery of the BFC Gateway under "bfc/docker/scriptlogic/samples/processwritejob.js". The MQTT client must confirm write jobs before their lifetime expires.

Replace the "ClientID" specified in line 22 in the example: "801ae1a4-e219-4ce5-9c44-2663c5a57d51" by the MQTT client ID generated by the BFC Gateway when the corresponding MQTT client was created.

Note

Thoroughly test the source code generated when developing script logics and verify it before productive use.

Parameters

Parameters	Description
① Select middleware type	
Select middleware type...*	Selection of the middleware
① Define basic configuration	
Type scriptlogic name...*	ID of the script logic
Type middleware logic	Executing logic in JavaScript
④ Define wild data configuration	
Type a new wild data topic...	Topic of the external MQTT client processed in the logic in the "onWildData(state, message)" function
⑤ Define write configuration	
Pick a client ID	Client ID of the write jobs to be processed

*: Obligatory data

Pattern: The character string to be used for the respective entries can be found in the input windows

Procedure

1. "Select middleware type"

- In the "Middlewares (Logic)" area, click on "+" to add a middleware.

↑ Middlewares (Logic)



- Select the entry "scriptlogic" from the drop-down list.
- Click on "Next".

Add new middleware

1 Select middleware type

Select middleware type... *

Scriptlogic

Next

2. "Define basic configuration"

- To perform the basic configuration, specify a name for the scriptlogic and the code.
- Insert a function in the editor which converts the received data into a valid format and publishes it internally.
- Click on "Next".

1 Define basic configuration

Type scriptlogic name... *

machine9voltage

Supported input: lowercase letters, numbers, - (must start with letter) 15 / 50

Type middleware logic:

```

1 function onWildData(state, message) {
2   var topicElements = message.Topic.split(".");
3   var clientID = topicElements[2] // 3rd element, index 2, is "machine9"
4   var nowDate = new Date();
5   var voltage = JSON.parse(message.Data)
6
7   var setInfo = {
8     "Source": {
9       "ProjectID": "genericcorp",
10      "DeviceType": "sensor",
11      "ApplicationType": "mqttclient",
12      "ClientID": clientID
13    },
14    "Set": {
15      "Time": nowDate.toISOString(),
16      "Name": "VoltageData",
17      "Data": [
18        {
19          "Name": "Voltage",
20          "Type": "float",
21          "Value": voltage
22        }
23      ]
24    }
25  };
26  publishReadingSet(setInfo);
27 }

```

Next

3. "Define wild data configuration"

- To configure the topic that is to be used by the scriptlogic to receive data from the MQTT client, populate the field.
- From the remote MQTT client, send all data to this topic. The topic structure is geared toward AMQP.
- Click on "+" to save the topic.

4 Define wild data configuration

Optional

Type a new wild data topic...

site1.room4.machine9.voltage



Type a new wild data topic...

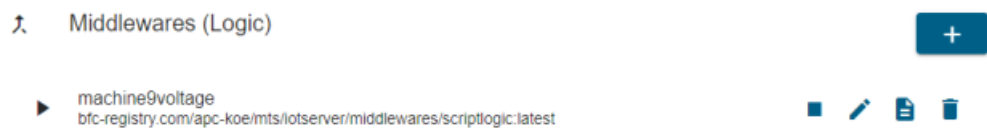


Next

4. "Define write configuration"

- Select the client ID for which write jobs should be received.
- Click on "+" to save the client ID.
- Click on "Save" to save the scriptlogic.

5. The scriptlogic has been successfully created and is displayed in the "Middlewares (Logic)" area of the overview.



6.4.3.15 MQTT client - calling connection data

In order to send data to the MQTT client you require a login on the MQTT broker. You can obtain the required information from "Show credentials".

Requirement

- Configuring a scriptlogic
More information can be found in Chapter: MQTT client - configuring scriptlogic (Page 189).
- The "Commissioning" area is open.

Procedure

To call the connection data, proceed as follows:

1. "Key from client..."

- In the "Client (Import)" area, click on the "Show credentials" icon on the row of the MQTT client.



2. "Find in login"

- Find the user name "key" and the password "secret" for the MQTT connection.
- Use the "Copy" button to copy a JSON with the following format into the clipboard:


```
{
  "key": "<key>",
  "secret": "<secret>"
}
```

Key from client "machine9-voltage"

Key:wild:987c483d-cb55-423b-8d5b-1129765e58ea

Secret:8868d11b-800d-431e-94a9-eea5efebaaa2



6.4.3.16 Creating an SFTP client

The SFTP client allows files to be accessed on devices that support the SFTP protocol (SFTP server).

Perform the following steps to create an SFTP client:

- Step 1: "Select client type"
- Step 2: "Define basic configuration"
- Step 3: "Define file permissions"
- Step 4: "Define advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

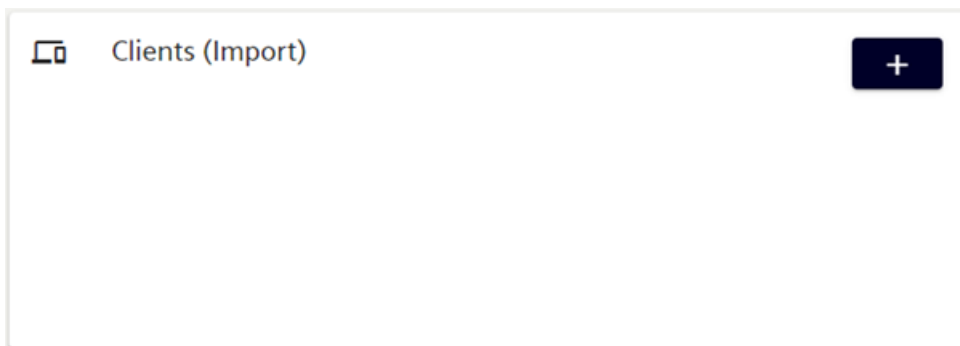
Parameter	Description
① Select client type	
Select application type...*	Selection of the client
② Define basic configuration	
Type client ID...*	ID of the SFTP client
Type client description...	Description of the machine
Type SFTP URL... *	URL or IP address of the SFTP server (with port)
Type SFTP server username	Specify the user name of the SFTP server
Type SFTP server password	Specify the password of the SFTP server
③ Define file permissions	
Set the access rights for file operations.	
④ Define advanced configuration / Optional	
Remark:	
These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path...*	Path for SFTP docker image
Type username to access the image path...	User name to access the ImagePath
Type password to access the image path...	Password to access the image path
Type filesystem address...*	Designates the internal address for program transfer. Generally does not have to be changed.
Type comma separated list of allowed key exchange algorithms...	List of permissible key exchange algorithms. If this field remains empty, then as default ALL algorithms are accepted.
Type comma separated list of allowed cipher algorithms...	List of permissible encryption algorithms. If this field remains empty, then as default ALL algorithms are accepted.
Type comma separated list of allowed MAC algorithms...	List of permissible message authentication code algorithms. If this field remains empty, then as default ALL algorithms are accepted.
Type sftp server public key signature (RFC 4253)...	Public key of the remote device that is used to authenticate the remote end point.
Type new environment variable...	New environment variable for the image

*: Obligatory data

For information on configuring access rights for file operations, see Chapter: Configuring access rights (Page 467).

Procedure

1. In the "Client (Import)" area, click on "+" to add a new client.



2. Step 1: "Select client type"
 - Select entry "SFTP client" from the drop-down list.
 - Click on "Next".

Add new client

1 Select client type

Select application type... *

SFTP client

Unlimited activations available

Next

3. Step 2: "Define basic configuration"

- Populate the fields to perform the basic configuration.
- Click on "Next".

2 Define basic configuration

Type client ID... *
SFTP_Client
Supported input: letters, numbers, -, _ (must start with letter or number) 11 / 50

Type client description...
SFTP_Client_Description
23 / 100

Type SFTP URL... *
sftp://<your_SFTP_Server_IP>:2222
e. g.: sftp://<host>:<port>

Type SFTP server username...
username

Type SFTP server password...

Next

4. Step 3: "Define file permissions"

- Configure the access rights for file operations.
- Click on "Next".

5. Step 4: "Define advanced configuration" (Optional)

- As a rule, no entries/changes are required.

Note

Advanced configuration

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click on "Save" to save the SFTP client.

4 Define advanced configuration
Optional

Type image path... *

bfc-registry.com/apc-koe/mts/iotserver/clients/sftp:v1.1.0

Type username to access the image path...

Type password to access the image path...

Type file system address... *

:7979

Type comma separated list of allowed key exchanges algorithms...

Type comma separated list of allowed cipher algorithms...

Type comma separated list of allowed MAC algorithms...

Type sftp server public key signature (RFC 4253)...

6. The SFTP client was successfully created and is shown in the overview in the "Clients (Import)" area.

6.4.3.17 Defining a dataset configuration

When creating a client, configure datasets in step "Define dataset configuration". You define variables and the type of access there.

Parameterization is realized in 3 steps:

1. New Dataset
2. New Datapoint
3. Trigger Condition (optional)

Parameters

Parameters	Description
① New Dataset	
Select variable list...	Selecting the variable list
Type dataset name...*	Dataset name
Select reading mode...*	Selecting the reading mode: - Interval The set time is called "Interval time" The configured variable is read cyclically according to the interval defined in the BFC gateway, regardless of whether the value of the variable has changed. - On Change The set time is called "Debounce time". The configured variable is read if the value changes. - On Trigger The configured variables are read when initiating the trigger, which is defined under Step 3 "Configure trigger".
② New Datapoint	
Type datapoint name...*	Name of the datapoint
Type datapoint address...*	Address of the datapoint Remark: If an invalid address is used when configuring the BFC client, error messages are output in the client's log file during runtime. To diagnose these incorrect entries, check the log file of the BFC client after every configuration change using the "Show logfile" function.
Select data type...*	Selecting the data type: - Integer - Float - String (not available for the Modbus client) - Bool
③ Trigger Condition (optional)	
Name	Variable name, as defined in the datapoint
Select trigger variable...*	Dataset name

Parameters	Description		
Select trigger condition... *	Trigger condition, the following conditions are possible:		
	Name	Trigger initiation	Type of datapoint
	Any change	If the value changes	all
	=	If the value corresponds to the comparison value	int, bool, string
	!=	If the value does not correspond to the comparison value	int, bool, string
	>	If the value is higher than the comparison value	int, float
	>=	If the value is higher than or equal to the comparison value	int
	<	If the value is less than the comparison value	int, float
	<=	If the value is less than or equal to the comparison value	int
	contains	If the value contains the comparison value	string
Value	Comparison value		
Reset Value	The value to which the variable is reset after the trigger is initiated. The variable is not reset if a value is not specified. To use this function, the variable must be able to be written to, and released under Point "Define allowlist configuration".		
Logic			
Logic function	Logic function, the following elements can be used to logically link the trigger: • AND • OR • NOT • (•)		

*: Obligatory data

Procedure

1. Select "Define Dataset configuration".

3 Define dataset configuration

Select browsing file...

None

1 New Dataset:

Type dataset name... *

Select reading mode... *

On Trigger

Set interval time in milliseconds... *

500

Configure datapoints

2 New Datapoint:

3 Trigger condition:

Add current dataset

No datasets configured yet Import

Automated variable suggestions are generated depending on the client type. For the following clients, these are automatically taken from the control:

- OPC UA client
- MTConnect client

For the other clients, a variable list can be transferred, so that also here suggestions for variables can be generated. These variable lists are available for the following clients:

- BFC client
- FANUC client
- Heidenhain client

Creating additional address lists is described in Chapter "Custom Variable List" area (Page 319). If there is an address list for the client, then this can be selected under "Select variable list".

Select variable list

None

S7Client

S7ClientDefault

Manage existing datasets using the import and export functions.
More information is provided in Chapter Using configured datasets (Page 206).

2. Step 1: "New Dataset"
 - Specify the dataset name

6.4 "Commissioning" area

- Select the reading mode:
On Change: The dataset is transferred as soon as a value of the configured datapoints changes.
Interval: The dataset is cyclically transferred, independent of changes to the values.
On Trigger: The dataset is transferred depending on the values of the datapoints. The trigger is defined in Step 3 "Trigger Condition".
- Define a debounce time in the range 200-86400000ms.
- Click on "Configure datapoints" to configure the datapoints belonging to the dataset.

3. Step 2: "New Datapoint"

- Add one or several datapoints.
- Enter a name, an address and a datatype.
- If variable lists are available for the client, then these are displayed as list. You can select the variable and add using the "+" button.

Note

Manually adapting addresses

Note that several addresses can still contain internal parameters, e.g. for the axis or the channel. You must manually adapt this.

- Click on "Add current datapoint" to add the datapoint.
Enter all of the required datapoints in this way.
- If you're reading mode is set to "OnChange" or "OnInterval", you can now add your dataset using the "Add current dataset" button.

- If you're reading mode is set to "OnTrigger", you can start the trigger configuration using the "Configure trigger" button.

New Datapoint:

Search...

Name: addinfo

Description: Additional information

Name: max_axis

Description: Maximum controlled axes. This is the maximum controlled axes in CNC.

Address: /focas/cncsysinfo/max_axis

Type: int

Unit: -

Name: cnc_type

Description: Kind of CNC (ASCI0 - '15' Series 15/15i/15i; '16' Series 16/16i; '18' Series 18/18i; '21' Series 21/21i; '30' Series 30i; '31' Series 31i; '32' Series 32i; '35' Series 35i; '0' Series 0; 'T0' Power Mate I-0; 'T1' Power Mate I-1)

Type datapoint name... *

Type datapoint address... *

Select data type... *

Type datapoint name... *

cnc_type

Type datapoint address... *

/focas/cncsysinfo/cnc_type

Select data type... *

String

Add current datapoint

1 datapoint(s) configured - do not forget to add dataset!

Datapoint name	Address	Type	Value
max_axis	/focas/cncsysinfo/max_axis	int	

Configure trigger

Type datapoint name... *

max_axis

Type datapoint address... *

/focas/cncsysinfo/max_axis

Select data type... *

Integer

Add current datapoint

1 datapoint(s) configured - do not forget to add dataset!

Datapoint name	Address	Type	Value
cnc_type	/focas/cncsysinfo/cnc_type	string	

Configure trigger

4. Step 3: "Configure trigger"

- Compile one or more triggers.
- Link the trigger to a logical operation using "AND", "OR", "NOT", "(", ")".
- Add your dataset using "Add current dataset".

3

Trigger condition: T1 && ! T2

Name *	Select trigger variable...	Select trigger condition...	Value *	Reset value
T1	max_axis	>	10	0
T2	cnc_type	contains	test	Reset value

+ Add trigger

Operators:

AND

OR

NOT

{ }

T1

T2

Triggers:

Logic function:

T1

AND

NOT

T2

Add current dataset

Timing

Only the trigger is read when a change takes place if a data set is read "On Trigger". Additional data of the set is only read once the trigger condition has been fulfilled. This can result in delays until the change in the trigger is known, and then until reading the variable has been completed.

As a result of this response, it is possible that the sent data set no longer contains the trigger condition.

The timing for a trigger is shown with a Boolean variable in the following diagram. The data set should be sent if variable "Bool1" = "true".

How the variables respond over time is shown above in blue, the internal sequences of the trigger evaluation are shown below.

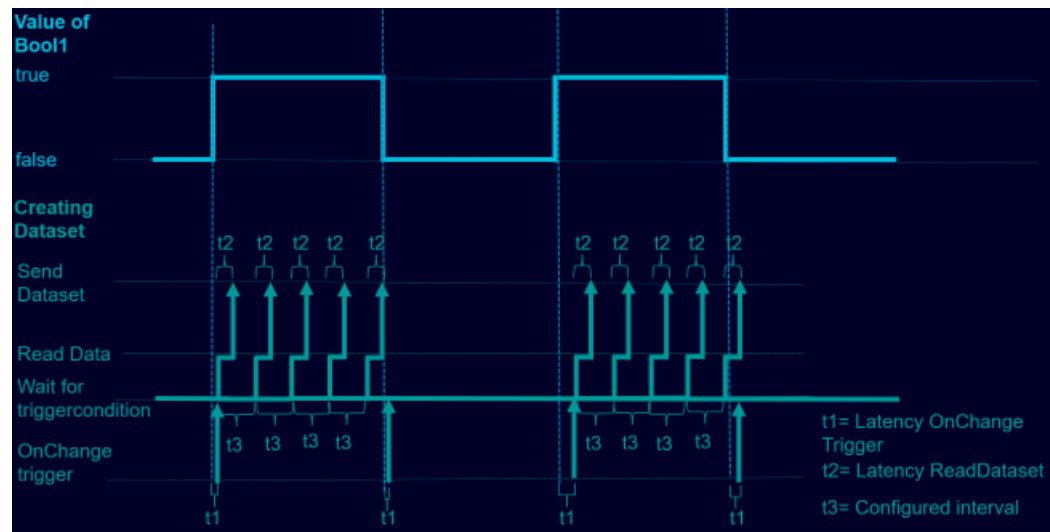
After changing the variable, it can be seen that a time difference (t1) occurs until the value is signaled in the BFC Gateway. All data of the data set are then read out, as the trigger condition is satisfied. This also takes a certain time (t2). This data set is then sent.

As long as the trigger is active, the data set is regularly sent in the configured time interval.

Note

Individual script logic

If the triggers you require are too complex for you to completely create them using this mechanism, then you can also create an individual script logic. The description to create the script logic, including templates, is provided in Chapter Creating script logic (Page 211).



6.4.4 Displaying/editing the client

You can access configured clients as follows:

- Start and stop client
- Edit client

6.4 "Commissioning" area

- Display live data in real-time
- Display log files
- Delete client

Procedure

- Click on one of the icons behind the client to edit the client or to display data and information.



6.4.5 Using configured datasets

In the user interface, 2 functions are available to export and import configured data sets within the clients:

- "Export" function
Configured data sets are exported from client A to an external file.
- "Import" function
Configured data sets of client A are imported from the external file into client B.

Use the functions if you want to create several clients with the same data sets. This eliminates the need to repeatedly create identical data sets for each individual client.

Both functions are supported by all client types.

The "Import" and "Export" functions are available for configurations:

- Creating a new client
- Editing an already created client

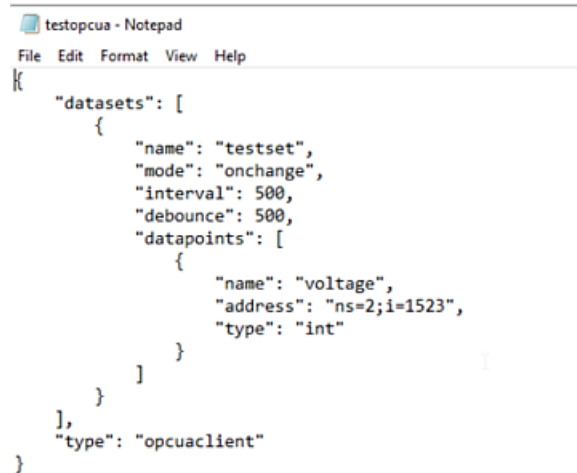
Structure of the export file

You can open and edit the exported data set configuration file in a text editor. This makes it possible to add further data points to the data set configuration.

Parameter	Description
"datasets": [	Marking data set
"name": "testset",	Name of the data set
"mode": "onchange",	Data set mode
"interval": 500,	Interval time
"debounce": 500,	Debounce time
"datapoints": [	Configured data points

Parameter	Description
"name": "voltage"	Name of the datapoint
"address": "ns=2;i=1523"	Address of the datapoint
"type": "int"	Datapoint type
"type" : "opcuaclient"	Type of client

The file is in JSON format.



```
testopcua - Notepad
File Edit Format View Help
{
  "datasets": [
    {
      "name": "testset",
      "mode": "onchange",
      "interval": 500,
      "debounce": 500,
      "datapoints": [
        {
          "name": "voltage",
          "address": "ns=2;i=1523",
          "type": "int"
        }
      ]
    }
  ],
  "type": "opcuaclient"
}
```

6.4.5.1 Exporting datasets from a client

Export configured data sets from a client to an external file.

Requirement

- Configured data sets are created in a client.
Further information can be found in Chapter: Creating clients (Import) (Page 96).

Procedure

1. Navigate via the areas "Commissioning" > "Clients (Import)" to editing step 3: "Define Dataset configuration".

Note

Navigation

- Create new client:
You are automatically navigated to this editing step.
- Edit created client:
Select the client in the Clients (Import) area and click the "Edit" icon. Navigate to editing step 3.

2. Click the "Export" button to export a configured data set.

Note

Export

At least one data set must be configured.

Add new client

2 New Datapoint:

1 dataset(s) configured Export Import

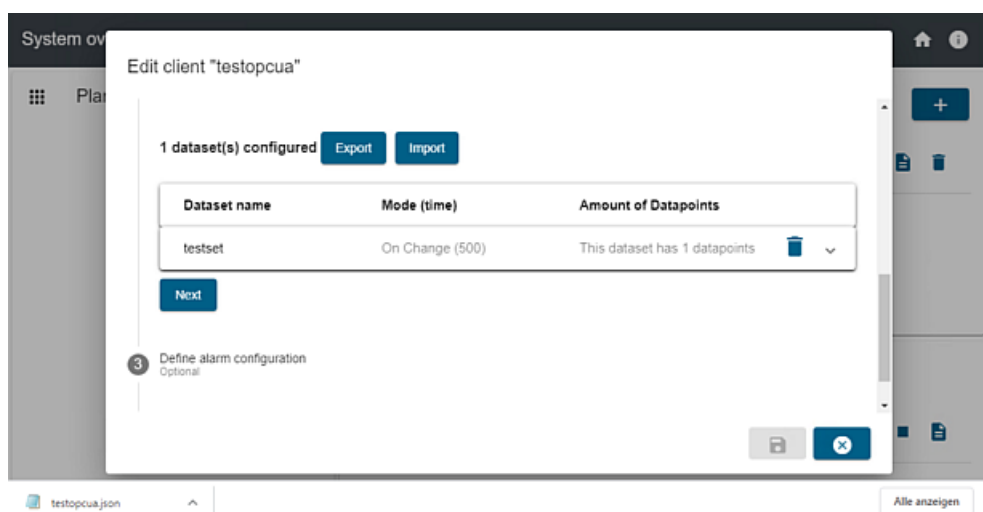
Dataset name	Mode (time)	Amount of Datapoints
testset	On Change (500)	This dataset has 1 datapoints 🗑️ ⌵

Next

4 Define alarm configuration
Optional

📁 ✕

3. The configured data set is automatically downloaded from the web browser.
The name of the downloaded external file follows the pattern: <Assigned Client ID>.json.



6.4.5.2 Importing data sets into a client

Import configured data sets into a client.

Requirement

- A configured data set of a client has been exported and is saved as an external file. Further information can be found in Chapter: Exporting datasets from a client (Page 207).

Procedure

1. Navigate via the areas "Commissioning" > "Clients (Import)" to editing step 3: "Define Dataset configuration".

Note

Navigation

- Create new client:
You are automatically navigated to this editing step.
 - Edit created client:
Select the client in the Clients (Import) area and click the "Edit" icon. Navigate to editing step 3.
-

2. Click the "Import" button to import a saved configured data set.

Add new client

2 New Datapoint

No datasets configured yet **Import**

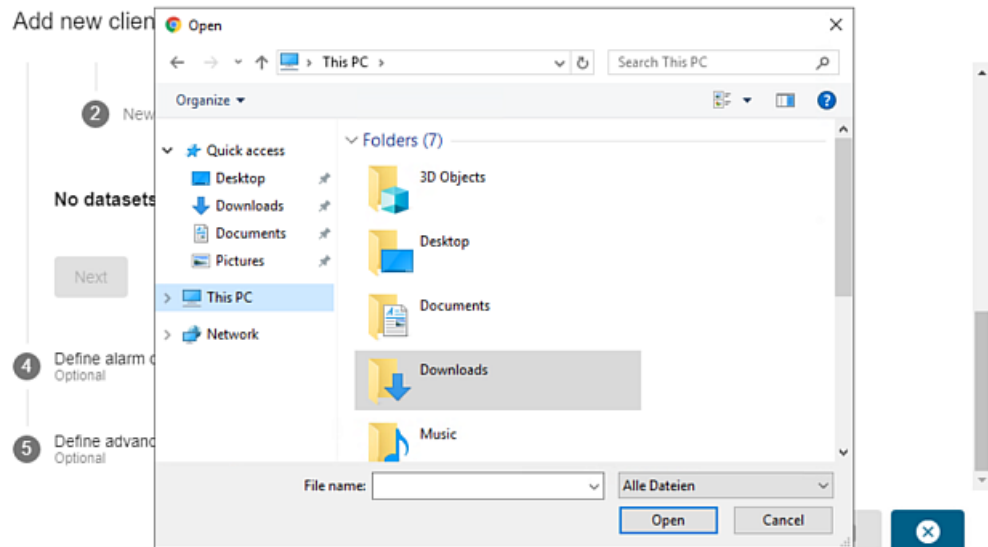
Next

4 Define alarm configuration
Optional

5 Define advanced configuration
Optional

Save Close

3. The file manager opens.
 - Select the configured data set saved as an external file.
 - Click the "Open" button to accept the configured data set.



4. The configured data set is transferred to the client.

6.4.6 Creating Middlewares (Processing)

A middleware processes the data that is received from clients and transfers this on. You can create the following middlewares in the "Commissioning" area in the "Middlewares (Logic)" area:

- Script logic
- Internal middleware (placeholder for project-specific extensions)
- External middleware (placeholder for project-specific extensions)

6.4.6.1 Creating script logic

Data can be processed and linked with one another using script logic.

Note

Thoroughly test the source code generated when developing script logics and verify it before productive use.

Procedure

1. Step 1: "Select middleware type"
 - Select entry "Scriptlogic" from the drop-down list.
 - Click on "Next".

1 Select middleware type

Select middleware type... *

Scriptlogic

Next

2. Step 2: "Define basic configuration"
 - To perform the basic configuration, specify a name for the scriptlogic and the code.
 - When required, select a template.
 - Insert the script or adapt the script to the functions that you require.
 - Click on "Next".

2 Define basic configuration

Select template... *

read

Type scriptlogic name... *

name_of_script

Type middleware logic:

```
5 startIntervalWorker(function (state) {
6   console.log("simpleread.js - startIntervalWorker - state:", 350M.strin
7
8   var res = readData({
9     "target": {
10      "ProjectID": "connectivity",
11      "DeviceType": "device",
12      "ApplicationType": "modbusclient",
13      "ClientID": "modbustest"
14    },
15    "ttl": 1000,
16    "data": [{
17      "address": "h:1,1",
18      "type": "int"
19    }]
20  });
21
22  if (res === false) {
23    console.log("simpleread.js - Could not writeData");
24    return;
25  }
26
27  counter++;
28  if (counter > 100) {
29    counter = 0;
30  }
31
32  console.log("simpleread.js - writeData:", res);
33
34 }, 10000);
```

Next

Download

3. Step 3: "Define dataset configuration"

- Select a client from which data should be processed in the script logic.
- Select the datasets that should be processed. When selecting "All datasets", all datasets are transferred to the script logic.
- Click on "+" to save.
- Repeat the process as required for additional clients.
- Click on "Next".

4. Step 4: "Define wild data configuration"

- Enter all topics outside the datasets, which should be processed in the script logic. For example, these can be topics that are provided from an MQTT client. Other script logics can provide individual data here.
- Click on "+" to save.
- Repeat the process as required for additional topics.
- Click on "Next".

5. Step 5: "Define alarm configuration"

- Select a client from which alarms should be processed in the script logic.
- Click on "+" to save.
- Repeat the process as required for additional clients.
- Click on "Next".

6. Step 6: "Define state configuration"

- Select a client from which state messages should be processed in the script logic.
- Click on "+" to save.
- Repeat the process as required for additional clients.
- Click on "Next".



7. Step 7: "Define write configuration"

- Select a client from which write tasks should be processed in the script logic.
- Click on "+" to save.
- Repeat the process as required for additional clients.
- Click on "Next".



8. Step 8: "Advanced configuration"

- You can make project-specific adaptations in this area.
- Save the middleware.

6.4.7 Creating the plant hierarchy

In the "Plant hierarchy" area you define your plant structure like this: Plant - Line - Cell - Machine.

4 level types are available to you:

- Level 1: Plant
- Level 2: Line
- Level 3: Cell
- Level 4: Machine

In the display bar of the BFC gateway, you can see the level in which you are in.

Click on the name of the hierarchy in the display bar to navigate within the levels.



Requirement

The "Commissioning" area is open.

Parameter

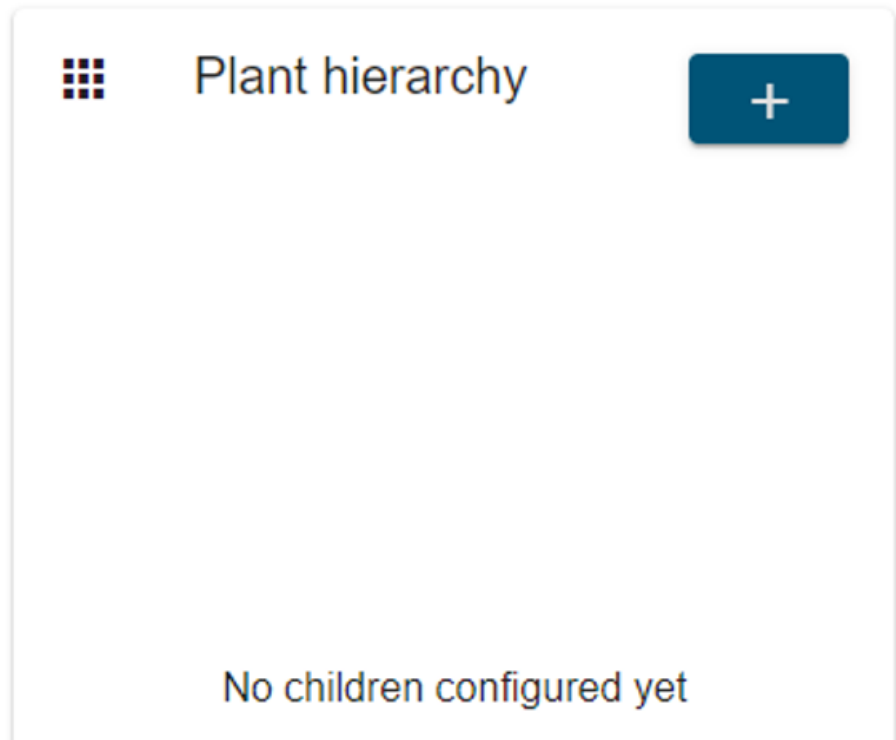
Parameter	Description
Type equipment name... *	
Type: Select type... *	The following can be selected: • Machine • Cell • Line • Plant
Description: Type equipment description...	Description of the selection
Select a new parent equipment...	Assignment of the selection to a target location

*: Obligatory data

Pattern: The character sequence to be used for the various entries is provided in the input windows.

Procedure

1. Click the "+" button in the "Plant hierarchy" area to add a plant object.



2. The window to enter the plant structure opens.
 - Select the required entry from the drop-down list:
 - Level 1: Plant
 - Level 2: Line
 - Level 3: Cell
 - Level 4: Machine

Note

Flexible structure

You can create several plants horizontally at the uppermost level.

You can also assign multiple lines/cells to each plant and multiple machines to each line/cell vertically.

- Click on "Save" to save the entry.

Add new Equipment

Type equipment name... *

Supported input: letters, numbers, spaces, /, #, _ - 0 / 50

Select type... *

Type equipment description...

0 / 100

- Horizontal plant structure:
To horizontally create a new element at a level, click the "+" button again.
The input window opens.
Enter your plant data as described under point "1."
- Vertical plant structure:
You have 2 options to create one or several sublevels for a specific level:
 - Option 1:
Click on the designation created at the relevant level to directly create a new sublevel.
"Plant 1" in the example.

 **Plant1**
This is our Plant1

The input window opens.

Enter your plant data as described under point "1."

- Option 2:
Click the "+" button to create additional levels, initially horizontally.
The input window opens.
Enter your plant data as described under point "1."
Click on "Move..." to shift the plant that was entered to the correct level.
From the drop-down list select the target location and save your entry by clicking on "Save".

Move equipment "Plant1"

Select new parent equipment...

6.4.8 Creating gateways (Export)

In the "Gateway (Export)" area you connect the client with gateways to export the machine data to higher-level systems, e.g. MindSphere, or into local company networks.

6.4.8.1 Gateways (Export)

You can create the following gateways in the area "Commissioning" > "Gateways (Export)":

- MindSphere
- MySQL
- AMP 4.1
- MQTT
- InfluxDB
- Elasticsearch
- HTTP
- Custom gateway
Placeholder for extensions

Note

Parameterization step 5 "Define write configuration"

With this optional function you can configure external write jobs for export gateways.

The following gateway supports this optional function:

- OPC UA
-

6.4.8.2 Creating a MindSphere gateway

Perform the following steps to create a MindSphere gateway:

- Step 1: "Select gateway type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define alarm configuration"
- Step 5: "Advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select gateway type	
Select target Store...*	Selection of the gateway
② Define basic configuration	
Type gateway name...*	Gateway name
Type gateway description...	Description of the gateway
Paste MindSphere connection info...*	Connection information Enter the MindSphere connection string in the JSON format
More information on the format of the MindSphere connection string is available on the Internet: "Agent Management Service - API Specification (https://developer.mindsphere.io/apis/connectivity-agentmanagement/api-agentmanagement-api-swagger-v3-4-0.html)". Segment: "Boarding Options" > "Get boarding configuration".	
Type proxy address if needed...	Enter the proxy address if a proxy is required for Internet access Format: <Protocol>://<IP address>:<Port>
③ Define dataset configuration / Optional	
Pick a client ID	Client ID of the client, from which data is to be sent to MindSphere. Remark: All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
Pick a dataset	Select the client data set that should be sent to MindSphere. Remark: All data sets of the selected client ID are displayed
- OR -	
All dataset	All client data is sent to MindSphere
④ Define alarm configuration / Optional	
Pick a client ID	All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
⑤ Advanced configuration / Optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path...*	Path to MindSphere docker image
Type username to access the image path...	User name to access the image path
Type password to access the image path...	Password to access the image path
Set message TTL (time to live) in milliseconds...*	AMQP lifetime of the data
Set queue expiration in milliseconds...*	AMQP queue expiration time
Set max queue size...*	AMQP maximum queue size
Set max queue size bytes...*	AMQP maximum queue size in bytes
Select queue mode...*	AMQP queue mode

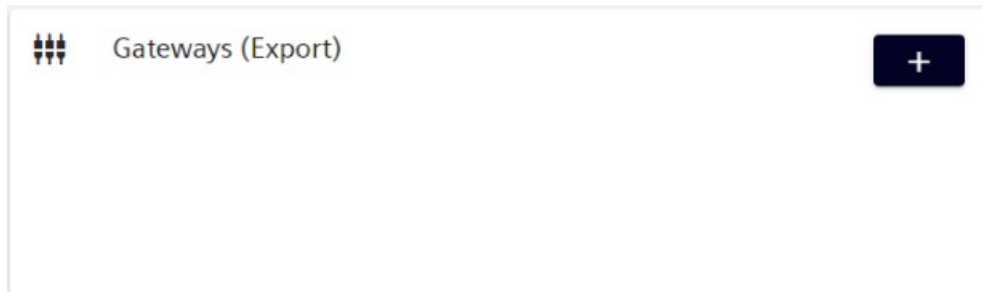
Parameter	Description
Set prefetch count...*	AMQP prefetch counter
Type alarm topic*	MQTT topic alarm event
Type new reading topic*	MQTT topic new dataset
Type new environment variable*	New environment variable for the image

*: Obligatory data

The character sequence to be used for the various entries is provided in the input windows.

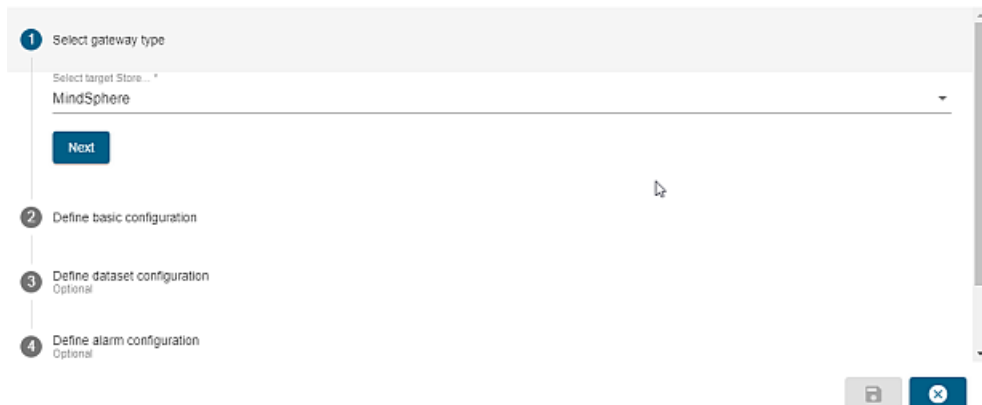
Procedure

1. In the "Gateway (Export)" area, click on "+" to add a new gateway.



2. Step 1: "Select gateway type"
 - From the drop-down list, select entry "MindSphere".
 - Click on "Next".

Add new gateway



3. Step 2: "Define basic configuration"

- Populate the fields to perform the basic configuration.
Copy "MindSphere connection info".

Note**"MindSphere connection info" not available**

Take the "MindSphere connection info" from the onboarding data of the MindConnectLib in the MindSphere user interface.

More information can be found in Chapters:

- Creating the MindConnectLib asset (Page 224)
- Generating connection information of the assets (Page 488)

- Click on "Next".

Add new gateway "gatewaytest"

The screenshot shows the 'Define basic configuration' step of the gateway setup process. The interface includes a progress bar with two steps: '1 Select gateway type' and '2 Define basic configuration'. The 'Define basic configuration' step is active. The form contains the following fields:

- Type gateway name... ***: The value 'gatewaytest' is entered. A note indicates 'Supported input: lowercase letters, numbers, - (must start with letter)' and '11 / 50'.
- Type gateway description...**: The value 'mindsphere connection test' is entered. A note indicates '26 / 100'.
- Paste MindSphere connection info... ***: A placeholder icon is shown.
- Type proxy address if needed...**: This field is empty.

At the bottom right, there are two buttons: a save icon (floppy disk) and a cancel icon (X).

4. Step 3: "Define dataset configuration" / Optional
With this step, you define which data is to be sent to MindSphere.
 - Select one data set, several or all data sets.

Note

Several entries

- Click on "+" to add an additional entry.

Note

Automatic mapping is not possible if the MindSphere variable includes an Umlaut (a special German character).

- Click on "Next".

Add new gateway "gatewaytest"

The screenshot displays a vertical sequence of five steps for gateway commissioning:

- 1 Select gateway type
- 2 Define basic configuration
- 3 Define dataset configuration
Optional
- 4 Define alarm configuration
Optional
- 5 Advanced configuration
Optional

Step 3 is the active step. It contains two input fields:

- Pick a client ID:** Contains the text "testopcua".
- Pick a dataset:** Contains the text "**All datasets**". To the right of this field is a blue "+" button for adding more datasets.

Below the input fields is a blue "Next" button.

At the bottom right of the interface are two blue buttons: a save icon (floppy disk) and a close icon (X).

5. Step 4: "Define alarm configuration" / Optional

With this step, you define which alarms are to be sent to MindSphere.

- Select the alarms.

Note**Several entries**

- Click on "+" to add an additional entry.

- Click on "Next"

Add new gateway "gatewaytest"

The screenshot shows a vertical progress bar on the left with five steps:

- 1 Select gateway type
- 2 Define basic configuration
- 3 Define dataset configuration
Optional
- 4 Define alarm configuration
Optional
- 5 Advanced configuration
Optional

Step 4 is currently active. Below the progress bar, the text "Pick a client ID" is followed by a text input field containing "testopcu". To the right of the input field is a trash icon. Below the input field is a blue "Next" button. A mouse cursor is hovering over the "Next" button.

At the bottom right of the form, there are two blue buttons: one with a square icon and one with an 'X' icon.

6.4 "Commissioning" area

6. Step 5: "Advanced configuration" / Optional
With this step, you can optionally configure high-frequency data.
Further configurations are usually not necessary.

Note

Advanced configuration

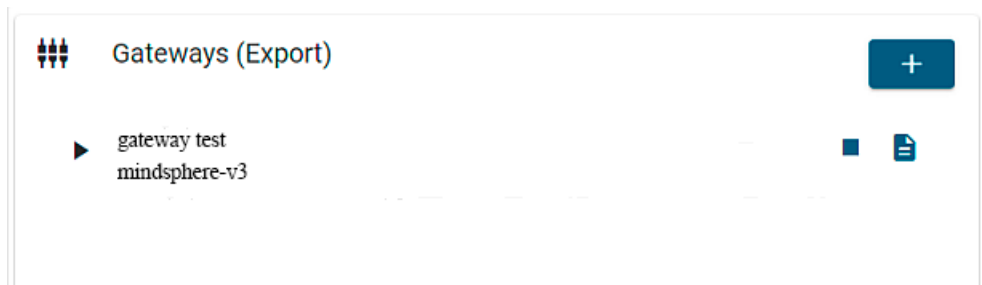
Entries and changes are only made in the case of service, and only performed by the hotline.

- Click on "Save" to save the MindSphere gateway.

Add new gateway "gatewaytest"

The screenshot shows a vertical wizard with five steps. Step 5, 'Advanced configuration' (Optional), is selected and highlighted with a blue circle. Below the steps, there are two input fields. The first is labeled 'Type image path... *' and contains the text 'apc-docker-registry.dlinkddns.com/apc-koe/mts/iotserver/gateways/mindsphere:v1.0.0'. The second is labeled 'Type username to access the image path...'. At the bottom right of the form, there are two buttons: a blue 'Save' button with a floppy disk icon and a grey 'Cancel' button with an 'X' icon.

7. The MindSphere gateway was successfully created, and is shown in the overview in the "Gateways (Export)" area.



6.4.8.3 Creating the MindConnectLib asset

Proceed as described below to create a new "MindConnectLib type" asset in MindSphere.

Requirement

You are in the configuration user interface of your MindSphere access.

Parameters

Parameter	Description
Type information	
Parent type:	For MindConnectLib types, always the designation: "core.mclib" Remark: The input cannot be changed.
Type ID:	<MindSphere Tenant Name>.Name Remark: The name cannot be changed.
Name:*	Name of the type
Description:	Description of the type

* optional

Procedure

1. Create a new MindConnectLib type.

Note

Exception: "BFCGateway type" is available

If a "BFCGateway type" was selected, then step 1 is no longer applicable.

- As requested in step 2, create an asset of this type, and then follow the additional instructions.

If no "BFCGateway type" is available, then you must create a new MindConnectLib type:
Populate the fields.

Create type

— Type information

Parent type:

core.mclib

Parent type due to hierarchical order

Type ID:

rcdevpl.BFCGateway

Type ID cannot be changed after creation

Name: *

BFCGateway

Description:

This Type is for BFC Gateways

226 characters left

2. Create a new asset of the new type.
3. Open "MindConnect Lib Plugin".



4. Select option "SHARED_SECRET".
5. Click on "Generate onboarding key".
Copy the complete "key".

6.4.8.4 MySQL export gateway

Using the BFC Gateway, you can transfer selected data, such as "Datasets" and "Alarms", to a MySQL database.

For the data transfer you have to create and configure a MySQL export gateway.

System requirements for MySQL database

MySQL Server 8.0 must be installed to be able to use the DB export gateway.

Configuring MySQL database

The following MySQL database settings are required:

- The MySQL database must be accessible via TCP/IP.
- The user who logs into the database must have the access privileges to create the tables in the database.

MySQL configuration example:

The following procedure creates a database with name "bfcdatabase". A user "bfc4mysql" with password "dbpassword" is then created for this. This user is assigned all of the rights necessary to access a database from a BFC Gateway with IP address "192.168.3.79".

1. Log in as root at the system database 'mysql'.
2. CREATE DATABASE bfcdatabase;
3. CREATE USER 'bfc4mysql'@'192.168.3.79' IDENTIFIED BY 'dbpassword';
4. GRANT ALL PRIVILEGES ON bfcdatabase.* TO 'bfc4mysql'@'192.168.3.79';
5. update db set Host='192.168.3.79' where Db='bfcdatabase';
6. update user set Host='192.168.3.79' where user='bfc4mysql';
7. flush privileges;

Database schema

The tables in which the data is stored have the following structure:



Entry data source

In this schema, an entry for the data source is created in the table "bfc_sources", if not already existing.

Note

Entries

- The table names and column names are normalized to match the limitations of the MySQL database, i.e. all special characters are replaced by _.
- Prefixes are added to prevent collisions with the BFC names.
Table names have the prefix DATASET_.
Column names have the prefix DATAPOINT_.

Alarms

Each incoming or outgoing alarm is created as an entry in the table "bfc_alarms".

This entry references the source from the table "bfc_sources".

DataSets

Each entry in a DataSet table references the data source from the table "bfc_sources".

High-frequency data

The table name is formed analogous to ReadingSets with "DATASET_<HF Set Name>".

The following columns are included in the scheme for high-frequency data:

Column name	Data type	Data source
BFC_SOURCE_ID	bigint	ID from the bfc_sources table for which AppName and DeviceID are stored, respectively
BFC_TIMESTAMP	datetime(6)	Start time of recording and offset
BFC_CONTENT_TYPE	text	"hfDataSet"
BFC_HFDATA_ID	text	ID of the NC trace session
BFC_CONTEXT_<ContextVar1>	<Type of Context Var1>	<Value of ContextVar1>
BFC_CONTEXT_<ContextVar2>	<Type of Context Var2>	<Value of ContextVar2>
...	...	...
BFC_CONTEXT_<ContextVarN>	<Type of Context VarN>	<Value of ContextVarN>
DATAPOINT_<SignalName1>	<SignalType1>	<Value of Var1>
DATAPOINT_<SignalName2>	<SignalType2>	<Value of Var2>
...	...	...
DATAPOINT_<SignalNameN>	<SignalTypeN>	<Value of VarN>

The columns BFC_SOURCE_ID and BFC_TIMESTAMP are used to create a unique index during creation.

This ensures that only **one** entry exists for a point in time for a source and speeds up queries to the table.

Resource utilization (quantity structure)

The following configuration is the basis for the resource utilization of the MySQL database:

With the configuration specified in the following table, 75,000 data sets per minute are thus exported from the BFC Gateway to the MySQL database. This corresponds to a memory requirement of approximately 96 MB per minute.

Note**Sufficient resources**

The operator of the database must ensure that sufficient resources are available and that data that are no longer required are deleted.

Description	Quantity/value
Clients connected to a BFC Gateway	60
Datasets	5
Data points per data set (sum of the data points)	10 (50)
Memory required per data set	1.31 KB
Frequency of the data points to be read	200 ms

Note**Sum of the data points**

The sum of all data points results from the multiplication of data sets and data points.

6.4.8.5 Creating a MySQL export gateway

Perform the following steps to create a MySQL export gateway:

- Step 1: "Select gateway type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define alarm configuration"
- Step 5: "Advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select gateway type	
Select target Store...*	Selection of the gateway

Parameter	Description
② Define basic configuration	
Type gateway name...*	Gateway name Remark: The designation for "gateway name" is freely selectable.
Type gateway description...	Description of the gateway Remark: The designation for "gateway description" is freely selectable.
Type MySQL connection...*	Connection data to the MySQL database Format: username:password@tcp({address:port})/database name
Type MySQL maximal connections...*	Maximum number of connections to the database
③ Define dataset configuration / Optional	
Pick a client ID	Client ID of the client from which data is to be sent to the MySQL export gateway. Remark: All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
Pick a dataset	Select the client data set that should be sent to the MySQL export gateway. Remark: All data sets of the selected client ID are displayed.
- OR -	
All dataset	All client data is sent to the MySQL export gateway.
④ Define alarm configuration / Optional	
Pick a client ID	All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
⑤ Advanced configuration / Optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path...*	Path to the gateway docker image
Type username to access the image path...	Username to access the image path
Type password to access the image path...	Password to access the image path
Set message TTL (time to live) in milliseconds...*	AMQP lifetime of the data
Set queue expiration in milliseconds...*	AMQP queue expiration time
Set max queue size...*	AMQP maximum queue size
Set max queue size bytes...*	AMQP maximum queue size in bytes
Select queue mode...*	AMQP queue mode
Set prefetch count...*	AMQP prefetch counter

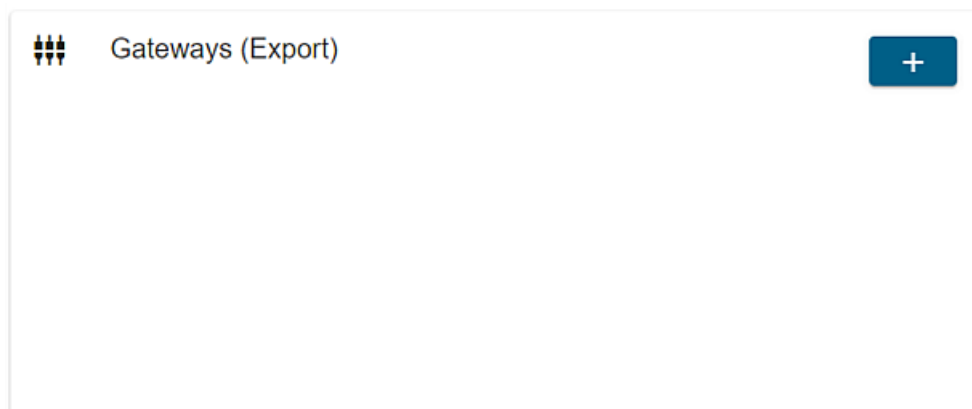
Parameter	Description
Type alarm topic*	Configuration of the topic: *.*. {deviceId}. *.events Example of a FANUC client with the name "Fanuc01": *. *. Fanuc01. *.events
Type new reading topic*	Configuration of the topic: *. *. {deviceId}. *.record. * Example of a FANUC client with the name "Fanuc01": *. *. Fanuc01. *.record. *
Type new environment variable*	New environment variable for the image

*: Obligatory data

The character sequence to be used for the various entries is provided in the input windows.

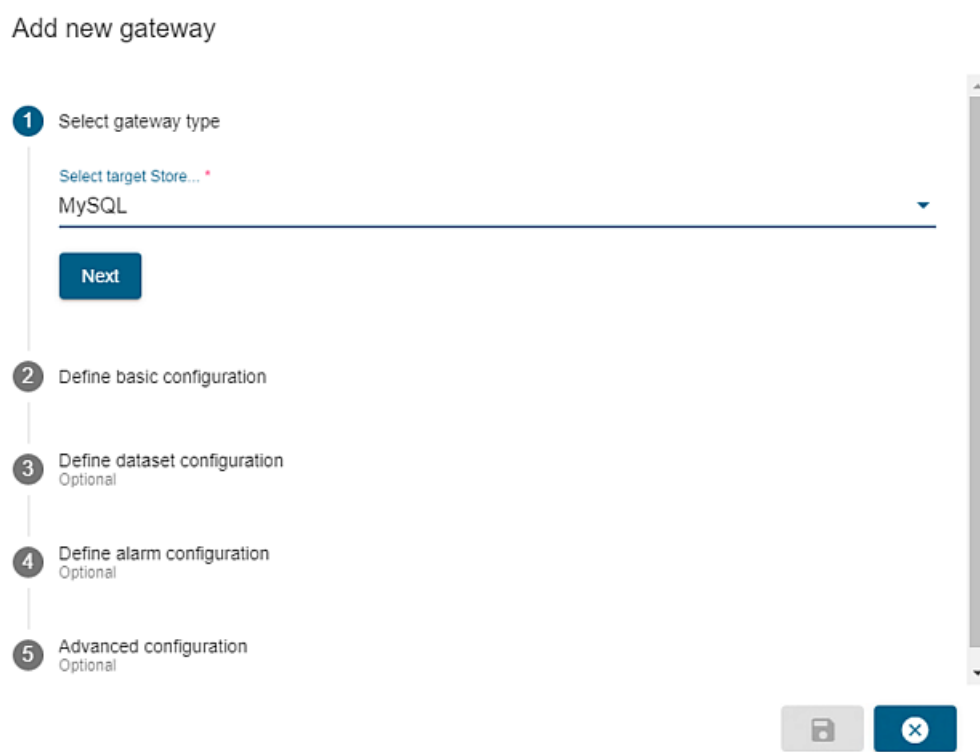
Procedure

1. In the "Gateway (Export)" area, click on "+" to add a new gateway.



2. Step 1: "Select gateway type"
 - Select the entry "MySQL" from the drop-down list.
 - Click on "Next".

Add new gateway



3. Step 2: "Define basic configuration"
 - Populate the fields to perform the basic configuration.
 - Click on "Next".

Add new gateway "dbexporttest"

The screenshot shows a web-based configuration interface for the BFC Gateway. It is divided into two main sections: Step 2, 'Define basic configuration', and Step 3, 'Define dataset configuration'. Step 2 is the active section and contains four text input fields. The first field, 'Type gateway name...', contains 'dbexporttest' and has a character count of 12 / 50. The second field, 'Type gateway description...', contains 'database export test' and has a character count of 20 / 100. The third field, 'Type MySQL connection...', contains 'root:secretpassword@tcp(192.168.5.95:3306)/bfcdatabase'. The fourth field, 'Type MySQL maximal connections...', contains '50'. A blue 'Next' button is located below the input fields. Step 3 is labeled 'Optional' and is currently inactive. At the bottom right of the interface, there are two buttons: a blue button with a save icon and a blue button with a close icon. A vertical scrollbar is visible on the right side of the configuration area.

2 Define basic configuration

Type gateway name... *
dbexporttest 12 / 50

Type gateway description... *
database export test 20 / 100

Type MySQL connection... *
root:secretpassword@tcp(192.168.5.95:3306)/bfcdatabase

Type MySQL maximal connections... *
50

Next

3 Define dataset configuration
Optional

Save Close

4. Step 3: "Define dataset configuration" / Optional
With this step, you define which data is to be sent to the MySQL export gateway.

- Select one data set, several or all data sets.

Note**Several entries**

- Click on "+" to add an additional entry.
-

- Click on "Next".

Add new gateway "dbexporttest"

1 Select gateway type

2 Define basic configuration

3 Define dataset configuration
Optional

Pick a client ID
dbexporttest

Pick a dataset

Next

4 Define alarm configuration
Optional

5 Advanced configuration
Optional

Save Close

5. Step 4: "Define alarm configuration" / Optional
With this step, you define which alarms are to be sent to the MySQL export gateway.
 - Select the alarms.
 - Click on "Next"

Add new gateway "dbexporttest"

The screenshot shows a five-step configuration wizard for a new gateway named "dbexporttest". The steps are listed on the left, with step 4, "Define alarm configuration", highlighted in blue and marked as "Optional". Below the steps, a text input field labeled "Pick a client ID" contains the text "dbexporttest". To the right of the input field is a blue plus icon. Below the input field is a blue "Next" button. At the bottom right of the wizard are two blue buttons: one with a save icon and another with a close (X) icon.

1 Select gateway type

2 Define basic configuration

3 Define dataset configuration
Optional

4 Define alarm configuration
Optional

Pick a client ID
dbexporttest

Next

5 Advanced configuration
Optional

Save Close

6. Step 5: "Advanced configuration" / Optional
With this step, you can optionally configure high-frequency data.
Further configurations are usually not necessary.

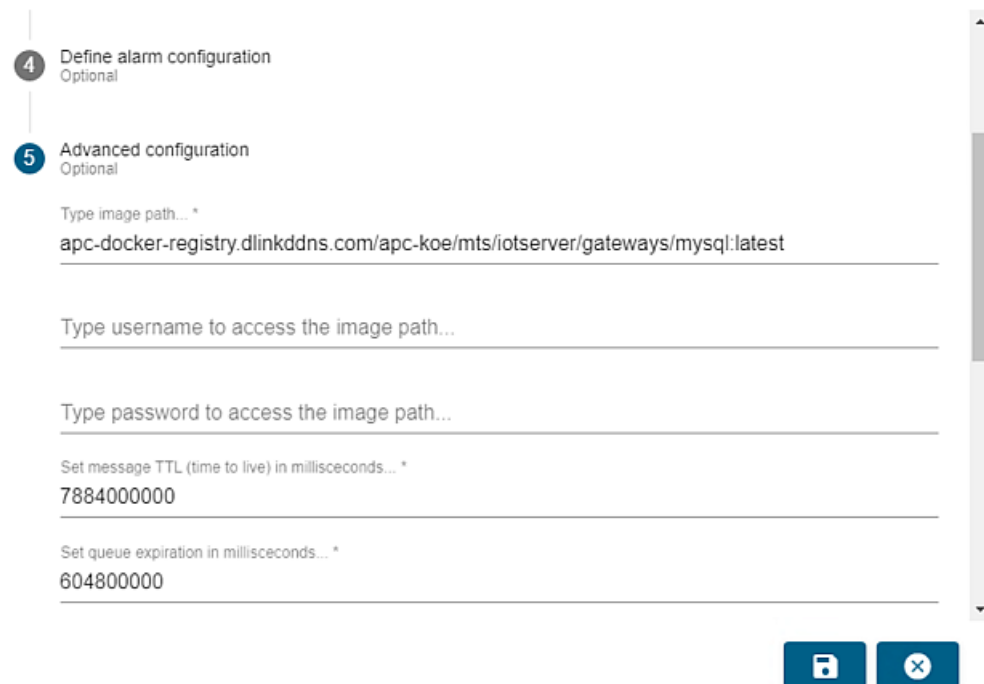
Note

Advanced configuration

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click on "Save" to save the DB export gateway.

Add new gateway "dbexporttest"

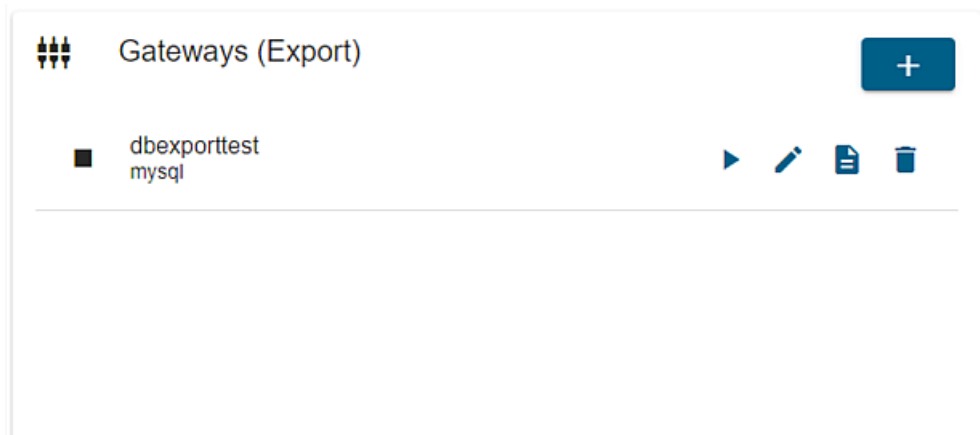


The screenshot shows a configuration interface with two steps: '4 Define alarm configuration Optional' and '5 Advanced configuration Optional'. Step 5 is active. It contains several input fields:

- 'Type image path... *' with the value 'apc-docker-registry.dlinkddns.com/apc-koe/mts/iotserver/gateways/mysql:latest'
- 'Type username to access the image path...'
- 'Type password to access the image path...'
- 'Set message TTL (time to live) in milliseconds... *' with the value '7884000000'
- 'Set queue expiration in milliseconds... *' with the value '604800000'

At the bottom right, there are two buttons: a blue 'Save' button and a blue 'Cancel' button.

7. The MySQL export gateway has been created successfully and is shown in the overview in the "Gateways (Export)" area.



6.4.8.6 Creating the AMP 4.1 export gateway

Perform the following steps to create an AMP 4.1 export gateway:

- Step 1: "Select gateway type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define alarm configuration"
- Step 5: "Advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select gateway type	
Select target Store...*	Selection of the gateway
② Define basic configuration	
Type gateway name...*	Gateway name Remark: The designation for "gateway name" is freely selectable.
Type gateway description...	Description of the gateway Remark: The designation for "gateway description" is freely selectable.
Type AMP URI...*	Address of the AMP server
Type AMP port...*	Port of the AMP server
Type AMP path...*	Path of the AMP server to receive data
Type AMP light name...*	Parameterized AMP light name
Type AMP machine name...	Name of the machine
③ Define dataset configuration / Optional	
Pick a client ID	Client ID of the client from which the data is to be sent to the AMP 4.1 export gateway. Remark: All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
Pick a dataset	Selection of the client data set to be sent to the AMP 4.1 export gateway. Remark: All data sets of the selected client ID are displayed.
- OR -	

Parameter	Description
All dataset	All client data is sent to the AMP 4.1 export gateway.
④ Define alarm configuration / Optional	
Pick a client ID	All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
⑤ Advanced configuration / Optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path...*	Path to the gateway docker image
Type username to access the image path...	Username to access the image path
Type password to access the image path...	Password to access the image path
Set message TTL (time to live) in milliseconds...*	AMQP lifetime of the data
Set queue expiration in milliseconds...*	AMQP queue expiration time
Set max queue size...*	AMQP maximum queue size
Set max queue size bytes...*	AMQP maximum queue size in bytes
Select queue mode...*	AMQP queue mode
Set prefetch count...*	AMQP prefetch counter
Set AMP keep alive interval in seconds...*	Send interval for the ConnectionCheck telegrams in seconds
Set AMP timeout in seconds...*	Timeout in seconds
Set AMP sending period in seconds...*	New signals are not sent directly to AMP. Instead, new signals are buffered for a time according to the number of seconds set here and sent together in a block. Remark: The "AMP sending period" parameter must be at least twice as large as "AMP timeout in seconds". If not, the "sending period in seconds" is automatically adapted. A corresponding message appears in the log.
Disable AMP buffering...*	If buffering is to be switched off completely (type of acquisition "online"), set this parameter to "true".
Set AMP buffer...*	Maximum number of signals that should be buffered Remark: The parameter must be greater than the value "prefetch count" of the gateway.
Type AMP CA certificate for https connections...	The CA certificate used for https connections ("\\n" substitutes line breaks)
Enable AMP passthrough mode...*	If signals are to be sent directly to the AMP server without consideration of "sending period in seconds", change this parameter to "true"(type of acquisition "online").

6.4 "Commissioning" area

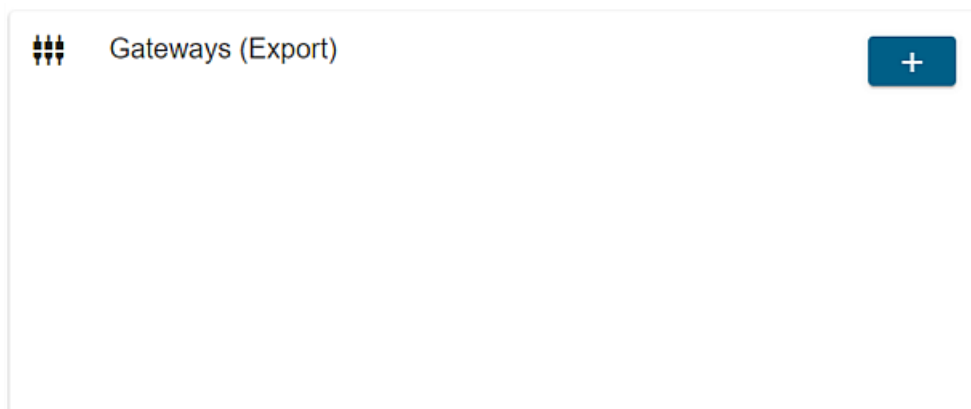
Parameter	Description
Type alarm topic...*	Configuration of the topic: *. *. {deviceId}. *.events Example of a FANUC client with the name "Fanuc01": *. *. Fanuc01. *.events
Type new reading topic...*	Configuration of the topic: *. *. {deviceId}. *.record.* Example of a FANUC client with the name "Fanuc01": *. *. Fanuc01. *.record.*
Type new environment variable...	New environment variables for the image. - Enter the following: "USE_SINGLE_QUEUE=TRUE" - Click on the symbol "+" on the right.

*: Obligatory data

The character sequence to be used for the various entries is provided in the input windows.

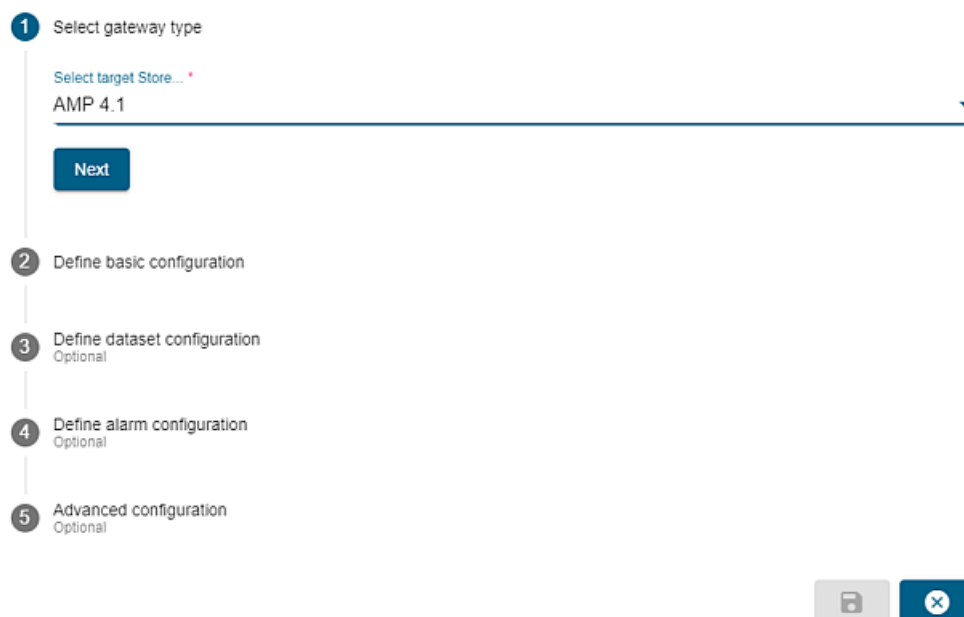
Procedure

1. In the "Gateway (Export)" area, click on "+" to add a new gateway.



2. Step 1: "Select gateway type"
 - Select the entry "AMP 4.1" from the drop-down list.
 - Click on "Next".

Add new gateway



3. Step 2: "Define basic configuration"
 - Populate the fields to perform the basic configuration.
 - Click on "Next".

Add new gateway

2 Define basic configuration

Type gateway name... *

amp-41

Supported input: lowercase letters, numbers, - (must start with letter) 6 / 50

Type gateway description...

This is a AMP 4.1 gateway.

26 / 100

Type AMP URI... *

192.168.6.19

Type AMP port... *

3560

Type AMP path... *

/mcis/vl/http_MDA_MAIN__5.xml

Type AMP light name... *

IOTTEST

Type AMP machine name...

Fanuc32i

Next

4. Step 3: "Define dataset configuration" / Optional
With this step, you define which data is to be sent to the AMP 4.1 export gateway.

- Select one data set, several or all data sets.

Note

Several entries

- Click on "+" to add an additional entry.
-

- Click on "Next".

Add new gateway

1 Select gateway type

2 Define basic configuration

3 Define dataset configuration
Optional

Pick a client ID
amp-client

Pick a dataset +

Next

4 Define alarm configuration
Optional

5 Advanced configuration
Optional

Save Close

6.4 "Commissioning" area

5. Step 4: "Define alarm configuration" / Optional

With this step, you define which alarms are to be sent to the AMP 4.1 export gateway.

- Select the alarms.
- Click on "Next".

Add new gateway

The screenshot shows a vertical progress bar on the left with five steps:

- 1 Select gateway type
- 2 Define basic configuration
- 3 Define dataset configuration
Optional
- 4 Define alarm configuration
Optional
- 5 Advanced configuration
Optional

Step 4 is currently active. Below the progress bar, there is a text input field with the placeholder "Pick a client ID" and the text "amp-client" entered. To the right of the input field is a blue plus sign (+). Below the input field is a blue button labeled "Next". At the bottom right of the form, there are two blue buttons: one with a save icon (floppy disk) and one with a close icon (X).

6. Step 5: "Advanced configuration" / Optional
With this step, you can optionally configure high-frequency data.
Further configurations are usually not necessary.

Note

Advanced configuration

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click on "Save" to save the AMP 4.1 export gateway.

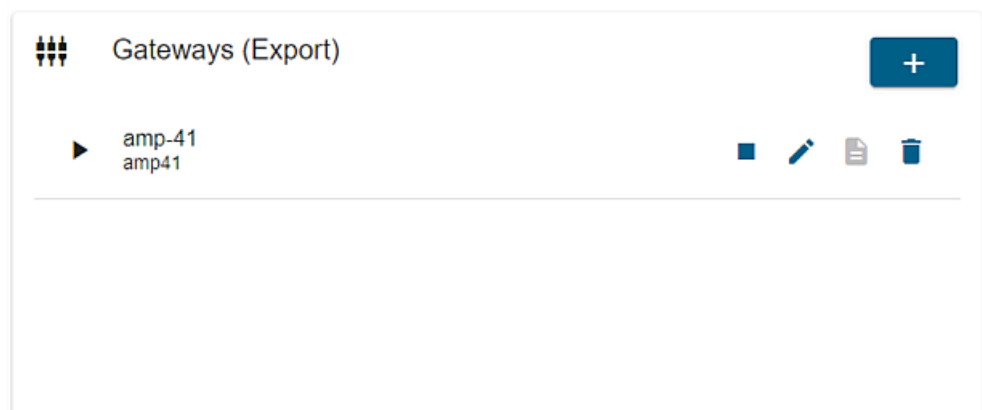
Add new gateway

The screenshot shows a vertical step indicator on the left with five steps: 1. Select gateway type (active), 2. Define basic configuration, 3. Define dataset configuration (Optional), 4. Define alarm configuration (Optional), and 5. Advanced configuration (Optional). The form fields are as follows:

- Type image path... *: `apc-docker-registry.dlinkdns.com/apc-koe/mts/iotserver/gateways/amp-4.1:latest`
- Type username to access the image path...
- Type password to access the image path...
- Set message TTL (time to live) in milliseconds... *: `7884000000`

At the bottom right, there are two buttons: a blue 'Save' button and a grey 'Cancel' button.

7. The AMP 4.1 export gateway has been created successfully and is shown in the overview in the "Gateways (Export)" area.



6.4.8.7 Creating an MQTT export gateway

Perform the following steps to create an MQTT export gateway:

- Step 1: "Select gateway type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define alarm configuration"
- Step 5: "Advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select gateway type	
Select target Store...*	Selection of the gateway
② Define basic configuration	
Type gateway name...*	Gateway name Remark: The designation for "gateway name" is freely selectable.
Type gateway description...	Description of the gateway Remark: The designation for "gateway description" is freely selectable.
Type MQTT broker ip...*	IP address of the MQTT broker
Type MQTT broker client ID...	Unique client ID If no ID is specified, a random ID is generated.
Type MQTT broker username...	User name for authentication on the MQTT broker
Type MQTT broker password...	Password for authentication on the MQTT broker
Type MQTT topic prefix...*	This topic is placed in front of every published message.
Select MQTT QoS...*	Sets the QoS level for publishing messages to the broker.

Parameter	Description
Paste PEM block of CA certificate...	If the connection is established via TLS and the broker uses a certificate signed by an untrusted CA, you must provide this CA as a PEM block. Use the escape sequence ("\\n") to display line feeds and line breaks when entering the certificate. Remark: The server certificate provided by the broker must contain its IP/host name as Subject Alternative Name if this is not sufficient as Common Name.
Paste PEM block of client certificate...	If the broker requires client authentication through a certificate, you must specify a valid clientCert Use the escape sequence ("\\n") to display line feeds and line breaks when entering a client certificate.
Paste PEM block of client key...	If the broker requires client authentication through a certificate, you must specify an un-encrypted clientKey in PEM format. Use the escape sequence ("\\n") to display line feeds and line breaks when entering a key.
③ Define dataset configuration / Optional	
Pick a client ID	Client ID of the client from which data should be sent to the MQTT export gateway. Remark: All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
Pick a dataset	Selection of the client dataset that is to be sent to the MySQL export gateway. Remark: All datasets of the selected client ID are displayed.
- OR -	
All dataset	All client data is sent to the MQTT export gateway.
④ Define alarm configuration / Optional	
Pick a client ID	All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
⑤ Advanced configuration / Optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path...*	Path to the gateway docker image
Type username to access the image path...	User name to access the image path
Type password to access the image path...	Password to access the image path
Set message TTL (time to live) in milliseconds...*	AMQP lifetime of the data
Set queue expiration in milliseconds...*	AMQP queue expiration time

6.4 "Commissioning" area

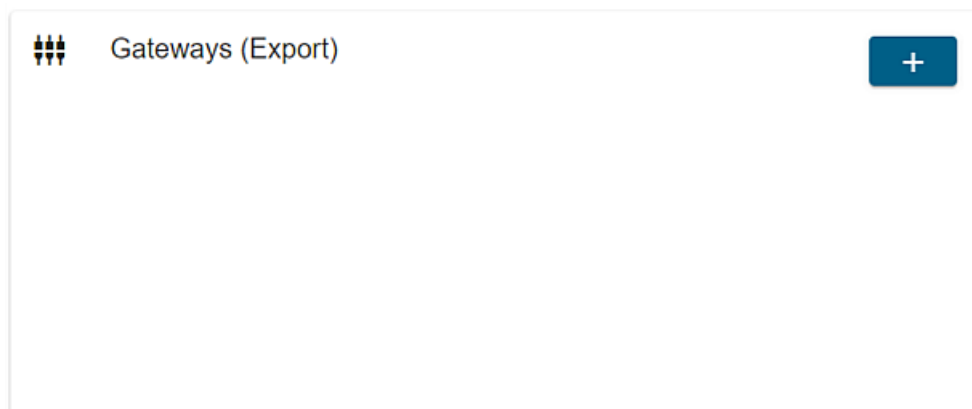
Parameter	Description
Set max queue size...*	AMQP maximum queue size
Set max queue size bytes...*	AMQP maximum queue size in bytes
Select queue mode...*	AMQP queue mode
Set prefetch count...*	AMQP prefetch counter
Set MQTT timeout in nanoseconds...*	Setting the timeout in nanoseconds
Skip TLS verification...*	If you select "true", any TLS verification will be skipped. Remark: Do not use in production if possible
Set the retained flag...*	If you select "true", the message last sent is saved at the broker.
Type alarm topic*	Configuration of the topic: *.*. {deviceId}. *.events Example of a FANUC client with the name "Fanuc01": *. *. Fanuc01. *.events
Type new reading topic*	Configuration of the topic: *. *. {deviceId}. *.record.* Example of a FANUC client with the name "Fanuc01": *. *. Fanuc01. *.record.*
Type new high frequency reading topic ... e.g. '*. *. {clientId}. *.hfddata.*'	Configuration of the topic: *. *. {deviceId}. *.hfddata.* Example of a FANUC client with the name "Fanuc01": *. *. Fanuc01. *.hfddata.*
Type new environment variable*	New environment variable for the image

*: Obligatory data

The character string to be used for the respective entries can be found in the input windows

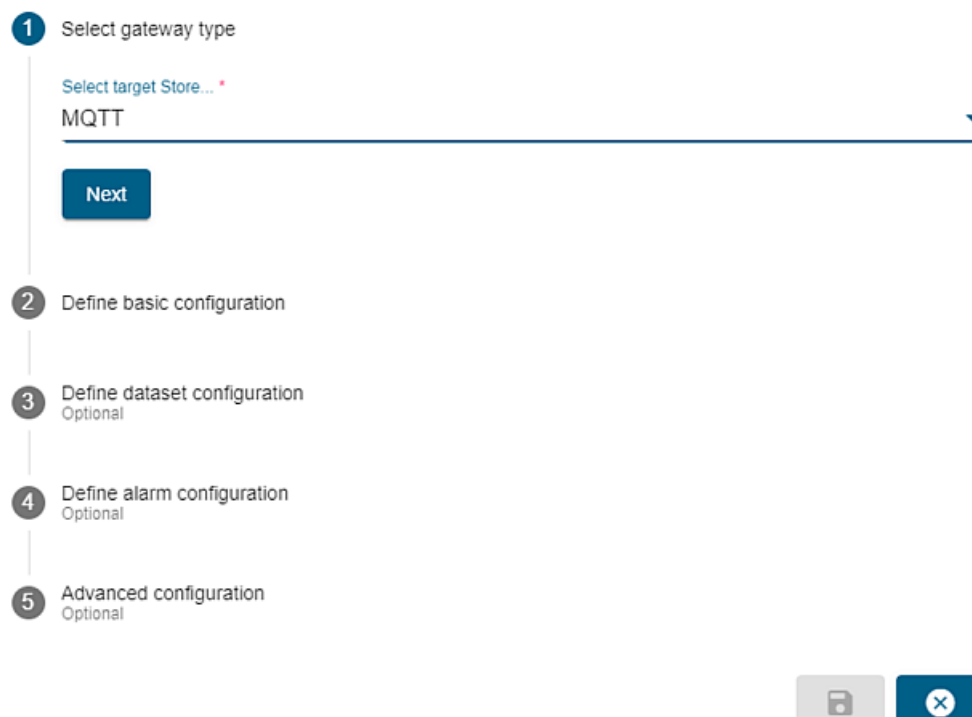
Procedure

1. In the "Gateway (Export)" area, click on "+" to add a new gateway.



2. Step 1: "Select gateway type"
 - Select the entry "MQTT" from the drop-down list.
 - Click on "Next".

Add new gateway



3. Step 2: "Define basic configuration"
 - Populate the fields to perform the basic configuration.
 - Click on "Next".

Add new gateway

2
Define basic configuration

Type gateway name... *

mqtt-gateway

Supported input: lowercase letters, numbers, - (must start with letter) 12 / 50

Type gateway description...

0 / 100

Type MQTT broker ip... e.g. 'tcp://<IP-address>:1883' or 'tls://<IP-address>:8883' *

tcp://192.168.5.93:1883

23 / 100

Type MQTT client ID ...

0 / 100

Type MQTT broker username...

0 / 100

Type MQTT broker password...

0 / 100

Type MQTT topic prefix... *

bfc

3 / 100

Select MQTT QoS... *

1

▼

In case of secured connection provide the certificates:

Paste PEM block of CA certificate...

//

Paste PEM block of client certificate...

//

Paste PEM block of client key...

//

Next

📁
✕

4. Step 3: "Define dataset configuration" / Optional
With this step, you define which data is to be sent to the MQTT export gateway.

- Select one dataset, several or all datasets.

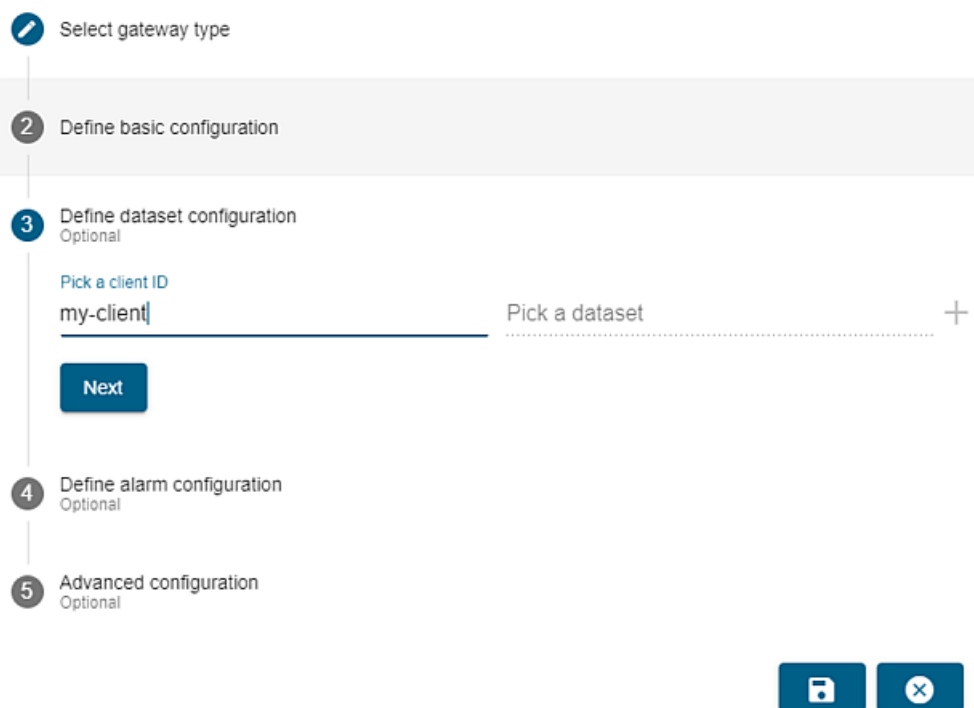
Note

Several entries

- Click on "+" to add an additional entry.
-

- Click on "Next".

Add new gateway



Select gateway type

2 Define basic configuration

3 Define dataset configuration
Optional

Pick a client ID
my-client

Pick a dataset +

Next

4 Define alarm configuration
Optional

5 Advanced configuration
Optional

Save Close

5. Step 4: "Define alarm configuration" / Optional
With this step, you define which alarms are to be sent to the MQTT export gateway.
 - Select the alarms.
 - Click on "Next".

Add new gateway

The screenshot shows a vertical progress bar on the left with five steps:

- 1. Select gateway type (indicated by a pencil icon)
- 2. Define basic configuration
- 3. Define dataset configuration (Optional)
- 4. Define alarm configuration (Optional) - This step is currently active and highlighted with a blue circle.
- 5. Advanced configuration (Optional)

Below the progress bar, the "Define alarm configuration" section is visible. It includes a label "Pick a client ID" in blue text, followed by a text input field containing "my-client". To the right of the input field is a blue plus sign (+). Below the input field is a blue button labeled "Next".

At the bottom right of the interface, there are two blue buttons: one with a save icon (floppy disk) and one with a close icon (X).

6. Step 5: "Advanced configuration" / Optional
 With this step, you can optionally configure high-frequency data.
 Further configurations are usually not necessary.

Note

Advanced configuration

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click on "Save" to save the MQTT export gateway.

Add new gateway

5

Advanced configuration
Optional

Type image path... *

bfc-registry.com/apc-koe/mts/iotserver/gateways/mqtt:latest

Type username to access the image path...

Type password to access the image path...

Set message TTL (time to live) in milliseconds... *

7884000000

Set queue expiration in milliseconds... *

604800000

Set max queue size... *

15000000

Set max queue size bytes... *

5000000000

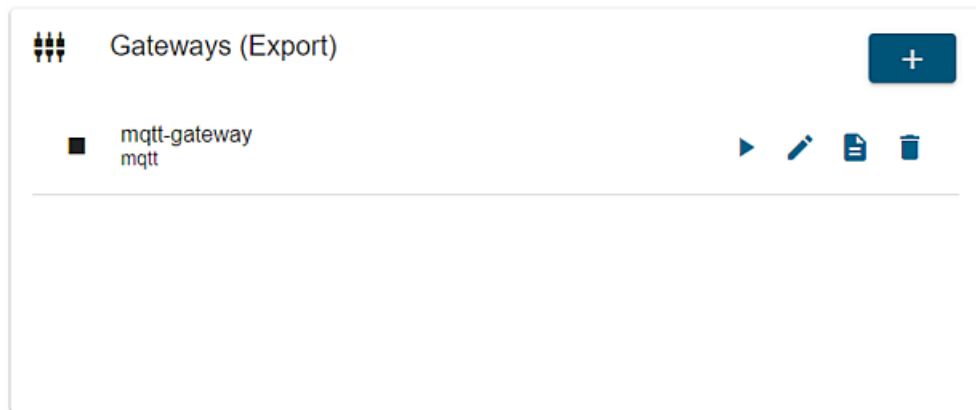
Select queue mode... *

lazy

Save

Cancel

7. The MQTT export gateway has been successfully created and is displayed in the overview in the "Gateways (Export)" area



6.4.8.8 Elasticsearch gateway

Using the BFC service you can forward data to Elasticsearch.

For the data transfer you have to create and configure an Elasticsearch gateway.

Indices

Every day, the Elasticsearch gateway creates an index with the following format:

`{indexPrefix}-{content_type}-{year}-{month}-{day}`

Parameter	Description
<code>{year}</code>	The information is obtained from the received message.
<code>{month}</code>	
<code>{day}</code>	
<code>{indexPrefix}</code>	The specification is set via the configuration and has the default value "bfc".
<code>{content_type}</code>	Templates that map the data have been created for each "content_type" Each mapping has the following properties: • @timestamp • clientID • applicationName • content_type: – dataset - OR - – alarmstatechange

6.4.8.9 Creating an Elasticsearch gateway

To create an Elasticsearch gateway, perform the following steps:

- Step 1: "Select gateway type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define alarm configuration"
- Step 5: "Advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select gateway type	
Select target Store... *	Selection of the gateway
② Define basic configuration	
Type gateway name... *	Gateway name Remark: The designation for "gateway name" is freely selectable.
Type gateway description...	Description of the gateway Remark: The designation for "gateway description" is freely selectable.
Type Elasticsearch URLs (comma separated)...	Address of the ElasticsearchDB Remark: Multiple addresses are separated by commas.
Example of Elasticsearch URLs: https://192.168.0.42:32426,https://mysecondinstance	
Type Elasticsearch username...	User name of the ElasticsearchDB
Type Elasticsearch password...	Password for the user name
Type Elasticsearch index prefix... *	
Paste PEM block of CA certificate...	CA certificate for establishing an encrypted connection to the target system
③ Define dataset configuration / Optional	
Pick a client ID	Client ID of the client from which data is to be sent to the Elasticsearch gateway. Remark: All IDs of the clients, which were created under this "Plant hierarchy", are displayed.

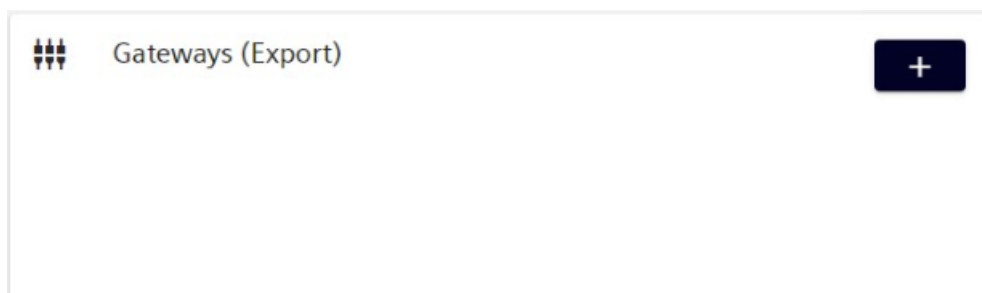
Parameter	Description
Pick a dataset	Selection of the client dataset that is to be sent to the Elasticsearch gateway. Remark: All datasets of the selected client ID are displayed.
- OR -	
All dataset	All client data is sent to the Elasticsearch gateway.
④ Define alarm configuration / Optional	
Pick a client ID	All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
⑤ Advanced configuration / Optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path...*	Path to the Elasticsearch gateway Docker image
Type username to access the image path...	Username to access the image path
Type password to access the image path...	Password to access the image path
Set message TTL (time to live) in milliseconds...*	AMQP lifetime of the data
Set queue expiration in milliseconds...*	AMQP queue expiration time
Set max queue size...*	AMQP maximum queue size
Set max queue size bytes...*	AMQP maximum queue size in bytes
Select queue mode...*	AMQP queue mode
Set prefetch count...*	AMQP prefetch counter
Set Elasticsearch saving period in milliseconds...*	Received data is not posted directly in the database. Instead, the data is buffered for a time corresponding to the number of seconds set here and posted together in the database. Remark: The parameter "Elasticsearch saving period" must be at least 200.
Set Elasticsearch timeout in seconds...*	Timeout in seconds
Skip TLS verification...*	Select "true" in order to trust every server.
Type alarm topic*	MQTT topic alarm event
Type new reading topic*	MQTT topic new dataset
Type new high frequency reading topic*	MQTT topic new high frequency dataset
Type new environment variable*	New environment variable for the image

*: Obligatory data

The character string to be used for the respective entries can be found in the input windows

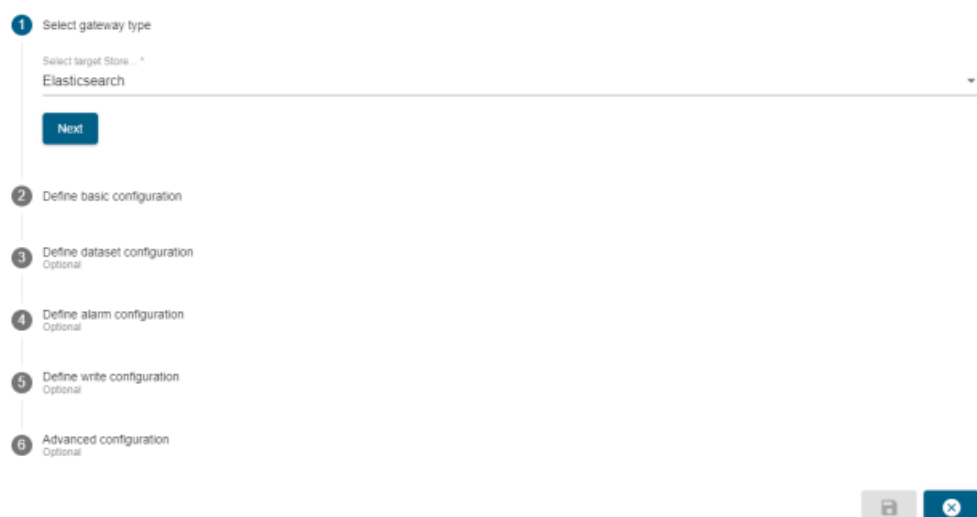
Procedure

1. In the "Gateway (Export)" area, click on "+" to add a new gateway.



2. Step 1: "Select gateway type"
 - From the drop-down list, select entry "Elasticsearch".
 - Click on "Next".

Add new gateway



6.4 "Commissioning" area

3. Step 2: "Define basic configuration"

- Populate the fields to perform the basic configuration.
- Click on "Next".

Add new gateway

1

Select gateway type

2

Define basic configuration

3

Define dataset configuration

Optional

Type gateway name... *

elasticsearch

Supported input: lowercase letters, numbers, - (must start with letter) 13 / 50

Type gateway description...

This is a Elasticsearch Gateway

31 / 100

Type Elasticsearch URLs (comma separated) ... *

http://myaddress:8086,http://192.168.0.115:8086

Type Elasticsearch username...

user

Type Elasticsearch password...

Type Elasticsearch index prefix... *

bfc

Paste PEM block of CA certificate...

Next

Save

Cancel

4. Step 3: "Define dataset configuration" / Optional
With this step, you define which data is to be sent to Elasticsearch.

- Select one data set, several or all data sets.

Note

Several entries

- Click on "+" to add an additional entry.
-

- Click on "Next".

Add new gateway

1 Select gateway type

2 Define basic configuration

3 Define dataset configuration
Optional

Pick a client ID
testclient

Pick a dataset
All datasets

Next

4 Define alarm configuration
Optional

5 Define write configuration
Optional

6 Advanced configuration
Optional

Save Close

5. Step 4: "Define alarm configuration" / Optional
With this step, you define which alarms are to be sent sent to Elasticsearch.

- Select the alarms.

Note

Several entries

- Click on "+" to add an additional entry.
-

- Click on "Next"

Add new gateway

1 Select gateway type

2 Define basic configuration

3 Define dataset configuration
Optional

4 Define alarm configuration
Optional

Pick a client ID
testclient

Next

5 Define write configuration
Optional

6 Advanced configuration
Optional

Document icon Close icon

6. Step 5: "Advanced configuration" / Optional
With this step, you can optionally configure high-frequency data.
Further configurations are usually not necessary.

Note

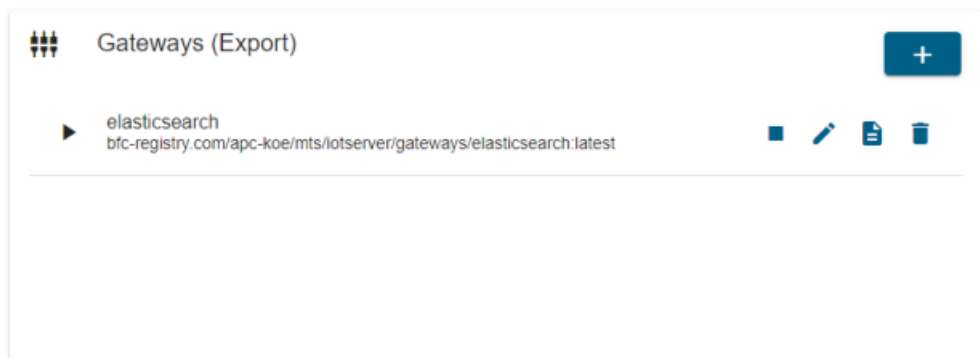
Advanced configuration

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click "Save" to save the Elasticsearch gateway.

Add new gateway

7. The Elasticsearch gateway has been successfully created and is shown in the "Gateways (Export)" area of the overview.



6.4.8.10 InfluxDB gateway

Using the BFC Gateway, you can transfer data to an InfluxDB.

For the data transfer you have to create and configure an InfluxDB gateway.

Requirement

The following InfluxDB versions are supported:

- v1.6.x
- v1.7.x
- v1.8.x

Datasets

The following tags are available:

- deviceID
- applicationName
- contentType → dataSet
- name → {DataSetName}

Each contained datapoint is stored in the measurement as a field with the name {DataSetName}.

Alarms

The following tags are available:

- deviceID
- applicationName
- contentType → alarmStateChange
- alarmNumber
- alarmState

Alarms are saved in the measurement "bfc_alarms".

The following fields are available:

- alarmMessage
- data_ + "any" data key

HFDData

The following tags are available:

- deviceID
- applicationName
- contentType → hfDataSet
- name → {HFDDataSetConfigName}
- hfDataID → contains unique ID of a measurement

Each contained datapoint is stored in the measurement as a field with the name {HFDDataSetConfigName}. Exception: The datapoint is set to the value "zero".

6.4.8.11 Creating an InfluxDB gateway

Perform the following steps to create an InfluxDB gateway:

- Step 1: "Select gateway type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define alarm configuration"
- Step 5: "Advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select gateway type	
Select target Store...*	Selection of the gateway
② Define basic configuration	
Type gateway name...*	Gateway name Remark: The designation for "gateway name" is freely selectable.
Type gateway description...	Description of the gateway Remark: The designation for "gateway description" is freely selectable.
Type InfluxDB URI...	Address of the InfluxDB
Type InfluxDB username...	User name for logging onto the InfluxDB
Type InfluxDB password...	Password for the specified user name
Type InfluxDB database...	Name of the database in which the data is stored
Paste PEM block of CA certificate...	CA certificate for establishing an encrypted connection to the target system
③ Define dataset configuration / Optional	
Pick a client ID	Client ID of the client from which data is to be sent to the InfluxDB gateway. Remark: All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
Pick a dataset	Selection of the client dataset that is to be sent to the InfluxDB gateway. Remark: All datasets of the selected client ID are displayed.
- OR -	

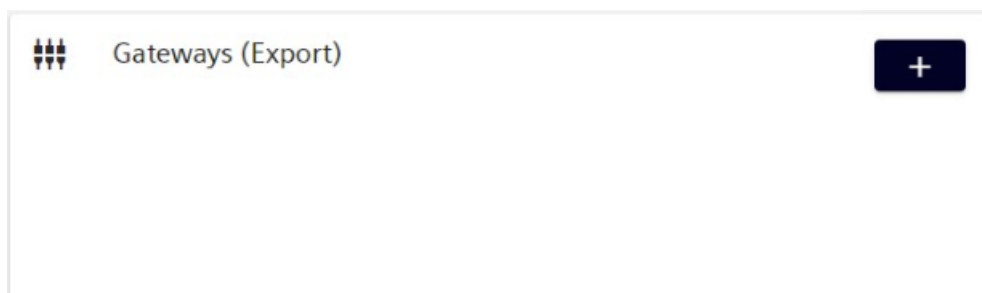
Parameter	Description
All dataset	All client data is sent to the InfluxDB gateway.
④ Define alarm configuration / Optional	
Pick a client ID	All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
⑤ Advanced configuration / Optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path...*	Path to the InfluxDB gateway Docker image
Type username to access the image path...	Username to access the image path
Type password to access the image path...	Password to access the image path
Set message TTL (time to live) in milliseconds...*	AMQP lifetime of the data
Set queue expiration in milliseconds...*	AMQP queue expiration time
Set max queue size...*	AMQP maximum queue size
Set max queue size bytes...*	AMQP maximum queue size in bytes
Select queue mode...*	AMQP queue mode
Set prefetch count...*	AMQP prefetch counter
Set InfluxDB saving period in milliseconds...*	Received data is not posted directly in the database. Instead, the data is buffered for a time corresponding to the number of seconds set here and posted together in the database. Remark: The parameter "InfluxDB saving period" must be at least 200.
Set InfluxDB timeout in seconds...*	Timeout in seconds
Type alarm topic*	MQTT topic alarm event
Type new reading topic*	MQTT topic new dataset
Type new high frequency reading topic*	MQTT topic new high frequency dataset
Type new environment variable*	New environment variable for the image

*: Obligatory data

The character string to be used for the respective entries can be found in the input windows

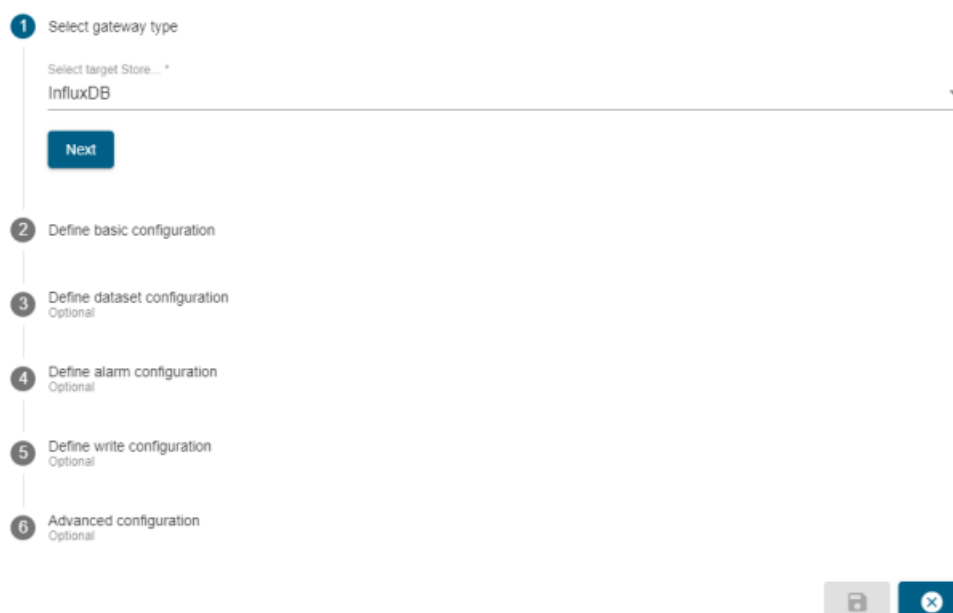
Procedure

1. In the "Gateway (Export)" area, click on "+" to add a new gateway.



2. Step 1: "Select gateway type"
 - From the drop-down list, select entry "InfluxDB".
 - Click on "Next".

Add new gateway

A screenshot of a multi-step wizard titled "Add new gateway". Step 1, "Select gateway type", is active. It contains a dropdown menu labeled "Select target Store..." with "InfluxDB" selected. A "Next" button is below the dropdown. A vertical progress bar on the left shows steps 1 through 6. Steps 2-6 are labeled "Define basic configuration", "Define dataset configuration Optional", "Define alarm configuration Optional", "Define write configuration Optional", and "Advanced configuration Optional" respectively. At the bottom right are "Save" and "Close" buttons.

3. Step 2: "Define basic configuration"
 - Populate the fields to perform the basic configuration.
 - Click on "Next".

Add new gateway

1

Select gateway type

2

Define basic configuration

3

Define dataset configuration
Optional

Type gateway name... *

influxdb

Type gateway description...

This is a InfluxDB Gateway

Type InfluxDB URI... *

http://mydbaddress:8086

Type InfluxDB username...

dbuser

Type InfluxDB password...

Type InfluxDB database... *

bfcdatabase

Paste PEM block of CA certificate...

Next

Save

Cancel

4. Step 3: "Define dataset configuration" / Optional
With this step, you define which data is to be sent to the InfluxDB.

- Select one data set, several or all data sets.

Note**Several entries**

- Click on "+" to add an additional entry.
-

- Click on "Next".

Add new gateway

The screenshot shows a vertical wizard with six steps. Step 3, 'Define dataset configuration', is highlighted in grey and marked as 'Optional'. It contains two input fields: 'Pick a client ID' with the value 'testclient' and 'Pick a dataset' with the value '**All datasets**'. A blue '+' icon is to the right of the dataset field. A blue 'Next' button is below the inputs. Steps 1-2 are completed, and steps 4-6 are optional. At the bottom right are 'Save' and 'Close' buttons.

1 Select gateway type

2 Define basic configuration

3 Define dataset configuration
Optional

Pick a client ID
testclient

Pick a dataset
All datasets

Next

4 Define alarm configuration
Optional

5 Define write configuration
Optional

6 Advanced configuration
Optional

Save Close

6.4 "Commissioning" area

5. Step 4: "Define alarm configuration" / Optional
With this step, you define which alarms are to be sent to InfluxDB.

- Select the alarms.

Note

Several entries

- Click on "+" to add an additional entry.
-

- Click on "Next"

Add new gateway

1 Select gateway type

2 Define basic configuration

3 Define dataset configuration
Optional

4 Define alarm configuration
Optional

Pick a client ID
testclient

Next

5 Define write configuration
Optional

6 Advanced configuration
Optional

Save Cancel

6. Step 5: "Advanced configuration" / Optional
With this step, you can optionally configure high-frequency data.
Further configurations are usually not necessary.

Note

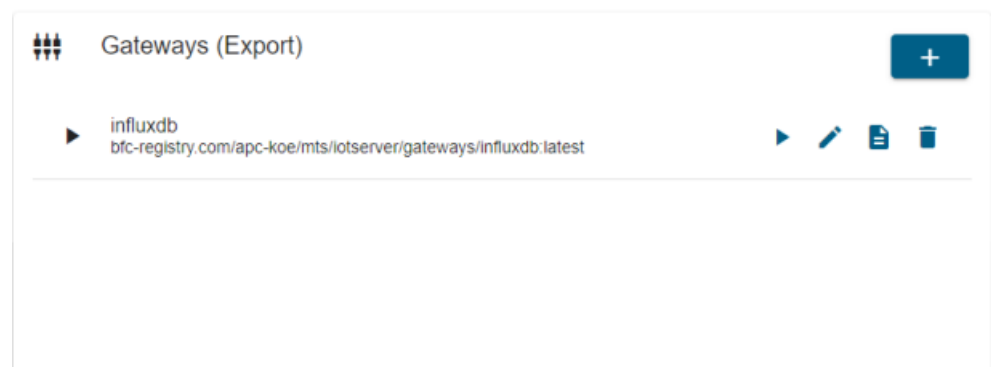
Advanced configuration

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click "Save" to save the InfluxDB gateway.

Add new gateway

7. The InfluxDB gateway has been successfully created and is shown in the "Gateways (Export)" area of the overview.



6.4.8.12 HTTP gateway

Using the BFC service you can forward data via HTTP. In this way you can send data to Node-Red, for example.

For the data transfer you have to create and configure an HTTP gateway.

All data is forwarded to the specified end points.

The following end points can be configured:

- datasets
- hfdatasets
- alarmlists
- alarmstatechanges

The following path is used for an end point that is not configured:

`/clients/{clientID}/types/{applicationType}/{endpoint}`

The POST method is used as standard.

6.4.8.13 Creating an HTTP gateway

Perform the following steps to create an HTTP gateway:

- Step 1: "Select gateway type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define alarm configuration"
- Step 6: "Advanced configuration"

Requirement

The "Commissioning" area is open.

Parameter

Parameter	Description
① Select gateway type	
Select target Store...*	Selection of the gateway
② Define basic configuration	
Type gateway name...*	Gateway name Remark: The designation for "gateway name" is freely selectable.
Type gateway description...	Description of the gateway Remark: The designation for "gateway description" is freely selectable.
Type HTTP webserver URL...	Address of the HTTP web server
Type HTTP webserver username...	User name for logging onto the web server
Type HTTP webserver password...	Password for the user name

Parameter	Description
Paste PEM block of CA certificate...	CA certificate for establishing an encrypted connection to the target system
Type HTTP webserver path for datasets...	Address to which the datasets are to be forwarded Remark: The HTTP methods "PUT" and "POST" are available.
Type HTTP webserver path for hfdatasets...	Address to which the hfdatasets are to be forwarded Remark: The HTTP methods "PUT" and "POST" are available.
Type HTTP webserver path for alarmlists...	Address to which the alarm lists are to be forwarded Remark: The HTTP methods "PUT" and "POST" are available.
Type HTTP webserver path for alarmstatechanges...	Address to which the alarm state changes are to be forwarded Remark: The HTTP methods "PUT" and "POST" are available.
③ Define dataset configuration / Optional	
Pick a client ID	Client ID of the client from which data is to be sent to the HTTP gateway. Remark: All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
Pick a dataset	Selection of the client dataset that is to be sent to the HTTP gateway. Remark: All datasets of the selected client ID are displayed.
- OR -	
All dataset	All client data is sent to the HTTP gateway.
④ Define alarm configuration / Optional	
Pick a client ID	All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
⑤ Define write configuration / Optional	
⑥ Advanced configuration / Optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path...*	Path to the HTTP gateway Docker image
Type username to access the image path...	Username to access the image path
Type password to access the image path...	Password to access the image path
Set message TTL (time to live) in milliseconds...*	AMQP lifetime of the data

6.4 "Commissioning" area

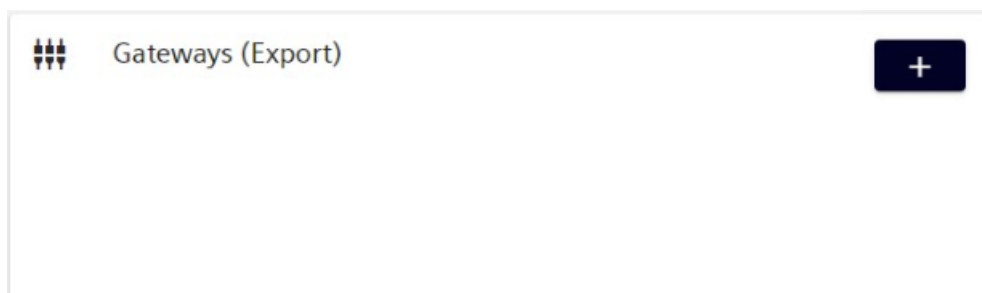
Parameter	Description
Set queue expiration in milliseconds...*	AMQP queue expiration time
Set max queue size...*	AMQP maximum queue size
Set max queue size bytes...*	AMQP maximum queue size in bytes
Select queue mode...*	AMQP queue mode
Set prefetch count...*	AMQP prefetch counter
Set HTTP webserver timeout in seconds...*	Timeout in seconds
Skip TLS verification...*	Select "true" in order to trust every server.
Type alarm topic*	MQTT topic alarm event
Type new reading topic*	MQTT topic new dataset
Type new high frequency reading topic*	MQTT topic new high frequency dataset
Type new environment variable*	New environment variable for the image

*: Obligatory data

The character string to be used for the respective entries can be found in the input windows

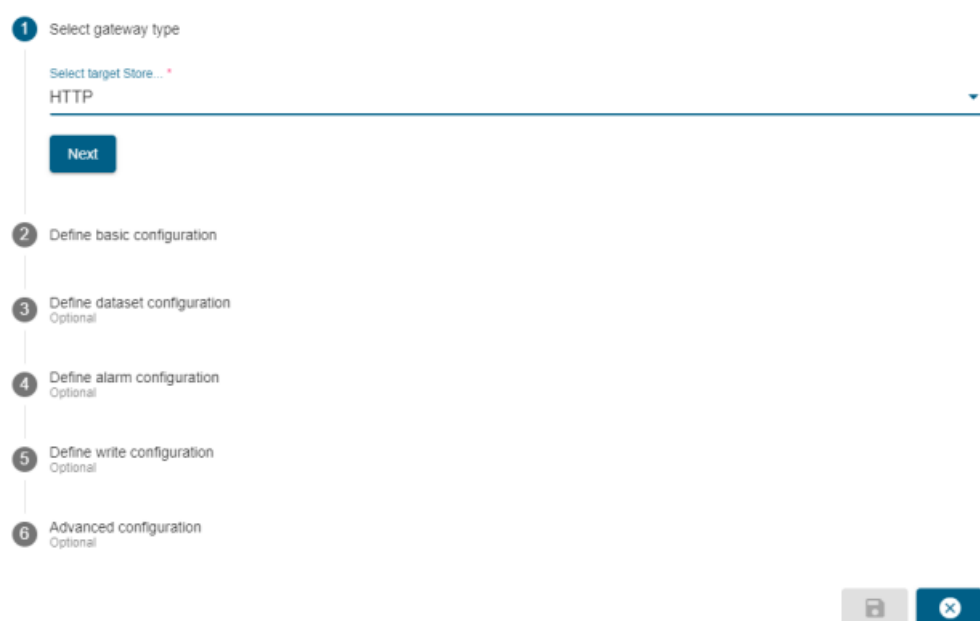
Procedure

1. In the "Gateway (Export)" area, click on "+" to add a new gateway.



2. Step 1: "Select gateway type"
 - From the drop-down list, select entry "HTTP".
 - Click on "Next".

Add new gateway



3. Step 2: "Define basic configuration"
 - Populate the fields to perform the basic configuration.
 - Click on "Next".

Add new gateway

1 Select gateway type

2 Define basic configuration

3 Define dataset configuration
Optional

4 Define alarm configuration
Optional

5 Define write configuration
Optional

Type gateway name... *

http

Supported input: lowercase letters, numbers, - (must start with letter)

Type gateway description...

This is a HTTP Gateway

Type HTTP webservice URL... *

http://192.168.6.12:80

Type HTTP webservice username...

User

Type HTTP webservice password...

Paste PEM block of CA certificate...

The following endpoints may be configured:

- datasets
- hfdatasets
- alarmlists
- alarmstatechanges

By default the path "/clients/{clientID}/types/{applicationType}/{endpoint}" will be used.
The default method is POST.

Type HTTP webservice path for datasets...	Select HTTP request mode... *
/my/data/datasets	POST ▼
Type HTTP webservice path for hfdatasets...	Select HTTP request mode... *
/my/data/hfdatasets	PUT ▼
Type HTTP webservice path for alarmlists...	Select HTTP request mode... *
/my/alarms/alarmlists	PUT ▼
Type HTTP webservice path for alarmstatechanges...	Select HTTP request mode... *
/my/alarms/alarmstatechanges	POST ▼

Next

⏮
✖

4. Step 3: "Define dataset configuration" / Optional
With this step, you define which data is to be sent via HTTP.

- Select one data set, several or all data sets.

Note**Several entries**

- Click on "+" to add an additional entry.
-

- Click on "Next".

Add new gateway

The screenshot shows a vertical wizard with six steps. Step 3, 'Define dataset configuration', is highlighted and labeled as 'Optional'. It contains two input fields: 'Pick a client ID' with the value 'testclient' and 'Pick a dataset' with the value '**All datasets**'. A blue '+' icon is to the right of the dataset field. A blue 'Next' button is below the inputs. Steps 4 through 6 are listed below: 'Define alarm configuration' (Optional), 'Define write configuration' (Optional), and 'Advanced configuration' (Optional). At the bottom right are 'Save' and 'Close' buttons.

1 Select gateway type

2 Define basic configuration

3 Define dataset configuration
Optional

Pick a client ID
testclient

Pick a dataset
All datasets

Next

4 Define alarm configuration
Optional

5 Define write configuration
Optional

6 Advanced configuration
Optional

Save Close

5. Step 4: "Define alarm configuration" / Optional
With this step, you define which alarms are to be sent via HTTP.
 - Select the alarms.

Note

Several entries

- Click on "+" to add an additional entry.
-

- Click on "Next"

Add new gateway

1 Select gateway type

2 Define basic configuration

3 Define dataset configuration
Optional

4 Define alarm configuration
Optional

Pick a client ID

testclient

Next

5 Define write configuration
Optional

6 Advanced configuration
Optional

Save Close

6. Step 6: "Advanced configuration" / Optional
With this step, you can optionally configure high-frequency data.
Further configurations are usually not necessary.

Note

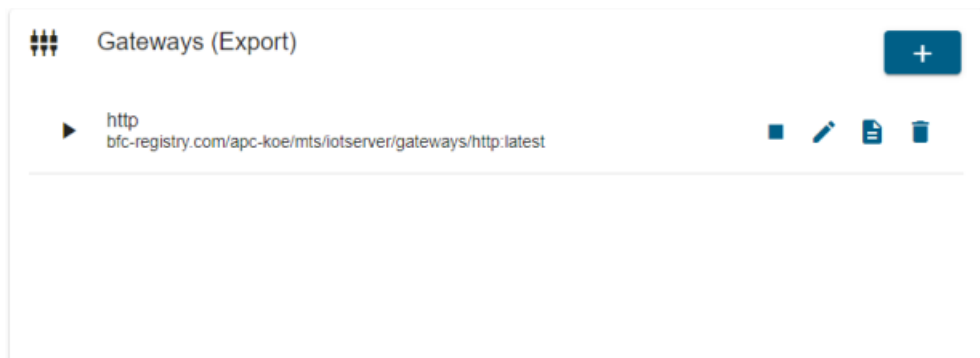
Advanced configuration

Entries and changes are only made in the case of service, and only performed by the hotline.

- Click on "Save" to save the HTTP gateway.

Add new gateway

7. The HTTP gateway has been successfully created, and is shown in the "Gateways (Export)" area of the overview.



6.4.8.14 MS SQL gateway

MS SQL gateway allows you to write selected data such as "Datasets" and "Alarms" to a database of a Microsoft SQL Server database management system (DBMS).

For the data transfer you have to create and configure an MS SQL gateway.

System requirements for MS SQL Server database

The MS SQL gateway is tested and approved for Microsoft SQL Server 2019.

To use the MS SQL gateway, you need an installed Microsoft SQL Server on a Windows system or Microsoft SQL Server as a docker container as the target system. Server licenses and client access licenses may be required to use the Microsoft SQL Server.

Logging in to the database can be done via a database or Windows login. Other login methods are not supported.

The MS SQL Server gateway supports fixed TCP ports which differ from the default port 1433/tcp and dynamic ports if the "SQL Server Browser" service is enabled and reachable (1434/udp). Specification of the SQL Server instance is only supported with an active "SQL Server Browser" and dynamic ports.

The MS SQL Server gateway supports unencrypted and encrypted data transmission to the Microsoft SQL Server.

To check the server certificate, the CA/issuer certificate must be present at the BFC gateway. This is usually done by importing the CA/issuer certificate as a PEM file into the Linux operating system's certificate list of trusted certificate authorities in the "/etc/ssl/certs" directory. Refer to the documentation of the Linux distribution used for importing CA/issuer certificates.

Note

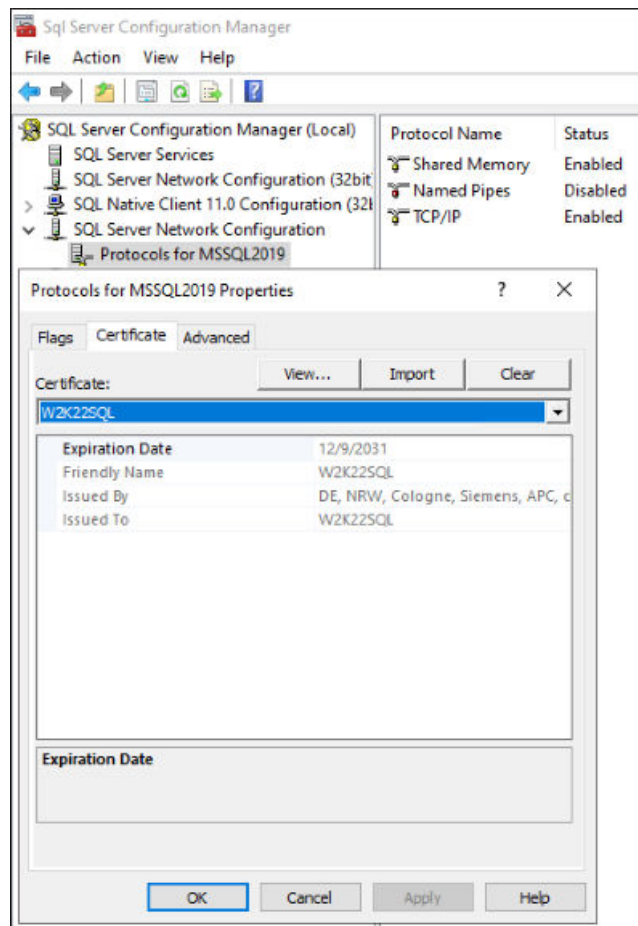
Time limit of the certificates

Observe the time limit of the certificates and organize the certificate changes in time before their expiry date.

Setting up encrypted communication on the MS SQL Server on Windows

For encrypted communication, the MS SQL Server must be configured with a certificate. This is configured with the "SQL Server Configuration Manager" on the MS SQL Server.

1. Right-click to open the properties dialog of the database instance used under "SQL Server Network Configuration".
Open the "Certificate" tab and assign a certificate from the Windows computer's certificate store.



2. The configuration becomes active only after restarting the MS SQL Server instance.

Setting up encrypted communication on the MS SQL Server docker container

1. Create an MS SQL Server Docker Container with certificates for encrypted communication with the following Docker Run command:


```
docker run -e 'ACCEPT\_EULA=Y'
-e 'SA\_PASSWORD=ChangeMeImPublic!'
-p 1433:1433
-e 'MSSQL\_PID=Standard'
--name sql1
-h sql1
-v sql1data:/var/opt/mssql
-v /home/apc/mssql/cert/mssql.pem:/var/opt/mssql/mssql.pem
-v /home/apc/mssql/cert/mssql.key:/var/opt/mssql/mssql.key
-v /home/apc/mssql/mssql.conf:/var/opt/mssql/mssql.conf
-d mcr.microsoft.com/mssql/server:2019-latest
```

 - The command was divided into several lines for better readability.
 - The certificates are configured using volume mappings of the key and certificate files and the configuration file "mssql.conf".
The file "mssql.conf" contains, for example:


```
[network]
tlscert = /var/opt/mssql/mssql.pem
tlskey = /var/opt/mssql/mssql.key
tlsprotocols = 1.2
forceencryption = 0
```
 - Adapt the paths that begin with "/home/apc/" to the actual paths of your file system. The files must be readable by the docker process.
2. First, change the SA password if administrators of the docker system should not automatically also be administrators of the MS SQL Server.


```
docker exec -it sql1 /opt/mssql-tools/bin/sqlcmd -S localhost
-U SA -P "ChangeMeImPublic!"
-Q "ALTER LOGIN SA WITH PASSWORD='theSecret!'"
```

 - The command was divided into several lines for better readability.
 - Use "Microsoft SQL Server Management Studio" for further administration of the database instance.

Setting up the database and database login

1. In production systems, create the database with administrative SQL Server privileges before using it for the first time. The database login in this database must be assigned the necessary rights of the [db_owner] role.

```
--use master
```

- Alternatively, you can grant the database login used the right to create new databases (not recommended for production systems).

```
--GRANT CREATE ANY DATABASE TO [bfclogin] - not recommended
```

- The default value for the database name is "bfc". If you enter a different name, use the same database name in the MS SQL gateway configuration.

2. Create a new database and database login (Windows/Docker) as follows:

```
[master]
create login [bfclogin] with password='theSecret'
create database [bfc]
goALTER DATABASE [bfc] SET RECOVERY SIMPLE
go
use [bfc]
CREATE USER [bfcuser] FOR LOGIN [bfclogin]
ALTER ROLE [db\_owner] ADD MEMBER [bfcuser]
```

- OR -

On a Windows-hosted MS SQL Server database, you can alternatively use a Windows login. Observe the password guidelines (maximum validity) of the system and organize any password changes that may be required.

Create a new database and grant access for an existing Windows login as follows:

```
[master]
CREATE LOGIN [W2K22SQL\bfcwinlogin] FROM WINDOWS
create database [bfc]
go
ALTER DATABASE [bfc] SET RECOVERY SIMPLE
go
use [bfc]
CREATE USER [bfcwinuser] FOR LOGIN [W2K22SQL\bfcwinlogin]
ALTER ROLE [db\_owner] ADD MEMBER [bfcwinuser]
```

Note

Any values

The values [bfc], [bfclogin], [bfcuser], [W2K22SQL\bfcwinlogin] and [bfcwinuser] are examples. You can choose the values on the production system as required.

Note

"Full" or "Bulk logged" recovery setting

If you back up and shorten the transaction logs on the production Microsoft SQL Server, you can also operate the database with "Full" or "Bulk logged" recovery setting.

Database structure

The MS SQL gateway automatically creates the database tables with the relationships.

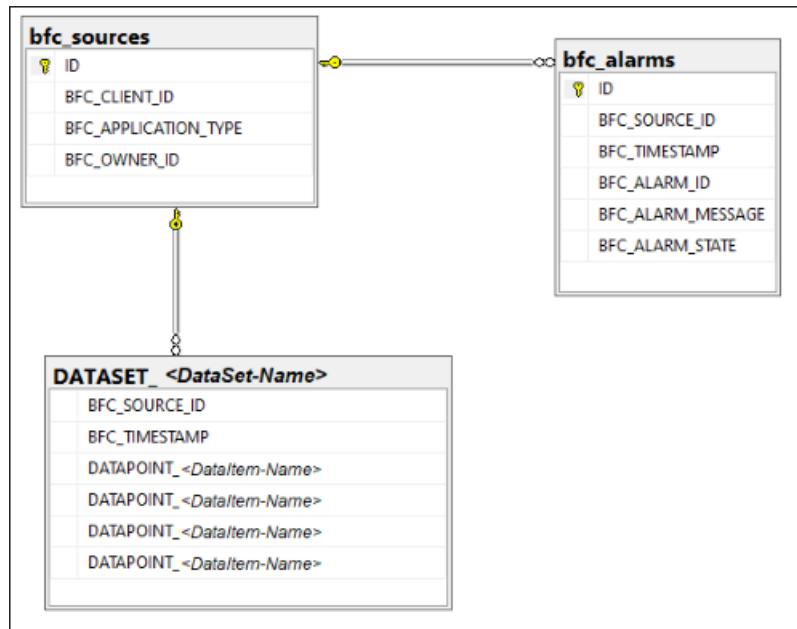


Table "bfc_sources"

The table "bfc_sources" contains a unique identifier for the combination of client ID and the sending application on the data bus.

Column	Meaning
BFC_CLIENT_ID	Identifier or identification of the machine in BFC, e.g. "ConveyorPLC0376"
BFC_APPLICATION_TYPE	Sender identifier of the application, e.g. "s7client"
BFC_OWNER_ID	Currently not used

Table "bfc_alarms"

In the table "bfc_alarms", all alarms of all clients are stored.

Column	Meaning
BFC_SOURCE_ID	External key for column [ID] in [bfc_sources]
BFC_TIMESTAMP	Time stamp of message
BFC_ALARM_ID	Alarm number (e.g. 3000) or identifier (character string)
BFC_ALARM_MESSAGE	Alarm text (e.g. "Emergency stop")
BFC_ALARM_STATE	Contains the values "active" or "dismissed"

To free up memory space, delete the processed lines from the table using the reading application. The time stamps are taken from the alarm source.

Tables "DATASET_ *DataSet-Name*"

The reported DataSets that are configured for recording in the MS SQL Server gateway settings are stored in the "DATASET_ *DataSet-Name*" tables. A separate table is created for

each DataSet name. When a new data point/item is added to a DataSet, a new column with the appropriate data type is automatically added. The data type of the table column cannot be changed automatically. In case of a data type change of an item in a DataSet, the table must be deleted or the column must be renamed or deleted manually. Unused DATAPOINT columns of deleted or renamed items are not automatically removed.

Column	Meaning
BFC_SOURCE_ID	External key for column [ID] in [bfc_sources]
BFC_TIMESTAMP	Timestamp of the data message
DATAPOINT_Item name	
...	

The "DATAPOINT" columns are defined for the following database data types depending on the stored data type in the BFC gateway:

Type in DataSet	Type in database
"string"	nvarchar(max) (up to 2 GB)
"int"	bigint (64 bit)
"bool"	bit
"float"	float (64 bit)

To free up memory space, delete the processed lines from the table using the reading application.

Tables that are not automatically reduced or cleared can occupy the entire database storage space over time.

6.4.8.15 Creating an MS SQL gateway

Perform the following steps to create an MS SQL gateway:

- Step 1: "Select gateway type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define alarm configuration"
- Step 5: "Define write configuration"
- Step 6: "Advanced configuration"

Requirement

The "Commissioning" area is open.

Parameters

Parameters	Description
① Select gateway type	
Select target Store...	Selection of the gateway
② Define basic configuration	
Type gateway name...	Gateway name Remark: The designation for "gateway name" is freely selectable.
Type gateway description...	Description of the gateway Remark: The designation for "gateway description" is freely selectable.
Type MS SQL DB IP or hostname...	TCP/IP address, computer name or DNS name of the MS SQL Server Optionally, the port can be configured (e.g.: 192.168.1.15:14330). If no port is specified, then the default value 1433 is used. As an alternative to the port, the instance name of the database server can be specified (e.g.: 192.168.1.15/ PRODUCTION). The use of dynamic SQL Server ports is possible if the "SQL Server Browser" service is active and accessible on the database server.
Type MS SQL protocol type...	Currently, only the default value "sqlserver" is accepted.
Type MS SQL DB username...	The login account used to access the database is specified here. The login can be a Windows login, which is authorized for access in the SQL server, or a database login. For Windows logins, the domain or server name must be specified (e.g.: DBSRVPROD\bfclogin for Windows login, bfclogin for database login) For more information about the necessary rights of the login, see Chapter: MS SQL gateway (Page 277). The use of the SA login is not recommended.
Type MS SQL DB password...	Password for the previously configured login
Type MS SQL DB parameters...	Additional parameters can be specified here. Insert "&" as separator between parameters (e.g.: "database=bfcdata&encrypt=true")
• database	Name of the database Only necessary if it differs from the default name "bfc".
• app name	Facilitates the assignment of database sessions on the MS SQL server. (e.g.: app name=mssqlgateway)
• Workstation ID	Facilitates the assignment of database connections on the MS SQL server. (e.g.: Workstation ID=bfcgateway)

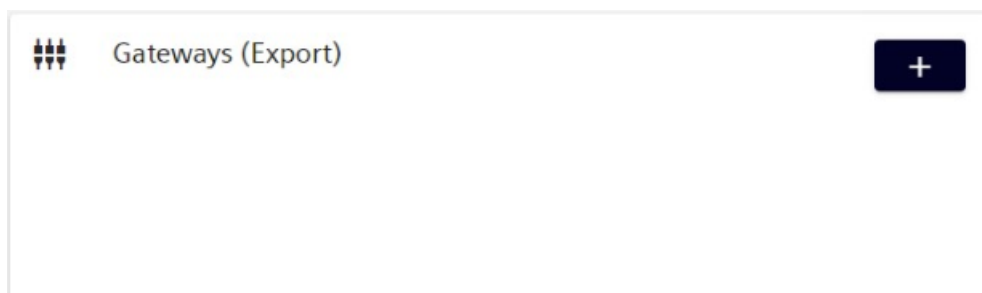
Parameters	Description
encrypt	"true" activates the certificate-based encryption of the communication channel. This requires a correct certificate configuration on the MS SQL server and BFC gateway. For more information on using certificates with MS SQL servers, see Chapter: MS SQL gateway (Page 277). (e.g.: encrypt=true)
hostNameInCertificate	If the server certificate is issued for DNS or system names, but IP addresses are used for addressing the MS SQL Server, the name contained in the certificate can be specified here for certificate validation. (e.g.: hostNameInCertificate=sql1.example.net)
TrustServerCertificate	Disables the verification of the server certificate, but allows encrypted communication with insecure certificates. Caution: This option can undermine communication security through encryption. An incorrect certificate configuration will not be corrected by this option. Use this option only in case of extreme emergency for certificates that are invalid for a short time (expired, not yet valid, etc.). (e.g.: TrustServerCertificate=false)
Type MS SQL maximal connections...	Maximum number of parallel connections to the MS SQL server. Parallel connections can increase the update speed and system load (network, CPU, RAM, disk).
③ Define dataset configuration / Optional	
Pick a client ID	Name of the client or machine from which data is to be transferred to the MS SQL Server database.
Pick a dataset	Selection of all datasets (not recommended) or a specific dataset (recommended) of the machine. Do not fill the database with data that you do not use. The reading process should automatically delete the processed data to avoid unlimited growth of database tables.
④ Define alarm configuration / Optional	
Pick a client ID	Name of the client or machine from which data is to be transferred to the MS SQL Server database. Do not fill the database with data that you do not use. The reading process should automatically delete the processed data to avoid unlimited growth of database tables.
⑤ Define write configuration / Optional	
Writing data in direction of the machine is not supported.	
⑥ Advanced configuration / Optional	
Remark:	
These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path...*	Path to the gateway docker image
Type username to access the image path...	User name to access the image path
Type password to access the image path...	Password to access the image path

Parameters	Description
Set message TTL (time to live) in milliseconds...	AMQP lifetime of the data in the queue
Set queue expiration in milliseconds...	AMQP queue expiration time
Set max queue size...	AMQP maximum queue size
Set max queue size bytes...	AMQP maximum queue size in bytes
Select queue mode...	AMQP queue mode
Set prefetch count...	AMQP prefetch counter
Do not create new Database...	In the default setting, the gateway creates the target database if it does not already exist.
Do not create new Tables...	In the default setting, the gateway creates tables for the data in the target database.
Type alarm topic...	Configuration of the topic: *. *. {deviceId}. *. events To adapt the selection defined in step ④.
Type new reading topic...	Configuration of the topic: *. *. {deviceId}. *. record. * To adapt the selection defined in step ③.
Type new high frequency reading topic...	For configuring high-frequency data: *. *. {clientId}. *. hfdata. # (currently not available)
Type a new environment variable...	New environment variable for the image (e.g.: TRACE=TRUE)

*: Obligatory data

Procedure

1. In the "Gateway (Export)" area, click on "+" to add a new gateway.



2. Step 1: "Select gateway type"
 - Select the entry "MS SQL" from the drop-down list.
 - Click on "Next".

Add new gateway

A screenshot of the "Add new gateway" process, specifically step 1: "Select gateway type". It shows a form with a label "Select target Store..." and a dropdown menu displaying "MS SQL". A dark blue "Next" button is located below the dropdown.

3. Step 2: "Define basic configuration"
 - Populate the fields to perform the basic configuration.
 - Click on "Next".

2 Define basic configuration

Type gateway name... *

dataToMSSQL

Supported input: letters, numbers, -, _ (must start with letter or number) 11 / 50

Type gateway description...

0 / 100

Type MS SQL DB IP or hostname... *

DBSERVER/PRODUCTIONDATA

Type MS SQL protocol type... *

sqlserver

Type MS SQL DB username... *

bfclogin

8 / 100

Type MS SQL DB password... *

.....

Type MS SQL DB parameters... e.g. 'param1=value1&...&paramN=valueN'

database=production&encrypt=true&app name=BFC

Type MS SQL maximal connections... *

1

Next

4. Step 3: "Define dataset configuration" / Optional
With this step, you define which data is to be sent to the MS SQL gateway.

- Select one data set, several or all data sets.

Note

Several entries

- Click on "+" to add an additional entry.

- Click on "Next".

3 Define dataset configuration
Optional

Pick a client ID
870

Pick a dataset
DS1

Next

5. Step 4: "Define alarm configuration" / Optional
With this step, you define which alarms are to be sent to the MS SQL gateway.

- Select the alarms.
- Click on "Next".

4 Define alarm configuration
Optional

Pick a client ID
870

Next

6. Step 5: "Define write configuration" / Optional
The MS SQL gateway does not currently support write operations in direction of the machine.

- Click on "Next".

5 Define write configuration
Optional

Pick a client ID

Type address you want to write...

Select address type...
Float

Next

6.4 "Commissioning" area

7. Step 6: "Advanced configuration" / Optional
Further configurations are usually not necessary.

Note

Advanced configuration

Entries and changes are only made in the case of service and are exclusively performed by the hotline.

- Click on "Save" to save the MS SQL gateway.

6

Advanced configuration Optional

Type image path... *

bfc-registry.com/apc-koe/mts/iotserver/gateways/mssql:latest

Type username to access the image path...

Type password to access the image path...

Set message TTL (time to live) in milliseconds... *

7884000000

Set queue expiration in milliseconds... *

604800000

Set max queue size... *

15000000

8. The MS SQL gateway has been successfully created and is shown in the "Gateways (Export)" area of the overview.



6.4.8.16 Kafka gateway

A Kafka gateway enables connection to external Kafka brokers within a Kafka cluster. The Kafka gateway works as a producer and thus enables data export from the BFC gateway.

Possible use case:

- Data forwarding from an existing BFC data provider to an external Kafka cluster

Requirements

- You are familiar with the general Kafka technology.
- The Kafka cluster is accessible from the BFC gateway.
- The access data for the Kafka cluster are known.
- The Kafka cluster allows access via SASL/PLAIN, SASL/SCRAM256, or SASL/SCRAM512 if authentication is required.

Kafka topics

The Kafka gateway publishes the values under the following topic structure:

```
{base} . {srcID} . {srcAppType} . {msgType} . {setName}
```

- OR -

```
{base} . {srcID} . {srcAppType} . {msgType}
```

Examples:

- Data sets with the name `temperatures` from an OPC UA client with the `srcID` `machine1` are published under the following topic:
 - `bfc.machine1.opcuaclient.dataset.temperatures`
- Alarm lists from an OPC UA client with the `srcID` `machine1` are published under the following topic:
 - `bfc.machine1.opcuaclient.alarmlist`
- Alarm changes from an OPC UA client with the `srcID` `machine1` are published under the following topic:
 - `bfc.machine1.opcuaclient.alarmstatechange`

Supported message types:

- `dataset`
- `alarmlist`
- `alarmstatechange`
- `hfddata`

Kafka partitions

The `srcID` (see Kafka topics) is always used as `MessageKey`. Together with the balancer hash, the data of a source are written into the same partition.

6.4.8.17 Creating a Kafka gateway

Perform the following steps to create a Kafka gateway:

- Step 1: "Select gateway type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"

6.4 "Commissioning" area

- Step 4: "Define alarm configuration"
- Step 5: "Advanced configuration"

Requirement

The "Commissioning" area is open.

Parameters

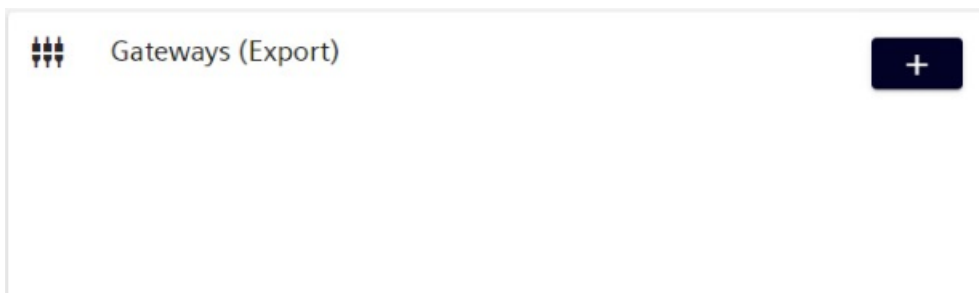
Parameters	Description
① Select gateway type	
Select target Store...	Selection of the gateway
② Define basic configuration	
Type gateway name...*	Gateway name Remark: The designation for "gateway name" is freely selectable.
Type gateway description...	Description of the gateway Remark: The designation for "gateway description" is freely selectable.
Type Kafka broker IPs...	Addresses of the Kafka broker as comma separated list Examples: • 192.168.6.66:29092,kafka-1:9092 • broker:29092
Select Kafka type...*	Selection of the login type: • SASL/PLAIN • SASL/SCRAM256 • SASL/SCRAM512 • NONE
Type Kafka broker username...	User name for the Kafka broker
Type Kafka broker password...	Password for the Kafka broker
Type Kafka topic prefix...*	Text that precedes the topic Default: bfc
Upload CA certificate	If the connection is established via TLS and the broker uses a certificate signed by an untrusted CA, you must provide this CA as a PEM block. Use the escape sequence ("\\n") to display line feeds and line breaks when entering the certificate. Remark: The server certificate provided by the broker must contain its IP/host name as Subject Alternative Name if this is not sufficient as Common Name.
③ Define dataset configuration / Optional	

Parameters	Description
Pick a client ID	Client ID of the client from which data is to be sent to the Kafka gateway. Remark: All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
Pick a dataset	Select the client data set that should be sent to the MySQL export gateway. Remark: All datasets of the selected client ID are displayed.
④ Define alarm configuration / Optional	
Pick a client ID	All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
⑤ Advanced configuration / Optional	
Select Kafka balancer...*	Optional selection of a balancer: • hash • leastbytes • roundrobin • crc32 • murmur2 Default: hash
Set maximum batch size...*	Maximum number of messages that are buffered before they are sent to a partition Default: 100
Set maximum batch bytes...*	Maximum size of a message in bytes before it is sent to a partition Default: 1048576
Set maximum attempts...*	Maximum number of attempts to deliver a message Default: 10
Select required acknowledgements...*	Number of acknowledgements of partition replicas required to receive a response to a message Default: -1 (all replicas are waited for) A value above 0 is required to specify how many replicas should acknowledge a message to be considered successful. 0= <i>fire-and-forget</i>
Set batch timeout in milliseconds...*	Time limit on the frequency with which incomplete message batches are sent to Kafka Default: 1000
Set read timeout in milliseconds...*	Read timeout Default: 10000
Set write timeout in milliseconds...*	Write timeout Default: 10000
Skip TLS verification...*	When set to "true", any TLS verification is skipped. Must not be used in productive operation.

*: Obligatory data

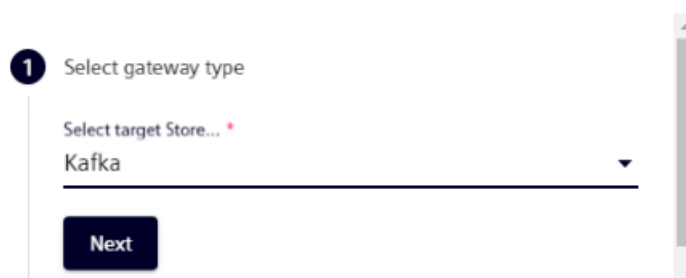
Procedure

1. In the "Gateway (Export)" area, click on "+" to add a new gateway.



2. Step 1: "Select gateway type"
 - Select the entry "Kafka" from the drop-down list.
 - Click on "Next".

Add new gateway



3. Step 2: "Define basic configuration"

- Populate the fields to perform the basic configuration.
- Click on "Next".

2 Define basic configuration

Type gateway name... *
to-kafka
Supported input: letters, numbers, -, _ (must start with letter or number) 8 / 50

Type gateway description...
0 / 100

Type Kafka broker IPs... e.g. '<IP-address>:9092,<IP2-address>:9092' *
kafka-1:9092,kafka-2:9092
25 / 100

Select Kafka type... *
SASL/PLAIN

Type Kafka broker username...
kafka-user
10 / 100

Type Kafka broker password...

Type Kafka topic prefix... *
bfc
3 / 100

In case of secured connection provide the certificates:
Upload CA certificate

4. Step 3: "Define dataset configuration" / Optional

With this step, you define which data is to be sent to the Kafka gateway.

- Select one data set, several or all data sets.

Note**Several entries**

- Click on "+" to add an additional entry.

- Click on "Next".

3 Define dataset configuration
Optional

Pick a client ID
machine 1

Pick a dataset
..... +

Next

6.4 "Commissioning" area

5. Step 4: "Define alarm configuration" / Optional
With this step, you define which alarms are to be sent to the Kafka export gateway.
 - Select the alarms.
 - Click on "Next".

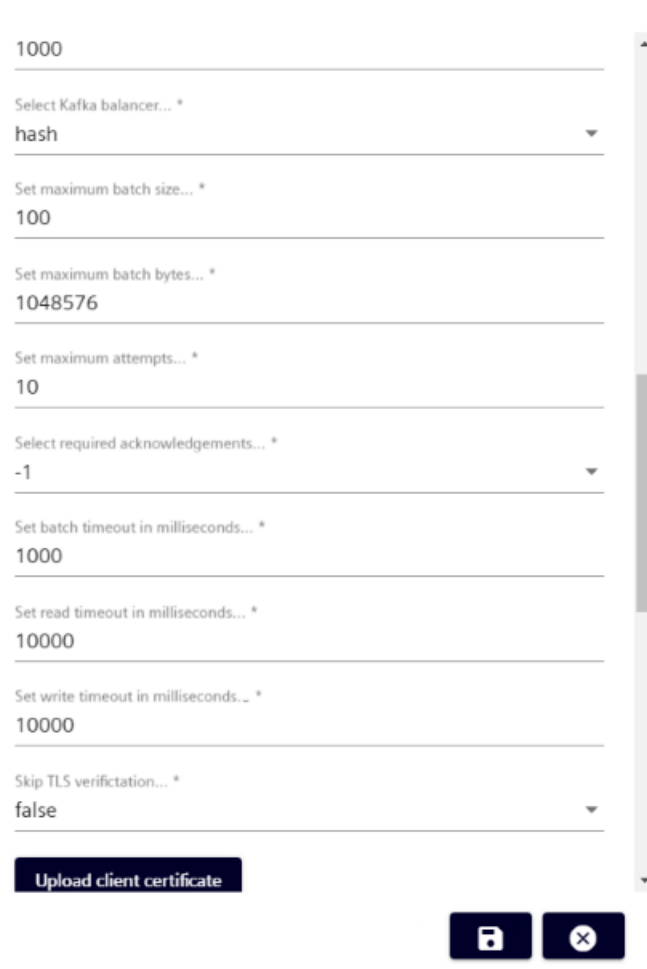


4 Define alarm configuration
Optional

Pick a client ID
machine1

Next

6. Step 5: "Advanced configuration" / Optional
With this step, you can optionally configure high-frequency data.
Further configurations are usually not necessary.
 - Click on "Save" to save the Kafka export gateway.



1000

Select Kafka balancer... *

hash

Set maximum batch size... *

100

Set maximum batch bytes... *

1048576

Set maximum attempts... *

10

Select required acknowledgements... *

-1

Set batch timeout in milliseconds... *

1000

Set read timeout in milliseconds... *

10000

Set write timeout in milliseconds... *

10000

Skip TLS verification... *

false

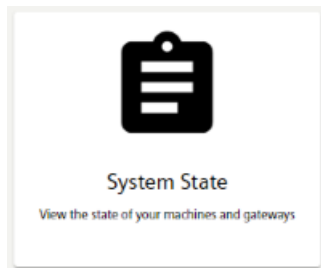
Upload client certificate

Save Cancel

6.5 "System State" area

6.5.1 Displaying information

In area "System State" you obtain an overview of the overall status of all clients, middlewares and gateways. Start the area from the "System State" tile on the home page of the BFC Gateway.



The following system statuses are displayed using these system messages:

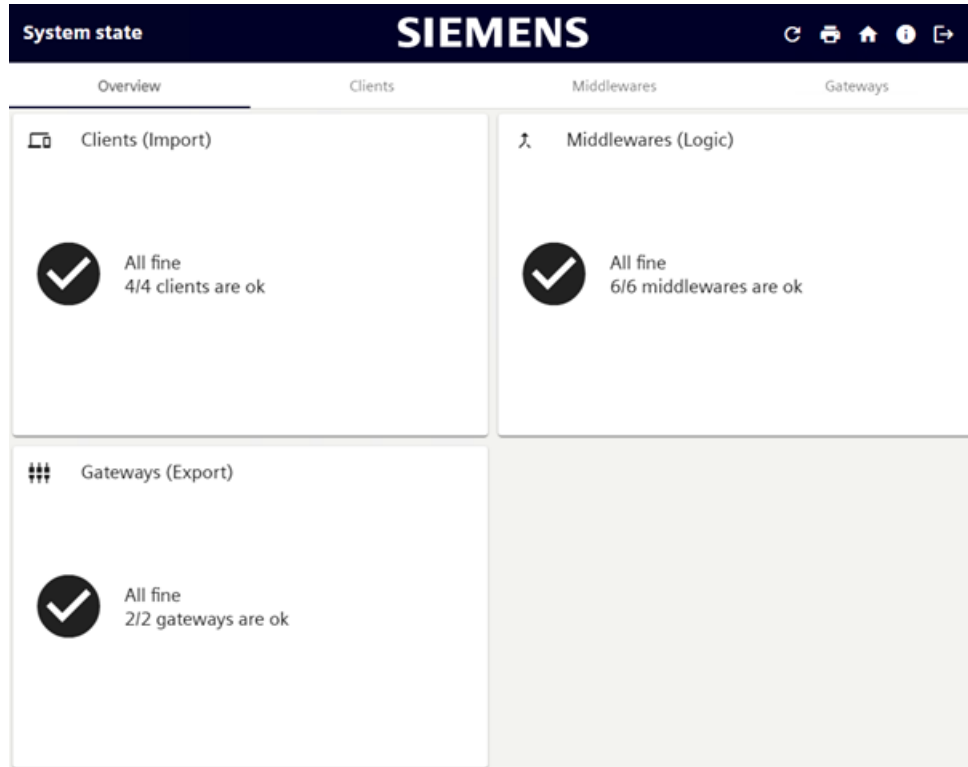
- "All fine": All clients, middleware and gateways are functioning without any error.
- "Warning": At least one client, middleware or gateway has a warning.
- "Problem detected": At least one client, middleware or gateway has a problem.

Requirement

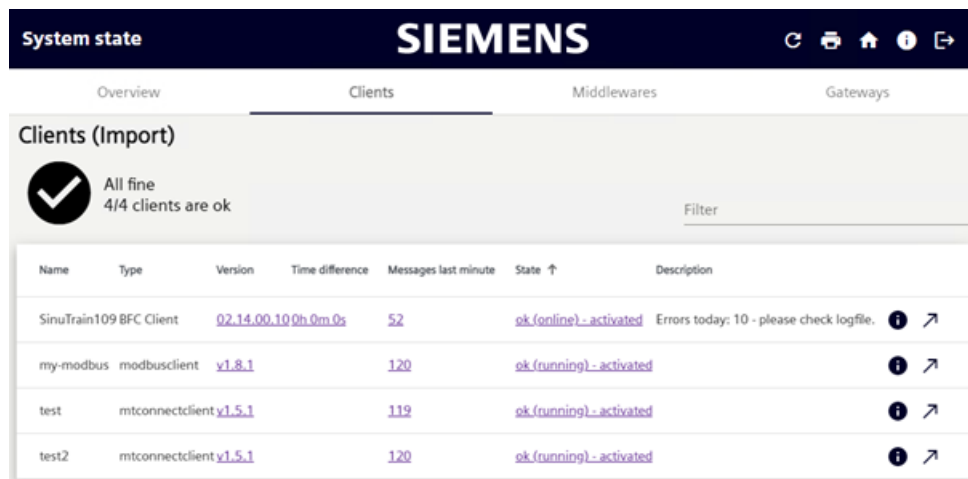
Area "System State" is open.

Procedure

1. Proceed as follows to obtain a tabular overview of clients, middlewares and gateways:
 - Click on "Clients", "Middlewares" or "Gateways" below the display bar.
 - OR -
 - Click on areas "Clients (Import)", "Middlewares (Logic)" or "Gateways (Export)".



2. The overview window opens.
In this example, the table lists details of individual clients.



3. Click on area "Overview" to return to the "System State" page.

Note**Connection interrupted**

If the connection to a target system is interrupted, e.g. due to a network failure, the data is buffered in the BFC Gateway. The buffer size is limited to 5 GB per configured gateway.

6.5.2 Area "Clients (Import)"

Detailed information about the configured clients is shown in the "Clients (Import)" area. All of the configured clients are listed in a table.

Requirement

Area "System State" > "Clients (Import)" is open.

Parameters

The following table provides an overview of all of the parameters.

Parameters	Description
Name	Client name
Type	Type of client, e.g. • Modbus client • OPC UA client • BFC client • ...
Version	Client version Remark: If a version newer than the installed version exists, then a symbol is displayed together with a check box to update the client.
Last minute messages	Number of messages that the client has sent within the last minute.
State	Client status • ok (online/offline/running/stopped): No error • not ok: Client has an error
Description	More information about the client

Procedure

- In the "Clients (Import) " area, click on the icons in the table in order to obtain detailed information about the clients.

Status: All fine → no error

"Overview" area

System state

SIEMENS

Overview

Clients

Middlewares

Gateways

Clients (Import)

✓

All fine
4/4 clients are ok

Middlewares (Logic)

✓

All fine
6/6 middlewares are ok

Gateways (Export)

✓

All fine
2/2 gateways are ok

Area "Clients (Import)"

System state

SIEMENS

Overview

Clients

Middlewares

Gateways

Clients (Import)

✓

All fine
4/4 clients are ok

Filter

Name	Type	Version	Time difference	Messages last minute	State ↑	Description
SinuTrain109 BFC Client		02.14.00.10 0h 0m 0s	52	ok (online) - activated	Errors today: 10 - please check logfile.	i ↗
my-modbus	modbusclient	v1.8.1	120	ok (running) - activated		i ↗
test	mtconnectclient	v1.5.1	119	ok (running) - activated		i ↗
test2	mtconnectclient	v1.5.1	120	ok (running) - activated		i ↗

Status: Problem detected → error

"Overview" area

System state

SIEMENS

Overview

Clients

Middlewares

Gateways

Clients (Import)

✖

Problems detected
1/4 clients are not ok

Middlewares (Logic)

✓

All fine
6/6 middlewares are ok

Gateways (Export)

✓

All fine
2/2 gateways are ok

Area "Clients (Import)"

System state

SIEMENS

Overview

Clients

Middlewares

Gateways

Clients (Import)

✖

Problems detected
1/4 clients are not ok

Filter

Name	Type	Version	Time difference	Messages last minute	State ↑	Description
my-modbus	modbusclient	v1.8.1		120	✖ not ok (unknown)	State: expected state: running ⓘ ↗
SinuTrain109 BFC Client		02.14.00.10 0h 0m 0s		60	ok (online) - activated	Errors today: 10 - please check logfile. ⓘ ↗
test	mtconnectclient	v1.5.1		119	ok (running) - activated	ⓘ ↗
test2	mtconnectclient	v1.5.1		120	ok (running) - activated	ⓘ ↗

Status: Alarm

"Overview" area

System state

SIEMENS

Overview

Clients

Middleware

Gateway

Clients (Import)

1/4 clients have a time difference > 10min
1/4 clients have version issues
4/4 clients are ok

Middlewares (Logic)

All fine
6/6 middlewares are ok

Gateways (Export)

All fine
2/2 gateways are ok

Area "Clients (Import)"

System state

SIEMENS

Overview

Clients

Middleware

Gateway

Clients (Import)

1/4 clients have version issues
4/4 clients are ok

Filter

Name	Type	Version	Time difference	Messages last minute	State ↑	Description	
SinuTrain109BFC Client		02.14.01.01	0h 39m 33s		ok (online) - activated		
my-modbus	modbusclient	v1.8.1	0		ok (running) - activated		
test	mtconnectclient	v1.5.0	118		ok (running) - activated	Version: expected version: v1.5.1	
test2	mtconnectclient	v1.5.1	116		ok (running) - activated		

1. In area "Clients", click on the time difference to obtain additional information.
2. Window "Time differences of BFC Client <Name of client>" opens.
The overview lists the time of the BFC Gateway with all available times of the SINUMERIK.

Time differences of BFC Client Sinutrain109

Name	Time	Time difference
BFC Gateway	2022-06-28T20:52:39.970897299Z	-
hmi	2022-06-28T23:25:00.363000+02:00	0h 32m 20s
plc	2022-06-28T22:13:07.304+02:00	0h 39m 33s



3. In area "Clients", click on the "Update" symbol to update the client.

Note

Updating several clients

To update several clients, activate the appropriate check box and click on the "Update" symbol in the header.

To update BFC clients, first upload a ZIP file using the arrow symbol.

Note

Time difference between the HMI and PLC

If a time difference between the HMI and PLC is displayed for a BFC client, then this can be ignored when operating the BFC. This only serves as note. For all the data to be captured, including alarms, the BFC client uses the HMI time stamp.

6.5.3 "Middlewares (Logic)" area

Detailed information about the configured middlewares is shown in the "Middlewares (Logic)" area. All configured middlewares are listed in a table.

Requirement

Area "System State" > "Middlewares (Logic)" is open.

Parameter

The following table provides an overview of all of the parameters.

Parameter	Description
Name	Name of the middleware
Type	Type of middleware, e.g. • middleware

6.5 "System State" area

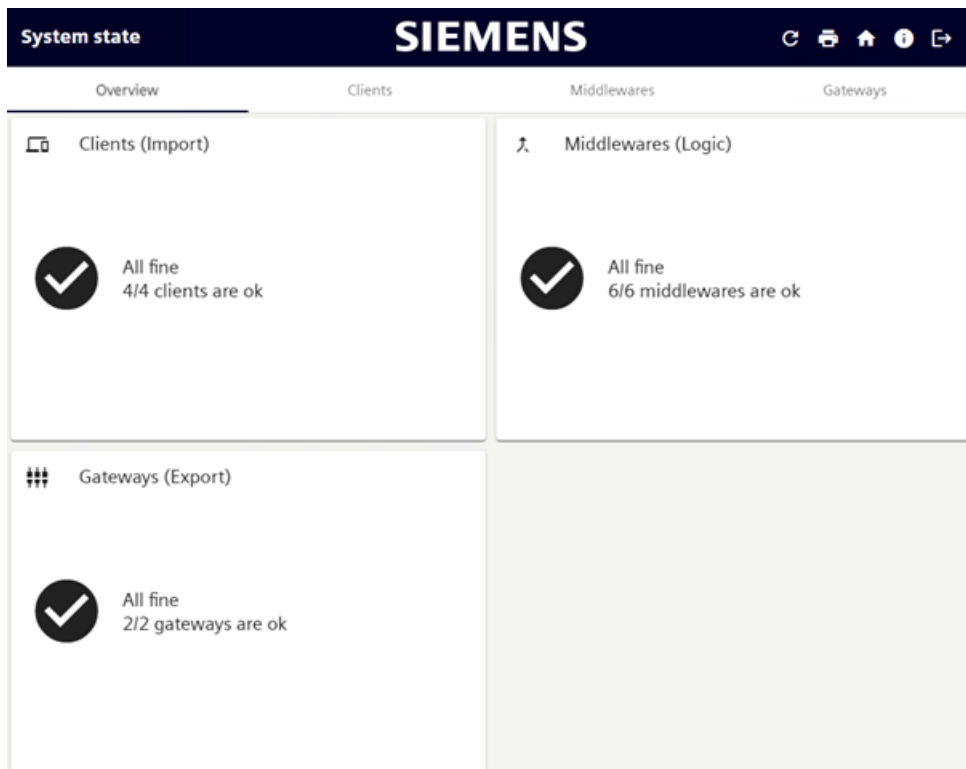
Parameter	Description
Version	Version of the middleware Remark: If a version newer than the installed version exists, then a symbol is displayed together with a check box to update the middleware.
Last minute messages	Number of messages that the middleware has sent within the last minute.
State	Status of the middleware - ok (running/stopped): No error - not ok: Middleware has an error
Description	More information about the middleware

Procedure

- In area "Middlewares (Logic)", click on the symbols in the table to obtain detailed information about the middlewares.

Status: All fine → no error

"Overview" area



"Middlewares (Logic)" area

System state

SIEMENS

Overview

Clients

Middlewares

Gateways

Middlewares (Logic)

✓

All fine
6/6 middlewares are ok

Filter

Name	Type	Version	Messages last minute	State ↑	Description
IBaseApp	middleware	1.1.2	ok (running)		Version: unknown image, could not check for newer version: bfc-registry.com/apc-koelmts/iotserver/applications/ibase/ibase-service
OptimizationCheckApp	middleware	1.1.1	ok (running)		Version: unknown image, could not check for newer version: bfc-registry.com/apc-koelmts/iotserver/applications/optimizationcheck/optimization-check-service
Scriptlogic	middleware	0.7.2	ok (running)		
State Service	middleware	1.1.7	ok (running)		
my-logic	middleware	1.7.51	ok (running)- activated		
SSA-Service	middleware	1.4.3	ok (stopped)		

Status: Problem detected -> error

"Overview" area

System state

SIEMENS

Overview

Clients

Middlewares

Gateways

Clients (Import)

✓

All fine
4/4 clients are ok

Gateways (Export)

✓

All fine
2/2 gateways are ok

Middlewares (Logic)

✗

Problems detected
1/6 middlewares are not ok

"Middlewares (Logic)" area

System state

SIEMENS

Overview

Clients

Middlewares

Gateways

Middlewares (Logic)

Problems detected
1/6 middlewares are not ok

Filter

Name	Type	Version	Messages last minute	State ↑	Description
my-logic	middleware	1.7.51		not ok (running) - activated	State: expected state: stopped
IBaseApp	middleware	1.1.2		ok (running)	Version: unknown image, could not check for newer version: bfc-registry.com/apc-koe/mts/iotserver/applications/ibase/ibase-service
OptimizationCheckApp	middleware	1.1.1		ok (running)	Version: unknown image, could not check for newer version: bfc-registry.com/apc-koe/mts/iotserver/applications/optimizationcheck/optimization-check-service
Scriptlogic	middleware	0.7.2		ok (running)	
State Service	middleware	1.1.7		ok (running)	
SSA-Service	middleware	1.4.3		ok (stopped)	

Status: Alarm

"Overview" area

System state

SIEMENS

Overview

Clients

Middlewares

Gateways

Clients (Import)

All fine
4/4 clients are ok

Gateways (Export)

All fine
2/2 gateways are ok

Middlewares (Logic)

1/6 middlewares have version issues
6/6 middlewares are ok

"Middlewares (Logic)" area

System state **SIEMENS**

Overview Clients **Middlewares** Gateways

Middlewares (Logic)

1/9 middlewares have version issues
9/9 middlewares are ok

Filter

Name	Type	Version	Messages last minute	State	Description
IBaseApp	middleware	y1.1.2		ok (running)	
OptimizationCheckApp	middleware	y1.1.1		ok (running)	
Scriptlogic	middleware	y0.7.2		ok (running)	
State Service	middleware	y1.1.7		ok (running)	
my-logic	middleware	▲ v1.7.0 1		ok (running) - activated	Version: expected version: v1.7.5
SSA-Service	middleware	y1.4.3		ok (stopped)	

In area "Middlewares", click on the "Update" symbol to update the middleware.

Note**Updating several middlewares**

To update several middlewares, activate the appropriate check box and click on the "Update" symbol in the header.

6.5.4 "Gateways (Export)" area

Detailed information about the configured gateways is shown in the "Gateways (Export) " area. All of the configured gateways are listed in a table.

Requirement

Area "System State" > "Gateways (Export)" is open.

Parameter

The following table provides an overview of all of the information:

Parameter	Description
Name	Gateway name
Type	Type of gateway, e.g. - MindSphere - OPC UA ...

6.5 "System State" area

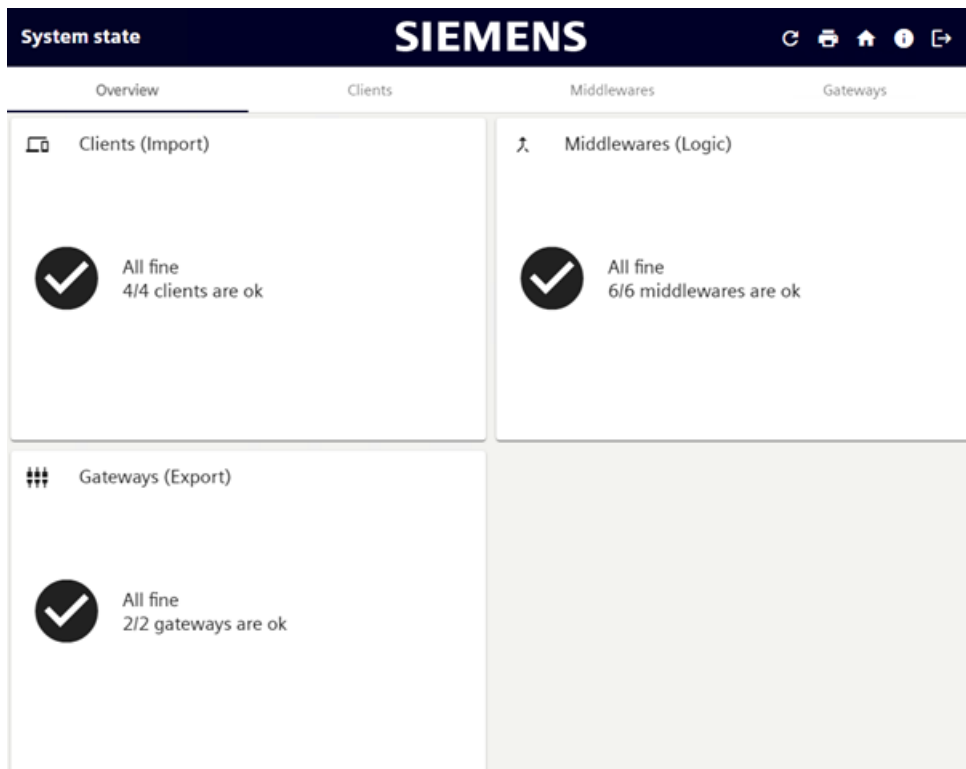
Parameter	Description
Version	Gateway version Remark: If a version newer than the installed version exists, then a symbol is displayed together with a check box to update the gateway.
Last minute messages	Number of messages that the gateway has sent within the last minute.
State	Gateway status - ok (running/stopped): No error - not ok: Gateway has an error
Description	More information about the gateway

Procedure

- In the "Gateway (Export) " area, click on the icons in the table in order to obtain detailed information about the gateways.

Status: All fine → no error

"Overview" area



"Gateways (Export)" area

System state

SIEMENS

Overview

Clients

Middlewares

Gateways

Gateways (Export)

All fine
2/2 gateways are ok

Filter

Name	Type	Version	Messages last minute	State ↑
OPC UA	gateway	v1.0.11		ok (running) - activated
to-influx	internalinfluxdb	v1.6.0	1	ok (running) - activated

Status: Problem detected → error

"Overview" area

System state

SIEMENS

Overview

Clients

Middlewares

Gateways

Clients (Import)

All fine
4/4 clients are ok

Middlewares (Logic)

All fine
6/6 middlewares are ok

Gateways (Export)

Problems detected
1/2 gateways are not ok

"Gateways (Export)" area

System state

SIEMENS

Overview

Clients

Middlewares

Gateways

Gateways (Export)

Problems detected

1/2 gateways are not ok

Filter

Name	Type	Version	Messages last minute	State ↑	Description
to-influx	internalinfluxdb	y1.6.0		not ok (running) - activated	State: expected state: stopped
OPC UA	gateway	y1.0.11		ok (running) - activated	

Status: Alarm

"Overview" area

System state

SIEMENS

Overview

Clients

Middlewares

Gateways

Clients (Import)

All fine

4/4 clients are ok

Middlewares (Logic)

All fine

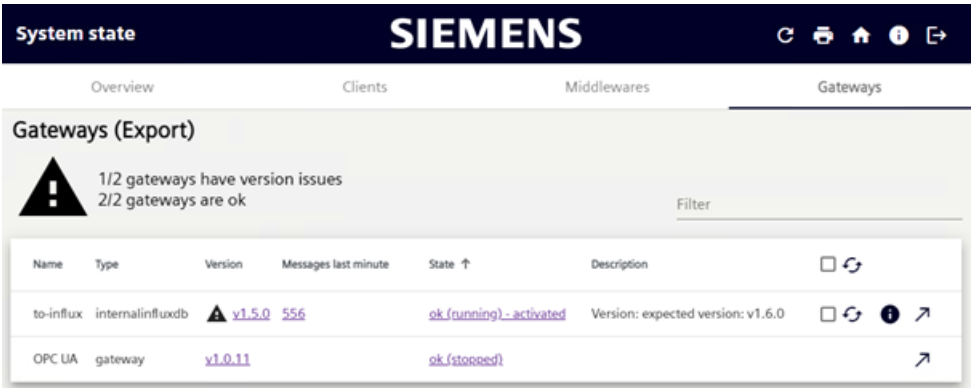
6/6 middlewares are ok

Gateways (Export)

1/2 gateways have version issues

2/2 gateways are ok

"Gateways (Export)" area



In area "Gateways", click on the "Update" symbol to update the gateway.

Note

Updating several gateways

To update several gateways, activate the appropriate check box and click on the "Update" symbol in the header.

6.5.5 System State Report

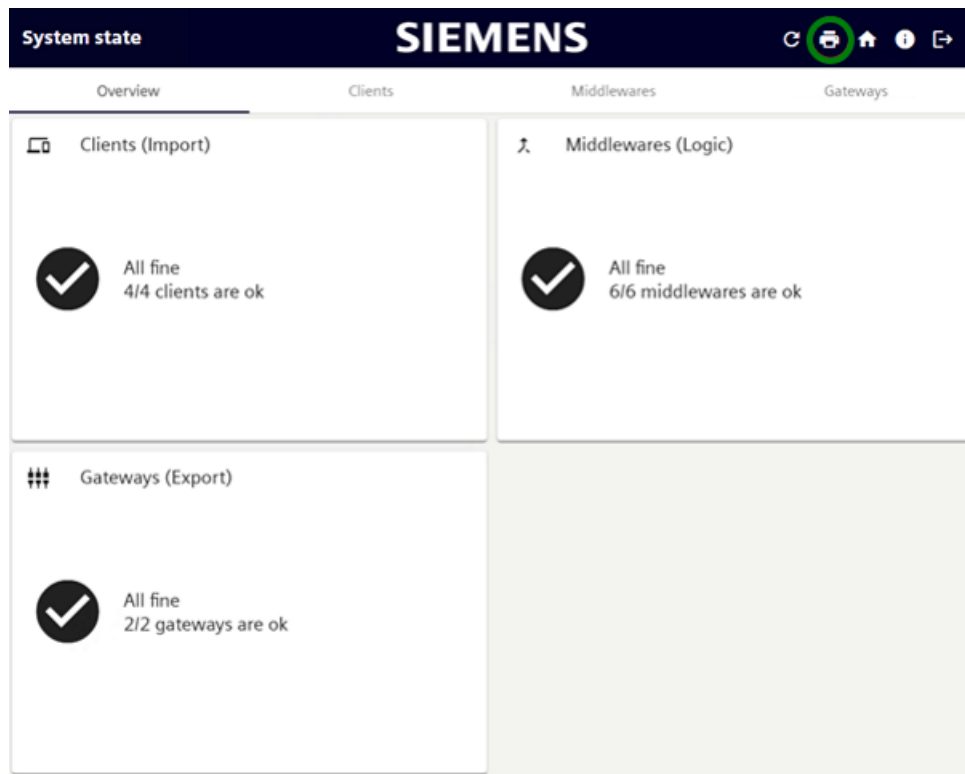
If you contact the hotline, it may also be necessary to send the "System State Report". This includes information about the configured Clients, Middlewares and Gateways.

Requirement

Area "System State" is open.

Procedure

1. In area "System state", click on the "Printer" symbol in the navigation bar. You can save "System State Report" as PDF.



6.6 "Usermanagement" area

You can access the user and rights management via the ConfigUI of the BFC. Start the management via the "Usermanagement" tile on the start page.



User management is split up into three areas. An upper area with the name "Users", and below it an area with the name "External Provider" and at the bottom, the area "Technical Users". Each of these areas comprises a filter function, a table and navigation elements below the table.

Note

Area "Technical Users" is used for information purposes only. New users cannot be created and user settings cannot be changed.

UserUI



Users



Filter

Name	Email	Amount of skills	
admin	admin@mail.com	2	  

Items per page: 10 1 - 1 of 1 |< < > >|

External Provider



Filter

Name	Created	Discovery URL

Items per page: 10 0 of 0 |< < > >|

Technical Users ?

Filter

Name	Displayname	Type	Amount of skills	
c4d575f2-6bb2-44d6-90d5-8151afda969d			2	
d6748c56-7134-4228-b021-13ba348a970a	do-nothing	scriptlogic:v1.5.0	1	
technicaluser			2	

Items per page: 10 1 - 5 of 5 |< < > >|

6.6.1 Creating a new user

Procedure

1. Click on the "Plus" symbol ("Add User").
2. An input window opens.
 - You must enter a user name and a password.
 - Optionally, you can also enter an email address, select the skills (user rights) and select whether the user that you created should be an administrator ("is admin").
 - Click on "Save" to save the entries.

The screenshot shows a dialog box titled "Add user". It contains several input fields: "Type user name..." with a red asterisk, "Type user password..." with a red asterisk, another "Type user password..." with a red asterisk, "Type user email...", and "Type new skill..." with a plus icon to its right. Below these fields is a checkbox labeled "is admin". At the bottom right, there are two buttons: a grey "Save" button with a floppy disk icon and a dark blue "Close" button with a white 'X' icon. A vertical scrollbar is visible on the right side of the dialog box.

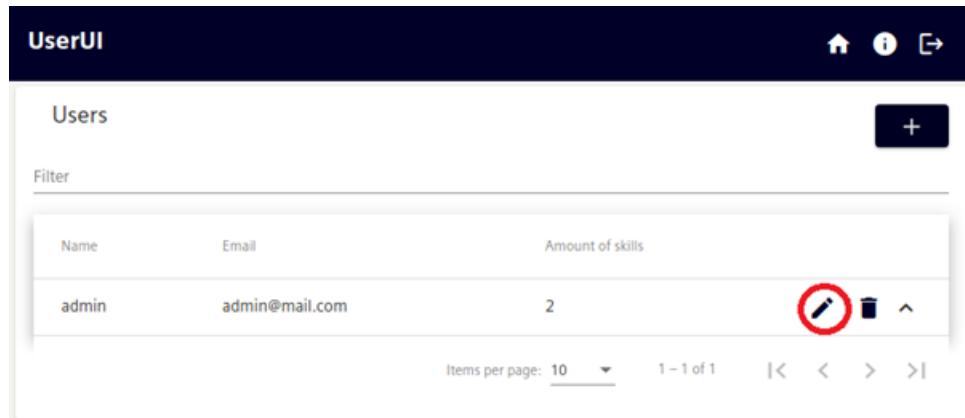
Note

Only users that have administrator rights can log into the ConfigUI.

6.6.2 Adapting an existing user

Procedure

1. Click on the "pen" icon next to the user.

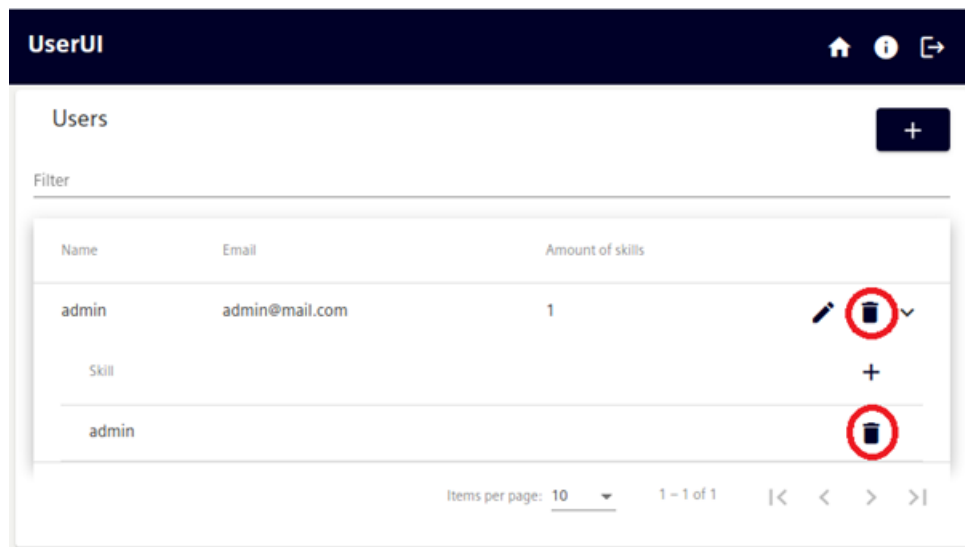


2. The same input window as for creating a user opens. Change the desired entries and click the "Save" button to save the changes.
3. Click on the respective user to assign skills to a user. You can then add a new skill using the "Plus" symbol ("Add Skill").

6.6.3 Deleting an existing user

Procedure

1. Click the "trash can" icon ("Delete User" / "Delete tag") next to the user or a skill to remove the entire user or one of the assigned skills.



6.6.4 Using an external IAM system

6.6.4.1 Requirements

To connect an external IAM "Identity and Access Management" to the user management of the BFC gateway, the following requirements must be met:

- A client that meets the following requirements has been created on the IAM:
 - Support for OpenID Connect 1.0 Authorization Code Flow
 - Support for OpenID Connect 1.0 Discovery (availability of the OpenID Provider Configuration Information)
 - Client ID and Client Secret
- Redirect URLs are stored in the client configuration.
In this way, you are redirected to the BFC gateway after logging onto the IAM. In the opposite direction, you are redirected to the logon window of the BFC gateway after logging off.
- The following URLs are stored for the logon and logoff processes:
 - Logon: URL of the return endpoint: `https://<BFC Gateway Hostname>:9877/users/api/openid/return`
 - Logoff: URL of the logon UI: `https://<BFC Gateway Hostname>:9877/`
- Users configured in the IAM are equipped with the attribute "skills" and the string value "admin".
- A Token Claim Mapping was created for the created client.
The content of the "skills" attribute is stored under the token claim name "skills" as multivalued JSON in the ID and access token.

6.6.4.2 Adjusting the configuration in the user management of the BFC gateway

To connect an external IAM to the BFC gateway, you must adjust the configuration in the user management of the BFC gateway.

Parameters

Parameters	Description
Type provider name...*	Name of the provider The name can be freely selected and is only used for display in the logon window
Type registered client ID...*	The registered client ID is taken over from the configuration in the IAM
Type registered client secret...*	The registered client secret is taken over from the configuration in the IAM
Type provider discoveryURL...*	The provider discoveryURL is obtained from the IAM
Type new scope	Further scopes are optional

Procedure

1. Click the "Usermanagement" tile on the landing page.
You are in the user management of the BFC gateway.
2. Click the "+" icon in the "External Provider" area to add a new external IAM.

3. Fill out the fields.
The fields marked "*" must be filled out.

Add provider

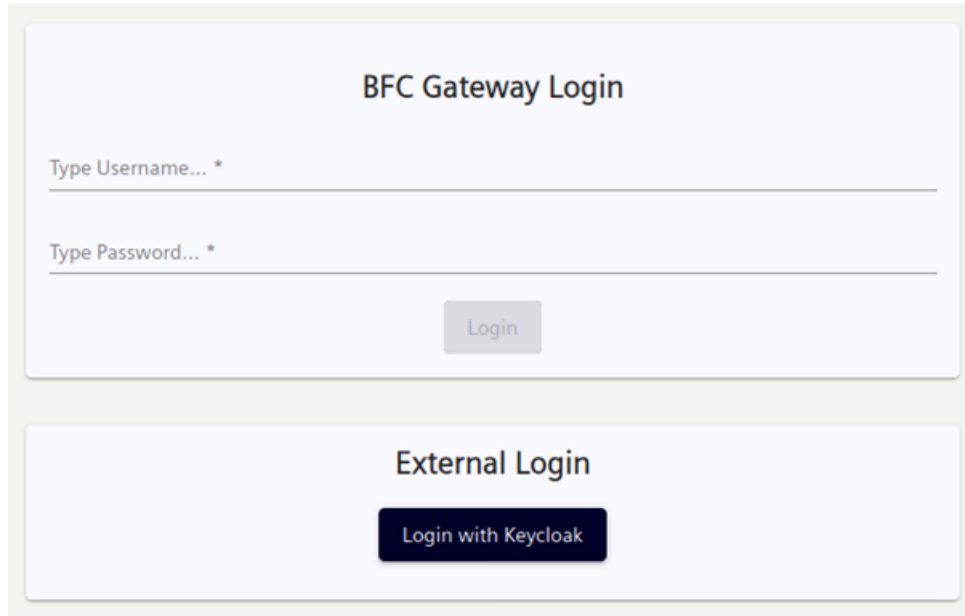
6.6.4.3 Logging on over an external IAM

Requirement

- You have configured the external IAM.
- The login window of the BFC gateway is open.

Procedure

1. Click the "Login with < ... >" button in the "External Login" area.
The logon window of the external IAM opens.

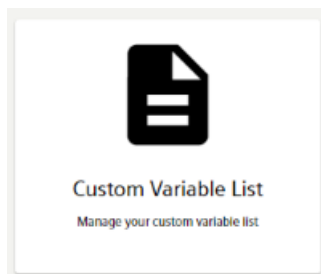


The image shows two login forms. The top form is titled "BFC Gateway Login" and contains two input fields: "Type Username... *" and "Type Password... *". Below these fields is a "Login" button. The bottom form is titled "External Login" and contains a single button labeled "Login with Keycloak".

2. Log on to the external IAM.
After successful logon, the landing page of the BFC gateway opens.

6.7 "Custom Variable List" area

You can add and manage your own variable lists via area "Custom Variable List" of the BFC. Start the area from the "Custom Variable List" tile on the home page of the BFC Gateway.



You can upload new browsing files using the "Upload" button.



The image shows a screenshot of the "Variable Browser UI" interface. It has a dark blue header with the "SIEMENS" logo. Below the header is a search bar and an "Upload" button. The main content area displays a table with columns for "Name", "Client type", and "Type".

Name	Client type	Type
brs-test	brs-test	json
default-brs-test	default-brs-test	json
STClient	STClient	json
STClientDefault	STClient	json

6.7 "Custom Variable List" area

Select the corresponding file in Explorer using the "Upload file" button. Then select the client type for which this variable list is applicable and assign a name.

Click on "Save" to confirm your entry.

Upload custom browsing file

Upload file

Select Client... *

Fairac

Type Filename... *

default-fairac-list

Saved variable lists can be selected when setting up the client under Point "Define dataset configuration".

More information can be found in Chapter Defining a dataset configuration (Page 198).

The table shows an overview of all of the saved files. Using the "Search" bar you can browse in the file names. The files can be downloaded, exchanged or deleted using the buttons on the right-hand side.

Note

The old file is permanently deleted as soon as you have confirmed delete or replace.

Using the BFC client

7.1 Installing a BFC client

7.1.1 Requirement

Only install the BFC client if you are very familiar with handling SINUMERIK 840D/840D sl/828D. This applies especially for the configuration of HMI-Advanced, SINUMERIK Operate Windows and SINUMERIK Operate Linux.

Note**Establishing the network configuration while installing the BFC client**

The network connection to the BFC gateway is checked while installing the BFC client on HMI-Advanced and SINUMERIK Operate under Windows (PCU). An appropriate message is displayed if a connection does not exist.

- Establish the network connection between the machine and the BFC gateway while installing the BFC client.
-

7.1.2 Installing the BFC client on SINUMERIK 840D sl / 828D

7.1.2.1 Preparing the installation

Source the latest BFC client software through PridaNet.

- BFC client for SINUMERIK

Observe any service packs or hotfix versions that have been made available.

Procedure

1. Open PridaNet.
2. Download the BFC client software and enter the information regarding the final destination. You obtain a zip archive with the name "BFC-Client-xx.xx.xx.xx.zip". The zip archive contains the following files:
 - BFC-Client-xx.xx.xx.xx.zip
You require this file to install the BFC client on the various target platforms for the first time.
 - readme.txt
 - Installationsanleitung (German)
 - Installation Instructions (English)
 - Liesmich (German)
 - Readme (English)
 - OSS_Readme (Open Source Software disclosure document)
3. Unzip the zip archive to install the BFC client for the first time on the required target platform. The zip archive contains the following files:
 - BFC-Client-Advanced-xx.xx.xx.xx.exe
This file is the Windows program for installation on PCU 50 with HMI-Advanced.
 - BFC-Client-Operate-Windows-xx.xx.xx.xx.exe
This file is the Windows program for installation on PCU 50 or IPC with SINUMERIK Operate (PCU/IPC).
 - BFC-Client-Operate-Linux-xx.xx.xx.xx
This installation file is the Linux application for installing on SINUMERIK Operate (NCU/PPU).
 - checksum.txt
This file contains the checksums (SHA1, SHA2-256, SHA2-512) of all files in order to identify any data manipulation after the software was delivered.

You can find additional information about installing the BFC client on the various target platforms in the following chapters.

7.1.2.2 Synchronizing date and time

Set the time and date on the target system.

1. Check the date and time of the target system for the correct settings. Potential consequences of an incorrect setting:
 - Difficulties relating to encrypted communication
 - Incorrect time stamp
 - Inconsistent data transfer
2. Check the settings of the time zone.
Correctly enter the settings for your location and ensure that they match the settings of the BFC gateway.

Note

Ensure that all of the connected devices operate with the same time settings.

7.1.3 Installing the BFC client on HMI-Advanced

For SINUMERIK control systems that use a PCU 50 with the HMI-Advanced operating software when installing the BFC client, select this installation file:

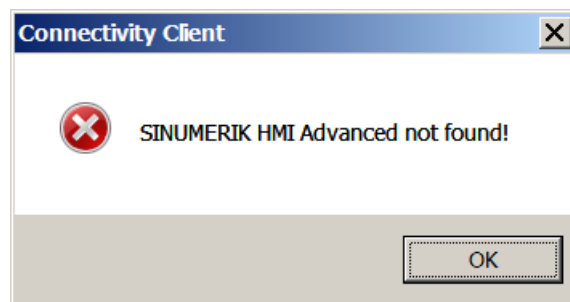
- BFC-Client-Advanced-xx.xx.xx.xx.exe

Note**Installing the BFC client**

The BFC client is installed in the service mode of the control system, and it is identical on the Windows 95, Windows NT, Windows XP and Windows 10 operating systems.

Procedure

1. Start HMI Advanced in the control system service mode.
2. Check the HMI-Advanced environment.
 - You will receive an appropriate message if the HMI-Advanced environment is not identified.
The installation will be canceled.



- Installation continues if the HMI-Advanced environment is identified.

7.1 Installing a BFC client

3. The input window opens.
Enter the following data in the input window:

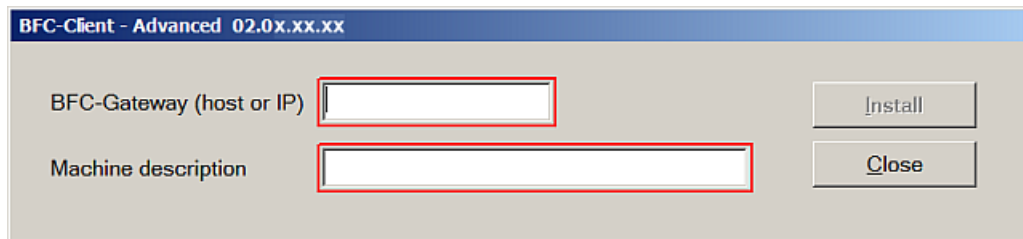
- BFC gateway: IP address or host name

Note**Selecting the entry**

When installing the BFC gateway, a decision is made as to whether the host name or the IP address is used for the TLS certificate.

- Depending on this decision, enter the host name or the IP address used in the certificate in the input window.

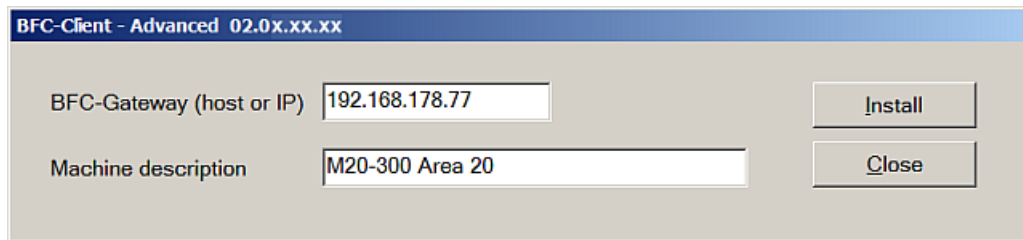
- Description of the machine
We recommend, in agreement with the operating company, a machine designation that is unique throughout the system.



4. The "Install" button is activated once the "BFC-Gateway" input field has been populated.
Carefully ensure the following:

- The data is correct.
- The BFC gateway can be accessed using this address.

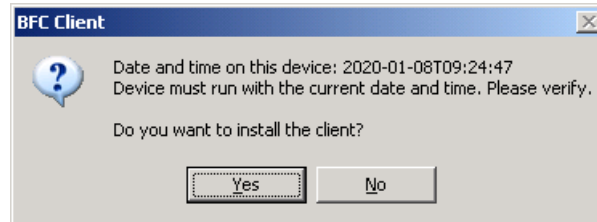
Click on "Install" if the data is correct.



5. You obtain a message whether the BFC client should be installed.

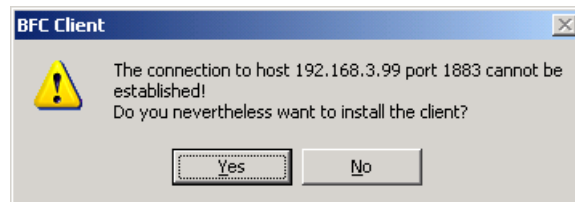
- Check the date and time at the control system.
The current date and time are shown in the message.
- Start the installation if the date and time are correct.
Click on "Yes".

Installation on the control system starts. Generally, the procedure lasts just a few seconds.

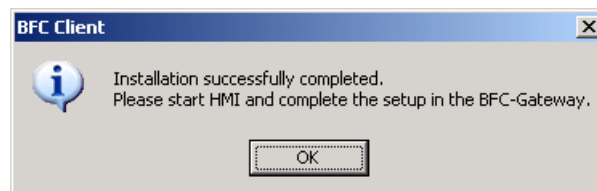


6. When installation starts, the network connection to the BFC Gateway is checked.

- You receive a message if the network connection was not able to be established.
In spite of a faulty connection to the BFC Gateway, installation can continue. Click on "Yes".
Additional information about network connection diagnostics is provided in Chapter: BFC client diagnostics (Page 351).
- You do not receive any message if the network connection to the BFC Gateway was able to be successfully established.



7. You receive a message if the installation was successfully performed.
Acknowledge the message with "OK" to complete the installation.



8. Restart HMI Advanced on the control system.

Note

Restart

The BFC client only becomes active after the control system is restarted.

9. After restarting HMI-Advanced, check whether the BFC client has successfully established a connection to the BFC gateway:
 - In the trace file, verify that an encrypted MQTT connection was successfully established via port 8883.
 - Back up the trace file to document that commissioning was successful.

Further information can be found in Chapter: Analyzing the BFC client (Page 351).

Additional information about errors that occurred during or after the installation, the causes and how they can be resolved is provided in Chapter: Trace (Page 356).

7.1.4 Installing the BFC client on SINUMERIK Operate under Windows (PCU)

For all SINUMERIK control systems that use SINUMERIK Operate as operating system running on a PCU 50 under Windows, when installing the BFC client, select this installation file:

- BFC-Client-Operate-Windows-xx.xx.xx.xx.exe

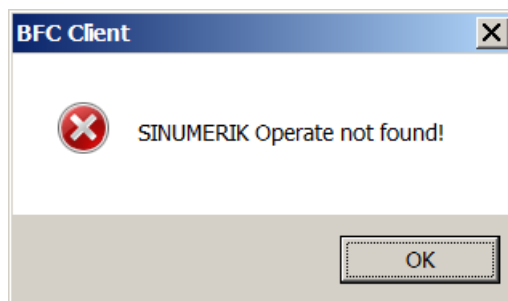
Note

Installing the BFC client

The BFC client is installed in the service mode of the control system, and it is identical on the Windows XP, Windows 7 and Windows 10 operating systems.

Procedure

1. Start SINUMERIK Operate in the control system service mode.
2. Check the SINUMERIK Operate environment
 - You receive a message if the SINUMERIK Operate environment is not identified. The installation will be canceled.



- Installation continues if the SINUMERIK Operate environment is identified.

3. An input window opens.
Enter the following data in the input window:
 - BFC gateway: IP address or host name

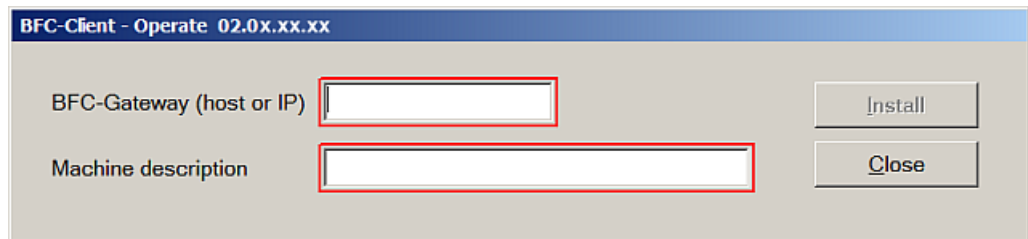
Note

Selecting the entry

When installing the BFC gateway, a decision is made as to whether the host name or the IP address is used for the TLS certificate.

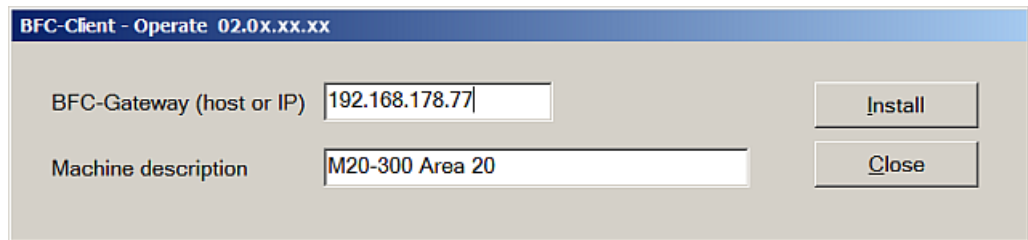
- Depending on this decision, enter the host name or the IP address used in the certificate in the input window.

- Description of the machine
We recommend, in agreement with the operating company, a machine designation that is unique throughout the system.



4. The "Install" button is activated once the "BFC-Gateway" input field has been populated.
Check the following:
 - The correctness of the data
 - The BFC gateway can be accessed using this address

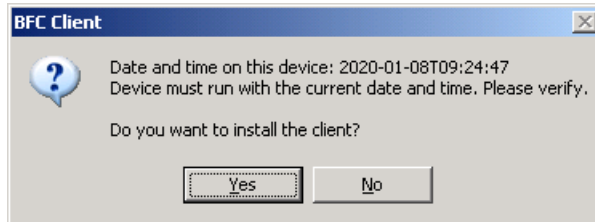
Click on "Install" if the check indicated everything is OK.



5. You obtain a message whether the BFC client should be installed.

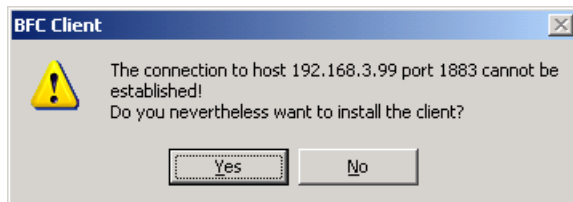
- Check the date and time at the control system.
The current date and time are shown in the message.
- Start the installation if the date and time are correct.
Click on "Yes".

Installation on the control system is started. Generally, the procedure lasts just a few seconds.

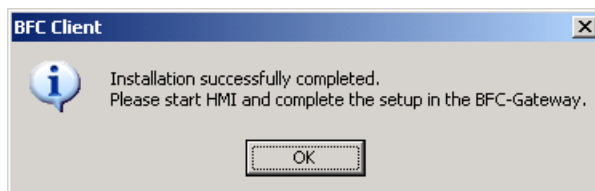


6. When installation starts, the network connection to the BFC Gateway is checked.

- You receive a message if the network connection was not able to be established.
In spite of a faulty connection to the BFC Gateway, installation can continue. Click on "Yes".
Additional information about network connection diagnostics is provided in Chapter: BFC client diagnostics (Page 351).
- You do not receive any message if the network connection to the BFC Gateway was able to be successfully established.



7. You receive a message that the installation was successfully performed.
Acknowledge the message with "OK" to complete the installation.



8. Restart SINUMERIK Operate on the control system.

Note

Restart

The BFC client only becomes active after the control system is restarted.

9. After SINUMERIK Operate has restarted, check whether the BFC client has successfully established a connection to the BFC gateway.
 - In the trace file, verify that an encrypted MQTT connection was successfully established via port 8883.
 - Back up the trace file to document that commissioning was successful.

Further information can be found in Chapter: Analyzing the BFC client (Page 351).

Additional information about errors that occurred during or after the installation, the causes and how they can be resolved is provided in Chapter: Analyzing the trace file (Page 357) , Paragraph: Connection status

7.1.5 Installing the BFC client on SINUMERIK Operate under Linux

For all SINUMERIK control systems that use SINUMERIK Operate as operating system running on an NCU/PPU under Linux, when installing the BFC client, select this installation file:

- BFC-Client-Operate-Linux-xx.xx.xx.xx

Requirement

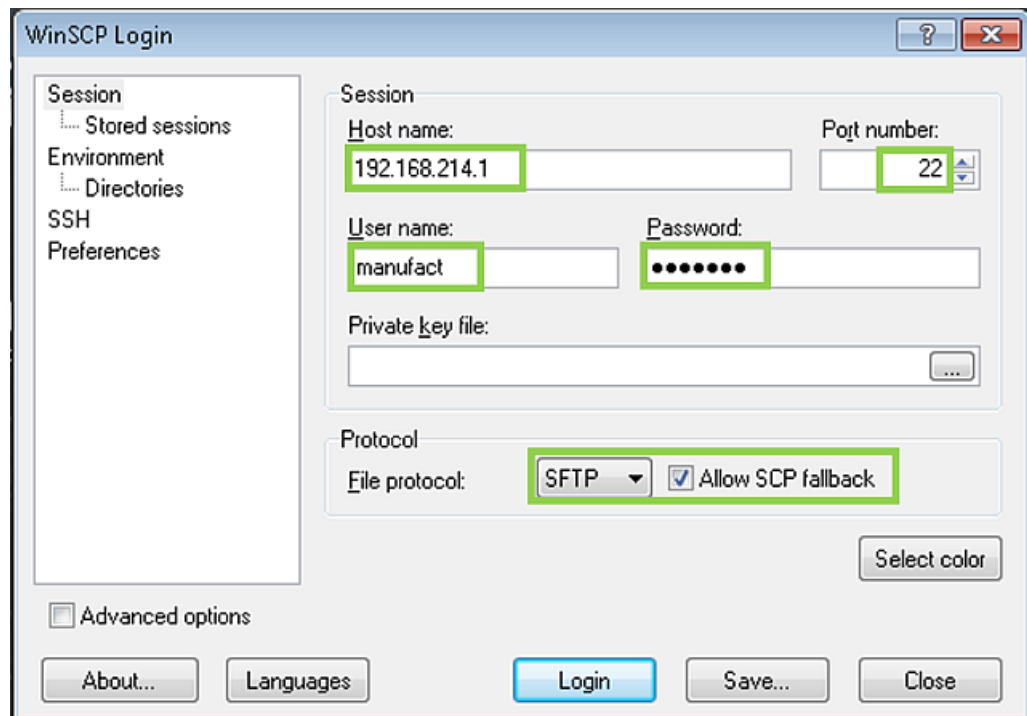
- Commissioning PC; e.g. SIMATIC Field PG
- The following tools are installed on the commissioning PC:
 - WinSCP
 - Additional information:**
SINUMERIK 840D sl SINUMERIK Operate (IM9) Commissioning Manual
Equipment Manual for SINUMERIK 828D PPU and components
 - PuTTY
- When installing for the first time, the commissioning PC must have a connection to the X120 interface (system network) of the NCU.

Procedure

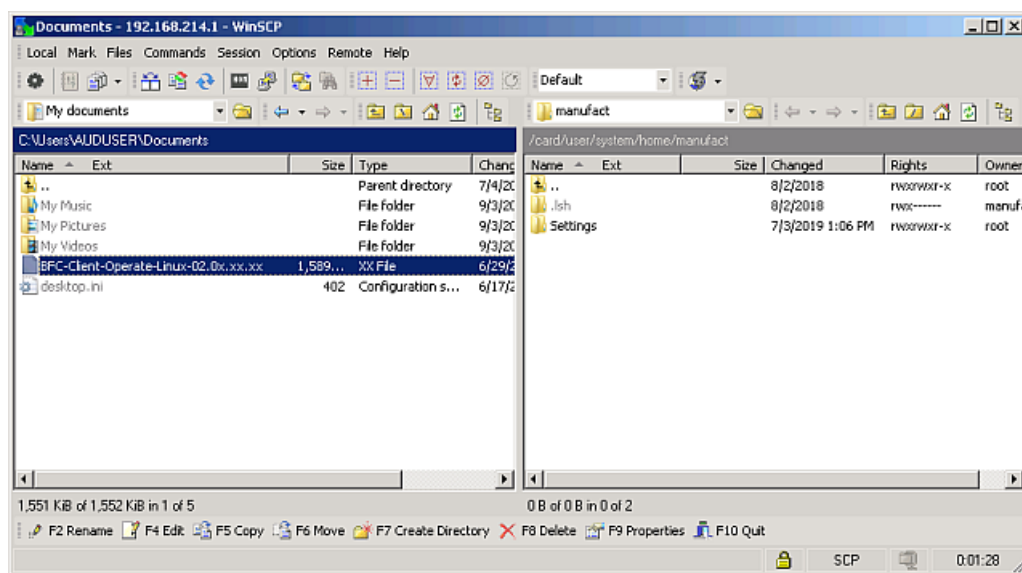
1. Start WinSCP.
Enter the following data to establish a connection to the NCU:

- Host name
- Port number
- User name
- Password
- Protocol

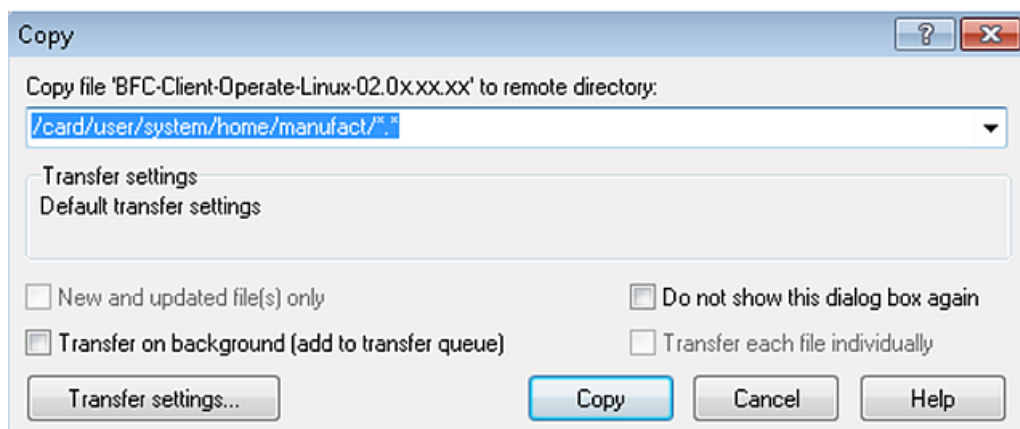
Click on "Login".



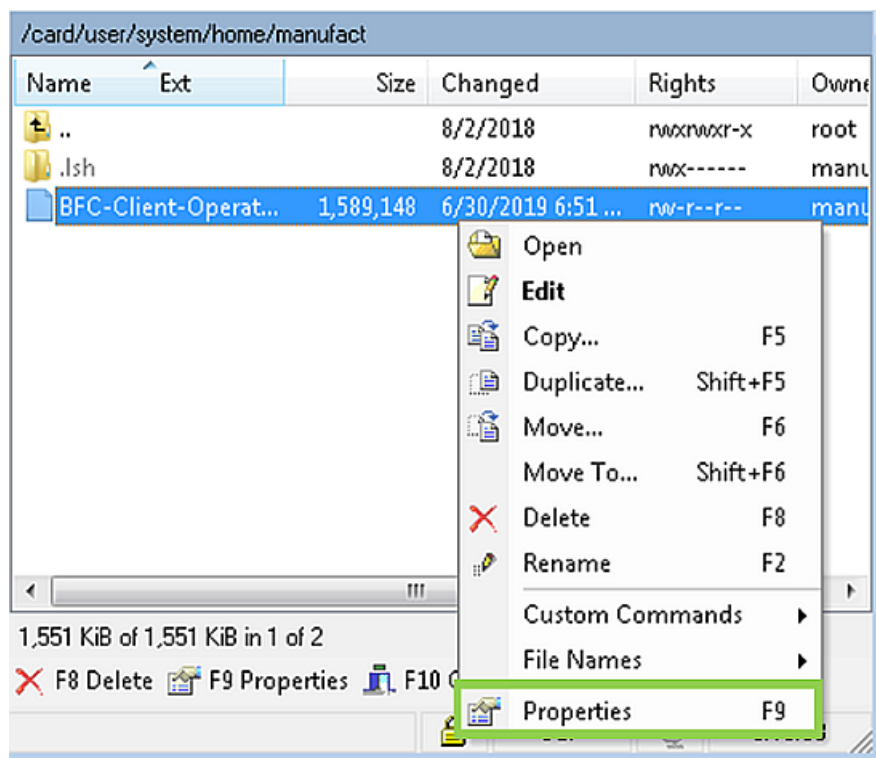
2. A window opens after successfully logging in.
 - To the left you can see the document directory of the industrial PC.
 - To the right you can see the home directory of user "manufact" in the Linux file system of the control.



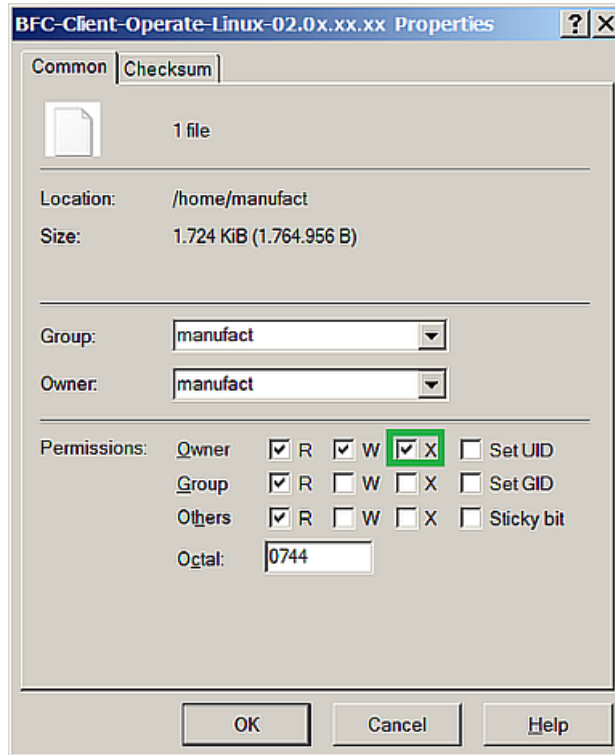
3. Copy file "BFC-Client-Operate-Linux-00.00.00.00" to the home directory of the NCU.



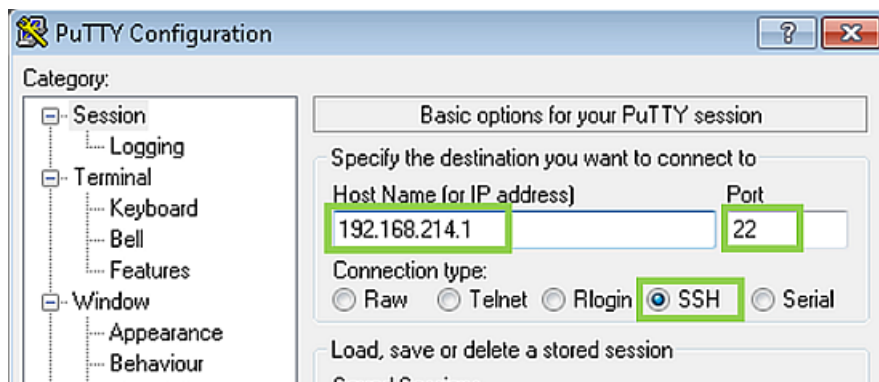
4. Permit file "BFC-Client-Operate-Linux-00.00.00.00" the right to "Execute" as follows:
 - Open the shortcut menu by right-clicking on the file.
 - Select "Properties" in the shortcut menu



5. To issue permission for "Execute", activate checkbox "X" in area "Permissions" for "Owner".
 - Confirm with "OK".The activities for WinSCP have been completed.

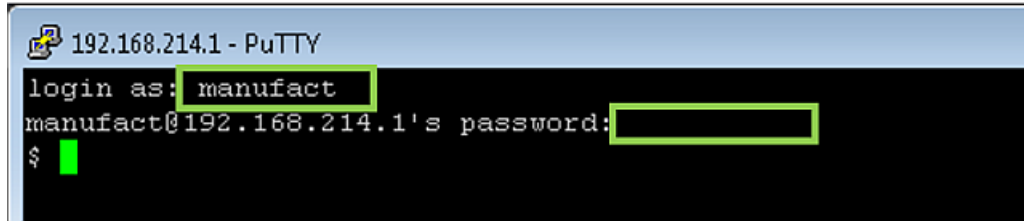


6. Start the "PuTTY" tool.
Enter the following data to establish a connection to the NCU:
 - Host name
 - Port
 - Connection typeConfirm the data with "Open".



7.1 Installing a BFC client

7. A command line window opens.
Enter the login data.



```
192.168.214.1 - PuTTY
login as: manufact
manufact@192.168.214.1's password:
$
```

8. You are in the home directory of user "manufact" on the CF card.
Start the installation with command: `./BFC-Client-Operate-Linux.02.xx.xx.xx`.
Enter the following data, and confirm each entry using the Enter key:
 - BFC gateway: IP address or host name

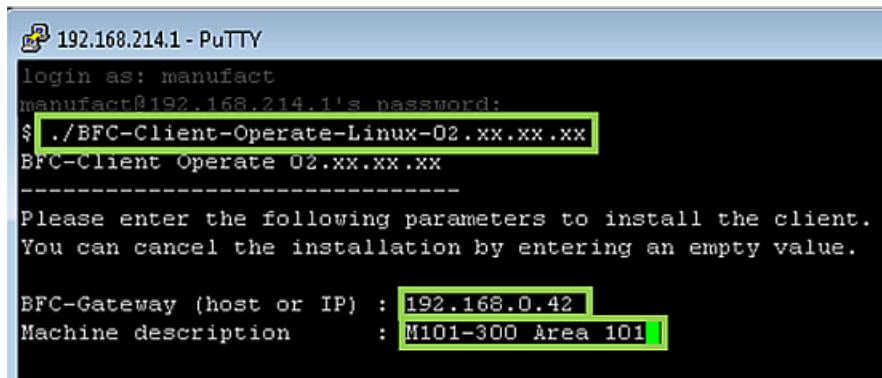
Note**Selecting the entry**

When installing the BFC gateway, a decision is made as to whether the host name or the IP address is used for the TLS certificate.

- Depending on this decision, enter the host name or the IP address used in the certificate in the input window.

- Description of the machine
We recommend, in agreement with the operating company, a machine designation that is unique throughout the system.

Confirm each entry using the ENTER key.



```
192.168.214.1 - PuTTY
login as: manufact
manufact@192.168.214.1's password:
$ ./BFC-Client-Operate-Linux-02.xx.xx.xx
BFC-Client Operate 02.xx.xx.xx
-----
Please enter the following parameters to install the client.
You can cancel the installation by entering an empty value.

BFC-Gateway (host or IP) : 192.168.0.42
Machine description      : M101-300 Area 101
```

9. To perform the installation you are prompted to enter character sequence "ok".
At the same time, check the date and the time on the control system. The actual date/time is shown in the display.
Confirm your entry using the Enter key.
The installation has been completed.

```
$ ./BFC-Client-Operate-Linux-02.0x.xx.xx
BFC-Client Operate 02.0x.xx.xx
-----
Please enter the following parameters to install the client.
You can cancel the installation by entering an empty value.

BFC-Gateway (host or IP) : 192.168.0.42
Machine description      : M101-300 Area 101

----- used parameters -----
BFC-Gateway (host or IP) : 192.168.0.42
Machine description      : M101-300 Area 101

Date and time on this device: 2020-01-09T11:00:46
Device must run with the current date and time. Please verify.

Do you want to install client? Type [ok] to confirm:ok
Installation successfully completed.
Please restart HMI and complete the setup in the BFC-Gateway.
$
```

10. When installation starts, the network connection to the BFC Gateway is checked.

- You receive a message if the network connection was not able to be established. See the yellow marking.
In spite of a faulty connection to the BFC Gateway, installation can continue. Enter "ok" and confirm the entry with the Enter key.
Additional information about network connection diagnostics is provided in Chapter: BFC client diagnostics (Page 351).
- You do not receive any message if the network connection to the BFC Gateway was able to be successfully established.

```
$ ./BFC-Client-Operate-Linux-02.0x.xx.xx
BFC-Client Operate 02.0x.xx.xx
-----
Please enter the following parameters to install the client.
You can cancel the installation by entering an empty value.

BFC-Gateway (host or IP) : 192.162.0.99
Machine description      : M101-300 Area 101

----- used parameters -----
BFC-Gateway (host or IP) : 192.162.0.99
Machine description      : M101-300 Area 101

Date and time on this device: 2020-01-09T11:27:23
Device must run with the current date and time. Please verify.

Do you want to install client? Type [ok] to confirm:ok
The connection to host 192.162.0.99 port 1883 cannot be established!
Do you nevertheless want to install the client? Type [ok] to confirm:ok
Installation successfully completed.
Please restart HMI and complete the setup in the BFC-Gateway.
$
```

11. Once installation has been completed, exit the "PuTTY" window using "exit".

12. Restart SINUMERIK Operate of the control system.

Note

Restart

The BFC client only becomes active after the control system is restarted.

13. After SINUMERIK Operate has restarted, check whether the BFC client has successfully established a connection to the BFC gateway.

- In the trace file, verify that an encrypted MQTT connection was successfully established via port 8883.
- Back up the trace file to document that commissioning was successful.

Further information can be found in Chapter: Analyzing the BFC client (Page 351).

Additional information about errors that occurred during or after the installation, the causes and how they can be resolved is provided in Chapter: Trace (Page 356).

7.1.6 Configuring data buffering

The data collected by the BFC client is continuously acquired according to the configuration and transmitted to the BFC gateway. Additional information on data transmission is provided in Chapter: Creating a BFC client (Page 97).

If the network connection to the BFC gateway is interrupted, the acquired data is buffered on the machine and reported after the connection has been re-established.

By default, a maximum of 500 MB memory on the hard disk of the HMI or CF card is used for buffering the data. It is ensured that at least 100 MB free memory remains available.

The time period during which data can be buffered without data loss depends on the configuration. About 0.5 KB per data point can be assumed as a guideline for memory utilization. If the maximum buffer size is reached, the oldest buffered entries are overwritten (FIFO).

These default settings can be modified as required in the configuration of the BFC client, file "mcvm.ini/slvlservice.ini".

The following figure shows a section of the configuration with regard to data buffering:

```
; -----  
; Enable/Disable message buffer                (Default: true)  
; -----  
BufferEnabled = true  
  
; -----  
; Maximum used disk space in MB                (Default: 500MB)  
; -----  
MaxBufferSize = 500  
  
; -----  
; Minimum remaining free disk space in MB      (Default: 100MB)  
; -----  
MinFreeDiskSize = 100  
  
; -----  
; Buffer file full path                        (Default: <inst-path>\buffer\buffer.db)  
; -----  
BufferFile =
```

7.1.7 BFC client update

If a BFC client has already been installed on a machine, then you can upgrade this to a new version via the BFC Gateway.

More information can be found in Chapter: BFC client update (Page 35).

7.2 Deinstalling a BFC client

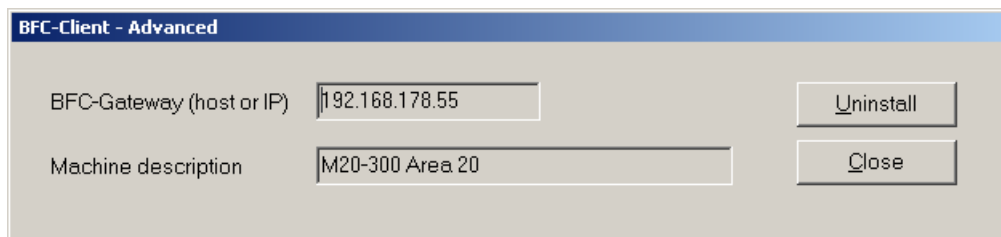
7.2.1 Deinstalling the BFC client on HMI-Advanced

Requirement

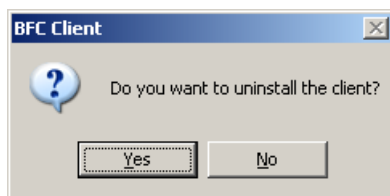
The BFC client is executed on the control system in the service mode.

Procedure

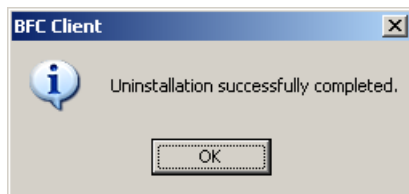
1. The setup recognizes the BFC client installation during the call.
2. If the BFC client is installed, the deinstall window opens.
To uninstall the BFC client, click on "Uninstall".



3. You receive a message to deinstall the BFC client.
Click "Yes" to start the deinstallation.



4. You receive a message indicating that deinstallation has been successfully completed.
Click on "OK" to exit deinstallation.



5. The configurations and data of this BFC client stored at the BFC Gateway are still available after successful uninstallation.
Remove the configurations and data for this client using the BFC Gateway user interface.

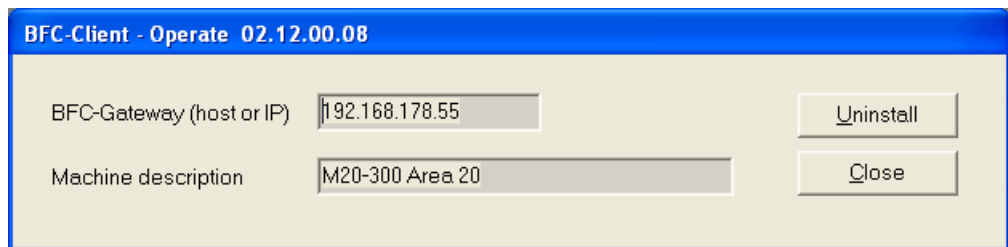
7.2.2 Deinstalling the BFC client on SINUMERIK Operate under Windows (PCU)

Requirement

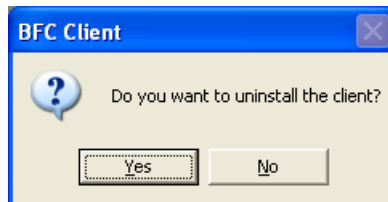
The BFC client is executed on the control system in the service mode.

Procedure

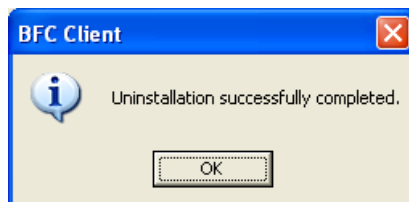
1. The setup recognizes the BFC client installation during the call.
2. If the BFC client is installed, the deinstall window opens.
To uninstall the BFC client, click on "Uninstall".



3. You receive a message to deinstall the BFC client.
Click "Yes" to start the deinstallation.



4. You receive a message indicating that deinstallation has been successfully completed.
Click on "OK" to exit deinstallation.



5. The configurations and data of this BFC client stored at the BFC Gateway are still available after successful uninstallation.
Remove the configurations and data for this client using the BFC Gateway user interface.

7.2.3 Uninstalling the BFC client from SINUMERIK Operate under Linux

Requirement

The BFC client was copied to the control system where it is executed.

Procedure

1. The setup recognizes the BFC client installation during the call.
2. If the installation program detects an existing installation on the system, the uninstallation dialog will start.
 - When prompted to do so, enter character sequence "ok".
 - Confirm the character sequence using the Enter key.
 - Deinstallation has been completed.

```

Using username "manufact".
$ ./BFC-Client-Operate-Linux-02.0x.xx.xx
BFC-Client Operate 02.0x.xx.xx
-----
BFC-Gateway (host or IP) : 192.168.0.42
Machine description      : M101-300 Area 101

Do you want to deinstall the client? Type [ok] to confirm: ok
Deinstallation successfully completed. Please restart HMI.
$ █

```

3. Exit the "PuTTY" window using "exit".
4. Restart the control system.

Note**Restart**

The BFC client is only completely removed from the memory when SINUMERIK Operate restarts.

5. The configurations and data of this BFC client stored at the BFC Gateway are still available after successful uninstallation.
Remove the configurations and data for this client using the BFC Gateway user interface.

7.3 Configuring high-frequency data acquisition

The BFC function "High-frequency data acquisition" allows you to acquire data with a high clock rate (a multiple of the interpolation cycle/position control cycle) from a SINUMERIK control system.

The configuration is carried out in the standard operating area "Diagnostics" > "Trace" of the SINUMERIK Operate operating software and thus directly on the machine.

The operating area offers you the following options:

- Compilation of the signals to be acquired
- Test of the created configuration directly on the machine

The data acquired during the execution of an NC program is displayed on the control in the form of a curve diagram.

By saving the created configuration as a file with a fixed name, this configuration is activated for acquisition by the BFC client.

The high-frequency data is acquired by the BFC client without manual operation on the control system. This can be done continuously or, for example, only at relevant points of the NC program.

On the BFC gateway, the high frequency data captured from SINUMERIK controls, can be processed through the following gateways:

- InfluxDB gateway - storing in an Influxdb database
- MySQL export gateway - storing in a MySQL database
- MQTT export gateway - forwarding data via the MQTT protocol
- MindSphere gateway - forwarding data to MindSphere
- AMP 4.1 export gateway – forwarding data to a SINUMERIK AMP 4.1 system
- Elasticsearch gateway – forwarding data to ElasticsearchDB
- HTTP gateway – forwarding data to an http web server

7.3.1 Requirement

The following requirements apply for the configuration:

- The basis for the configuration of high-frequency data is the NC trace function of the NCU. All controls are supported by the SINUMERIK Operate operating software:
 - SINUMERIK 828D
 - SINUMERIK 840D sl
 - SINUMERIK ONE

Further information can be found in the system requirements, Chapter: BFC client (Page 44).

- A maximum of 2 NC trace sessions with up to 20 variables each are supported. Exception: Only 1 trace session is supported under SINUMERIK 828D. Within an NC trace session you can use variables from different groups. Examples:
 - System variables
 - NC variables
 - PLC variables
 - Global user data
 - Servo variables

The maximum number of 20 variables per NC trace session limits the total number - regardless of which group the individual variable belongs to.

The following restriction applies for the number of variables in the individual groups:

- The maximum permissible number of servo variables is limited to 10 as standard by the machine data "MD 183373 MM_PROTOCOL_NUM_SERVO_DATA" .

An NC trace configuration must first be tested independently of the BFC client under SINUMERIK Operate.

You cannot activate this configuration for the BFC client until the NC trace configuration under SINUMERIK Operate supplies the desired data.

- To carry out the configuration, you will need technological expertise concerning the structure of the machine and knowledge of working with NC trace.

Note

HMI-Advanced

The "High-frequency data acquisition" function is not available for SINUMERIK controls with the HMI-Advanced operating software.

7.3.2 Configuration

7.3.2.1 Performing the configuration

Procedure

To configure "high-frequency data acquisition", proceed as follows:

1. In SINUMERIK Operate, navigate in operating area "Diagnostics" > "Trace".
Create a new NC trace configuration file using softkey "New trace (drive/NC)".
In the following screen form, assign a name to the configuration and activate option "NC and PLC variables".



Note

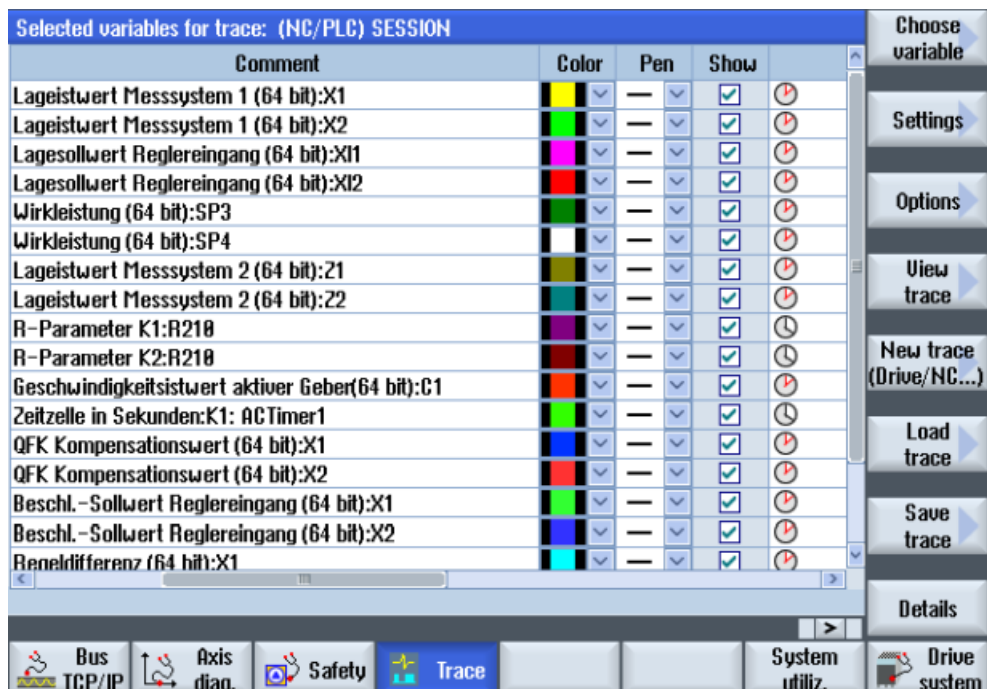
The BFC client does not support option "Drive variables".

Initially, you can freely select the name of the configuration file.

Only use the newly created configuration together with the BFC client once it has been completely created and tested independently of the BFC client.

By saving the configuration under an "Agreed name", the newly created configuration for the BFC client is activated (see Saving the configuration (Page 348)).

2. Configure the "High-frequency data acquisition" using the "Choose variable" and "Settings" softkeys.



For more information on the operating steps, refer to the Commissioning Manual SINUMERIK 840D sl SINUMERIK Operate (<https://support.industry.siemens.com/cs/ww/en/view/109777907>); Chapter: "Diagnostics and service" > "Trace".

7.3.2.2 Compiling signals

Compile the signals for the configuration.

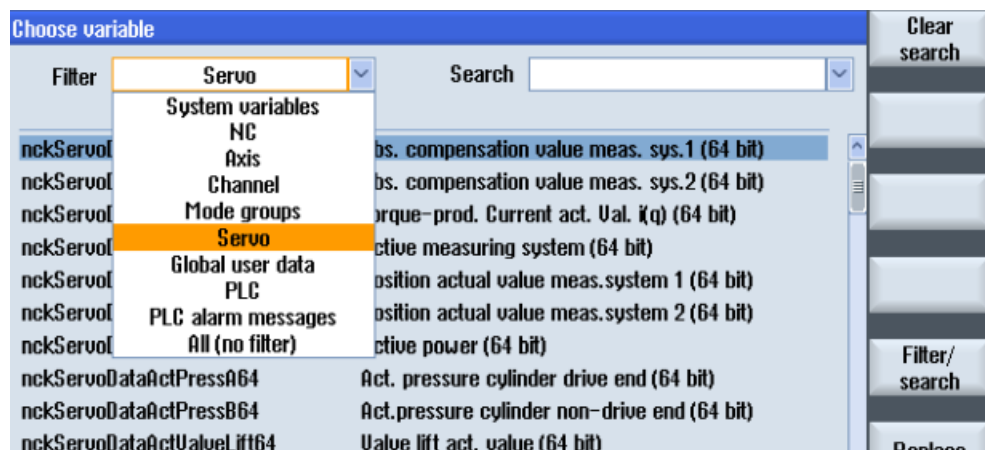
You can select variables that you want to insert from different areas.

- Signals from the servo area are acquired in the position control cycle (servo clock).
- All other signals are acquired in the interpolation cycle (IPO cycle).

A configuration can contain a maximum of 20 signals.

Procedure

1. Navigate to the operating area "Diagnostics" > "Trace".
2. Press the "Insert variable" softkey
3. Select the signals to be acquired.



As default setting, the signal names are generated based on the configured addresses.

Example:

Selected variables for trace: (NC/PLC) Session	
Variable	Comment
/Nck/ServoData/nckServoDataActPos1stEnc64[1]	Position actual value meas.system 1 (64 bit)
/Nck/ServoData/nckServoDataActPos1stEnc64[2]	Position actual value meas.system 1 (64 bit)

Names:

Nck_SD_nckServoDataActPos1stEnc64_u1_1, Nck_SD_nckServoDataActPos1stEnc64_u1_2

The signal names are used on the BFC gateway to identify the captured data. To establish a reference to the application, you can define the signal names in the comment column of the NC trace configuration. To do this, the required signal name is entered in the comment column with a leading ":". The signal names specified in this way must be unique in an NC trace configuration. Characters a - Z, numbers 0 - 9 and underscore "_" are permitted.

Example:

Selected variables for trace: (NC/PLC) Session	
Variable	Comment
/Nck/ServoData/nckServoDataActPos1stEnc64[1] ;XPos	
/Nck/ServoData/nckServoDataActPos1stEnc64[2] ;YPos	

Resulting signal names:
XPos, YPos

7.3.2.3 Defining a start/stop operation

You can define start and stop conditions for recordings.

Procedure

1. Navigate to the operating area "Diagnostics" > "Trace".
2. Press the "Settings" softkey.
3. Configure start and stop conditions according to your requirements.
You will find a description of the individual option buttons below.

Trace - (NC/PLC) Settings

Start trace

- ☐ At softkey Start trace
- ☒ Variable: \$AN_SLTRACE Relation: == Value: 1

Collecting Data to Hard Drive Rates: IP01 = 0.0160 s Servo = 0.0020 s

Stop trace

- ☐ At softkey Stop collecting
- ☐ Elapsed time
- ☐ At storage limit
- ☒ Variable: \$AN_SLTRACE Relation: == Value: 0

☐ Post-triggered with 0.500 s

☐ Discard oldest data at storage limit 123079 KB --> 18.9 min

Buttons on the right: Choose variable, Quick list, Sample rates, Cancel, OK

Bottom navigation bar: Bus TCP/IP, Axis diag., Safety, Trace, System utiliz., Drive system

Option button "Start trace with softkey"

If you select the "Start trace with softkey" option button in the "Start trace" area, this has the following effect in the context of the BFC client:

- Trace recording starts immediately after the machine is switched on.
- The recording started in this way is continuous, regardless of the operating mode and whether an NC program is being executed.

"If variable" option button

If you select the "If Variable" option button in the "Start trace" and "Stop trace" areas, this has the following effect:

- Trace recording is only carried out in the desired process stages if the defined start/stop condition applies.

You can use the following variables as trigger variables for start/stop operation:

- Predefined variable \$AN_SLTRACE
- The variable \$AN_SLTRACE is specially provided for starting and stopping a recording.
- Any GUD variable
- NCK variables, e.g. axis position

In most cases, you must set or reset the selected variable at the desired positions in the NC program.

```
STOPRE
```

```
$AN_SLTRACE=1 ; start trace
```

```
...
```

```
STOPRE
```

```
$AN_SLTRACE=0 ; stop trace
```

A trace recording covers the entire machining (NC program) or is performed specifically at one or more points in the NC program.

The duration of the machining time has different effects on the recording.

Short machining times

- The data is recorded during the entire machining process, but not during idle times.
- Data is only recorded in automatic mode.

Longer processing times

- The acquisition can be made at several points in the NC program if a critical machine situation is expected.

Note**Defining a start/stop operation without NC program changes**

A Siemens service technician can implement the start/stop of a trace recording project-specifically in a script in the BFC client.

In this case, no changes in the NC program are necessary.

"Elapsed time" option button

If you select the "Elapsed time" option button in the "Stop trace" area, trace recording stops after the defined time.

Combination of option buttons "If variable" and "Elapsed time"

The selection has the following advantage:

Recording is also terminated after a defined time if the NC program was interrupted during recording.

Additional settings**"Data acquisition"**

Always use the "on hard disk" option for the "Data acquisition" setting.

"Discard oldest data at storage limit"

The input field "Discard oldest data at storage limit" has no effect in the context of the BFC client.

Note**SINUMERIK Operate Version 2.6**

Trigger variable \$AN_SLTRACE is not available in SINUMERIK Operate, Version 2.6.

Alternatively, an R parameter can be used to start/stop a trace recording.

Example for parameter R0:

\$R[0]:CH1

7.3.2.4 Changing the sampling rate for signals

Use the "Sampling rates" softkey to increase the sampling rate for all signals in the configuration to a multiple of the position control cycle (servo clock) or interpolation cycle (IPO cycle). This can reduce the data volume.

Ensure that the control system is not overloaded by the configured data collection.

Procedure

1. Navigate to the operating area "Diagnostics" > "Trace".
2. Press the "Settings" softkey.
3. Press the "Sampling rate" softkey.
4. Change the sampling rate.

Sample Rate Settings			
	Base Rate	Factor	Effective Rate
IPO cycle	0.0080s x	2	= 0.0160s
Servo cycle	0.0020s x	1	= 0.0020s

7.3.2.5 Saving the configuration

You can save the created configuration as an XML file.

If you save the NC trace configuration under an "agreed name", this configuration is activated by the BFC client and the configured recording is performed automatically.

The following two configuration files are used by the BFC client as default:

- user\sinumerik\hmi\data\trace\PMON1.xml
- user\sinumerik\hmi\data\trace\PMON2.xml

You must specify the name: "PMON1" or "PMON2" when saving.

The directory "user\sinumerik\hmi\data\trace" corresponds to the default directory which is offered to the user when saving.

Note**Linux**

Note that the name of the configuration under Linux must be entered in upper case letters.

The BFC client continuously monitors the existence or changes of the named files. If one of these files is created, changed or deleted, the BFC client creates, reconfigures or deletes a corresponding trace recording. If necessary, you can configure the names "PMON1" and "PMON2" differently in the configuration of the BFC client (slvlservice.ini).

Procedure

1. Navigate to the operating area "Diagnostics" > "Trace".
2. Press the "Save trace" softkey.
3. Save the created configuration as an XML file.
The name of this file can be selected as required.

Save trace	
Destination folder:	/user/sinumerik/hmi/data/trace
Trace File Name:	PMON1.xml

Checking the status of an NC trace session

To check the current status of the NC trace session, open the "System State" area in the user interface of the BFC gateway. More information can be found in Chapter: "System State" area (Page 297).

The following table shows the possible states of the NC trace session.

"Description" column	Description
No text	The BFC client has not recognized any NC trace configuration.
NC trace session PMON1.xml/PMON2.xml is ready	The BFC client has recognized an NC trace configuration. Data is not acquired.
NC trace session PMON1.xml/PMON2.xml is active ^{*)}	The configured NC trace session is active. The start condition has occurred. Data is sent to the BFC gateway. The "Show live data" view shows the live data in real time. The signal names of an NC trace recording are additionally marked with "PMON1" or "PMON2" in this view. In the "System State" area and in the "Messages last Minute" column, the quantities of all messages sent by the BFC client are displayed. Click on this number to additionally display the proportion of "high-frequency messages".

^{*)} The display "NC Trace Session PMON1.xml/PMON2.xml is active" may not be displayed correctly in the web interface for short recordings such as <40 seconds

7.3.2.6 Accept configuration

You can transfer an NC trace configuration created on one machine to another machine using a USB flash drive.

We recommend manually testing the transferred NC trace configuration on the other machine.

The NC trace configuration cannot be transferred to another machine without a test. The reason for this can be, for example, a deviating axis configuration.

7.3.2.7 Test configuration

We recommend that you first test a newly created NC trace configuration independently of the BFC client. You should only activate the configuration for the BFC client when the NC trace configuration provides the required data in the test.

During the test, do not save the NC trace configuration under the "agreed" name "PMON1.xml" or "PMON2.xml", but under another freely selectable name such as "TEST1.xml".

Once the test has been completed successfully, you can activate the NC trace configuration for the BFC client by saving it under the name "PMON1.xml" or "PMON2.xml".

Procedure

To manually test an NC trace configuration independently of the BFC client, proceed as follows:

1. In the SINUMERIK Operate operating software, navigate to the "Diagnostics" > "Trace" operating area.
2. Press the "Start trace" softkey.
3. To start the NC trace recording, you must fulfill the previously configured start condition. The recording ends when the configured stop condition is reached.
4. To display the recorded data, press the "Display trace" softkey.

For further information on the operating steps, refer to the SINUMERIK 840D sl SINUMERIK Operate (<https://support.industry.siemens.com/cs/ww/en/view/109777907>) Commissioning Manual; Chapter: "Diagnostics and service" > "Trace".

Limitation to 2 active NC trace sessions

In the standard configuration of the SINUMERIK Operate, a maximum of 2 active NC trace sessions are possible.

If 2 NC trace sessions are already activated in the BFC client, you cannot manually start another NC trace session. An error message appears when recording is started.

In this case you have two options:

- Temporarily deactivate the active NC trace session in the BFC client.
- Contact the SINUMERIK hotline to increase the maximum number of NC trace sessions in SINUMERIK Operate. For the BFC client, the maximum number of NC trace sessions is as described in Chapter Requirement (Page 342). For the BFC client, it is not possible to increase the number of NC trace sessions that can be recorded.

Temporarily deactivating active NC trace recording

To temporarily deactivate an active NC trace recording in the BFC client, proceed as follows:

1. Navigate to the operating area "Commissioning" > "System files".
2. Temporarily rename one of the active NC trace configurations:
 - user\sinumerik\hmi\data\trace\PMON1.xml
 - user\sinumerik\hmi\data\trace\PMON2.xml

7.4 BFC client diagnostics

The following log files are provided for diagnosis of the BFC client:

- "install.log": Events that occur while installing the BFC client
- "mcvm.log": BFC client runtime information on HMI Advanced control systems
- "slvlservice.log": BFC client runtime information on SINUMERIK Operate control systems

7.4.1 Analyzing the BFC client

The installation of the BFC client is logged in the "install.log" log file.

The log file includes, for example, all installation steps, the generated configuration file and the message relating to the installation result.

Based on this information, you can reconstruct and understand the installation process and create the corresponding diagnostics.

Note

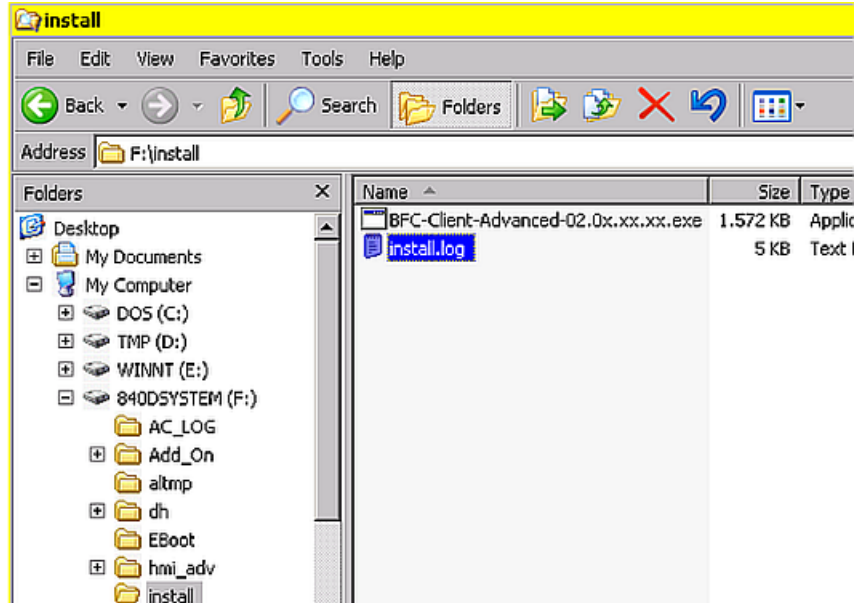
Archiving the log file

Always archive the log file.

Procedure

To reconstruct the installation, proceed as follows:

- Navigate to the folder in which the installation file was executed.



An example of a log file after a successfully completed installation is shown in the following diagram.

```

created.
2019-07-09T10:59:44.540 Configuration settings.
file:=F:\Add_On\mcvm.ini
[options]
Version = 02.0x.xx.xx
MachineId = alk1n7e4em
MachineDesc = M101-300 Area 101
ProjectId = connectivity
[mqtt]
Host = 192.168.3.42
Port = 1883
Username = lobby:guest
Password = encrypted:39AA5383BD24CE34D712FE16CC9F757C
CaCert =

2019-07-09T10:59:44.610 Regie startup configuration entry created.
(F:\Add_On\Regie.ini)
[StartupConfiguration]
Startup39 = name := mcvm, Cmdline := "/d", Timeout :=0
2019-07-09T10:59:44.610 Installation successfully completed.
  
```

HMI-Advanced folder structure

The following folder structure is generated on the PCU after installation:

F:\Add_On

```

    mcvm.exe
    mcvm.ini
    mcvm.log
    mcvm.bak
    runtime\data.ini
    mosquito-2.0.9.dll
    ReadMe_OSS.htm
    _Regie.ini

```

```

    -buffer
      buffer.db

```

```

    *.

```

```

    -mcx
      *.

```

```

    -update
      *.

```

mcvm.ini¹⁾

mcvm.log¹⁾

mcvm.bak¹⁾

runtime\data.ini¹⁾

./buffer/buffer.db¹⁾

./cert/*.¹⁾

./mcx

./update/*.¹⁾

¹⁾

Configuration file with all of the important settings

Log file of the BFC client

Log file of the BFC client

Runtime data

Buffer database

BFC client certificates for secure communication

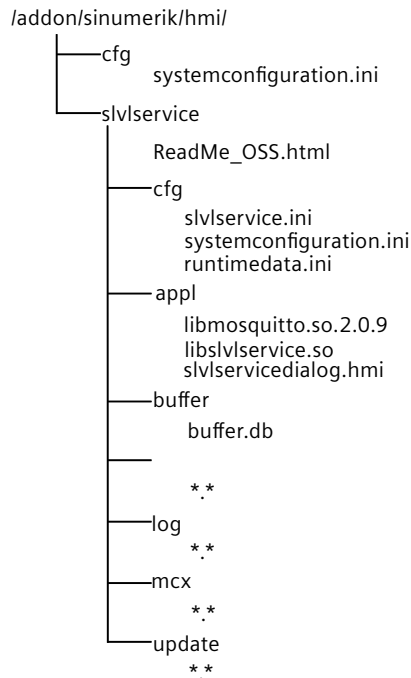
Routines for data acquisition and processing

Is used to upgrade the BFC client

These entries are generated when starting for the first time or created when required in operation.

SINUMERIK Operate folder structure

The following folder structure is generated on the hard disk or the CF card after installation:



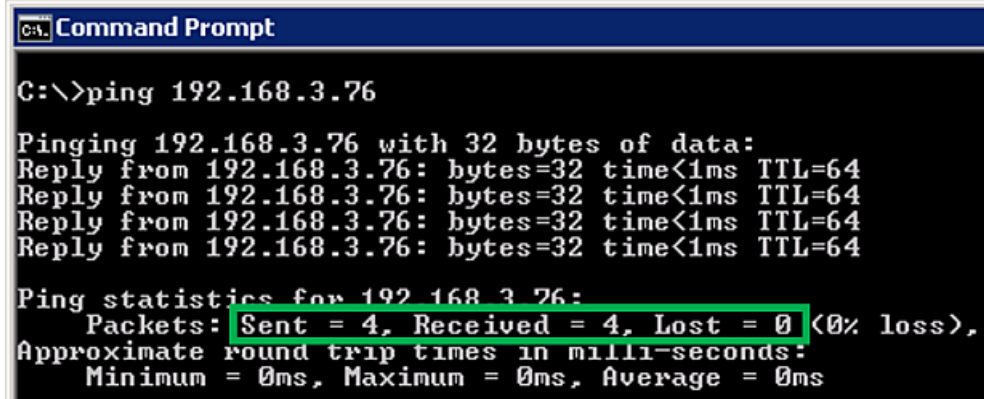
<code>./cfg/slvlservice.ini</code>	Configuration file with all of the important settings
<code>./log/slvlservice.log¹⁾</code>	Log file of the BFC client
<code>./log/slvlservice.bak¹⁾</code>	Log file of the BFC client
<code>runtimeodata.ini¹⁾</code>	Runtime data
<code>./buffer/buffer.db¹⁾</code>	Buffer database for power failure (optional)
<code>./cert/*.*¹⁾</code>	BFC client certificates for secure communication
<code>./mcx/*.*¹⁾</code>	Routines for data acquisition and processing
<code>./update/*.*¹⁾</code>	Is used to upgrade the BFC client
¹⁾	These entries are generated when starting for the first time or created when required in operation.

7.4.2 Checking the accessibility of the BFC gateway in the network

Ensure that the BFC gateway can be accessed in the network before installation on the control system.

Procedure

1. Use the command line command `ping` at the control system to test whether the BFC gateway can be accessed through your network.

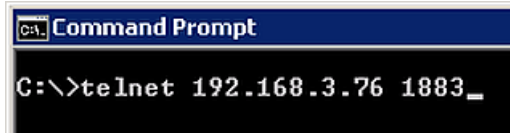


```
C:\>ping 192.168.3.76

Pinging 192.168.3.76 with 32 bytes of data:
Reply from 192.168.3.76: bytes=32 time<1ms TTL=64
Reply from 192.168.3.76: bytes=32 time<1ms TTL=64
Reply from 192.168.3.76: bytes=32 time<1ms TTL=64
Reply from 192.168.3.76: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.3.76:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

2. Also test the MQTT port using the command line tool "TELNET".



```
C:\>telnet 192.168.3.76 1883_
```

3. If the MQTT port on the BFC gateway is accessible in the network, an empty window opens after the command has been entered. Close the window.

Note

Access denied

Contact your local IT department if you have no access to the BFC gateway via the network.



```
Telnet 192.168.3.76
_
```

7.4.3 Trace

7.4.3.1 Activating the trace

Procedure

1. Open the corresponding configuration file using a text editor.
 - When using HMI-Advanced: "mcvm.ini"
 - When using SINUMERIK Operate: "slvlservice.ini"

Note**Open the configuration file**

If protection level "Manufacturer" is activated, you can also open the configuration file even when HMI-Advanced or SINUMERIK Operate is operational.

Note**Only change the configuration file when requested to do so**

Only change the configuration file if you are prompted to do so by the project manager or from the hotline.

2. Change the value for "TraceLevel" in section [options] as shown in the following example:

```
[options]

; -----
; Current client version. (Do not change this entry manually)
; -----
Version = 02.0x.xx.xx

; -----
; Trace level (default:0)
; 0 - trace information messages only
; 1 - trace all value changes of output variables
; 2 - trace all value changes of output and input variables
; 3 - trace all value changes of all variables
; 4 - trace additionally invoking of all variable logic functions
; 5 - trace additionally read/write access to data source
; -----
TraceLevel = 1

; -----
; Log file size in bytes
; (default: 600000)
; -----
LogFileSize = 600000
```

3. Restart HMI-Advanced or SINUMERIK Operate to accept the changes.
After the restart, the trace file contains messages with detailed information.
Further information can be found in Chapter: Analyzing the trace file (Page 357).

Note**Active trace**

Active trace with increased message depth can negatively impact the system performance.

- Only change this option if it is necessary or you are prompted to do so by the service department/hotline.
 - Ensure that there is sufficient memory on the data storage medium.
-

7.4.3.2 Analyzing the trace file

A trace file is generated if the BFC client is installed and HMI-Advanced was restarted.

Depending on the specific trace level, trace files contain different information:

- Informative data
- Messages that contain processed information and variables

The assignment of the trace files to the control systems is shown in the following overview:

SINUMERIK 840D with HMI-Advanced: "mcvm.log"
SINUMERIK 840D sl with SINUMERIK Operate: "slvlservice.log"

Information for trace level 0

Basic information and error messages are already generated for trace level = 0.

The following log file represents an example for log outputs when the control system is first started after the installation.

7.4 BFC client diagnostics

```

1 2020-01-09T14:55:05.452000+00:00 [I] Client Version: 02.08.00.00 Linux /card/addon/sinumerik,
2 2020-01-09T14:55:05.458000+00:00 [I] CardId=SPG2014071900576
3 2020-01-09T14:55:05.459000+00:00 [I] Runtime file not found, (/card/addon/sinumerik/hmi/slv1:
4 2020-01-09T14:55:05.459001+00:00 [I] Runtime is not valid.(Application runs in demo mode for
5 2020-01-09T14:55:05.478000+00:00 [I] Reading configuration file: /card/addon/sinumerik/hmi/s:
6 [options]
7 MachineId = 0bafxq6d63
8 MachineDesc = M101-300 Area 101
9 2020-01-09T14:55:05.522000+00:00 [I] Attempt to connect the MQTT broker. Host:=192.168.3.89
10 2020-01-09T14:55:05.522001+00:00 [I] MQTT: Client connectivity-0bafxq6d63 sending CONNECT
11 2020-01-09T14:55:05.540000+00:00 [I] MQTT: Client connectivity-0bafxq6d63 received CONNACK ((
12 2020-01-09T14:55:05.540001+00:00 [I] MQTT broker connected, Host:=192.168.3.89 Port:=1883 Use
13 2020-01-09T14:55:05.575000+00:00 [I] UpdateConfig done
14 [options]
15 MachineId = hfxxd1474e
16 [mqtt]
17 Password = encrypted:11f5e2b147ddc9de0480b6ea69704ef3
18 Port = 8883
19 Username = encryptedlobby:b047b1f4-c60b-4e3d-89bf-3c30603801d1
20 Password = encrypted:304c8df53f384def111cdac429bd100ebd0ed6fda17f40a8eb88c9c8c72b0643782b2a
21 CaCert = ca.crt
22 ClientCert =
23 ClientKey =
24 ClientKeyPW =
25 2020-01-09T14:55:06.588000+00:00 [I] MQTT: Certificate Information - Issuer: /C=DE/O=Siemens
26 2020-01-09T14:55:06.589000+00:00 [I] Attempt to connect the MQTT broker. Host:=192.168.3.89
27 2020-01-09T14:55:06.591000+00:00 [I] MQTT: Client connectivity-hfxxd1474e sending CONNECT
28 2020-01-09T14:55:06.615000+00:00 [I] MQTT: Client connectivity-hfxxd1474e received CONNACK ((
29 2020-01-09T14:55:06.628000+00:00 [I] MQTT broker connected, Host:=192.168.3.89 Port:=8883 Use

```

- Line 1 BFC client version
- Line 12 An unencrypted connection was established to the BFC gateway.
- Line 25 Information about the certificate used to establish an encrypted connection.
The validity date (from/to) of the certificate is also shown in this line.
The actual date of the control system (first column in the log file) must lie within this time interval.
- Line 29 An encrypted connection was established to the BFC gateway.

Information after a restart

After a restart you see different information, which can support you when analyzing the BFC client.

This information includes, for example:

- Client version
- Installation version
- Actual MachineId
- Information about the connection status between the BFC client and BFC gateway
 - MQTT broker connected
 - Host:=xxx.xxx.xxx.xxx
 - Port:=1883

Port 8883 is used as soon as secure communication is active.

MachineId

The MachineId is an important parameter for the BFC gateway as it internally assigns the data sent from the BFC client.

The MachineId, which is generated when the system is installed for the first time, is only used for initially establishing communication with the BFC gateway.

```
[options]
MachineId = 0bafxq6d63
MachineDesc = M101-300 Area 101
2020-01-09T14:55:05.522000+00:00 [I] Attempt to connect the MQTT broker. Host:=192.168.3.89 Port:=1883
2020-01-09T14:55:05.522001+00:00 [I] MQTT: Client connectivity-0bafxq6d63 sending CONNECT
2020-01-09T14:55:05.540000+00:00 [I] MQTT: Client connectivity-0bafxq6d63 received CONNACK (0)
2020-01-09T14:55:05.540001+00:00 [I] MQTT broker connected, Host:=192.168.3.89 Port:=1883
Username:=lobby:guest ClientId:=connectivity-0bafxq6d63
```

ClientId

If secure communication was activated at the BFC gateway, then the BFC gateway assigns a new, also automatically generated ClientId (<projectId>-<MachineId>).

```
2020-01-09T14:55:06.588000+00:00 [I] MQTT: Certificate Information - Issuer: /C=DE/O=Siemens AG/OU=APC
2020-01-09T14:55:06.589000+00:00 [I] Attempt to connect the MQTT broker. Host:=192.168.3.89 Port:=8883
2020-01-09T14:55:06.591000+00:00 [I] MQTT: Client connectivity-hfxxd1474e sending CONNECT
2020-01-09T14:55:06.615000+00:00 [I] MQTT: Client connectivity-hfxxd1474e received CONNACK (0)
2020-01-09T14:55:06.628000+00:00 [I] MQTT broker connected, Host:=192.168.3.89 Port:=8883
Username:=encryptedlobby:b847b1f4-c68b-4e3d-89bf-3c30683801d1 ClientId:=connectivity-hfxxd1474e
```

Once the manual first commissioning has been completed at the BFC gateway, the machine is assigned its final ClientId, and entry "Runtime is valid" is made.

```
2020-01-09T15:42:52.181001+00:00 [I] Runtime is valid, software: mqtt
2020-01-09T15:42:52.182000+00:00 [I] UpdateConfig done
[options]
MachineId = M101-300
[mqtt]
Password = encrypted:596BBEC396DD6C52A4E1F89CA63E09EEB66144D700248964D29864EA812805A479521B;
[options]
Software = encrypted:DDA6DC07FEC9B8E5077C782F91682BF3
MachineId = M101-300
Software = encrypted:DDA6DC07FEC9B8E5077C782F91682BF3
[mqtt]
Port = 8883
Username = iotserver
Password = encrypted:7C730AE8413E59DD2F0C2748C3672E50260097F6CB327AD925E294EDE5645C068ED673;
2020-01-09T15:42:53.185000+00:00 [I] MQTT: Certificate Information - Issuer: /C=DE/O=Siemens
2020-01-09T15:42:53.186000+00:00 [I] Attempt to connect the MQTT broker. Host:=192.168.3.89
2020-01-09T15:42:53.186001+00:00 [I] MQTT: Client connectivity-M101-300 sending CONNECT
2020-01-09T15:42:53.206000+00:00 [I] MQTT: Client connectivity-M101-300 received CONNACK (0)
2020-01-09T15:42:53.206001+00:00 [I] MQTT broker connected, Host:=192.168.3.89 Port:=8883
Username:=iotserver ClientId:=connectivity-M101-300
```

Note

ClientId

The ClientId comprises the ProjectId and MachineId. After the procedure has been completed, all three IDs are saved to the configuration file of the client on the machine.

Message when connection problems occur

Log outputs for different error scenarios regarding the network connection to the BFC gateway are subsequently shown.

The BFC gateway cannot be accessed in the network:

```
2020-01-09T15:42:53.206001+00:00 [I] MQTT broker connected, Host:=192.168.3.89 Port:=8883 Use
2020-01-09T16:28:37.220000+00:00 [I] MQTT broker disconnected Host:=192.168.3.89 Port:=8883
-> 14 Connection Refused: unknown reason. errno:=Broken pipe
```

Further information can be found in Chapter: Checking the accessibility of the BFC gateway in the network (Page 354)

Messages for an incorrect date

This log message appears if the date on the SINUMERIK control system lies outside the validity of the certificate:

```
2001-01-02T13:28:16.023000+00:00 [E] 3619 MQTT: OpenSSL Error:
error:1416F086:SSL
routines:tls_process_server_certificate:certificate verify failed
```

Messages for an incorrect certificate

This log message is shown if an incorrect certificate is being used:

```
2020-01-02T13:31:05.269000+00:00 [I] Attempt to connect the MQTT broker. Host:=192.168.3.89 Port:=8883 Username:=iotserver Cli
2020-01-02T13:31:05.285000+00:00 [I] MQTT: Client connectivity-Ellen sending CONNECT
2020-01-02T13:31:05.296000+00:00 [E] 3619 MQTT: OpenSSL Error: error:0407008A:rsa routines:RSA_padding_check_PKCS1_type_1:inva
2020-01-02T13:31:05.296001+00:00 [E] 3619 MQTT: OpenSSL Error: error:04067072:rsa routines:rsa_cssl_public_decrypt:padding chee
2020-01-02T13:31:05.297000+00:00 [E] 3619 MQTT: OpenSSL Error: error:0D0C5006:asn1 encoding routines:ASN1_item_verify:EVP lib
2020-01-02T13:31:05.297001+00:00 [E] 3619 MQTT: OpenSSL Error: error:1416F086:SSL routines:tls_process_server_certificate:cert:
2020-01-02T13:31:05.297002+00:00 [I] MQTT: Network thread shut down (8)
```



```
3619 MQTT: OpenSSL Error: error:0407008A:rs:
3619 MQTT: OpenSSL Error: error:04067072:rs:
3619 MQTT: OpenSSL Error: error:0D0C5006:as:
3619 MQTT: OpenSSL Error: error:1416F086:SS
```

Using an OPC UA server

8.1 Overview

The OPC UA server collects the data of all BFC Gateway clients and makes it available in a dedicated OPC UA address space.

The OPC UA server displays the current client values.

Specifications that are used

The OPC UA server uses the following specifications:

- Up to 5 connections to higher-level systems
- OPC UA standard according to specification OPC UA V1.03
- The "Standard OPC UA server profiles" are used.
- Data access according to IEC62541-8 "Data Access" is supported.
- Data access according to IEC62541-11 "Historical Access" is supported.
- Subscriptions according to IEC62541-4 "Services"
- Alarms
 - Alarms according to IEC62541-9 "Alarms and Conditions"
 - Alarms based on data type "AlarmConditionType"
- Connections with user name and password
- Binary OPC UA transfer protocol
- OPC UA security profiles:
 - Basic256Sha256 - Sign
 - Basic256Sha256 - Sign & Encrypt

The following specification is not supported by the OPC UA server:

- The XML-based transmission protocol

8.2 Configuring a certificate

To be able to establish an encrypted connection between the OPC UA server and the OPC UA client, the OPC UA server must know the certificate of the OPC UA client.

Certificates are configured in the ConfigUI.

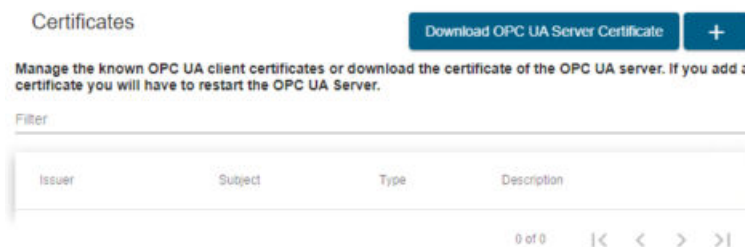
8.2.1 Uploading the client certificates in the ConfigUI

Procedure

1. Open the OPC UA server configuration view.



2. Upload the OPC UA client certificates.



3. Configure the client certificate

Add certificate

Type cert description ...
clientcert

Select cert type ... *
OPC UA Client

Paste pem string ... *

```
-----BEGIN CERTIFICATE-----
MIICODCCAaGgAwIBAgIBADANBgkqhkiG9w0BAQsFADA5MQswCQYDVQQGEwJ1czEM
MAoGA1UECAwDbnJ3MQ0wCwYDVQQKDAR0ZXN0MQ0wCwYDVQQDDAR0ZXN0MB4XD
MDMzMTEyMjg1NloXDTEyMDMzMTEyMjg1NlowOTELMAkGA1UEBhMCdXMxMDAKBg
NVBAGMA25ydZENMAAsGA1UECgwEdGVzdDENMAAsGA1UEAwwEdGVzdDCBnzANBg
kqhkiGIIUCAwEAAANQME4wHQYDVIR0OBByEFB3UbsKQyVaYic5YZavk/d5uELBN
MB8GA1UdIwQYMBaAFB3UbsKQyVaYic5YZavk/d5uELBNMAwGA1UdEwQFMAMB
A8wDQYJKoZIhvcNAQELBQADgYEAAuz1wMqJ0i0M8pDOQzKYSbJxG0rNixOZt
OQDWUhn3ylZqWQQA2w6J2Plc5/bSWtpvdKjuMafnhMYP0waTIFY8PbpPvkR6Ail
aa2sLRkcWcaNGMpxDlKodO6jEZFGMmyaC+xZiBoZNPmslr6UNSRnufAPIXtiqhe
SijoludWq/U=
-----END CERTIFICATE-----
```

Save Cancel

Option	Description
Type cert description	Certificate name, can be freely selected
Select cert type	Certificate type
Paste pem string	X509 certificate in the PEM format

4. Save the certificate.

Add certificate

Type cert description:
clientcert

Select cert type:
OPC UA Client

Paste pem string:

```
-----BEGIN CERTIFICATE-----
MIICODCCAAgAwIBAgIBADANBgkqhkiG9w0BAQsFADA5MQswCQYDVQQGEwJ1czEM
MAoGA1UECAwDbnJ3MQ0wCwYDVQQKZAR0ZXN0MQ0wCwYDVQQDDAR0ZXN0MB4XD
MDMzMTEyMjg1NloXDTEyMjg1Mjg1NloOTELMAkGA1UEBhMCdXMxDDAKBgNV
BAAwMA25ydzENMA5GA1UECgwEdGVzdDENMA5GA1UEAwwEdGVzdDCBnzANBgkqhkiG
IiUCAwEAaANQME4wHQYDVR0OBQYEFB3UbsKQyVaYic5YZavk/d5uELBNMB8GA1Ud
IwQYMBaAFB3UbsKQyVaYic5YZavk/d5uELBNMAwGA1UdEwQFMAMBAf8wDQYJKoZI
hvcNAQELBQADgYEAuz1wMqJ0I0M8pDQzKYSbJxG0rNixOZtOQDWUUh3yIZqWQQA2
w6J2Plc5bSWtpvdKjuMafnHMYP0waTIFY8PbpPvkR6AiLaa2sLRkcWcaNGMpXD
IKodO6jEZFgmmyaC+xZiBoZNPmsir6UNsrNufAPIIXftiqheSijoludWq/U=
-----END CERTIFICATE-----
```

5. Restart the OPC UA server.



8.3 OPC UA address space

The OPC UA server captures the actual data from all clients that have been created via the BFC gateway user interface. The display is realized in the OPC UA address space.

To be able to call data from the OPC UA address space, a connection to the OPC UA server must be established.

The OPC UA server uses the following address:

opc.tcp://<IP address of the BFC gateway>:4840/BFCGateway

Example: opc.tcp://192.168.0.10:4840/BFCGateway

OPC UA address space structure

The OPC UA address space has the following structure:

BFCGateway

root

<Equipment Name 1>

<Client Name 1>

Online

Alarms

<Alarm 1>

<Alarm 2>

```

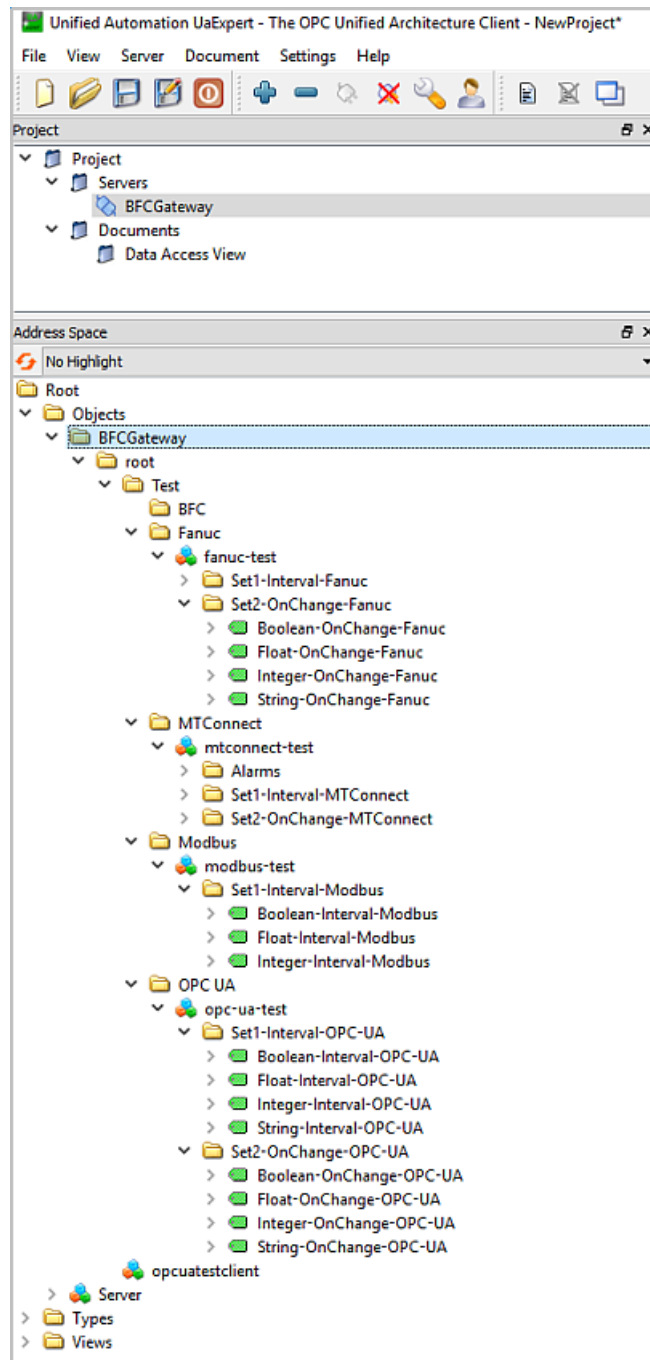
        .....
        <Readingset Name 1>
            <Variable 1>
            <Variable 2>
            <Variable 3>
        .....
        <ReadingSet Name 2>
            <Variable 1>
            <Variable 2>
            <Variable 3>
            <Variable 4>
        .....
        <Client Name 2>
        .....
    <Equipment Name 2>
    .....

```

Connection to the OPC UA client

The following diagram shows the connection between an OPC UA client and the BFC gateway OPC UA server.

The client data is displayed in the address space of the OPC UA server.



"BFCGateway" directory

The entry point in the OPC UA address space is the "BFCGateway" directory.

The "root" directory is in the "BFCGateway". The "root" directory contains all of the equipment and client data.

Each equipment can contain several clients. Every client can contain several readingsets.

The client data is structured just as it was configured in the BFC gateway user interface.

The basic structure of the address space is established when the OPC UA server starts for the first time.

All equipment and clients that were configured in the BFC gateway user interface are created in the address space.

Note

Readingsets

Readingsets and variables are only inserted in the address space once a configured client has sent at least one dataset.

After a client dataset has been received once, this structure is saved to a database, and directly restored the next time that the OPC UA server restarts.

The NodeIDs of already known elements, such as equipment, clients and variables remain constant. This means after a restart, equipment, clients and variables are always assigned the same NodeIDs.

The variable values are dynamically updated (refreshed) as soon as a new dataset was received from the data bus.

Online status variable

BFC clients regularly send a connection status on the data bus. As soon as the OPC UA server receives a valid connection status, an "online" variable is inserted in the OPC UA address space for the corresponding client. Boolean variable "Online" indicates the connection status of the client.

"Alarm" directory

An alarm directory is added to the OPC UA address space if the OPC UA server receives an event dataset from a configured client. In this case, all alarms from the event dataset are added to the alarm directory.

In addition, a standard OPC UA AlarmConditionState alarm is triggered; according to the OPC UA alarm specification, Part 9 "Alarms and Conditions". This can be evaluated by a connected OPC UA client if a subscription is created on the client or the "root" nodes. All alarms that were triggered from the clients are transferred to the "root" nodes so that an OPC UA client, which creates a subscription on the OPC UA "root" nodes, receives all of the client alarms of the system.

Dynamic updates of the OPC UA address space

If new clients or items of equipment are added via the BFC gateway user interface, the OPC UA server automatically attempts to insert these new elements in the OPC UA address space. Additional variables and reading sets are also directly assigned to the clients in the OPC UA address space as soon as an updated dataset has been received.

Deleted equipment and clients are directly removed from the OPC UA address space, including the associated elements such as reading sets and variables.

Note**Deleting**

Deleting reading sets or individual variables of a client already existing in the OPC UA address space in the BFC gateway user interface, means that the corresponding elements are not removed from the OPC UA address space. Instead, the last known status of the variables is kept, and is no longer updated (refreshed).

Identifying structure changes of the OPC UA address space

The OPC UA client is responsible for identifying changes to the structure in the OPC UA address space. To do this, the structure of the OPC UA address space must be regularly "browsed". Some OPC UA clients only identify changes when the connection to the server is reestablished.

Changes to variable values are automatically accepted.

8.4 Changing the system configuration while in operation

If there are changes to the system configuration during operation due to the creation or removal of clients and equipment, we recommend restarting the OPC UA server via the user interface of the BFC gateway.

Note**No restart**

If you only make changes to the client datasets during operation, you do not have to restart the OPC UA server.

Note**Planning configuration changes**

If changes are made to the system configuration while operational, the OPC UA server does not supply any data during this period.

- Plan any configuration changes with the company operating the system in order to prevent any potential data loss.
-

Requirement

You are in the user interface of the BFC gateway.

Procedure

1. In the "Commissioning" area, navigate to the "Gateways (Export)" area.
If the OPC UA server is running, stop the "OPC UA" service.
To do this, click on the "Stop Client" icon.



2. Restart the OPC UA server.
To do this, click on the "Start Client" icon.



8.5 Write access to variables

Client variables can be written to via the address space of the OPC UA server. Currently, the BFC client and the Modbus client support write access for variables.

The OPC UA server offers the following possibilities for writing to variables of the clients:

- Write method
- Direct access via the OPC UA variable

The variables that may be written to must be defined in release lists and stored in the configuration of the OPC UA server.

Information on the variables that are supported by the BFC client and may be written to can be found in Chapter: NC variables (Page 551).

In addition to OPC UA and the BFC client for SINUMERIK controls, the following BFC device drivers permit variables to be write accessed:

- ModbusTCP
- S7
- FANUC
- Heidenhain
- HTTP
- MQTT
- Beckhoff

You can find more information on the individual clients in Chapter: Operating the BFC gateway (Page 79).

8.5.1 Configuring the release list for write accesses

This chapter describes the configuration and activation of release lists for write accesses.

To be able to write to a client variable via the OPC UA server, you must store a release list in the OPC UA server.

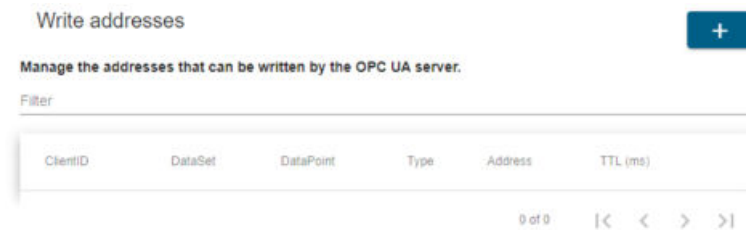
The release list is configured in the ConfigUI.

Procedure

1. Open the OPC UA server configuration view.



2. Configure the OPC UA release list.



3. Configure the OPC UA variable for write access.

Add write address

Type write clientid... *

Type write dataset... *

Type write datapoint... *

Select write type... *

Integer

Type write address... *

Type write TTL in ms... *

10000

Option	Description
Type write clientid	Client ID of the client that is permitted to make write accesses
Type write dataset	Name of the data set in which the variables to be written to are contained
Type write datapoint	Name of the variables in the data set
Select write type	Data type of the variable to be written to
Type write address	Address of the variable to be written to
Type write TTL in ms	Time-to-live value in milliseconds Specifies the maximum time that the execution of a write job can take

Example:

Add write address

opcuaclienttest

Type write dataset... *

Measurement1

Type write datapoint... *

Voltage

Select write type... *

Float

Type write address... *

ns=2,i=5873

Type write TTL in ms... *

10000



4. Save the variable.

Add write address

opcuaclienttest

Type write dataset... *

Measurement1

Type write datapoint... *

Voltage

Select write type... *

Float

Type write address... *

ns=2;i=5873

Type write TTL in ms... *

10000

Save Cancel

5. Complete the list.

Write addresses

Manage the addresses that can be written by the OPC UA server.

Filter

ClientID	DataSet	DataPoint	Type	Address	TTL (ms)	
opcuaclienttest	Measurement1	Voltage	float	ns=2;i=5873	10000	

1 - 1 of 1 |< < > >|

Repeat steps 1 to 4 until all variables for the write access have been saved.

8.5.2 Writing variables using the "WriteData" method

For all clients with configured release list for write operations, an OPC UA method called "WriteData" is created after the OPC UA server restarts.

Procedure

Each "WriteData" method is assigned to exactly one client.

After calling the method, the write operation is triggered for the associated client.

For all clients with configured release list for write operations, method "WriteData" is created after the OPC UA server starts for the first time.

The NodeId of the method is automatically generated and stored in a database. As long as the client exists in the system, the address of the write method remains unchanged.

Structure

The OPC UA address space has the following structure:

BFCGateway
- root

```

- <Equipment name 1>
- <Client name 1>
- WriteData
- Online
- Alarms
- <Alarm message 1>
- <Alarm message 2>
- .....
- <ReadingSet name 1>
- <Value 1>
- <Value 2>

```

8.5.3 Call method "WriteData"

The "WriteData" method can be called by any OPC UA client that supports the OPC UA methods.

Note

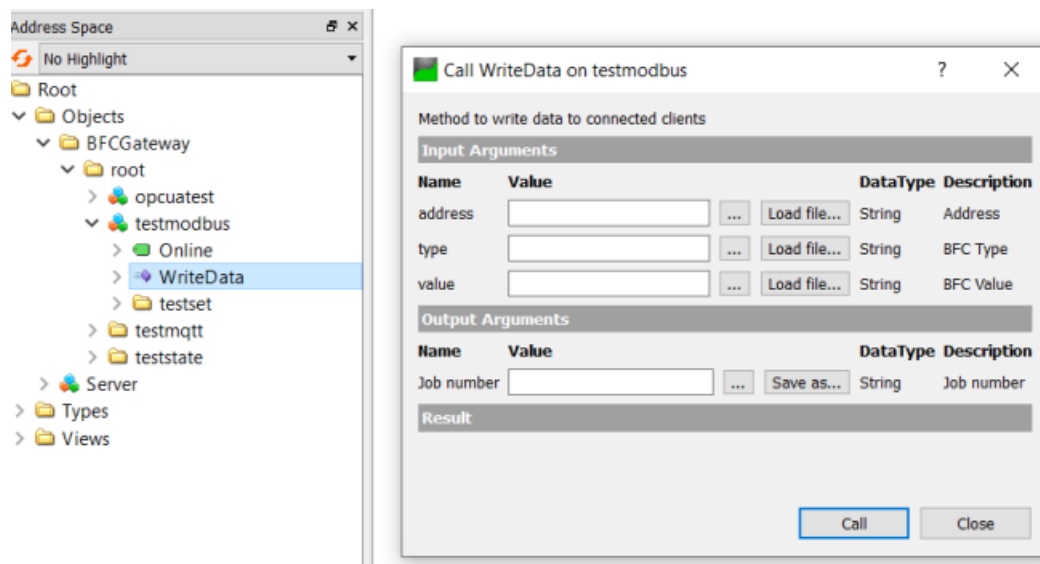
Specify values

Enter the following values in precisely the same way as they were defined in ConfigUI:

- address > ConfigUI "Address"
- type > ConfigUI "Type"

This is a check to determine whether the address in the target system has been released for write access.

The following example shows how the "WriteData" methods are called with the Unified Automation OPC UA client.



Parameter

Parameter	Description
address	Address of the variable to be written to
type	Data type of the variable to be written to
value	Value that should be written

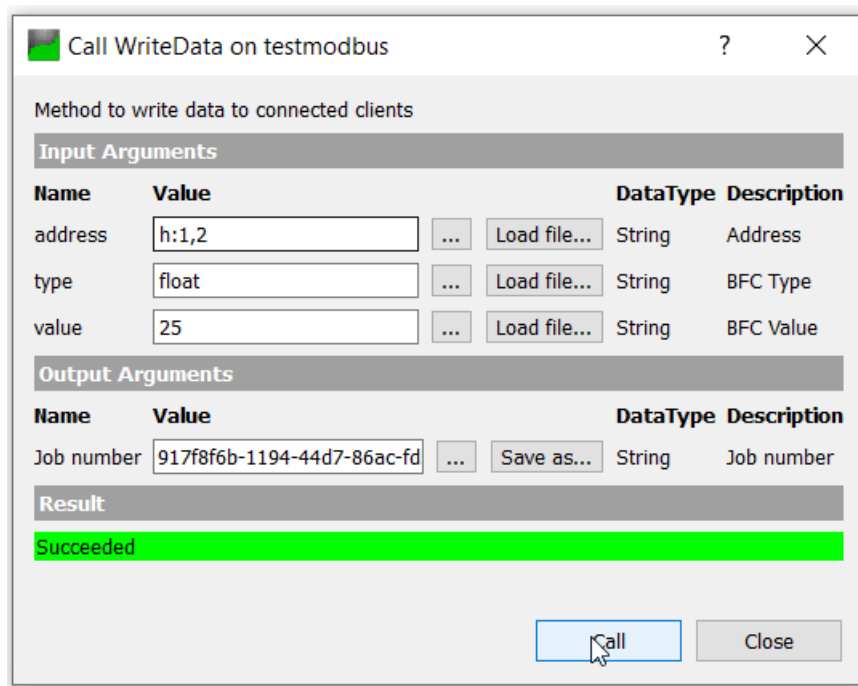
Return parameters	Description
Job number	Unique ID that identifies the write job. The ID is only set if the write job was successfully sent. Otherwise an empty "Job number" is returned.

Return value

After the OPC UA method "WriteData" has been called, a return value with a status code is returned to the calling OPC UA client. If the method is successfully called, the return parameter "Job number" is also set.

Status code	Description
BadInvalidArgument	Too few parameters have been transferred, the parameters are empty, or the parameters were specified in the wrong format.
BadNotWritable	The variable cannot be written to because the parameters differ from the data configured in the release list for write operations.
BadMethodInvalid	The write method could not be executed.
Good (Succeeded)	The write job was successfully sent.

The following example shows a successful call with the Unified Automation OPC UA client.



8.5.4 Feedback after write process

The status code that is returned to the OPC UA client when calling only informs the client that the write job has been received.

The status code does not indicate that the write job has been executed.

When the write job has been executed, the OPC UA server sends an event of the type "BaseEventType" to signal the OPC UA client that the write job has been executed.

Requirement

To process the feedback, the connected OPC UA client must create a subscription to the client node.

- OR -

To receive feedback messages from all clients, the connected OPC UA client must create a subscription to the root node of the OPC UA address space.

Parameter

Parameter	Description
id	ID of the write job. This ID corresponds to the ID that was returned after the method was called.
timestamp	Time stamp of the message.

Parameter	Description
ttlms	Time to live value in milliseconds. Specifies the maximum time allowed for the write job to be executed.
status	Status of the write job: "ok" if the write job was successfully completed. Otherwise an error code is stored.
data	A state is returned for each variable written.
data.address	Address of the written variables
data.type	Data type of the written variables
data.value	Current value of the written variables
data.status	Status of the write job: "ok", if the variables were executed successfully.

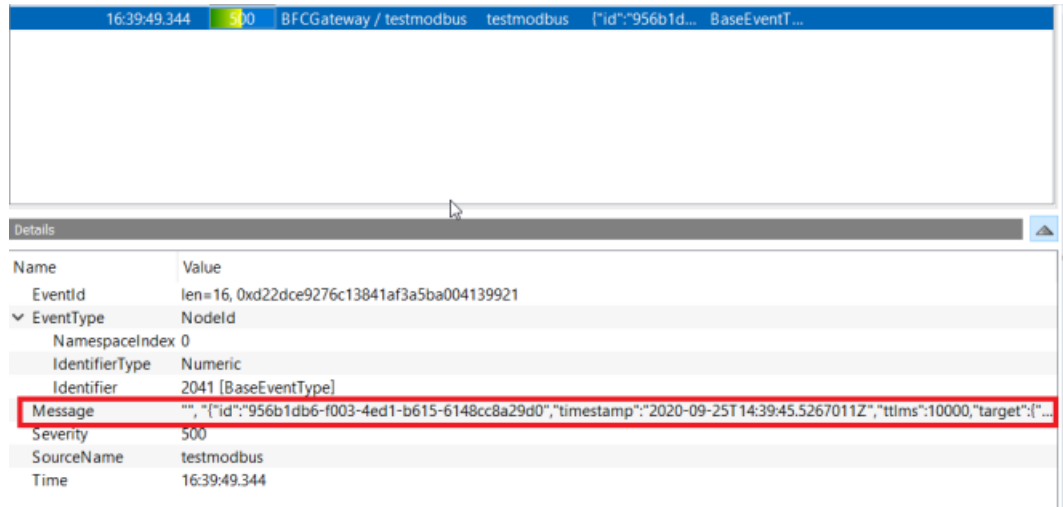
Procedure

In the "Message" element of the event you can see detailed information about the executed write job:

```
"id" : "....."  
"timestamp": "2020-05-07T08:50:12.420Z",  
"ttlms": 1000,  
"status": "ok",  
"data": [  
  {  
    "address": "/NC/SOME/STRING/VARIABLE",  
    "type": "string",  
    "value": "stringvaluegoeshere",  
    "status": "ok"  
  },  
  ...  
]
```

Example

The following example shows an OPC UA event as feedback after the write process.



8.5.5 Direct write access via variables

The variables stored in the release list for write operations can also be written to directly. For this purpose, a write access to the corresponding variable must be made.

Parameter

Status code	Description
Good	The write job was successfully sent.
Bad	The write job could not be executed because the write enable for the variable is missing.
BadMethodInvalid	The write job could be transferred to the system.

Procedure

The feedback after execution of the write job is also given when accessing via the variables. More information can be found in Chapter: Feedback after write process (Page 374).

8.6 Historical Access

The OPC UA server supports function "Historical Access", in the same way that it is described in the OPC UA specification, Part 11.

Using this function, not only can the actual values of a variable be read, but also previous values (historical values). To do this, the OPC UA server accesses an internal database of the BFC gateway in which the data is stored.

An "Internal InfluxDB" gateway must be used when using function "Historical Access". The name of the InfluxDB database must be "bfcdatabase". Retention Policy "sixmonths" must be used.

Note

Only variables from reading sets are supported. Historical alarms or high-frequency data cannot be read using the "Historical Access" feature.

The OPC UA server supports the following functions of the OPC UA specification, Section 11 "Historical Access":

- Basic functionality to call up raw data (<http://opcfoundation.org/UA-Profile/Server/HistoricalRawData>)

Every variable in the OPC UA address space supports "Historical Access". You can identify this property at the "HistoryRead" attribute.

The display of the Unified Automation OPC UA Client is shown as example in the following.

AccessLevel	CurrentRead, CurrentWrite, HistoryRead
UserAccessLevel	CurrentRead, CurrentWrite, HistoryRead

An OPC UA client can read historical data from the server if the OPC UA client supports the "Historical Access" profile (<http://opcfoundation.org/UA-Profile/Client/HistoricalAccess>).

When an OPC UA client reads historical data, then the OPC UA server accesses an internal database and sends the requested data to the OPC UA client.

When doing this, the following return values can occur:

Return value	Description
Good	The data call was successful
GoodNoData	No historical data are stored in the database
BadDataUnavailable	The database cannot be accessed or is not operational (the App infrastructure of the BFC gateway has not been started.)
BadUnexpectedError	An error occurred when calling the data
BadNoData	The "InfluxDB" database cannot be addressed

Data on the time stamp

When an OPC UA client reads data from the server, all time stamps must be available in the UTC format.

Continuation Points

If an OPC UA client calls up data from the server, only 100 datasets are provided from the database (this is always the case).

If additional data exist in the requested interval, then the OPC UA server creates a "Continuation Point" to signal the client that there is still more data to be read.

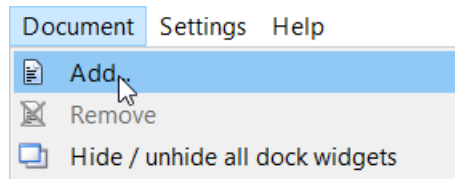
The next time the reading method is called, the client can use the "Continuation Point" to continue to read data at the location where the previous call was canceled.

More information on this is provided in the OPC UA specification, Section 11 "Historical Access" in Chapter "Continuation Points".

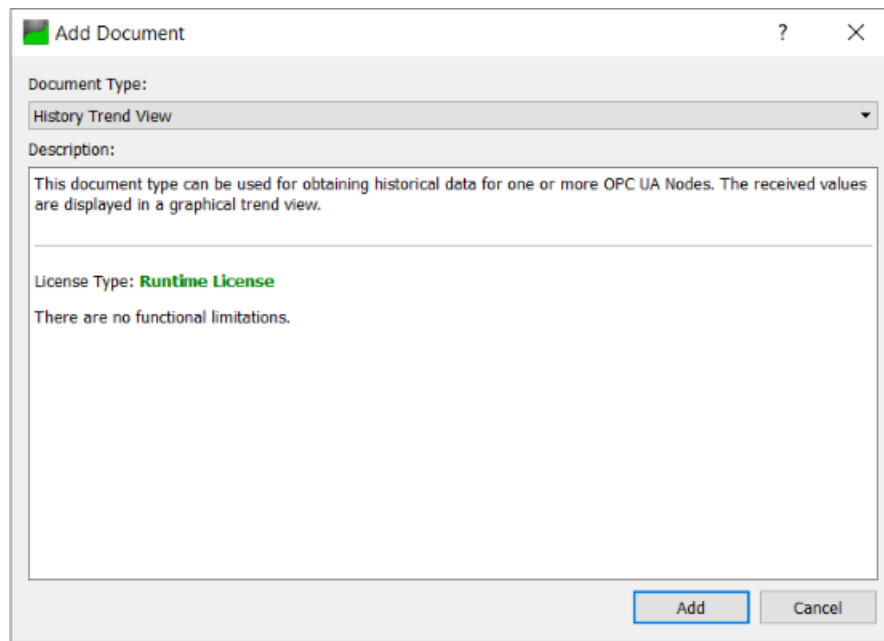
Example: Data call with the Unified Automation UA Expert OPC UA client

Historical data is called from the OPC UA server using the Unified Expert OPC UA client.

1. Using the Unified Automation UA Expert OPC UA client, establish a connection to the OPC UA server.
2. The individual variables are displayed in the address space of the OPC UA server.
3. Add a new view for the historical data using "Document" > "Add".



4. Under "Document Type", select value "History Trend View".



5. In the document view "History Trend View", insert a variable from the OPC UA address space.

The screenshot shows the 'History Trend View' configuration window. The 'Configuration' tab is active, showing a table with columns: Server, DisplayName, Node Id, and Color. The table contains one entry: 'UaServerC...' with 'ByteWithHi...' as the DisplayName, 'NS4[String]...' as the Node Id, and 'cyan' as the Color. To the right of the table are two radio buttons: 'Single Update' (selected) and 'Cyclic Update'. Below 'Single Update' are 'Start Time' (26.10.20 09:27:34.906) and 'End Time' (05.11.20 20:00:00.000) dropdowns, with 'Get Start Time' and 'Update' buttons. Below 'Cyclic Update' are 'Timespan' (00:01:00) and 'Update Interval' (00:00:01) dropdowns, with a 'Start' button.

Below the configuration window is the 'History Data' table. It has a tab labeled 'Numeric Values' and a dropdown menu showing 'NS4[String]Demo.History.ByteWithHistory'. The table has four columns: Source Timestamp, Server Timestamp, Value, and Status. It contains 20 rows of data, all with a 'Good' status.

Source Timestamp	Server Timestamp	Value	Status
2020-10-27 17:46:30.052	2020-10-27 17:46:30.052	173	Good (0x00000000)
2020-10-27 17:46:29.551	2020-10-27 17:46:29.551	179	Good (0x00000000)
2020-10-27 17:46:29.053	2020-10-27 17:46:29.053	171	Good (0x00000000)
2020-10-27 17:46:28.550	2020-10-27 17:46:28.550	163	Good (0x00000000)
2020-10-27 17:46:28.048	2020-10-27 17:46:28.048	155	Good (0x00000000)
2020-10-27 17:46:27.552	2020-10-27 17:46:27.552	147	Good (0x00000000)
2020-10-27 17:46:27.053	2020-10-27 17:46:27.053	139	Good (0x00000000)
2020-10-27 17:46:26.549	2020-10-27 17:46:26.549	131	Good (0x00000000)
2020-10-27 17:46:26.051	2020-10-27 17:46:26.051	123	Good (0x00000000)
2020-10-27 17:46:25.551	2020-10-27 17:46:25.551	115	Good (0x00000000)
2020-10-27 17:46:25.053	2020-10-27 17:46:25.053	107	Good (0x00000000)
2020-10-27 17:46:24.553	2020-10-27 17:46:24.553	99	Good (0x00000000)
2020-10-27 17:46:24.049	2020-10-27 17:46:24.049	91	Good (0x00000000)
2020-10-27 17:46:23.552	2020-10-27 17:46:23.552	83	Good (0x00000000)
2020-10-27 17:46:23.052	2020-10-27 17:46:23.052	75	Good (0x00000000)
2020-10-27 17:46:22.549	2020-10-27 17:46:22.549	67	Good (0x00000000)
2020-10-27 17:46:22.051	2020-10-27 17:46:22.051	59	Good (0x00000000)
2020-10-27 17:46:21.553	2020-10-27 17:46:21.553	51	Good (0x00000000)
2020-10-27 17:46:21.052	2020-10-27 17:46:21.052	43	Good (0x00000000)

6. Click on button "Get Start Time" to initialize the data call.
7. Click on button "Update" to call the data.
8. All historical data are listed in table "History Data" that are saved for the configured variable.

8.7 Companion Specification

Introduction

OPC UA supports the use of information models from Companion Specifications. The information models are stored in XML files and can be processed by the OPC UA server. The OPC UA server inserts all elements from the information model into the address space.

Data is allocated using a script logic adapted to the information model.

Companion Specification information model

The OPC UA server supports information models that are based on the rules of the OPC UA specification, Section 3 "Address Space Model".

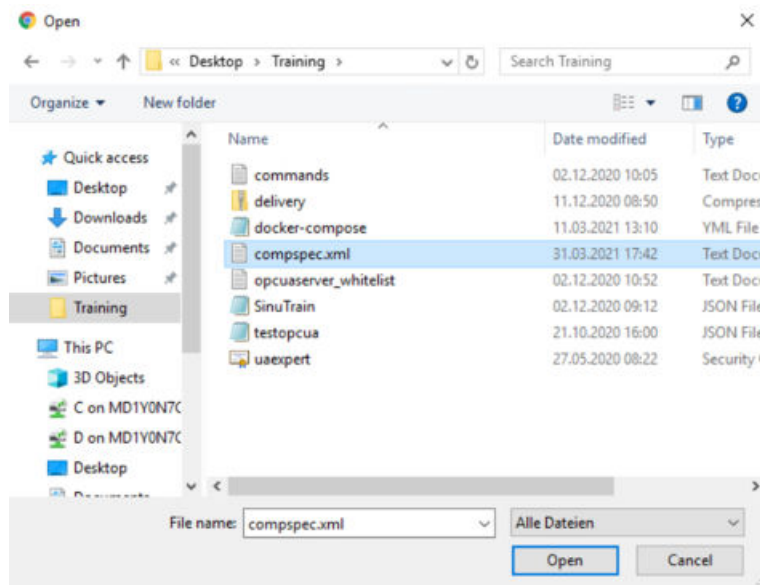
Models that were created using the "Siemens SIOME" ([https://support.industry.siemens.com/cs/document/109755133/siemens-opc-ua-modeling-editor-\(siome\)-for-implementing-opc-ua-companion-specifications?dti=0&lc=en-WW](https://support.industry.siemens.com/cs/document/109755133/siemens-opc-ua-modeling-editor-(siome)-for-implementing-opc-ua-companion-specifications?dti=0&lc=en-WW)) software or the Unified Automation UA Modeler are supported.

Preparing the OPC UA server for using a Companion Specification

1. Configure the Companion Specification in the ConfigUI in the configuration view of the OPC UA server.



2. Load the specification files to the BFC Gateway via field "Companion Spec".
3. In the dialog, select a valid Companion Spec file.



4. Restart the OPC UA server once the Companion Specification has been successfully loaded to the BFC gateway.

If a connection is established to the OPC UA server, then the new information model is displayed in the OPC UA address space. The structure and number of elements in the OPC UA address space are specified by the information model being used.

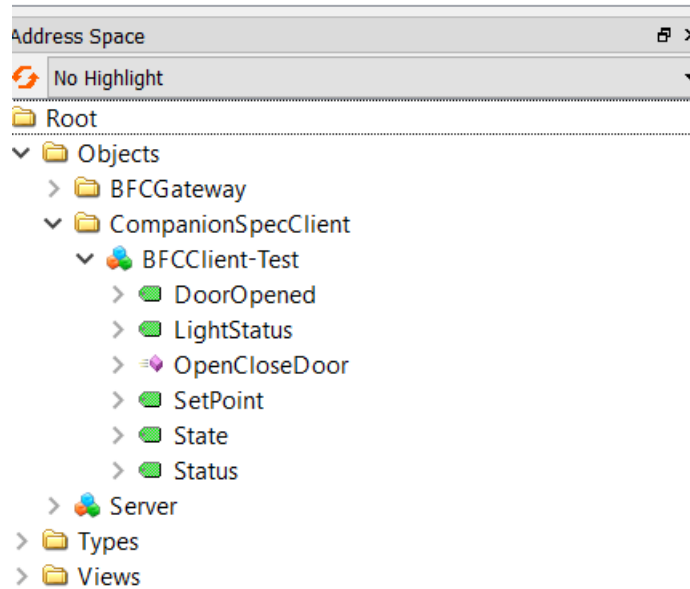
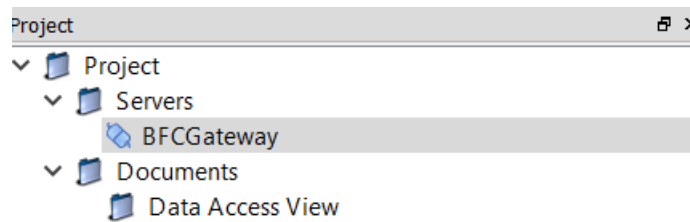
Loading the Companion Specification is also documented in the log file of the OPC UA server.

Example:

```
2021-01-12T11:01:52.778+01:00 [INFO] Companion spec option set.
Adding companion spec nodes.
2021-01-12T11:01:52.780+01:00 [INFO] Companion spec nodeset file:
BFCGatewayOpcUaServer/Testset.NodeSet2.xml
```

Address space:

The following diagram shows the address space of the OPC UA server with a loaded Companion Specification "CompanionSpecClient".



If the content of the information model is not displayed in the address space, then check the log file of the OPC UA server. Also check the information model in an OPC UA modeler software regarding its validity.

Note

If the information model references other submodels, then you must also copy the XML files of the submodels to the BFC gateway. Copy the files into the main model directory.

It is not permissible that the Companion Specification uses namespace "http://siemens.com/BFCGateway". This namespace is used for the static elements in directory "BFCGateway" of the OPC UA server.

The person that created the Companion Specification data model must ensure that the data model does not have any errors, such as nodes with duplicated NodeIDs or other structural errors. This can result in errors in the OPC UA server. When loading the data model, the OPC UA server cannot identify all defects in the Companion Specification data model.

Data types of a Companion Specification that are supported

The OPC UA server supports the following OPC UA data types:

- Int16, Int32, Int64
- UInt16, UInt32, UInt64
- Boolean
- Float, Double
- Byte, SByte
- String, ByteString, LocalizedText
- QualifiedName
- Guid
- DateTime
- Enumeration

Restrictions when using a Companion Specification

- The OPC UA server does not support array and matrix variables. Although these elements are displayed in the address space, data cannot be written to.
- If the Companion Specification uses alarms and events, then only OPC UA "BaseEventType" and "CncAlarmType" are supported.
- If user-defined data types are used, then only standard OPC UA data types as basic classes can be used.
- If user-defined, complex data types are used, then only "EnumFields" are supported. Structure, StructureWithOptionalFields, Union and StructureField data types are not supported.

Data mapping for a Companion Specification

The information model from a Companion Specification can contain any data and structures.

This is the reason that a connection must be established between the data of a BFC gateway reading set and the elements of the information model.

To do this, the OPC UA server uses the mapping objects from the reading set of a BFC client. These optional objects refer to a system-wide, unique address for a datum (also called NodeID) from the information model in the companion specification.

The OPC UA server assigns the value from the reading set of the stored NodeID.

Reading set example:

```
{
  "name" : "<Name of readingset>",
  "time": "<Timestamp of data>",
  "data" :
  [
    {
      "mappings" : {
```

```
      "opcua" : {
        "nodeid" : "<Nodeid to map, e.g. ns=5; i=1234>",
      },
    },
    "name" : "<Name of variable>",
    "time" : "<Individual timestamp of value>",
    "value" : true,
    "type" : "<BFC datatype, e.g. bool>"
  }
]
}
```

Notes relating to mapping objects:

- mappings: Mapping information
- opcua: Mapping information for the OPC UA server
- nodeid: Destination NodeID - where the value from the reading set is mapped to

Event set example:

```
{
  "name" : "<Name of eventset>",
  "time" : "<Timestamp of event>",
  "events" :
  [
    {
      "mappings" : {
        "opcua" : {
          "nodeid" : "<Nodeid of event source to map, e.g. ns=5;
i=1234>",
        },
      },
      "id" : "<Event id>",
      "source" : "<Event source>",
      "text" : "<Event text>",
      "state" : true,
      "timestamp" : "<Event timestamp>"
    }
  ]
}
```

Notes relating to mapping objects:

- mappings: Mapping information
- opcua: Mapping information for the OPC UA server
- nodeid: Destination-event-source node ID (a system-wide, unique address for a datum in the companion specification, e.g. a data source or a data destination).

If the OPC UA server receives a reading set with mapping information, then the OPC UA nodes from the information model are added to the values.

The log file of the OPC UA server contains information about the mapping that was performed:

```
2021-01-12T11:01:56.492+01:00 [INFO] Setting value of companion
target ns=5;i=15045 TestString
```

```

2021-01-12T11:01:56.496+01:00 [INFO] Setting value of companion
target ns=5;i=15080 TestInt16
2021-01-12T11:01:56.496+01:00 [INFO] Setting value of companion
target ns=5;i=15081 TestInt32
2021-01-12T11:01:56.497+01:00 [INFO] Setting value of companion
target ns=5;i=15082 TestInt64
2021-01-12T11:01:56.498+01:00 [INFO] Setting value of companion
target ns=5;i=15083 TestUInt16
2021-01-12T11:01:56.499+01:00 [INFO] Setting value of companion
target ns=5;i=15084 TestUInt32
2021-01-12T11:01:56.500+01:00 [INFO] Setting value of companion
target ns=5;i=15085 TestUInt64
2021-01-12T11:01:56.500+01:00 [INFO] Setting value of companion
target ns=5;i=15085 TestUInt64
2021-01-12T11:01:56.500+01:00 [INFO] Setting value of companion
target ns=5;i=15048 TestBool
2021-01-12T11:01:56.501+01:00 [INFO] Setting value of companion
target ns=5;i=15039 TestDouble
2021-01-12T11:01:56.502+01:00 [INFO] Setting value of companion
target ns=5;i=15090 TestFloat
2021-01-12T11:01:56.504+01:00 [INFO] Setting value of companion
target ns=5;i=15086 TestByte
2021-01-12T11:01:56.505+01:00 [INFO] Setting value of companion
target ns=5;i=15087 TestSByte
2021-01-12T11:01:56.505+01:00 [INFO] Setting value of companion
target ns=5;i=15088 TestLocalizedText
2021-01-12T11:01:56.506+01:00 [INFO] Setting value of companion
target ns=5;i=15089 TestQualifiedName
2021-01-12T11:01:56.507+01:00 [INFO] Setting value of companion
target ns=5;i=15095 TestGuid
2021-01-12T11:01:56.507+01:00 [INFO] Setting value of companion
target ns=5;i=15096 TestDateTime
2021-01-12T11:01:56.508+01:00 [INFO] Setting value of companion
target ns=5;i=15037 TestEnumeration
2021-01-12T11:01:56.509+01:00 [ERROR] Target node ns=5;i=15100 is an
array or matrix. Could not update value !!

```

An error is output in the log file if a variable cannot be mapped in the model.

Example: Script logic with mapping objects

One option of inserting mapping objects into the reading sets is to use a script logic.

The script logic subsequently shown integrates a mapping object for each variable. You can use the script logic and adapt it to suit your own applications.

```

////////////////////////////////////
/////
/
//
// CompanionMapper.js
//
// BFC Gateway Scriptlogic script to map data to a companion spec

```

```
model
//
//
////////////////////////////////////
////

var testString = "Hello World !!";
var testBool = false;
var testByte = 0;
var testDateTime = "2021-01-01T00:00:00Z";
var testDouble = 5.0;
var testFloat = 1.0;
var testGuid = "936DA01F-9ABD-4D9D-80C7-02AF85C822A8";
var testInt = 0;
var testLocalizedText = "Hello World !!";
var testQualifiedName = "Qualified Name";
var testSByte = 1;
var testEnumeration = -1;

// This function will be called for every received ReadingSet
function onNewReadingSet(state, set) {
    testBool = !testBool;
    testByte = 1;
    testDouble = testDouble + 1;
    testFloat = testFloat + 1;
    testInt = testInt + 1;
    testEnumeration = testEnumeration + 1;
    if (testEnumeration > 2)
    {
        testEnumeration = 0;
    }
    // Create set
    var nowDate = new Date();

    var setInfo = {
        "Source": {
            "ProjectID": set.Source.ProjectID,
            "DeviceType": set.Source.DeviceType,
            "ApplicationType": set.Source.ApplicationType,
            "ClientID": set.Source.ClientID
        },
        "Set": {
            "Time": nowDate.toISOString(),
            "Name": "TestMapping"
        }
    };

    var v = [{
        "Name": "TestString",
        "Type": "string",
        "Value": testString,
        "mappings": {
```

```
        "opcua": {
            "nodeid": "ns=5;i=15045"
        }
    },
    {
        "Name": "TestInt16",
        "Type": "int",
        "Value": testInt,
        "mappings": {
            "opcua": {
                "nodeid": "ns=5;i=15080"
            }
        }
    },
    {
        "Name": "TestInt32",
        "Type": "int",
        "Value": testInt,
        "mappings": {
            "opcua": {
                "nodeid": "ns=5;i=15081"
            }
        }
    },
    {
        "Name": "TestInt64",
        "Type": "int",
        "Value": testInt,
        "mappings": {
            "opcua": {
                "nodeid": "ns=5;i=15082"
            }
        }
    },
    {
        "Name": "TestUInt16",
        "Type": "int",
        "Value": testInt,
        "mappings": {
            "opcua": {
                "nodeid": "ns=5;i=15083"
            }
        }
    },
    {
        "Name": "TestUInt32",
        "Type": "int",
        "Value": testInt,
        "mappings": {
            "opcua": {
                "nodeid": "ns=5;i=15084"
            }
        }
    }
}
```

```
    }
  },
  {
    "Name": "TestUInt64",
    "Type": "int",
    "Value": testInt,
    "mappings":{
      "opcua":{
        "nodeid":"ns=5;i=15085"
      }
    }
  },
  {
    "Name": "TestBool",
    "Type": "bool",
    "Value": testBool,
    "mappings":{
      "opcua":{
        "nodeid":"ns=5;i=15048"
      }
    }
  },
  {
    "Name": "TestDouble",
    "Type": "float",
    "Value": testDouble,
    "mappings":{
      "opcua":{
        "nodeid":"ns=5;i=15039"
      }
    }
  },
  {
    "Name": "TestFloat",
    "Type": "float",
    "Value": testFloat,
    "mappings":{
      "opcua":{
        "nodeid":"ns=5;i=15090"
      }
    }
  },
  {
    "Name": "TestByte",
    "Type": "int",
    "Value": testByte,
    "mappings":{
      "opcua":{
        "nodeid":"ns=5;i=15086"
      }
    }
  }
}
```

```

    },
    {
      "Name": "TestSByte",
      "Type": "int",
      "Value": testSByte,
      "mappings": {
        "opcua": {
          "nodeid": "ns=5;i=15087"
        }
      }
    },
    {
      "Name": "TestLocalizedText",
      "Type": "string",
      "Value": testLocalizedText,
      "mappings": {
        "opcua": {
          "nodeid": "ns=5;i=15088"
        }
      }
    },
    {
      "Name": "TestQualifiedName",
      "Type": "string",
      "Value": testQualifiedName,
      "mappings": {
        "opcua": {
          "nodeid": "ns=5;i=15089"
        }
      }
    },
    {
      "Name": "TestGuid",
      "Type": "string",
      "Value": testGuid,
      "mappings": {
        "opcua": {
          "nodeid": "ns=5;i=15095"
        }
      }
    },
    {
      "Name": "TestDateTime",
      "Type": "string",
      "Value": testDateTime,
      "mappings": {
        "opcua": {
          "nodeid": "ns=5;i=15096"
        }
      }
    },
    {

```



```

    "Name": "TestEnumeration",
    "Type": "int",
    "Value": testEnumeration,
    "mappings":{
      "opcua":{
        "nodeid":"ns=5;i=15037"
      }
    }
  },
  {
    "Name": "TestArrayNotWorking",
    "Type": "int",
    "Value": testInt,
    "mappings":{
      "opcua":{
        "nodeid":"ns=5;i=15100"
      }
    }
  }
}]

setInfo.Set["Data"] = v;
console.log("\nCompanionMapper.js - publishing:",
JSON.stringify(setInfo));

// Publish machine state
publishReadingSet(setInfo);
}

```

8.8 Other OPC UA server settings

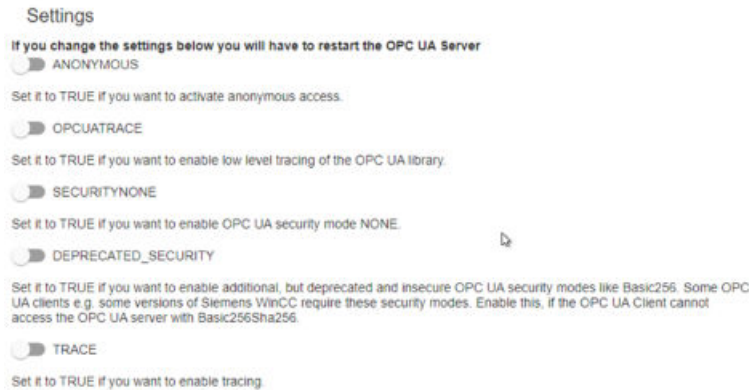
In the ConfigUI in the OPC UA server configuration view, you can set additional OPC UA server options, e.g. you can enable safety profiles.

Procedure

1. Open the OPC UA server configuration view.



2. Set the individual settings according to your specific requirements.



Setting	Description
ANONYMOUS	Activates an "Anonymous" profile in the OPC UA server, which facilitates access without requiring a user name and password.
OPCUATRACE	Activates the output of extended messages of the OPC UA stack in the log file.
SECURITYNONE	Activates the connection profile "None" in the OPC UA server, which allows a connection to be established to the OPC UA server without encryption.
DEPRECATED_SECURITY	Activates the connection profiles "Basic256" and "Basic128Rsa15" in the OPC UA server. These connection profiles are out of date and not secure. Only use these profiles if older OPC UA clients require these profiles to establish a connection.
TRACE	Activates the output of trace messages of the OPC UA server in the log file.

3. Restart the OPC UA server.
Changes to the OPC UA server settings are only adopted after a restart.



Client interface with HTTP REST protocol

9.1 Overview

Fundamentals

The BFC gateway includes a southbound interface, to which clients on the machine side can send data and retrieve write jobs using the HTTP protocol. Data that is sent is then processed. After being successfully executed, write jobs must be confirmed.

For this purpose, a service is implemented in the BFC gateway that comprises the REST Input Protocol (RIP), currently with 4 different data formats to call up write jobs:

- Data sets whose data are encoded into a fixed array data structure for a point in time as JSON
- Events that process an event or alarm list as a predefined data structure as JSON
- HFDatasets, which contain data in a predefined table structure for several consecutive points in time
- Raw data in XML, JSON, plain text or HTML form format without a predefined structure, which is processed into a data set with the raw data as a string data point.
- Write jobs are retrieved and confirmed as JSON-coded structure

The service has the start address:

- "http://{BFC-Gateway}:9876/rip-service/api (<http://{BFC-Gateway}:9876/rip-service/api//XmlEditoAdresse>)"
- or
- "https://{BFC-Gateway}:9877/rip-service/api (<http://{BFC-Gateway}:9877/rip-service/api>)".

This address must be accessible from the HTTP client applications.

If you use TLS or HTTPS, the issuer certificate of the server certificate must be known at the HTTPS client.

The HTTP client application must check the server certificate for validity.

9.2 Test and documentation

The HTTP REST service provides an integrated web page based on Swagger and Open-API 2.0. A registration is not necessary.

Areas and parameters

The test and documentation page (/rip-service/api/help) is divided into the following sections:

- Short description with download link for the interface definition as file "open-api.json"
- Login button "Authorize"

- 3 sections with the addresses of the following methods:
 - Datasets
 - Events
 - HFDataSets
 - WriteJobs
- Several data structure models:
 - AlarmList
 - DataSet
 - HFDataSet
 - Models for WriteJobs

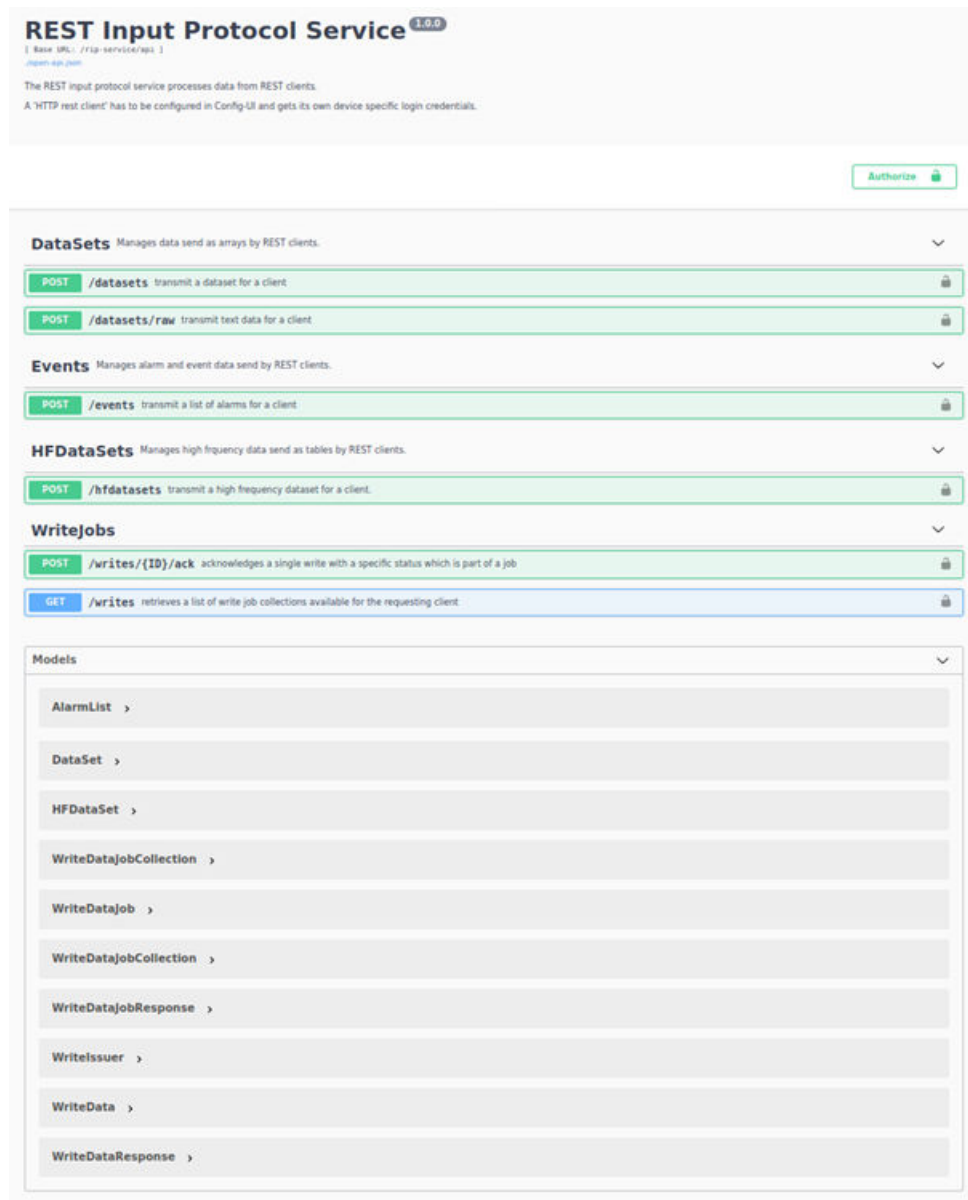
With the interface description file "open-api.json" you can generate clients in different programming languages with "Swagger Codegen".

REST Input Protocol Service for the API

Call the web page under the following address:

"https://{BFC-Gateway}:9877/rip-service/api/swagger-ui/ (<https://{BFC-Gateway}:9877/rip-service/api/swagger-ui/>)".

The "REST Input Protocol Service" web page opens:



9.3 Authorization

To execute the methods interactively, you must log on with the login credentials (Key and Secret) of an HTTP REST Client entry from the configuration interface.

The individual methods can now be executed with the stored login credentials. The login credentials determine to which machine the sent information belongs. Multiple HTTP client applications may also send data in parallel for the same machine, e.g. an automatic data acquisition and a user interface.

Requirement

If you use bearer tokens, you must first create them using the User Service API.

Creating bearer tokens

For example, a bearer token can be retrieved with the following "curl command".

Specify this command as a single line:

```
curl -s http://$BFC_HOSTS:9876/users/api/oauth/token -u$KEY:$SECRET  
--data-urlencode grant_type=client_credentials --data-urlencode  
client_id=$KEY --data-urlencode client_secret=$SECRET
```

Replace the variables \$BFC_HOST, \$KEY and \$SECRET with the appropriate values of the target system.

Example:

Below is an example of a complete HTTP request with header and request body at TCP level:

```
POST /users/api/oauth/token HTTP/1.1  
Host: 127.0.0.1:9876  
Authorization: Basic ZD[...]==  
User-Agent: curl/7.64.0  
Accept: */*  
Content-Type: application/x-www-form-urlencoded  
  
grant_type=client_credentials&client_id=d8[...]ae&client_secret=0b2d[...]  
]183e
```

If you call the client service with valid login credentials, you receive a bearer access token embedded in a JSON structure as a response:

```
{"access_token":"eyJhbG[...]","expires_in":7200,"token_type":"Bearer"}
```

Note

Temporary validity

The token has only a limited validity. Please note the validity period and generate a new token in time.

Method calls with expired bearer token are rejected by the HTTP REST service (HTTP status code: 401). Pay particular attention to this when implementing the program.

Procedure

1. Click on the "Authorization" button.
2. The "Available authorizations" window opens.
In the window area "Basic authorization":
 - In the "Username:" input field, enter the login data "key" of the HTTP REST client.
 - In the "Password" input field, enter the login data "secret" of the HTTP REST client.
 - Click "Authorize".
 - OR -
 - In the window area "BearerAuth (apiKey)":
 - Enter the bearer token in the form "Bearer eyXYZ...".
3. Click the "Authorize" button.
The selected login mode and login credentials in the active browser session are used to access the methods.
4. Click the "Close" button to close the window.

Available authorizations

Basic authorization

Get credentials from Config-UI. Use 'key' as username and 'secret' as password.

Username:

Password:

Authorize

Close

BearerAuth (apiKey)

Get an access-token by POST call to method 'oauth/token' of [user service](#).
You must enter 'Bearer' and a single space in front of the actual 'access_token'.

Name: Authorization

In: header

Value:

Bearer eyJhbGciOiJSUzUxMiI

Authorize

Close

9.4 Format for time stamp

At the interface, time stamps are encoded as text in the form according to ISO 8601 relative to the UTC time zone.

Format: "YYYY-MM-DDThh:mm:ss.fffZ"

In HTTP REST client applications, the conversion from local time to UTC time zone must be implemented.

Parameter

Designation	Description
YYYY	Year, 4-digit specification
MM	Month, 2-digit specification
DD	Day, 2-digit specification
hh	Hour, 2-digit specification
ss	Second, 2-digit specification
fff	Fractions of a second
Z	UTC time zone

9.5 Method POST /datasets

Parameter

Area	Description
datapoints	The area can contain several data points.
timestamp	Formatting of the time stamp is specified on the test interface (/rip-service/api/help) with the current time. The time stamps have the UTC time as time zone. The time stamp is not checked for plausibility by the system, the data can also be reported subsequently. Pay attention to the chronologically correct order so that target systems (e.g. MindSphere or AMP) can process the data correctly.

Procedure

1. Open the method "POST /datasets"
2. Click the "Try it out" button.
The text box opens for editing, e.g.:

```
{
  "set": {
    "datapoints": [
      {
        "name": "myNumber",
        "type": "int",
        "value": 1
      }
    ],
    "name": "myDataset",
    "timestamp": "{timestamp}"
  }
}
```

Make sure that the JSON structure is syntactically correct.

3. To send the JSON structure to the HTTP REST service, click the "Execute" button.
An equivalent curl command line and the return value of the call are output as result.
Note the "DataSet" data model on the test web page (/rip-service/api/help).

POST /datasets transmit a dataset for a client

Parameters Try it out

Name	Description
body	Example Value Model
object (body)	<pre>{ "set": { "datapoints": [{ "name": "string", "type": "int" }], "name": "string", "timestamp": "2020-08-26T07:34:17.962Z" } }</pre>

Parameter content type
application/json

Responses
Response content type application/json

9.6 Method POST /datasets/raw

The test execution procedure is similar to the method "POST /datasets". However, this method gives you more freedom with regard to the data sent. The data can be evaluated and further processed in the BFC using a script logic.

The received text is stored in JSON with replaced special characters for line feed, quotation marks, etc. A specification of the time stamp is not possible. The time of the BFC gateway is always used.

Overview

The following formats/MIME data types are allowed:

- text/plain
- application/json
- application/xml
- application/x-www-form-urlencoded

MIME data type "text/plain"

When a message is sent as MIME data type "text/plain", a message is published on the data bus in the following format:

```
{
  "name" : "rawdata",
  "time" : "{aktueller Zeitstempel}",
  "data" : [ {
    "name" : "encoding",
    "value" : "plain",
    "type" : "string"
  }, {
    "name" : "body",
    "value" : "{eingegebener Text}",
    "type" : "string"
  } ]
}
```

MIME data type "application/json"

When a message is sent as MIME data type "application/json", a message is published on the data bus in the following format:

```
{
  "name" : "rawdata",
  "time" : "{aktueller Zeitstempel}",
  "data" : [ {
    "name" : "encoding",
    "value" : "json",
    "type" : "string"
  }, {
    "name" : "body",
    "value" : "{eingegebenes JSON}",
    "type" : "string"
  } ]
}
```

Script logic

With a script logic, the contained data can then be further processed by converting the JSON into a JavaScript object using the `JSON.parse()` method.

```

function onNewReadingSet(state, s) {
  try {
    var set = s.Set;
    if (set.Name != "rawdata")
      return;

    var encoding, body;
    for (var i=0; i<set.Data.length; i++){
      var item = set.Data[i];
      switch(item.Name){
        case "encoding":
          encoding = item.Value;
          break;
        case "body":
          body = item.Value;
          break;
      }
    }
    if(encoding=="json"){
      console.log("\n received JSON: \n", body);
      var data = JSON.parse(body);
      if(data)
        console.log(" stringattribute = ",
data.list.stringattribute);
    }
  }
  catch (e) {
    console.log("onNewReadingSet() error: ", e.message);
  }
}

```

Note**Knowledge of the data structure required**

To use the "data" object according to the JSON.parse() method, knowledge of its data structure is required.

MIME data type "application/xml"

When a message is sent as MIME data type "application/xml", a message is published on the data bus in the following format:

```

{
  "name" : "rawdata",
  "time" : "{aktueller Zeitstempel}",
  "data" : [ {
    "name" : "encoding",
    "value" : "xml",
    "type" : "string"
  }, {
    "name" : "body",
    "value" : "{eingegebenes XML}",

```

```

        "type" : "string"
    } ]
}

```

MIME data type "application/x-www-form-urlencodedapplication/xml"

When a message is sent as MIME type "application/x-www-form-urlencoded", a message is published on the data bus in the following format:

```

{
  "name" : "formdata",
  "time" : "{aktueller Zeitstempel}",
  "data" : [ {
    "name" : "myID",
    "value" : "1",
    "type" : "string"
  }, {
    "name" : "myValue",
    "value" : "some text",
    "type" : "string"
  } ]
}

```

In the HTTP post body, the data is sent as a URL-encoded parameter list, e.g.:

```
myID=1&myValue=some%20text
```

The individual parameters are always evaluated in the value list as data type "string". A script logic can be used for the conversion to other data types.

It is not possible to specify a time stamp; the time of the BFC gateway is always used.

Note

Using unique names

BFC requires that the parameters are listed with a unique name. A parameter name must not be contained more than once in the parameter list.

HTML page

Below is a simple HTML page for sending the sample data shown above:

```

<html>
  <head><title>Form example</title></head>
  <body>
    <form method="POST" action="http://bfc:9876/rip-service/api/
dataset/raw">
      <input type="text" name="myID">
      <input type="text" name="myValue">
      <input type="submit" value="send">
    </form>
  </body>
</html>

```

The received values can be processed in BFC with a script logic, e.g.:

```
function onNewReadingSet(state, s) {
  try {
    var set = s.Set;
    if (set.Name != "formdata")
      return;

    var myId, myValue;
    for (var i=0; i<set.Data.length; i++) {
      var item = set.Data[i];
      switch (item.Name) {
        case "myId":
          myId = parseInt(item.Value);
          break;
        case "myValue":
          myValue = item.Value;
          break;
      }
    }
    console.log(" myId = ", myId) ;
    console.log(" myValue = ", myValue) ;
  }
  catch (e) {
    console.log("onNewReadingSet() error:", e.message);
  }
}
```

9.7 Method POST /events

This method can be used to report alarms and messages or event lists.

Parameter

The alarm lists must be sent in chronologically correct order.

Parameter	Description
alarms	The individual messages are reported with their respective start time. If a list no longer contains a previously contained alarm, or the start time has changed, the previous message is interpreted as finished. If no more alarms or messages should be pending, an empty list should be sent.

Note the "AlarmList" data model on the test web page (</rip-service/api/help>) for additional optional parameters.

Data structure with an alarm event

```

{
  "list": {
    "alarms": [
      {
        "id": "3000",
        "message": "Emergency Stop",
        "timestamp": "{Alarm Startzeit}"
      }
    ],
    "timestamp": "{Nachrichtsendezeit}"
  }
}client

```

9.8 Method POST /hfdataset

This method is used to report data from high-frequency recordings in one block to the BFC gateway. This is particularly necessary for data that is recorded at very short intervals. By combining the individual data series into tables, the number of messages is reduced.

Parameter

Parameter	Description
time	Specification of the message send point for each message
hfDataInfo	Contains meta-information about the column names and types contained and about the recording start. The recording start is identical over several messages.
values	Relative to the start of recording, the time offset is required for each data line as a floating-point number of the seconds since start
signalNames signalTypes values	The number of columns in "signalNames", "signalTypes" and "values" must be identical.
timeOffset	The value of "timeOffset" must increment in the chronologically correct order. Negative values and jumps back within a message are not allowed. In order for target systems to process the data without problems, the parameters should ascend over several messages, in chronologically correct order.

Further optional parameters for the data model "HFDataSet" are documented on the test web page (</rip-service/api/help>).

Data structure with recorded data

```

{
  "time": "{Nachrichtensendezeit}",
  "hfDataInfo": {
    "blockNo": 0,
    "config": "pmon",
    "id": "Name",
    "moreData": true,
    "signalNames": [ "myFloat", "myInt" ],
    "signalTypes": [ "float", "int" ],
    "startTime": "{Aufzeichnungsstart}"
  },
  "data": [
    {
      "timeOffset": 0,
      "values": [ null, null ]
    },
    {
      "timeOffset": 0.1,
      "values": [ 1.23, 1 ]
    },
    {
      "timeOffset": 0.2,
      "values": [ 12.3, 2 ]
    }
  ]
}

```

9.9 GET /writes method

Using this method, all write jobs available for the client and currently still not confirmed are called in the BFC gateway in the JSON format. The elements contained in the "jobs" array are used to confirm jobs (Page 404).

A detailed description of the parameters is provided on the test web page (</ripservice/api/help>).

Data structure with two write jobs

```

{
  "jobs": [
    {
      "id": "writejob-1",
      "timestamp": "2019-10-12T07:20:50.52Z",
      "ttlms": 30000,
      "issuer": {
        "ownerID": "2702a725-e96e-47c2-a7b6-68c14fb4eab3",
        "application": "opcua"
      },
      "data": [
        {
          "address": "/var1",
          "type": "string",
          "value": "dp1"
        }
      ]
    },
    {
      "id": "writejob-2",
      "timestamp": "2019-10-12T07:22:50.52Z",
      "ttlms": 10000,
      "issuer": {
        "ownerID": "2702a725-e96e-47c2-a7b6-68c14fb4eab3",
        "application": "opcua"
      },
      "data": [
        {
          "address": "/var2",
          "type": "int",
          "value": 10
        },
        {
          "address": "/var3",
          "type": "float",
          "value": 13.37
        }
      ]
    }
  ]
}

```

9.10 POST /writes/{ID}/ack method

Using this method, write jobs can be confirmed in the JSON format. To do this, a write job is required from the call to query write jobs (Page 403).

Parameter

Parameter	Description
ID	URL parameter to specify the write job to be confirmed

Data structure with the write job to be confirmed

```
{
  "id": "writejob-1",
  "timestamp": "2019-10-12T07:20:50.52Z",
  "ttlms": 30000,
  "issuer": {
    "ownerID": "2702a725-e96e-47c2-a7b6-68c14fb4eab3",
    "application": "opcu"
  },
  "status": "ok",
  "data": [
    {
      "address": "/var1",
      "type": "string",
      "value": "dp1",
      "status": "ok",
      "description": "Value changed"
    }
  ]
}
```


BFC gateway API

The BFC gateway includes an interface for configuring clients, middlewares and gateways using external applications as well. To do this, you can use SwaggerUI that is provided.

10.1 SwaggerUI

BFC gateway

The full SwaggerUI can be found in the BFC gateway:

```
https://<gateway-address>:<gateway-port>/openness/  
components/api/v1/ui/
```

Local

The SwaggerUI documentation is included in the software delivery and is located in the following directory:

```
swagger/clientconfig/swagger.json
```

Import this JSON file into any Swagger editor to display the available end points.

Swagger editor example (<https://editor.swagger.io/>)

10.2 Authentication

Procedure

1. Generate an `access_token` that can be used for authentication at the end points of the API components.
POST `https://<gateway-address>:<gateway-port>/openness/iam/api/v1/token`
{
 "grant_type": "client_credentials",
 "client_id": "{{client_id}}",
 "client_secret": "{{client_secret}}"
}

2. Enter your login information at `client_id` and `client_secret`.

3. The response will be as follows:

```
{
  "access_token":
    "eyJhbGciOiJSUzUxMiIsImtpZCI6IjEifQ.eyJhdWQiOiJlc2VyYWRTaW4iLCJleH
    AiOiJlE2NDY5Mjg2MzQuMzE3ODYxOCwiaWF0IjoxNjQ2OTIwNDM0LjMxNzg2MTgsIm
    lzc2Vyczo4MDgwL3VzZXJzL2FwaS9vYXV0aC90b2tlbl9rZXl2IiwianRpIjoiaMTA5NmU4YmQtNzg2Mi00OTAwLTg3ZDgtOTA0Y2E1ZDQ1MGY3Iiwic2tp
    bGxzIjpbImFkbWluIl0sInN1YiI6ImFkbWluIn0.eyJdnJ8x9Zd4JlsWAq4OXN0ohGpD
    OhF4_gX_Q82atABRzs-
    Ktc74UHTuJ__7VEwCPAYMtX6pmQaaGe7FJ987vCZSGclpGW__VIItTCTDL19EtLvc__
    XoCtoT38Sg9U0Era7sSt6Rs-
    csx19RpNKjB7PLYNvNhpaepRwXJWHwt1j9aKc303Trgfuj98uEEsX63gMTcz6fd6z_
    uTjwCTMuir-
    ZfXSUH7uq2Ki9z5ItOZgtjZA3LL64zRAuJ3iqYFwotOcxCjjwCf7nVOsu080iis8LT
    1NhEVhNc50A0RTFKGDvjyPxpYw1Y6i5igoooYhcV8oSJuuc-
    apq84jkgThtTAEBLORQGxpIW7So3X0vBypJy4mZ4ffjcUGPas_AL8rIb5C0hseKXy3
    R1F8ket6EWrw2_6jMPaFjHPuGNhthPG26dO6Kh4-
    h7kIOFJH4pI9gH8m6BOPQL8uPpNZSq4qgpzQg6bkWBY7dLU1ldF4l_R7t3e-
    qSjORkMdU_tnmGKOllsCHhENn16h6fgMRD42DahzfIquYMCJgS18Zefs7_59FTxWNZ
    7PskC3KM6bURNUiPH1000xO-
    E5f2Kg_lXk6ajPp5v793DCGUvRIQ5Dj3yJ9gIPPOpHINQve5Sq96HIzTAFJzThB52w
    wvmfBkjwFrzoW23R-JfW3V3xeBmX_D3jmO8",
  "expires_in": 7200,
  "token_type": "Bearer"
}
```

4. Use the value of `access_token` to authenticate Bearer at the end points of the API components. This token contains all the rights assigned to your user.

10.3 Client configuration

Client details

You require the following rights:

- `com.siemens.bfc.skills.openness.components.manager`
- OR -
- `com.siemens.bfc.skills.openness.components.viewer`

All clients:

GET `https://<gateway-address>:<gateway-port>/openness/components/api/v1/components/clients`

At this end point you obtain detailed information about all clients.

Example:

```
[
  {
    "clientConfig": {
      "opcuaClientConfig": {
        "url": "opc.tcp://opc.mtconnect.org:4840"
      }
    }
  }
]
```

```
    },
    "publishing": {
      "alarms": {
        "addresses": [],
        "alarmNoProperty": 0,
        "alarmNumberPropertyInfo": {}
      },
      "readingSets": {
        "test": {
          "mappings": [
            {
              "address": "test",
              "name": "test",
              "type": "int"
            }
          ],
          "readingMode": {
            "debounce": 500,
            "interval": 500,
            "mode": "onchange"
          }
        }
      },
      "whitelistedWriteData": []
    }
  },
  "config": {
    "envs": [],
    "ports": []
  },
  "id": "76d553d4-397b-41d6-babc-8d8f55374f66",
  "name": "test",
  "ownerID": "393fd4c0-6b12-45d4-bdb3-8912fce75e8f",
  "skills": null,
  "state": {
    "serverTimestamp": "0001-01-01T00:00:00.000Z",
    "status": "stopped",
    "timestamp": "2022-03-10T14:33:59.808Z"
  },
  "type": "opcuacclient"
},
{
  "clientConfig": {
    "fanucClientConfig": {
      "clientDescription": "string",
      "port": 8193,
      "requestTimeout": 30,
      "url": "192.168.0.2"
    },
    "publishing": {
      "alarms": {
        "addresses": null,
```

```

        "alarmNumberPropertyInfo": {},
        "data": {
            "alarmSampleTime": 10000
        }
    },
    "readingSets": {
        "TestSet": {
            "mappings": [
                {
                    "address": "I0.0",
                    "name": "TEST_BIT",
                    "type": "int"
                }
            ],
            "readingMode": {
                "debounce": 500,
                "mode": "onchange"
            }
        }
    },
    "whitelistedWriteData": [
        {
            "address": "Byte0",
            "type": "int"
        }
    ]
},
"config": {
    "envs": [
        {
            "key": "TEST_ENV",
            "value": "TRUE"
        }
    ],
    "ports": [
        {
            "type": "tcp",
            "value": 8080
        }
    ]
},
"id": "9eaf9357-7a16-4d41-93e4-e9a257c89f79",
"name": "postman-fanuc-client",
"ownerID": "rootid",
"skills": [
    {
        "name": "Focas"
    },
    {
        "name":
"com.siemens.bfc.skills.basic.publish.reading.sets"

```

```

    }
  ],
  "state": {
    "activated": true,
    "serverTimestamp": "2022-03-10T14:33:48.739Z",
    "status": "disconnected",
    "timestamp": "2022-03-10T14:33:48.723Z"
  },
  "type": "fanucclient"
},
{
  "clientConfig": {
    "publishing": {
      "alarms": {
        "addresses": null,
        "alarmNumberPropertyInfo": {}
      },
      "readingSets": {
        "TestSet": {
          "mappings": [
            {
              "address": "I0.0",
              "name": "TEST_BIT",
              "type": "int"
            }
          ],
          "readingMode": {
            "interval": 500,
            "mode": "interval"
          }
        }
      },
      "whitelistedWriteData": [
        {
          "address": "Byte0",
          "type": "int"
        }
      ]
    },
    "s7ClientConfig": {
      "clientDescription": "string",
      "connectionTimeout": 5000,
      "connectionType": 12,
      "devicePassword": "string",
      "devicePort": 244,
      "numberOfLocalTSAP": 5000,
      "numberOfRack": 5000,
      "numberOfRemoteTSAP": 5000,
      "numberOfSlot": 5000,
      "receiveTimeout": 5000,
      "requestSizePDU": 240,
      "sendTimeout": 5000,
    }
  }
}

```

```

        "url": "192.168.0.2"
      }
    },
    "config": {
      "envs": [
        {
          "key": "TEST_ENV",
          "value": "TRUE"
        }
      ],
      "ports": [
        {
          "type": "tcp",
          "value": 8080
        }
      ]
    },
    "id": "f520f372-3427-4547-8d95-69c0f2da1b78",
    "name": "postman-s7-client",
    "ownerID": "rootid",
    "skills": [
      {
        "name": "S7"
      },
      {
        "name":
"com.siemens.bfc.skills.basic.publish.reading.sets"
      }
    ],
    "state": {
      "activated": true,
      "serverTimestamp": "2022-03-10T14:33:35.428Z",
      "status": "disconnected",
      "timestamp": "2022-03-10T14:33:35.417Z"
    },
    "type": "s7client"
  },
  {
    "clientConfig": {
      "publishing": {
        "whitelistedWriteData": []
      }
    },
    "id": "Sinutrain",
    "meta": {
      "com.siemens.bfc.meta.bfcclient.hmi.type": "Operate",
      "com.siemens.bfc.meta.bfcclient.hmi.version":
"04.07.03.01",
      "com.siemens.bfc.meta.bfcclient.nck.type": "840D s1"
    },
    "name": "Sinutrain",
    "ownerID": "881a497a-dc6c-43b3-8298-09ecf0b59bad",

```



```
    "skills": null,  
    "state": {  
      "activated": true,  
      "serverTimestamp": "2022-03-10T14:33:56.322Z",  
      "status": "disconnected",  
      "timestamp": "2022-03-10T14:33:56.314Z"  
    },  
    "type": "bfcclient"  
  }  
]
```

Single client:

GET https://<gateway-address>:<gateway-port>/openness/
components/api/v1/components/clients/<id>

Example:

```
{  
  "clientConfig": {  
    "fanucClientConfig": {  
      "clientDescription": "string",  
      "port": 8193,  
      "requestTimeout": 30,  
      "url": "192.168.0.2"  
    },  
    "publishing": {  
      "alarms": {  
        "addresses": null,  
        "alarmNumberPropertyInfo": {},  
        "data": {  
          "alarmSampleTime": 10000  
        }  
      },  
      "readingSets": {  
        "TestSet": {  
          "mappings": [  
            {  
              "address": "I0.0",  
              "name": "TEST_BIT",  
              "type": "int"  
            }  
          ],  
          "readingMode": {  
            "debounce": 500,  
            "mode": "onchange"  
          }  
        }  
      },  
      "whitelistedWriteData": [  
        {  
          "address": "Byte0",  
          "type": "int"  
        }  
      ]  
    }  
  }  
}
```

```

        ]
      }
    },
    "config": {
      "envs": [
        {
          "key": "TEST_ENV",
          "value": "TRUE"
        }
      ],
      "ports": [
        {
          "type": "tcp",
          "value": 8080
        }
      ]
    },
    "id": "9eaf9357-7a16-4d41-93e4-e9a257c89f79",
    "name": "postman-fanuc-client",
    "ownerID": "rootid",
    "skills": [
      {
        "name": "Focas"
      },
      {
        "name":
"com.siemens.bfc.skills.basic.publish.reading.sets"
      }
    ],
    "state": {
      "activated": true,
      "serverTimestamp": "2022-03-10T14:33:48.739Z",
      "status": "disconnected",
      "timestamp": "2022-03-10T14:33:48.723Z"
    },
    "type": "fanucclient"
  }
}

```

Creating a client

You require the following rights:

- com.siemens.bfc.skills.openness.components.manager

POST <https://<gateway-address>:<gateway-port>/openness/components/api/v1/components/clients>

Updating a client

You require the following rights:

- com.siemens.bfc.skills.openness.components.manager

```
PUT https://<gateway-address>:<gateway-port>/openness/  
components/api/v1/components/clients/<id>
```

Removing the client

You require the following rights:

- `com.siemens.bfc.skills.openness.components.manager`

```
DELETE https://<gateway-address>:<gateway-port>/openness/  
components/api/v1/components/clients/<id>
```

Client configuration examples

S7 client

```
{  
  "clientConfig": {  
    "publishing": {  
      "readingSets": {  
        "TestSetInterval": {  
          "mappings": [  
            {  
              "address": "I0.0",  
              "name": "TEST_BIT",  
              "type": "int"  
            }  
          ],  
          "readingMode": {  
            "interval": 500,  
            "mode": "interval"  
          }  
        },  
        "TestSetOnChange": {  
          "mappings": [  
            {  
              "address": "I0.0",  
              "name": "TEST_BIT",  
              "type": "int"  
            }  
          ],  
          "readingMode": {  
            "debounce": 2000,  
            "mode": "onchange"  
          }  
        }  
      },  
      "TestSetOnTrigger": {  
        "mappings": [  
          {  
            "address": "I0.0",  
            "name": "TEST_BIT",  
            "type": "int"  
          }  
        ]  
      }  
    }  
  }  
}
```

```

        "type": "int"
      }
    ],
    "readingMode": {
      "interval": 500,
      "mode": "ontrigger"
    },
    "triggercomposition": "T1",
    "triggers": [
      {
        "name": "T1",
        "referenceDatapoint": "TEST_BIT",
        "condition": "=",
        "referenceValue": 200
      }
    ]
  },
  "whitelistedWriteData": [
    {
      "address": "Byte0",
      "type": "int"
    }
  ]
},
"s7ClientConfig": {
  "clientDescription": "string",
  "connectionTimeout": 5000,
  "connectionType": 12,
  "devicePassword": "string",
  "devicePort": 244,
  "numberOfLocalTSAP": 5000,
  "numberOfRack": 5000,
  "numberOfRemoteTSAP": 5000,
  "numberOfSlot": 5000,
  "receiveTimeout": 5000,
  "requestSizePDU": 240,
  "sendTimeout": 5000,
  "url": "192.168.0.2"
},
"config": {
  "envs": [
    {
      "key": "TEST_ENV",
      "value": "TRUE"
    }
  ],
  "ports": [
    {
      "type": "tcp",
      "value": 8080
    }
  ]
}

```

```
    }  
  ],  
  },  
  "name": "postman-s7-client",  
  "ownerID": "rootid",  
  "type": "s7client"  
}
```

Fanuc client

```
{  
  "clientConfig": {  
    "publishing": {  
      "alarms": {  
        "data": {  
          "alarmSampleTime": 10000  
        }  
      },  
    },  
    "readingSets": {  
      "TestSetInterval": {  
        "mappings": [  
          {  
            "address": "I0.0",  
            "name": "TEST_BIT",  
            "type": "int"  
          }  
        ],  
        "readingMode": {  
          "interval": 500,  
          "mode": "interval"  
        }  
      },  
      "TestSetOnChange": {  
        "mappings": [  
          {  
            "address": "I0.0",  
            "name": "TEST_BIT",  
            "type": "int"  
          }  
        ],  
        "readingMode": {  
          "debounce": 2000,  
          "mode": "onchange"  
        }  
      },  
      "TestSetOnTrigger": {  
        "mappings": [  
          {  
            "address": "I0.0",  
            "name": "TEST_BIT",  
            "type": "int"  
          }  
        ],  
      }  
    }  
  }  
}
```

```

        "readingMode":{
            "interval":500,
            "mode":"ontrigger"
        },
        "triggercomposition":"T1",
        "triggers":[
            {
                "name":"T1",
                "referenceDatapoint":"TEST_BIT",
                "condition":"=",
                "referenceValue":200
            }
        ]
    },
    "whitelistedWriteData":[
        {
            "address":"Byte0",
            "type":"int"
        }
    ],
    "filePermissions":{
        "patterns":[
            {
                "path":"readPath",
                "read":true
            },
            {
                "path":"readCreatePath",
                "read":true,
                "create":true
            },
            {
                "path":"readCreateWritePath",
                "read":true,
                "create":true,
                "write":true
            },
            {
                "path":"readCreateWriteDeletePath",
                "read":true,
                "create":true,
                "write":true,
                "delete":true
            }
        ]
    },
    "fanucClientConfig": {
        "clientDescription": "string",
        "port": 8193,
        "requestTimeout": 30,
    }
}

```

```
        "url": "192.168.0.2"
      },
    },
    "config": {
      "envs": [
        {
          "key": "TEST_ENV",
          "value": "TRUE"
        }
      ],
      "ports": [
        {
          "type": "tcp",
          "value": 8080
        }
      ]
    },
    "name": "postman-fanuc-client",
    "ownerID": "rootid",
    "type": "fanucclient"
  }
}
```

Focas1 client

```
{
  "clientConfig":{
    "publishing":{
      "alarms":{
        "data":{
          "alarmSampleTime":10000
        }
      },
      "readingSets":{
        "TestSetInterval":{
          "mappings":[
            {
              "address":"I0.0",
              "name":"TEST_BIT",
              "type":"int"
            }
          ],
          "readingMode":{
            "interval":500,
            "mode":"interval"
          }
        },
        "TestSetOnChange":{
          "mappings":[
            {
              "address":"I0.0",
              "name":"TEST_BIT",
              "type":"int"
            }
          ]
        }
      }
    }
  }
}
```

```

        ],
        "readingMode":{
            "debounce":2000,
            "mode":"onChange"
        }
    },
    "TestSetOnTrigger":{
        "mappings":[
            {
                "address":"I0.0",
                "name":"TEST_BIT",
                "type":"int"
            }
        ],
        "readingMode":{
            "interval":500,
            "mode":"ontrigger"
        },
        "triggercomposition":"T1",
        "triggers":[
            {
                "name":"T1",
                "referenceDatapoint":"TEST_BIT",
                "condition":"=",
                "referenceValue":200
            }
        ]
    }
},
"focas1ClientConfig":{
    "clientDescription":"string",
    "port":8193,
    "timeout":30,
    "url":"192.168.0.2"
},
"config":{
    "envs":[
        {
            "key":"TEST_ENV",
            "value":"TRUE"
        }
    ],
    "ports":[
        {
            "type":"tcp",
            "value":8080
        }
    ]
},
"name":"postman-focas1-client",
"ownerID":"rootid",

```



```
        "type": "focas1client"
    }
}
```

Heidenhain client

```
{
  "clientConfig": {
    "publishing": {
      "alarms": {
        "data": {
          "alarmSampleTime": 10000
        }
      },
      "readingSets": {
        "TestSetInterval": {
          "mappings": [
            {
              "address": "I0.0",
              "name": "TEST_BIT",
              "type": "int"
            }
          ],
          "readingMode": {
            "interval": 500,
            "mode": "interval"
          }
        },
        "TestSetOnChange": {
          "mappings": [
            {
              "address": "I0.0",
              "name": "TEST_BIT",
              "type": "int"
            }
          ],
          "readingMode": {
            "debounce": 2000,
            "mode": "onchange"
          }
        },
        "TestSetOnTrigger": {
          "mappings": [
            {
              "address": "I0.0",
              "name": "TEST_BIT",
              "type": "int"
            }
          ],
          "readingMode": {
            "interval": 500,
            "mode": "ontrigger"
          }
        }
      }
    }
  }
}
```

```

        "triggercomposition":"T1",
        "triggers":[
            {
                "name":"T1",
                "referenceDatapoint":"TEST_BIT",
                "condition":"=",
                "referenceValue":200
            }
        ]
    },
    "whitelistedWriteData":[
        {
            "address":"Byte0",
            "type":"int"
        }
    ],
    "filePermissions":{
        "patterns":[
            {
                "path":"readPath",
                "read":true
            },
            {
                "path":"readCreatePath",
                "read":true,
                "create":true
            },
            {
                "path":"readCreateWritePath",
                "read":true,
                "create":true,
                "write":true
            },
            {
                "path":"readCreateWriteDeletePath",
                "read":true,
                "create":true,
                "write":true,
                "delete":true
            }
        ]
    },
    "heidenhainClientConfig":{
        "clientDescription":"string",
        "customHeidenheinConnectionConfiguration":"test",
        "doNotSwapBytes":true,
        "logOutAfterEveryPLCOperation":true,
        "requestTimeout":15000,
        "url":"192.168.0.2:19000"
    }
}

```

```
    },
    "config":{
      "envs":[
        {
          "key":"TEST_ENV",
          "value":"TRUE"
        }
      ],
      "ports":[
        {
          "type":"tcp",
          "value":8080
        }
      ]
    },
    "name":"postman-heidenhain-client",
    "ownerID":"rootid",
    "type":"heidenhainclient"
  }
}
```

Modbus client

```
{
  "clientConfig":{
    "publishing":{
      "readingSets":{
        "TestSetInterval":{
          "mappings":[
            {
              "address":"I0.0",
              "name":"TEST_BIT",
              "type":"int"
            }
          ],
          "readingMode":{
            "interval":500,
            "mode":"interval"
          }
        },
        "TestSetOnChange":{
          "mappings":[
            {
              "address":"I0.0",
              "name":"TEST_BIT",
              "type":"int"
            }
          ],
          "readingMode":{
            "debounce":2000,
            "mode":"onchange"
          }
        },
        "TestSetOnTrigger":{
```

```

        "mappings":[
            {
                "address":"I0.0",
                "name":"TEST_BIT",
                "type":"int"
            }
        ],
        "readingMode":{
            "interval":500,
            "mode":"ontrigger"
        },
        "triggercomposition":"T1",
        "triggers":[
            {
                "name":"T1",
                "referenceDatapoint":"TEST_BIT",
                "condition":"=",
                "referenceValue":200
            }
        ]
    },
    "whitelistedWriteData":[
        {
            "address":"Byte0",
            "type":"int"
        }
    ]
},
"modbusClientConfig":{
    "clientDescription":"string",
    "unitID":2,
    "url":"192.168.0.90:502"
},
"config":{
    "envs":[
        {
            "key":"TEST_ENV",
            "value":"TRUE"
        }
    ],
    "ports":[
        {
            "type":"tcp",
            "value":8080
        }
    ]
},
"name":"postman-modbus-client",
"ownerID":"rootid",

```

```
    "type": "modbusclient"  
  }
```

MTConnect client

```
{  
  "clientConfig": {  
    "publishing": {  
      "alarms": {  
        "addresses": [  
          "con1",  
          "con2"  
        ]  
      },  
      "readingSets": {  
        "TestSetInterval": {  
          "mappings": [  
            {  
              "address": "I0.0",  
              "name": "TEST_BIT",  
              "type": "int"  
            }  
          ],  
          "readingMode": {  
            "interval": 500,  
            "mode": "interval"  
          }  
        },  
        "TestSetOnChange": {  
          "mappings": [  
            {  
              "address": "I0.0",  
              "name": "TEST_BIT",  
              "type": "int"  
            }  
          ],  
          "readingMode": {  
            "debounce": 2000,  
            "mode": "onchange"  
          }  
        },  
        "TestSetOnTrigger": {  
          "mappings": [  
            {  
              "address": "I0.0",  
              "name": "TEST_BIT",  
              "type": "int"  
            }  
          ],  
          "readingMode": {  
            "interval": 500,  
            "mode": "ontrigger"  
          }  
        }  
      }  
    }  
  }
```

```

        "triggercomposition":"T1",
        "triggers":[
            {
                "name":"T1",
                "referenceDatapoint":"TEST_BIT",
                "condition":"=",
                "referenceValue":200
            }
        ]
    },
    "mtconnectClientConfig":{
        "clientDescription":"string",
        "password":"pass",
        "requestTimeout":30000,
        "samplingTime":200,
        "url":"192.168.0.90",
        "username":"user"
    }
},
"config":{
    "envs":[
        {
            "key":"TEST_ENV",
            "value":"TRUE"
        }
    ],
    "ports":[
        {
            "type":"tcp",
            "value":8080
        }
    ]
},
"name":"postman-mtconnect-client",
"ownerID":"rootid",
"type":"mtconnectclient"
}
}

```

OPC UA client

```

{
    "clientConfig":{
        "publishing":{
            "alarms":{
                "addresses":[
                    "adr1",
                    "adr2"
                ],
                "alarmNoProperty":3,
                "alarmNumberPropertyInfo":{
                    "address":"ads",

```

```
        "name": "test"
      },
    },
    "readingSets": {
      "TestSetInterval": {
        "mappings": [
          {
            "address": "I0.0",
            "name": "TEST_BIT",
            "type": "int"
          }
        ],
        "readingMode": {
          "interval": 500,
          "mode": "interval"
        }
      },
      "TestSetOnChange": {
        "mappings": [
          {
            "address": "I0.0",
            "name": "TEST_BIT",
            "type": "int"
          }
        ],
        "readingMode": {
          "debounce": 2000,
          "mode": "onchange"
        }
      },
      "TestSetOnTrigger": {
        "mappings": [
          {
            "address": "I0.0",
            "name": "TEST_BIT",
            "type": "int"
          }
        ],
        "readingMode": {
          "interval": 500,
          "mode": "ontrigger"
        },
        "triggercomposition": "T1",
        "triggers": [
          {
            "name": "T1",
            "referenceDatapoint": "TEST_BIT",
            "condition": "=",
            "referenceValue": 200
          }
        ]
      }
    }
  }
}
```

```

    },
    "whitelistedWriteData":[
      {
        "address":"Byte0",
        "type":"int"
      }
    ]
  },
  "opcuaClientConfig":{
    "clientDescription":"string",
    "password":"pass",
    "url":"opc.tcp://192.168.0.90:4820",
    "username":"user"
  }
},
"config":{
  "envs":[
    {
      "key":"TEST_ENV",
      "value":"TRUE"
    }
  ],
  "ports":[
    {
      "type":"tcp",
      "value":8080
    }
  ]
},
"name":"postman-opcua-client",
"ownerID":"rootid",
"type":"opcuaclient"
}

```

Beckhoff client

```

{
  "clientConfig":{
    "publishing":{
      "readingSets":{
        "TestSetInterval":{
          "mappings":[
            {
              "address":"I0.0",
              "name":"TEST_BIT",
              "type":"int"
            }
          ],
          "readingMode":{
            "interval":500,
            "mode":"interval"
          }
        }
      }
    }
  },

```



```
    "TestSetOnChange":{
      "mappings":[
        {
          "address":"I0.0",
          "name":"TEST_BIT",
          "type":"int"
        }
      ],
      "readingMode":{
        "debounce":2000,
        "mode":"onchange"
      }
    },
    "TestSetOnTrigger":{
      "mappings":[
        {
          "address":"I0.0",
          "name":"TEST_BIT",
          "type":"int"
        }
      ],
      "readingMode":{
        "interval":500,
        "mode":"ontrigger"
      },
      "triggercomposition":"T1",
      "triggers":[
        {
          "name":"T1",
          "referenceDatapoint":"TEST_BIT",
          "condition":"=",
          "referenceValue":200
        }
      ]
    }
  ],
  "whitelistedWriteData":[
    {
      "address":"Byte0",
      "type":"int"
    }
  ]
},
"beckhoffClientConfig":{
  "clientDescription":"string",
  "localRoute":"string",
  "maxBuffer":52428800,
  "requestTimeout":5000,
  "sampleTime":30000,
  "url":"192.168.0.90"
}
},
```

```

    "config":{
      "envs":[
        {
          "key":"TEST_ENV",
          "value":"TRUE"
        }
      ],
      "ports":[
        {
          "type":"tcp",
          "value":8080
        }
      ]
    },
    "name":"postman-beckhoff-client",
    "ownerID":"rootid",
    "type":"beckhoffclient"
  }

```

Omron client

```

{
  "clientConfig":{
    "publishing":{
      "readingSets":{
        "TestSetInterval":{
          "mappings":[
            {
              "address":"I0.0",
              "name":"TEST_BIT",
              "type":"int"
            }
          ],
          "readingMode":{
            "interval":500,
            "mode":"interval"
          }
        },
        "TestSetOnChange":{
          "mappings":[
            {
              "address":"I0.0",
              "name":"TEST_BIT",
              "type":"int"
            }
          ],
          "readingMode":{
            "debounce":2000,
            "mode":"onchange"
          }
        },
        "TestSetOnTrigger":{
          "mappings":[

```

```
        {
          "address": "I0.0",
          "name": "TEST_BIT",
          "type": "int"
        }
      ],
      "readingMode": {
        "interval": 500,
        "mode": "ontrigger"
      },
      "triggercomposition": "T1",
      "triggers": [
        {
          "name": "T1",
          "referenceDatapoint": "TEST_BIT",
          "condition": "=",
          "referenceValue": 200
        }
      ]
    }
  },
  "whitelistedWriteData": [
    {
      "address": "Byte0",
      "type": "int"
    }
  ]
},
"omronClientConfig": {
  "clientDescription": "string",
  "url": "192.168.1.10:9600",
  "sourceNetwork": 0,
  "sourceNode": 1,
  "sourceUnit": 0,
  "destinationNetwork": 0,
  "destinationNode": 0,
  "destinationUnit": 0,
  "requestTimeout": 30000
},
"config": {
  "envs": [
    {
      "key": "TEST_ENV",
      "value": "TRUE"
    }
  ],
  "ports": [
    {
      "type": "tcp",
      "value": 8080
    }
  ]
}
```

```

    ]
  },
  "name": "postman-omron-client",
  "ownerID": "rootid",
  "type": "omronclient"
}

```

Ethernet IP client

```

{
  "clientConfig": {
    "publishing": {
      "readingSets": {
        "TestSetInterval": {
          "mappings": [
            {
              "address": "I0.0",
              "name": "TEST_BIT",
              "type": "int"
            }
          ],
          "readingMode": {
            "interval": 500,
            "mode": "interval"
          }
        },
        "TestSetOnChange": {
          "mappings": [
            {
              "address": "I0.0",
              "name": "TEST_BIT",
              "type": "int"
            }
          ],
          "readingMode": {
            "debounce": 2000,
            "mode": "onchange"
          }
        },
        "TestSetOnTrigger": {
          "mappings": [
            {
              "address": "I0.0",
              "name": "TEST_BIT",
              "type": "int"
            }
          ],
          "readingMode": {
            "interval": 500,
            "mode": "ontrigger"
          },
          "triggercomposition": "T1",
          "triggers": [

```

```
        {
            "name": "T1",
            "referenceDatapoint": "TEST_BIT",
            "condition": "=",
            "referenceValue": 200
        }
    ]
},
"whitelistedWriteData": [
    {
        "address": "Byte0",
        "type": "int"
    }
]
},
"ethernetIPClientConfig": {
    "clientDescription": "string",
    "url": "192.168.1.15:44818",
    "deviceFamily": "None",
    "path": "1.0",
    "protocolType": "protocolType",
    "connectionTimeout": 300,
    "sendTimeout": 1000,
    "receiveTimeout": 3000
}
},
"config": {
    "envs": [
        {
            "key": "TEST_ENV",
            "value": "TRUE"
        }
    ],
    "ports": [
        {
            "type": "tcp",
            "value": 8080
        }
    ]
}
},
"name": "postman-eip-client",
"ownerID": "rootid",
"type": "eipclient"
}
```

SFTP client

```
{
    "clientConfig": {
        "publishing": {
            "readingSets": {},
            "whitelistedWriteData": [],

```

```

    "filePermissions":{
      "patterns":[
        {
          "path":"readPath",
          "read":true
        },
        {
          "path":"readCreatePath",
          "read":true,
          "create":true
        },
        {
          "path":"readCreateWritePath",
          "read":true,
          "create":true,
          "write":true
        },
        {
          "path":"readCreateWriteDeletePath",
          "read":true,
          "create":true,
          "write":true,
          "delete":true
        }
      ]
    },
    "sftpClientConfig":{
      "url":"sftp://192.168.5.90:2222",
      "clientDescription":"string",
      "username":"user",
      "password":"pass",
      "keyExchanges":"keyExchanges",
      "ciphers":"ciphers",
      "macs":"macs",
      "hostPublicKey":"hostPublicKey",
      "webdavPort":"","7979"
    },
    "config":{
      "envs":[
        {
          "key":"TEST_ENV",
          "value":"TRUE"
        }
      ],
      "ports":[
        {
          "type":"tcp",
          "value":8080
        }
      ]
    }
  ]

```

```
    },  
    "name": "postman-sftp-client",  
    "ownerID": "rootid",  
    "type": "sftpclient"  
  }  
}
```

HTTP script client

```
{  
  "clientConfig": {  
    "publishing": {  
      "readingSets": {  
        "TestSetInterval": {  
          "mappings": [  
            {  
              "address": "I0.0",  
              "name": "TEST_BIT",  
              "type": "int"  
            }  
          ],  
          "readingMode": {  
            "interval": 500,  
            "mode": "interval"  
          }  
        },  
        "TestSetOnChange": {  
          "mappings": [  
            {  
              "address": "I0.0",  
              "name": "TEST_BIT",  
              "type": "int"  
            }  
          ],  
          "readingMode": {  
            "debounce": 2000,  
            "mode": "onchange"  
          }  
        },  
        "TestSetOnTrigger": {  
          "mappings": [  
            {  
              "address": "I0.0",  
              "name": "TEST_BIT",  
              "type": "int"  
            }  
          ],  
          "readingMode": {  
            "interval": 500,  
            "mode": "ontrigger"  
          }  
        },  
        "triggercomposition": "T1",  
        "triggers": [  
          {  
            "address": "I0.0",  
            "name": "TEST_BIT",  
            "type": "int"  
          }  
        ]  
      }  
    }  
  }  
}
```

```

        "name": "T1",
        "referenceDatapoint": "TEST_BIT",
        "condition": "=",
        "referenceValue": 200
      }
    ]
  },
  "whitelistedWriteData": []
},
"httpScriptClientConfig": {
  "caCert": "ca",
  "clientDescription": "string",
  "deviceName": "deviceName",
  "logic": "{ \"this\": \"is my logic\" }",
  "logicConfig": "{ \"this\": \"is my logicconfig\" }",
  "logicPath": "env://IOT_HTTP_LOGIC",
  "password": "pass",
  "resultLanguage": "de-DE",
  "skipVerify": true,
  "url": "url",
  "username": "user"
}
},
"config": {
  "envs": [
    {
      "key": "TEST_ENV",
      "value": "TRUE"
    }
  ],
  "ports": [
    {
      "type": "tcp",
      "value": 8080
    }
  ]
},
"name": "postman-httpsclient",
"ownerID": "rootid",
"type": "httpsclient"
}

```

BFC client

Creating and removing the client is not supported, only the update.

```

{
  "clientConfig": {
    "publishing": {
      "readingSets": {
        "TestSetInterval": {
          "mappings": [

```



```
        {
          "address": "I0.0",
          "name": "TEST_BIT",
          "type": "int"
        }
      ],
      "readingMode": {
        "interval": 500,
        "mode": "interval"
      }
    },
    "TestSetOnChange": {
      "mappings": [
        {
          "address": "I0.0",
          "name": "TEST_BIT",
          "type": "int"
        }
      ],
      "readingMode": {
        "debounce": 2000,
        "mode": "onchange"
      }
    },
    "TestSetOnTrigger": {
      "mappings": [
        {
          "address": "I0.0",
          "name": "TEST_BIT",
          "type": "int"
        }
      ],
      "readingMode": {
        "interval": 500,
        "mode": "ontrigger"
      },
      "triggercomposition": "T1",
      "triggers": [
        {
          "name": "T1",
          "referenceDatapoint": "TEST_BIT",
          "condition": "=",
          "referenceValue": 200
        }
      ]
    }
  ],
  "whitelistedWriteData": [
    {
      "address": "Byte0",
      "type": "int"
    }
  ]
}
```

```

    ]
  }
},
"id": "postman-bfc-client",
"ownerID": "rootid",
"name": "postman-bfc-client",
"type": "bfcclient"
}

```

10.4 Middleware configuration

Middleware details

You require the following rights:

- com.siemens.bfc.skills.openness.components.manager
- OR -
- com.siemens.bfc.skills.openness.components.viewer

All Middlewares:

GET https://<gateway-address>:<gateway-port>/openness/components/api/v1/components/middlewares

At this end point you obtain detailed information about all middlewares.

Example:

```

[
  {
    "config": {
      "envs": [
        {
          "key": "IOT_MIDDLEWARE_TEMPLATE",
          "value": "default"
        },
        {
          "key": "IOT_OWNER_ID",
          "value": "f263c9bc-e1a0-4b04-a58b-8b6bf0e7497c"
        },
        {
          "key": "IOT_TYPE",
          "value": "middleware"
        },
        {
          "key": "IOT_NAME",
          "value": "pub"
        },
        {
          "key": "IOT_ID",
          "value": "4aa6eab6-2e2c-4fd2-bde4-5fcf748081f5"
        }
      ]
    }
  }
]

```

```

    {
      "key": "IOT_APP_ID",
      "value": "4aa6eab6-2e2c-4fd2-bde4-5fcf748081f5"
    },
    {
      "key": "IOT_DNS_NAMES",
      "value": "pubf253cdd7-b7ec-49e6-89b1-3cf021db2faf"
    },
    {
      "key": "TRACE",
      "value": "TRUE"
    },
    {
      "key": "IOT_INTERACT",
      "value": "TRUE"
    }
  ],
  "ports": [],
},
"id": "4aa6eab6-2e2c-4fd2-bde4-5fcf748081f5",
"name": "pubsub",
"ownerID": "rootid",
"scriptlogicConfig": {
  "logic": "\nfunction onNewData(topic,payload)
{\n console.log(\"onNewData ( topic:=\", topic,
\", payload:=\", payload, \" )\");\n}\n\nvar
done = subscribeData(\"connectivity.foo.*\",
onNewData );\nconsole.log(\"subscribeData done:=\", done);\n\n//
This function will be called after the
selected time period in ms.\nstartIntervalWorker(function
(state) {\n \n console.log(\"start\");\n
var ret = publishData(\"connectivity.machine.Kai-
NT.iotclient.cmd\", \"deleteBuffer\");\n console.log(\"done - \",
ret);\n\n}, 40000); // This function will be called every 40s",
  "subscriptions": {
    "alarms": [],
    "connectionStateChange": [],
    "readingSets": [],
    "wildData": [],
    "writeDataJobs": []
  }
},
"state": {
  "activated": false,
  "serverTimestamp": "2022-10-27T14:56:51.990Z",
  "status": "stopped",
  "timestamp": "2022-10-27T14:56:51.943Z"
},
"type": "scriptlogic"
},
{
  "config": {

```

```

    "envs": [
      {
        "key": "IOT_INTERACT",
        "value": "TRUE"
      },
      {
        "key": "IOT_MIDDLEWARE_TEMPLATE",
        "value": "Default"
      },
      {
        "key": "IOT_OWNER_ID",
        "value": "7ed7592d-e346-483b-84cf-2003618f437c"
      },
      {
        "key": "IOT_TYPE",
        "value": "middleware"
      },
      {
        "key": "IOT_NAME",
        "value": "trb-test"
      },
      {
        "key": "IOT_ID",
        "value": "7758590e-7fc4-4857-ab15-a06777b3cb63"
      },
      {
        "key": "IOT_APP_ID",
        "value": "7758590e-7fc4-4857-ab15-a06777b3cb63"
      },
      {
        "key": "IOT_DNS_NAMES",
        "value": "trb-testab37da9c-c045-4c40-a570-fd07f542c7b1"
      }
    ],
    "ports": [],
    "id": "7758590e-7fc4-4857-ab15-a06777b3cb63",
    "name": "test",
    "ownerID": "rootid",
    "scriptlogicConfig": {
      "logic": "var prev_values =
{};\n\nfunction onNewReadingSet(state, msg) {\n  //
  console.log(\"onNewReadingSet() : \", JSON.stringify(msg)); //
  trace all incoming messages\n\n  switch (msg.Set.Name) {\n
  case \"test\":\n    // To achieve a simple access to the
  variables of the data set, convert the 'Data' array to dataset
  object.\n    var dataset = {};\n    for (var i = 0; i <
  msg.Set.Data.length; i++) {\n      dataset[msg.Set.Data[i].Name]
  = msg.Set.Data[i];\n    }\n    var dataset_old =
  state[msg.Source.ClientID];\n    if (dataset_old ===
  undefined) {\n      dataset_old = dataset;\n    } else
  {\n      console.log(\"dataset.R0.Value:\", dataset.R0.Value,

```

```

\", dataset_old.R0.Value:\", dataset_old.R0.Value);\n      }
\n\n      // Check condition\n      var diff_R0 =
Math.abs(dataset.R0.Value - dataset_old.R0.Value);\n      var
diff_R1 = Math.abs(dataset.R1.Value - dataset_old.R1.Value);\n
console.log(\"diff_R0:\", diff_R0, \", diff_R1:\", diff_R1);\n
var diff = Math.max(diff_R0,diff_R1);\n      if (diff >
100) {\n      console.log(\"condition applies, diff_R0:\",
diff_R0, \", diff_R1:\", diff_R1);\n\n      var newMasg =
JSON.parse(JSON.stringify(msg)); // clone received message\n
newMasg.Set.Name = \"test_with_diff\"; // modify the dataset
name\n      // append additionally data point: 'DIFF'\n
newMasg.Set.Data.push({\n      Name: \"DIFF\", \n
Type: \"float\", \n      Value: diff, \n
Changed: true, \n      });\n      publishReadingSet(newMasg); //
publish dataset\n      }\n\n      state[msg.Source.ClientID]
= dataset;\n      persistState();\n\n      break;\n\n
default:\n      //console.error(\"Unexpected dataset received,
dataset:\", msg.Set.Name);\n      break;\n  }\n}\n\",
  \"subscriptions\": {
    \"alarms\": [],
    \"connectionStateChange\": [],
    \"readingSets\": [],
    \"wildData\": [],
    \"writeDataJobs\": []
  },
  \"state\": {
    \"activated\": true,
    \"serverTimestamp\": \"2022-11-03T08:36:55.307Z\",
    \"status\": \"ok\",
    \"timestamp\": \"2022-11-03T08:36:55.290Z\"
  },
  \"type\": \"scriptlogic\"
},
{
  \"config\": {
    \"envs\": [
      {
        \"key\": \"IOT_MIDDLEWARE_TEMPLATE\",
        \"value\": \"Default\"
      },
      {
        \"key\": \"IOT_OWNER_ID\",
        \"value\": \"eb3b6b5b-8537-4a89-acc1-e99ef1fa4cfa\"
      },
      {
        \"key\": \"IOT_TYPE\",
        \"value\": \"middleware\"
      },
      {
        \"key\": \"IOT_NAME\",
        \"value\": \"Heidenhain\"
      }
    ]
  }
}

```

442

```

console.log(\simplewrite.js - writeData successfull:\",
res);\n      }\n\t}\n}\n",
    "subscriptions": {
      "alarms": [],
      "connectionStateChange": [],
      "readingSets": [],
      "wildData": [],
      "writeDataJobs": []
    }
  },
  "state": {
    "activated": true,
    "serverTimestamp": "2022-11-03T08:37:12.368Z",
    "status": "ok",
    "timestamp": "2022-11-03T08:37:12.348Z"
  },
  "type": "scriptlogic"
},
{
  "config": {
    "envs": [
      {
        "key": "IOT_MIDDLEWARE_TEMPLATE",
        "value": "Default"
      },
      {
        "key": "IOT_OWNER_ID",
        "value": "43033834-b4b1-4d33-b3f5-7033226aac74"
      },
      {
        "key": "IOT_TYPE",
        "value": "middleware"
      },
      {
        "key": "IOT_NAME",
        "value": "test"
      },
      {
        "key": "IOT_ID",
        "value": "179e997a-7249-4b0b-ae9d-41824373568f"
      },
      {
        "key": "IOT_APP_ID",
        "value": "179e997a-7249-4b0b-ae9d-41824373568f"
      },
      {
        "key": "IOT_DNS_NAMES",
        "value": "test7058794f-624c-4901-9c6c-c93d7604838d"
      },
      {
        "key": "IOT_INTERACT",
        "value": "TRUE"
      }
    ]
  }
}

```

```

    }
  ],
  "ports": [],
},
"id": "179e997a-7249-4b0b-ae9d-41824373568f",
"name": "test",
"ownerID": "43033834-b4b1-4d33-b3f5-7033226aac74",
"scriptlogicConfig": {
  "logic": "sent=false;\r\n\r\nstartIntervalWorker(function
(state) {\r\n  console.log(\"example.js - startIntervalWorker
- state:\", JSON.stringify(state));\r\n  \r\n  if (sent ===
true) {\r\n    return;\r\n  }\r\n  var now = new Date();\r\n
var a = [{\r\n    \"Name\": \"subject\", \r\n
\"Type\": \"string\", \r\n    \"Value\": \"Greetings\" \r\n  },
{\r\n    \"Name\": \"body\", \r\n    \"Type\":
\"string\", \r\n    \"Value\": \"Hallo\" \r\n  }];\r\n  sent
= true;\r\n\r\n  publishReadingSet({\r\n    \"Source\":
{\r\n      \"ProjectID\": \"connectivity\", \r\n
\"DeviceType\": \"software\", \r\n    \"ApplicationType\":
\"scriptlogic\", \r\n    \"ClientID\":
\"emailer\" \r\n    }, \r\n    \"Set\": {\r\n
\"Time\": now.toISOString(), \r\n    \"Name\":
\"com.siemens.bfc.models.basic.notification\", \r\n
\"Data\": a \r\n    } \r\n  });\r\n  console.log(\"PUBLISHED!
\");\r\n}, 1000); // This function will be called every 40s",
  "subscriptions": {
    "alarms": [],
    "connectionStateChange": [],
    "readingSets": [],
    "wildData": [],
    "writeDataJobs": []
  }
},
"state": {
  "activated": false,
  "serverTimestamp": "2022-11-02T14:25:32.119Z",
  "status": "stopped",
  "timestamp": "2022-11-02T14:25:32.110Z"
},
"type": "scriptlogic"
},
{
  "config": {
    "envs": [
      {
        "key": "IOT_MIDDLEWARE_TEMPLATE",
        "value": "default"
      },
      {
        "key": "IOT_OWNER_ID",
        "value": "a85105a6-db58-49d1-9e7c-80d1391f3fb2"
      }
    ]
  }
}

```



```

    {
      "key": "IOT_TYPE",
      "value": "middleware"
    },
    {
      "key": "IOT_NAME",
      "value": "readTest"
    },
    {
      "key": "IOT_ID",
      "value": "2318c006-2ba3-420e-bb2b-2885e23b71a0"
    },
    {
      "key": "IOT_APP_ID",
      "value": "2318c006-2ba3-420e-bb2b-2885e23b71a0"
    },
    {
      "key": "IOT_DNS_NAMES",
      "value": "readTestbbbbebc5-2e50-46ed-a727-d7aaecb8246e"
    },
    {
      "key": "IOT_INTERACT",
      "value": "TRUE"
    },
    {
      "key": "IOT_MIDDLEWARE_TEMPLATE",
      "value": "default"
    }
  ],
  "ports": [],
},
"id": "2318c006-2ba3-420e-bb2b-2885e23b71a0",
"name": "readTest",
"ownerID": "a85105a6-db58-49d1-9e7c-80d1391f3fb2",
"scriptlogicConfig": {
  "logic": "// clientID used in configUI\nvar
clientID = \"test\";\n\n// supported ApplicationTypes for
writing data:\n// \t\"mtconnectclient\" \n// \t\"modbusclient\" \n//
\t\"opcuaclient\" \n// \t\"s7client\" \n// \t\"fanucclient\" \n//
\t\"focas1client\" \n// \t\"heidenhainclient\" \n//
\t\"beckhoffclient\" \n// \t\"omronclient\" \n// \t\"eipclient\" \n//
\t\"iotclient\" \n\nvar appType = \"mtconnectclient\";\n\n// Variable
addresses, values & data types to write\n// supported
types are: \"bool\", \"int\", \"float\" and \"string\" \n\nvar
address1 = \"Test01\";\nvar address2 = \"Test02\";\nvar
address3 = \"Test03\";\nvar type1= \"float\";\nvar type2=
\"float\";\nvar type3= \"float\";\n\n// intervalTime in ms to
read data\nvar intervalTime = 1000;\n\n// the IntervalWorker
is called every <intervalTime> ms\nstartIntervalWorker(function
(state) {\n  console.log(\"IntervalWorker: state ->
\", JSON.stringify(state));\n\n\t// DeviceType is for
BFC Client \"machine\". For all other clients

```

```
\\"device\\".\n\tvar deviceType = \\"device\\";\n\tif (appType ==  
\\"iotclient\\") {\n\t\tdeviceType = \\"machine\\"; \n\t}\n\n    var success = readData({\n        \"target\": {\n            \"ProjectID\": \\"connectivity\\",\n            \"DeviceType\": deviceType,\n            \"ApplicationType\": appType,\n            \"ClientID\": clientID\n        },\n        \"ttl\": 5000,\n        \"data\": [{\n            \"address\": address1,\n            \"type\": type1\n        },\n        {\n            \"address\": address2,\n            \"type\": type2\n        },\n        {\n            \"address\": address3,\n            \"type\": type3\n        }\n    ]});\n    for(var i = 0;  
i < success.data.length; i++) {\n        console.log(\\"Value of  
variable '\" + success.data[i].address + '\" (type: '\" +  
success.data[i].type + \") is: '\" + success.data[i].value);\n    }\n    console.log(\\"IntervalWorker(): readData  
successful.\");\n    return;\n}, intervalTime);\n",  
    "subscriptions": {\n        "alarms": [],\n        "connectionStateChange": [],\n        "readingSets": [],\n        "wildData": [],\n        "writeDataJobs": []\n    },  
    state: {\n        "activated": true,  
        "serverTimestamp": "2022-11-03T08:37:12.386Z",  
        "status": "ok",  
        "timestamp": "2022-11-03T08:37:12.372Z"  
    },  
    type: "scriptlogic"  
}
```

Single middleware:

```
GET https://<gateway-address>:<gateway-port>/openness/
components/api/v1/components/middlewares/<id>
```

Example:

```
{
  "config": {
    "envs": [
      {
        "key": "IOT_MIDDLEWARE_TEMPLATE",
        "value": "default"
      },
      {
        "key": "IOT_OWNER_ID",
        "value": "f263c9bc-e1a0-4b04-a58b-8b6bf0e7497c"
      },
      {
        "key": "IOT_TYPE",

```

```

        "value": "middleware"
      },
      {
        "key": "IOT_NAME",
        "value": "pub"
      },
      {
        "key": "IOT_ID",
        "value": "4aa6eab6-2e2c-4fd2-bde4-5fcf748081f5"
      },
      {
        "key": "IOT_APP_ID",
        "value": "4aa6eab6-2e2c-4fd2-bde4-5fcf748081f5"
      },
      {
        "key": "IOT_DNS_NAMES",
        "value": "pubf253cdd7-b7ec-49e6-89b1-3cf021db2faf"
      },
      {
        "key": "TRACE",
        "value": "TRUE"
      },
      {
        "key": "IOT_INTERACT",
        "value": "TRUE"
      }
    ],
    "ports": [],
  },
  "id": "4aa6eab6-2e2c-4fd2-bde4-5fcf748081f5",
  "name": "pubsub",
  "ownerID": "rootid",
  "scriptlogicConfig": {
    "logic": "\nfunction onNewData(topic,payload)
{\n console.log(\"onNewData ( topic:=\", topic,
\", payload:=\", payload, \" )\");\n}\n\nvar
done = subscribeData(\"connectivity.foo.*\",
onNewData );\nconsole.log(\"subscribeData done:=\", done);\n\n//
This function will be called after the
selected time period in ms.\nstartIntervalWorker(function
(state) {\n \n console.log(\"start\");\n
var ret = publishData(\"connectivity.machine.Kai-
NT.iotclient.cmd\", \"deleteBuffer\");\n console.log(\"done - \",
ret);\n\n}, 40000); // This function will be called every 40s",
    "subscriptions": {
      "alarms": [],
      "connectionStateChange": [],
      "readingSets": [],
      "wildData": [],
      "writeDataJobs": []
    }
  }
},

```

```
"state": {
  "activated": false,
  "serverTimestamp": "2022-10-27T14:56:51.990Z",
  "status": "stopped",
  "timestamp": "2022-10-27T14:56:51.943Z"
},
"type": "scriptlogic"
```

Creating middleware

You require the following rights:

- com.siemens.bfc.skills.openness.components.manager

POST <https://<gateway-address>:<gateway-port>/openness/components/api/v1/components/middlewares>

Updating middleware

You require the following rights:

- com.siemens.bfc.skills.openness.components.manager

PUT <https://<gateway-address>:<gateway-port>/openness/components/api/v1/components/middlewares/<id>>

Removing middleware

You require the following rights:

- com.siemens.bfc.skills.openness.components.manager

DELETE <https://<gateway-address>:<gateway-port>/openness/components/api/v1/components/middlewares/<id>>

Script logic configuration example

```
{
  "scriptlogicConfig": {
    "logic": "nisi officia in",
    "simulation": false,
    "subscriptions": {
      "alarms": [
        {
          "clientID": "clientA"
        }
      ],
      "connectionStateChange": [
        {
          "clientID": "clientA"
        }
      ],
      "readingSets": [
        {
          "clientID": "clientA",
          "readingSetNames": [
            "SetA",
            "SetB"
          ]
        }
      ],
      "wildData": [
        {
          "topic": "clientA"
        }
      ],
      "writeDataJobs": [
        {
          "clientID": "clientA"
        }
      ]
    }
  },
  "config": {
    "envs": [
      {
        "key": "TRACE",
        "value": "TRUE"
      }
    ],
    "ports": [
      {
        "type": "tcp",
        "value": 8080
      }
    ]
  },
  "ownerID": "rootid",
```

```

    "name": "postman-scriptlogic",
    "type": "scriptlogic",
  }

```

10.5 Gateway configuration

Gateway details

You require the following rights:

- com.siemens.bfc.skills.openness.components.manager
- OR -
- com.siemens.bfc.skills.openness.components.viewer

All gateways:

GET https://<gateway-address>:<gateway-port>/openness/components/api/v1/components/gateways

At this end point you obtain detailed information about all gateways.

Example:

```

[
  {
    "config": {
      "envs": [],
      "ports": []
    },
    "gatewayConfig": {
      "influxdbGatewayConfig": {
        "alarmTags": [],
        "database": "bfcdatabase",
        "password": "a6f5be82-17c9-4dfc-a14d-72debb16c982",
        "readingTags": [],
        "retentionPolicyDuration": "180d",
        "retentionPolicyName": "sixmonths",
        "savingPeriod": 1000,
        "timeout": 30,
        "url": "http://appdb:8086",
        "useRetentionPolicyAsDefault": false,
        "username": "appsuser"
      },
      "subscriptions": {
        "hfReadingSets": [],
        "readingSets": [
          {
            "clientID": "*",
            "readingSetNames": [
              "#"
            ]
          }
        ]
      }
    }
  }
]

```

```

    ],
    "whitelistedWriteData": []
  },
  {
    "id": "05cd6668-0b8f-4954-abea-8e81629c6341",
    "name": "all-to-internal",
    "ownerID": "rootid",
    "skills": [
      {
        "name":
"com.siemens.bfc.skills.basic.subscribe.reading.sets"
      },
      {
        "name":
"com.siemens.bfc.skills.basic.subscribe.alarm.statechanges"
      },
      {
        "name":
"com.siemens.bfc.skills.advanced.subscribe.hfdata.sets"
      }
    ],
    "state": {
      "activated": true,
      "serverTimestamp": "2022-11-03T08:52:52.459Z",
      "status": "ok",
      "timestamp": "2022-11-03T08:52:52.452Z"
    },
    "type": "influxdb"
  },
  {
    "config": {
      "envs": [],
      "ports": []
    },
    "gatewayConfig": {
      "mindsphereGatewayConfig": {
        "mindSphereConnectionInfo": "{ \"content\": { \"baseUrl\":
\"https://southgate.eu1.mindsphere.io\", \"iat\":
\"eyJraWQiaOiJrZXktaWQtMSIsInR5cCI6IkpXVCIsImFsZyI6IlJTMjU2In0.eyJpc3
MiOiJTB0kiLCJzdWIiOiJkYTNjM2U3YzlkOTE0ZDg5ODFhODU3ZjZhMGVmYTc4OCIsIm
F1ZCI6IkJFJQU0iLCJpYXQiOiJlbnR5bWVudC5mdWVudC5mdWVudC5mdWVudC5mdWVudC
oxNjY1OTg1ODAwLCJqdGkiOiI2NDc1NWE5MS1kMjc0LTQ4NjktOTcyNS0yZmM2MWM5NT
BjNmUiLCJzY29wZSI6IklBVCIsInRlbiI6Im5iZ21zaW90IiwidGVuX2N0eCI6Im1haW
4tdGVuYW50IiwiaWY2xpZW50X2NyZWRLbnRpbWVudC5mdWVudC5mdWVudC5mdWVudC5mdW
JFVCJdLCJzY2h1bWVudC5mdWVudC5mdWVudC5mdWVudC5mdWVudC5mdWVudC5mdWVudC
mnN7RwhpoKdyzSzYDm-qSUAdjg7MTCm7XTH_2-cCOJ-
kD9rCcg_bgz0ZvROD_Ul4tTl1P08WJOV3xxFeyicrBJYmqjlV2KurbZZ3C44XkK0aWAO
hKgxuMVSsIaDWGGvhDx04IKRPSRzuAL4u97rJ-z8MKQw9_hhtOOo-LEBlkaqdL-
N6ZXRAzdguWpiEAQ80seeCP8tSU5FwRcleYp4AQoB6pJ8A7Zi0409kNk6ca3XTZclIyB
NEdoBT9yw90yfp3JUTp3AnybZWUw183rmfDeghDS3ALEJZ1v-680_y5ftC_1ZJHets9M
s47SLAOvYJlHP5IjodREpQc7Wg\", \"clientCredentialProfile\":
[ \"SHARED_SECRET\" ], \"clientId\":

```

10.5 Gateway configuration

```

\"da3c3e7c9d914d8981a857f6a0efa788\",      \"tenant\":
\"nbgmsiot\"  },  \"expiration\": \"2022-10-17T05:50:00.000Z\"}
  },
  \"subscriptions\": {
    \"hfReadingSets\": [],
    \"readingSets\": [
      {
        \"clientID\": \"Heidenhain\",
        \"readingSetNames\": [
          \"#\"
        ]
      }
    ],
    \"whitelistedWriteData\": []
  }
},
\"id\": \"6b8bdf58-0552-48de-bc7e-5e8b75c9fa90\",
\"name\": \"testkeineFunktion\",
\"ownerID\": \"rootid\",
\"skills\": [
  {
    \"name\":
\"com.siemens.bfc.skills.basic.subscribe.reading.sets\"
  },
  {
    \"name\":
\"com.siemens.bfc.skills.basic.subscribe.alarm.statechanges\"
  },
  {
    \"name\":
\"com.siemens.bfc.skills.advanced.subscribe.hfdata.sets\"
  },
  {
    \"name\": \"com.siemens.bfc.skills.advanced.subscribe.files\"
  }
],
\"state\": {
  \"activated\": false,
  \"serverTimestamp\": \"2022-10-24T06:32:25.804Z\",
  \"status\": \"stopped\",
  \"timestamp\": \"2022-10-24T06:32:25.794Z\"
},
\"type\": \"mindsphere\"
},
{
  \"config\": {
    \"envs\": [],
    \"ports\": []
  },
  \"gatewayConfig\": {
    \"mysqlGatewayConfig\": {
      \"database\": \"bfcdatabase\",

```



```

        "maxConnections": 1,
        "password": "pw",
        "port": 3306,
        "protocolType": "tcp",
        "url": "192.168.5.95",
        "username": "user"
    },
    "subscriptions": {
        "hfReadingSets": [],
        "readingSets": [
            {
                "clientID": "*",
                "readingSetNames": [
                    "#"
                ]
            }
        ]
    },
    "whitelistedWriteData": []
}
},
"id": "9b6e3a6a-cc08-4549-8dff-e5de4a4e5f29",
"name": "all-to-mysql",
"ownerID": "rootid",
"skills": null,
"state": {
    "serverTimestamp": "0001-01-01T00:00:00.000Z",
    "status": "running",
    "timestamp": "2022-11-03T08:53:11.933Z"
},
"type": "mysql"
}
]

```

Single gateway:

GET https://<gateway-address>:<gateway-port>/openness/components/api/v1/components/gateways/<id>

Example:

```

{
  "config": {
    "envs": [],
    "ports": []
  },
  "gatewayConfig": {
    "mysqlGatewayConfig": {
      "database": "bfcdatabase",
      "maxConnections": 1,
      "password": "pw",
      "port": 3306,
      "protocolType": "tcp",
      "url": "192.168.5.95",
      "username": "user"
    }
  }
}

```

```

    },
    "subscriptions": {
      "hfReadingSets": [],
      "readingSets": [
        {
          "clientID": "*",
          "readingSetNames": [
            "#"
          ]
        }
      ],
      "whitelistedWriteData": []
    },
    "id": "9b6e3a6a-cc08-4549-8dff-e5de4a4e5f29",
    "name": "all-to-mysql",
    "ownerID": "rootid",
    "skills": null,
    "state": {
      "serverTimestamp": "0001-01-01T00:00:00.000Z",
      "status": "running",
      "timestamp": "2022-11-03T08:54:58.568Z"
    },
    "type": "mysql"
  }
}

```

Creating a gateway

You require the following rights:

- com.siemens.bfc.skills.openness.components.manager

POST <https://<gateway-address>:<gateway-port>/openness/components/api/v1/components/gateways>

Updating a gateway

You require the following rights:

- com.siemens.bfc.skills.openness.components.manager

PUT <https://<gateway-address>:<gateway-port>/openness/components/api/v1/components/gateways/<id>>

Remove gateway

You require the following rights:

- com.siemens.bfc.skills.openness.components.manager

DELETE <https://<gateway-address>:<gateway-port>/openness/components/api/v1/components/gateways/<id>>

Gateway configuration examples

MQTT gateway

```
{
  "gatewayConfig": {
    "mqttGatewayConfig": {
      "url": "mqttUrl",
      "mqttTopicPrefix": "bfc",
      "mqttQoS": 1,
      "mqttTimeout": 200000000000,
      "skipTLS": true,
      "mqttBrokerUsername": "user",
      "mqttBrokerPassword": "pass",
      "mqttClientID": "client",
      "caCert": "ca",
      "clientCert": "cert",
      "clientKey": "key",
      "retain": true
    },
    "subscriptions": {
      "alarm": "ClientA",
      "hfReadingSets": [
        {
          "clientID": "ClientA",
          "readingSetNames": [
            "SetA", "SetB"
          ]
        }
      ],
      "readingSets": [
        {
          "clientID": "ClientA",
          "readingSetNames": [
            "SetA", "SetB"
          ]
        }
      ],
      "whitelistedWriteData": [
        {
          "address": "Byte0",
          "clientID": "ClientA",
          "type": "int"
        }
      ]
    },
    "ownerID": "rootid",
    "name": "postman-mqtt-gateway",
    "type": "mqtt"
  }
}
```

```
{
  "gatewayConfig": {
    "mindsphereGatewayConfig": {
      "mindSphereConnectionInfo": "{ \"content\":
{ \"baseUrl\": \"https://southgate.eu1.mindsphere.io\",
\"iat\":
\"eyJraWQiOiJrZXktaWQtMSIsInR5cCI6IkpXVCIsImFsZyI6IlJTMjU2In0.eyJpc3MiOiJBTQ0kiLCJzdWIiOiJjM2FlMTI1ODcyNjE0NDZlOTNmZGNiMDRmZjY2MmQ4ZCIsImF1ZCI6IkJFJQU0iLCJpYXQoOiJlNjc3NjU4NDUsIm5iOiI6MTU2Nzc2NTg0NSwiZXhwIjoxNTY4MzcwNjQ1LCJqdGkiOiIyNzYyOTI3YS1hOWIyLTRmODItOWIwOS00NmFmMTU0NGU5MTkiLCJzY29wZSI6IklBVCIsInRlbiI6Im1vZGV2IiwidGVuX2N0eCI6Im1haW4tdGVuYW50IiwiiY2xpZW50X2NyZXXXWRLbnRpyWxzX3Byb2ZpbGUiOlsiU0hBUkVEX1NFQ1JFVCJdLCJzY2h1bWZlIjpbInVybjpzaWVtZW5zOm1pbmRzcGhlcmU6dGEiXX0.mf6tyKjhl_3T_eVA7i2_5izPYzk2r2zEBNLePRlLQRrGqsD4uT_2TvkizA_SBHI6TDZi_BaEr6tdQtekTXMuQjbhnaW7C4AArj7tcjRgt9fds6H5UdqI5Fcb62NqT_qQwbvzrcXlZvPbrtKHYFy34hwZEyuiViIvCb7Xhj096T-81a2c0-xhkB50U05_ejv1bVdkXMz9AJbrdcEYoIgv2SK1M7YzeOZUq2XEApP-OPKyYabmI-Zk6hg4iz5iKmzv2bXGg_q90-HTQhTt7yEukQ41id2PzdwhegMPB5nqobdXREKhaX15ViwZerD3Rv3CHL5d0ZZIXizTi4A55shMZw\", \"clientCredentialProfile\":
[ \"SHARED_SECRET\" ], \"clientId\":
\"c3ael2587261446e93fdbcb04ff662d8dx\", \"tenant\": \"XXXX\" },
\"expiration\": \"2019-09-13T10:30:45.000Z\"}]",
      "proxyAddress": "http://192.168.0.9:800"
    },
    "subscriptions": {
      "alarm": "ClientA",
      "hfReadingSets": [
        {
          "clientID": "ClientA",
          "readingSetNames": [
            "SetA", "SetB"
          ]
        }
      ],
      "readingSets": [
        {
          "clientID": "ClientA",
          "readingSetNames": [
            "SetA", "SetB"
          ]
        }
      ],
      "whitelistedWriteData": [
        {
          "address": "Byte0",
          "clientID": "ClientA",
          "type": "int"
        }
      ]
    }
  }
}
```

```
    },  
    "ownerID": "rootid",  
    "name": "postman-mindsphere-gateway",  
    "type": "mindsphere"  
  }  
}
```

MySQL gateway

```
{  
  "gatewayConfig": {  
    "mysqlGatewayConfig": {  
      "url": "url",  
      "port": 123,  
      "protocolType": "tcp",  
      "username": "user",  
      "password": "pass",  
      "database": "db",  
      "maxConnections": 5,  
      "parameters": "?test=ok"  
    },  
    "subscriptions": {  
      "alarm": "ClientA",  
      "hfReadingSets": [  
        {  
          "clientID": "ClientA",  
          "readingSetNames": [  
            "SetA", "SetB"  
          ]  
        }  
      ],  
      "readingSets": [  
        {  
          "clientID": "ClientA",  
          "readingSetNames": [  
            "SetA", "SetB"  
          ]  
        }  
      ],  
      "whitelistedWriteData": [  
        {  
          "address": "Byte0",  
          "clientID": "ClientA",  
          "type": "int"  
        }  
      ]  
    }  
  },  
  "ownerID": "rootid",  
  "name": "postman-mysql-gateway",  
  "type": "mysql"  
}
```

AMP4.1 Gateway

```

{
  "gatewayConfig": {
    "amp41GatewayConfig": {
      "url": "url",
      "port": 8080,
      "ampPath": "path",
      "ampLightName": "light",
      "ampKeepAliveInterval": 2000,
      "ampTimeout": 2000,
      "ampSendingPeriod": 2000,
      "ampDisableBuffering": true,
      "ampBuffer": 2000,
      "passthroughMode": true,
      "ampMachineName": "machineName",
      "caCert": "ca"
    },
    "subscriptions": {
      "alarm": "ClientA",
      "hfReadingSets": [
        {
          "clientID": "ClientA",
          "readingSetNames": [
            "SetA", "SetB"
          ]
        }
      ],
      "readingSets": [
        {
          "clientID": "ClientA",
          "readingSetNames": [
            "SetA", "SetB"
          ]
        }
      ],
      "whitelistedWriteData": [
        {
          "address": "Byte0",
          "clientID": "ClientA",
          "type": "int"
        }
      ]
    },
    "ownerID": "rootid",
    "name": "postman-amp41-gateway",
    "type": "amp41"
  }
}

```

InfluxDB gateway

```

{
  "gatewayConfig": {

```

```

"influxdbGatewayConfig": {
  "url": "url",
  "database": "db",
  "savingPeriod": 2000,
  "timeout": 2000,
  "retentionPolicyName": "name",
  "retentionPolicyDuration": "duration",
  "useRetentionPolicyAsDefault": false,
  "username": "user",
  "password": "pass",
  "caCert": "ca",
  "readingTags": [
    "readingTag1",
    "readingTag2"
  ],
  "alarmTags": [
    "alarmTag1",
    "alarmTag2"
  ]
},
"subscriptions": {
  "alarm": "ClientA",
  "hfReadingSets": [
    {
      "clientID": "ClientA",
      "readingSetNames": [
        "SetA", "SetB"
      ]
    }
  ],
  "readingSets": [
    {
      "clientID": "ClientA",
      "readingSetNames": [
        "SetA", "SetB"
      ]
    }
  ],
  "whitelistedWriteData": [
    {
      "address": "Byte0",
      "clientID": "ClientA",
      "type": "int"
    }
  ]
},
"ownerID": "rootid",
"name": "postman-influxdb-gateway",
"type": "influxdb"
}

```

Elasticsearch gateway

```
{
  "gatewayConfig": {
    "elasticsearchGatewayConfig": {
      "urls": "url1",
      "indexPrefix": "bfc",
      "savingPeriod": 2000,
      "timeout": 2000,
      "skipTLS": true,
      "username": "user",
      "password": "pass",
      "caCert": "ca"
    },
    "subscriptions": {
      "alarm": "ClientA",
      "hfReadingSets": [
        {
          "clientID": "ClientA",
          "readingSetNames": [
            "SetA", "SetB"
          ]
        }
      ],
      "readingSets": [
        {
          "clientID": "ClientA",
          "readingSetNames": [
            "SetA", "SetB"
          ]
        }
      ],
      "whitelistedWriteData": [
        {
          "address": "Byte0",
          "clientID": "ClientA",
          "type": "int"
        }
      ]
    }
  },
  "ownerID": "rootid",
  "name": "postman-elasticsearch-gateway",
  "type": "elasticsearch"
}
```

HTTP gateway

```
{
  "gatewayConfig": {
    "httpGatewayConfig": {
      "url": "url",
      "timeout": 2000,
      "skipTLS": false,

```



```

        "username": "user",
        "password": "pass",
        "caCert": "ca",
        "alarmlistsMethod": "PUT",
        "alarmlistsPath": "alarmlistsPath",
        "alarmstatechangesMethod": "PUT",
        "alarmstatechangesPath": "alarmstatechangesPath",
        "datasetsMethod": "PUT",
        "datasetsPath": "datasetsPath",
        "hfdatasetsMethod": "PUT",
        "hfdatasetsPath": "hfdatasetsPath"
    },
    "subscriptions": {
        "alarm": "ClientA",
        "hfReadingSets": [
            {
                "clientID": "ClientA",
                "readingSetNames": [
                    "SetA", "SetB"
                ]
            }
        ],
        "readingSets": [
            {
                "clientID": "ClientA",
                "readingSetNames": [
                    "SetA", "SetB"
                ]
            }
        ],
        "whitelistedWriteData": [
            {
                "address": "Byte0",
                "clientID": "ClientA",
                "type": "int"
            }
        ]
    },
    "ownerID": "rootid",
    "name": "postman-http-gateway",
    "type": "http"
}

```

MSSQL gateway

```

{
    "gatewayConfig": {
        "mssqlGatewayConfig": {
            "ip": "ip",
            "protocol": "sqlserver",
            "username": "user",
            "password": "pass",

```

```

        "params": "params=test",
        "maxConnections": 1,
        "doNotCreateDatabase": true,
        "doNotCreateTables": true
    },
    "subscriptions": {
        "alarm": "ClientA",
        "hfReadingSets": [
            {
                "clientID": "ClientA",
                "readingSetNames": [
                    "SetA", "SetB"
                ]
            }
        ],
        "readingSets": [
            {
                "clientID": "ClientA",
                "readingSetNames": [
                    "SetA", "SetB"
                ]
            }
        ],
        "whitelistedWriteData": [
            {
                "address": "Byte0",
                "clientID": "ClientA",
                "type": "int"
            }
        ]
    },
    "ownerID": "rootid",
    "name": "postman-mssql-gateway",
    "type": "mssql"
}

```

Kafka gateway

```

{
    "gatewayConfig": {
        "kafkaGatewayConfig": {
            "brokers": "brokers",
            "authorizationType": "NONE",
            "username": "user",
            "password": "pass",
            "baseTopic": "baseTopic",
            "caCert": "ca",
            "balancer": "leastbytes",
            "batchSize": 100,
            "batchBytes": 1048576,
            "maxAttempts": 10,
            "requiredAcks": -1,

```

```

        "batchTimeout": 1000,
        "readTimeout": 1000,
        "writeTimeout": 1000,
        "skipTLS": true,
        "clientCert": "clientCert",
        "clientKey": "clientKey"
    },
    "subscriptions": {
        "alarm": "ClientA",
        "hfReadingSets": [
            {
                "clientID": "ClientA",
                "readingSetNames": [
                    "SetA", "SetB"
                ]
            }
        ],
        "readingSets": [
            {
                "clientID": "ClientA",
                "readingSetNames": [
                    "SetA", "SetB"
                ]
            }
        ],
        "whitelistedWriteData": [
            {
                "address": "Byte0",
                "clientID": "ClientA",
                "type": "int"
            }
        ]
    },
    "ownerID": "rootid",
    "name": "postman-kafka-gateway",
    "type": "kafka"
}

```


File transfer

The BFC Gateway provides you with the opportunity of accessing the files of connected machines.

The following machine control systems are supported:

- SINUMERIK control systems that are connected via the BFC client
- FANUC control systems that are connected via the FOCAS2 interface
- Heidenhain control systems, if not inhibited by the machine OEM
- Control systems that can be connected using the SFTP protocol

NOTICE

Possible damage when modifying programs

With FANUC controls, directories below CNC_MEM can be accessed. A program that is in editing mode or currently active on the control may not be modified. The program file name and the name within the program data must also match.

For Heidenhain control systems, the directories below TNC:\ can be accessed. A program that is in editing mode or currently active on the control may not be modified.

Note

Maximum file size

The BFC Gateway supports files up to a maximum size of 200 MB. In some instances, reduced values are obtained for individual control system types:

- SINUMERIK control systems, max. 200 MB
- SFTP devices, max. 200 MB
- Heidenhain control systems, max. 10 MB
- FANUC control systems, max. 20 MB

For SINUMERIK control systems that are connected via the BFC client, NC files and files on the HMI (Windows/Linux) can be accessed. No network sharing is used on the machine for Windows-based HMI systems.

You can access the file of a connected machine using the WebDAV protocol (<https://en.wikipedia.org/wiki/WebDAV>). The WebDAV protocol is based on HTTP(S) and offers the same security standards (encryption with TLS 1.3 and authentication with user name and password). All typical operations relating to files and directories of a connected machine can be executed using the WebDAV protocol (reading, writing, renaming, deleting files and directories). Regarding files, the BFC Gateway does not have its own data management, and solely facilitates transparent access to the files of connected machines. Within the BFC configuration, access rights can be defined for individual directories (see Configuring access rights (Page 467)).

Using the WebDAV interface can be configured differently from the user's perspective. An overview of possible usage scenarios is provided in the following:

- In your application, a file is transferred to the machine using HTTP-PUT. To do this, in the customer application only an HTTP-PUT request has to be executed. In this particular case, it is not necessary to be knowledgeable about the WebDAV protocol. Examples are provided under Using the WebDAV interface with curl (Page 470).
- A WebDAV client library is used in your application in order to fully utilize the functional scope of the WebDAV interface. Numerous WebDAV client libraries exist for the various programming environments. When using a WebDAV client library, for example, the content of directories can be listed and file attributes determined (size and date of the last change).
- The WebDAV interface of the BFC Gateway is integrated as a network drive on your computer (Windows or Linux) (see Using the WebDAV interface as Windows network drive (Page 470) and Using the WebDAV interface under Linux (Page 472)). All the files of the connected machines are visible under the integrated network drive, and can be edited/processed in the usual way.
- You are using a WebDAV client application to perform manual file operations on the machines. For example, one of the numerous WebDAV client applications is WinSCP (see Using the WebDAV interface with WinSCP (Page 473)).

Several applications can use the WebDAV interface of the BFC Gateway in parallel. Access is realized via the following URL:

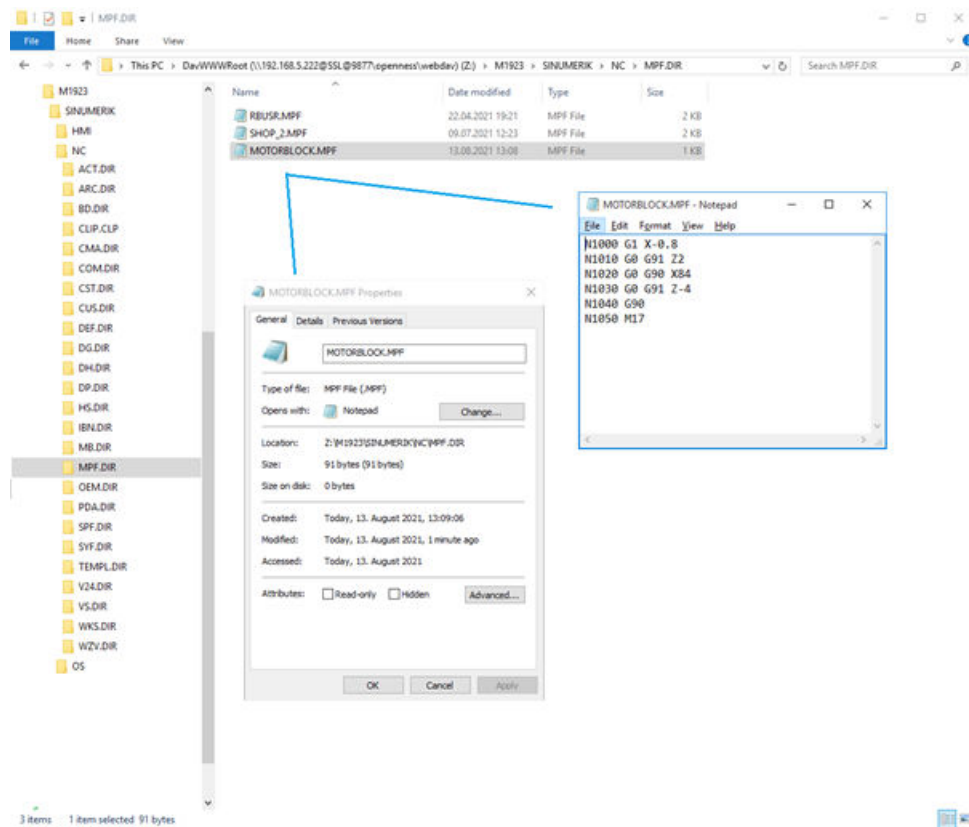
`https://<bfc-gateway>:9877/openness/webdav/<equipment-path>/<file-path>`

bfc-gateway	IP address or host name of the BFC Gateway
equipment-path	Machine path in the system hierarchy Example of a system hierarchy:  equipment-path: /M1923/SINUMERIK
file-path	Machine-specific path of a file Example for SINUMERIK NC file: /NC/MPF.DIR/MOTORBLOCK.MPF

Complete example of a WebDAV URL, and the associated display in Windows Explorer:

`https://192.168.5.222:9877/openness/webdav/M1923/SINUMERIK/NC/MPF.DIR/MOTORBLOCK.MPF`

11.1 Configuring access rights

**Note**

The following access data to the BFC Gateway are used in this chapter for documentation reasons:

User name: **admin**

Password: **admin**

11.1 Configuring access rights

The access rights are defined in BFC ConfigUI in the "Commissioning" area.



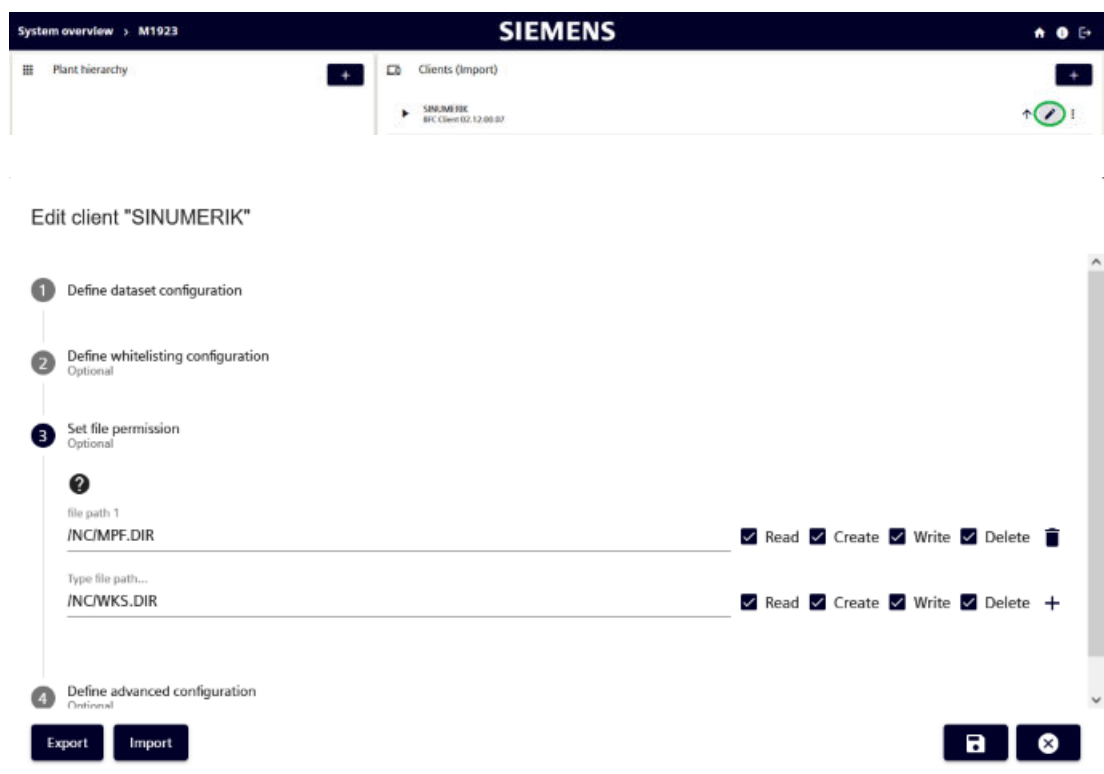
11.1 Configuring access rights

A directory can be assigned the following authorization levels:

- **READ:** Allows directories and files to be read.
- **CREATE:** Allows new directories and files to be created. However, existing files and directories cannot be changed.
- **WRITE:** Allows existing files to be overwritten and files and directories to be renamed. However, it is not permissible to delete files and directories.
- **DELETE:** Allows files and directories to be deleted.

The assigned authorization levels refer to the configured directories and all subdirectories.

An example is shown below in which the complete authorization levels for directories /NC/MPF.DIR and /NC/WKS.DIR were configured.



For a newly linked machine, as default setting, no authorization levels are configured. As a consequence, files cannot be read or written to. As a minimum, the READ authorization for a directory must first be configured before this directory can be accessed via WebDAV.

Directory "/" can be used to assign authorization levels to all files of a machine. All authorization levels for all files of a machine were configured in the following example.

Edit client "SINUMERIK"

- 1 Define dataset configuration
- 2 Define whitelisting configuration
Optional
- 3 Set file permission
Optional

Type file path...

/

☒ Read ☒ Create ☒ Write ☒ Delete +
- 4 Define advanced configuration
Optional

Export Import

NOTICE**Potential damage to the machine by changing or deleting existing data**

Access rights "Write" and "Delete", allow files in the control system to be changed and/or deleted. This can damage the machine. From version 1.11, when using "Write" and "Delete" access rights, it is necessary to confirm an appropriate note.

Example:

3 Set file permission
Optional

Type file path...

/NC/WORKS.DIR

☒ Read ☒ Create ☒ Write ☒ Delete +

☒ Warning regarding machine safety has been considered. Date: 24.10.2022, 11:29:15

With the access rights "Write" and "Delete" it is possible to change or delete files on the controller. This can lead to unwanted data loss and possible machine damage.

Make sure that the configured file access in your application scenario does not lead to any undesired behavior on the machine.

The manufacturer of the software (Siemens AG) is not liable for damages caused by improper file access.

Note

For SINUMERIK control systems, changes can never be made in the following directories:

- /NC/CST.DIR/ - standard NC cycles
- /NC/CMA.DIR/ - manufacturer NC cycles

11.2 Using the WebDAV interface with curl

curl is an open source tool that is available for Windows and Linux. This means that it is possible to manually execute HTTP-GET or HTTP-PUT requests. Below are two examples to read and write to/from file MOTORBLOCK.MPF using curl:

```

Microsoft Windows [Version 10.0.17763.2001]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users>curl -k -i -u admin:admin "https://192.168.5.222:9877/openness/webdav/M1923/SINUMERIK/NC/MPF.DIR/MOTORBLOCK.MPF"
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 13 Aug 2021 13:31:23 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 91
Connection: keep-alive
Accept-Ranges: bytes
Etag: "169adfac72b9be005b"
Last-Modified: Fri, 13 Aug 2021 13:09:07 GMT
Strict-Transport-Security: max-age=63072000
Content-Security-Policy: default-src 'self' 'unsafe-eval' 'unsafe-inline'; img-src 'self' data;;
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block

N1000 G1 X-0.8
N1010 G0 G91 Z2
N1020 G0 G90 X84
N1030 G0 G91 Z-4
N1040 G90
N1050 M17

C:\Users>

```

```

Microsoft Windows [Version 10.0.17763.2001]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users>curl -X PUT -k -i -u admin:admin -T "C:\Tests\MOTORBLOCK.MPF" "https://192.168.5.222:9877/openness/webdav/M1923/SINUMERIK/NC/MPF.DIR/MOTORBLOCK.MPF"
HTTP/1.1 100 Continue

HTTP/1.1 201 Created
Server: nginx
Date: Fri, 13 Aug 2021 13:41:30 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 7
Connection: keep-alive
Etag: "169adfac72b9be005b"
Strict-Transport-Security: max-age=63072000
Content-Security-Policy: default-src 'self' 'unsafe-eval' 'unsafe-inline'; img-src 'self' data;;
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block

Created
C:\Users>

```

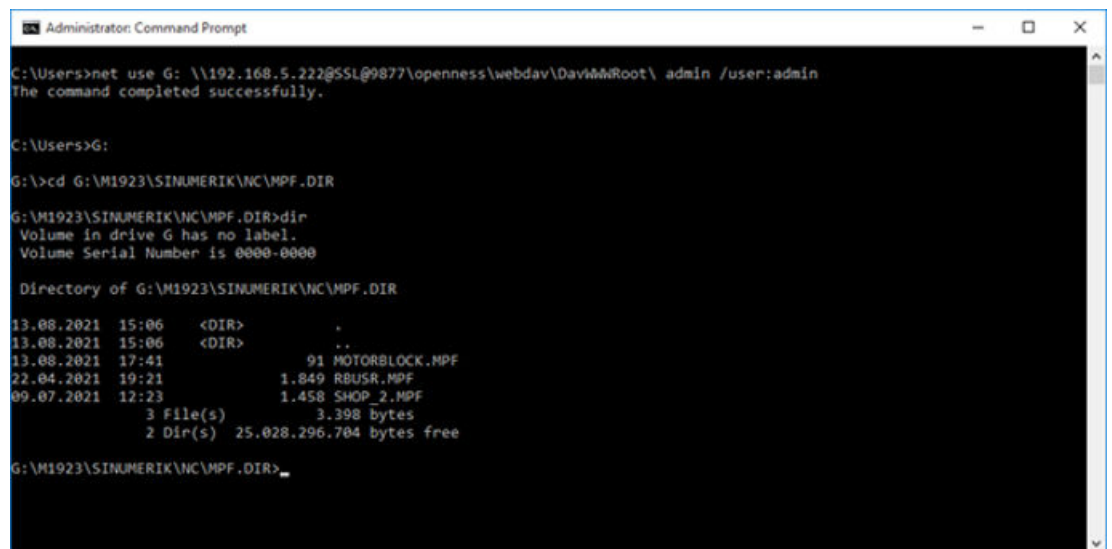
11.3 Using the WebDAV interface as Windows network drive

You can integrate a WebDAV data source as network drive in all current Windows versions (Windows 10, Windows Server 2016 and 2019). Command `net use` is used to do this.

Example:

```
net use [FREE DRIVE LETTER]: \\[IP ADDRESS OF THE
CONTROL]@SSL@9877\openness\webdav\DavWWWRoot\ admin /user:admin
```

11.3 Using the WebDAV interface as Windows network drive



```
Administrator: Command Prompt

C:\Users>net use G: \\192.168.5.222@SSL@9877\openness\webdav\DevWWWRoot\ admin /user:admin
The command completed successfully.

C:\Users>G:

G:\>cd G:\M1923\SINUMERIK\NC\MPF.DIR

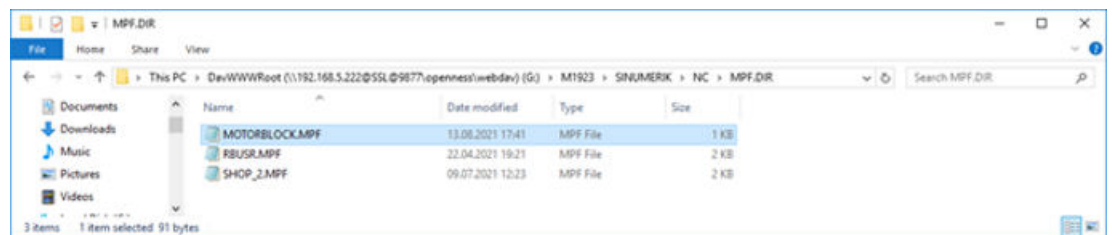
G:\M1923\SINUMERIK\NC\MPF.DIR>dir
Volume in drive G has no label.
Volume Serial Number is 0000-0000

Directory of G:\M1923\SINUMERIK\NC\MPF.DIR

13.08.2021  15:06    <DIR>          .
13.08.2021  15:06    <DIR>          ..
13.08.2021  17:41                91 MOTORBLOCK.MPF
22.04.2021  19:21             1.849 RBUSR.MPF
09.07.2021  12:23             1.458 SHOP_2.MPF
               3 File(s)              3.398 bytes
               2 Dir(s)  25.028.296.704 bytes free

G:\M1923\SINUMERIK\NC\MPF.DIR>
```

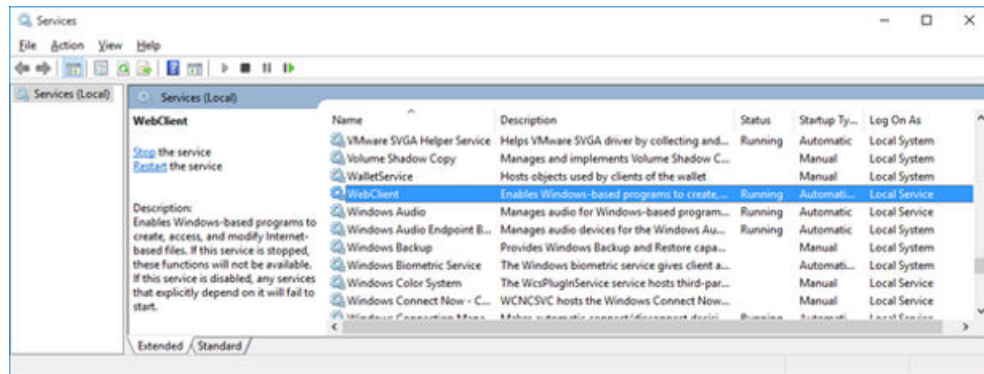
After executing the command, in the Windows Explorer drive G: is available, for example.



11.4 Using the WebDAV interface under Linux

The following steps must be performed before setting up a WebDAV network drive:

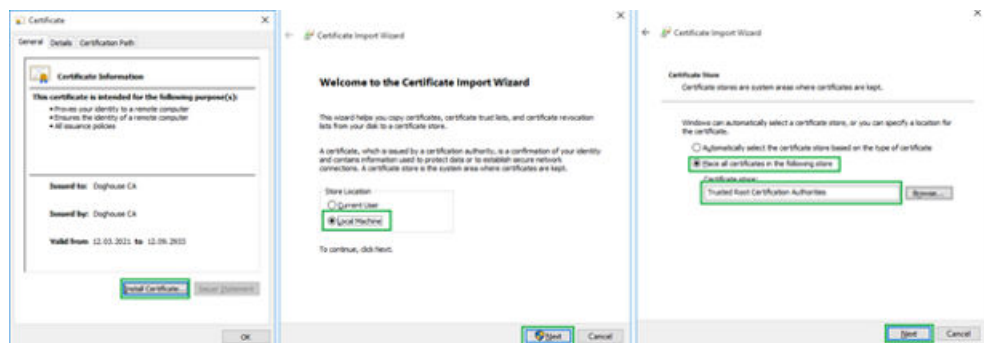
- Check whether the web client service is active on the Windows system on which you wish to setup the network drive.



- In order to be able to establish an encrypted connection from the Windows system – on which you wish to set up the network drive – to the BFC Gateway, the CA certificate of the BFC Gateway must be imported into this system. In the BFC ConfigUI, the CA certificate of the BFC Gateway can be downloaded under "Download CA Certificate".



The CA certificate that was downloaded must be installed on the Windows system on which you wish to set up the network drive. To do this, double-click on the CRT file.



- The following setting must be changed in the Windows registry:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WebClient\Parameters\SupportLocking
Set the variable to the value "0".

11.4 Using the WebDAV interface under Linux

Command `mount.davfs` is used to integrate a WebDAV data source into the Linux file system.

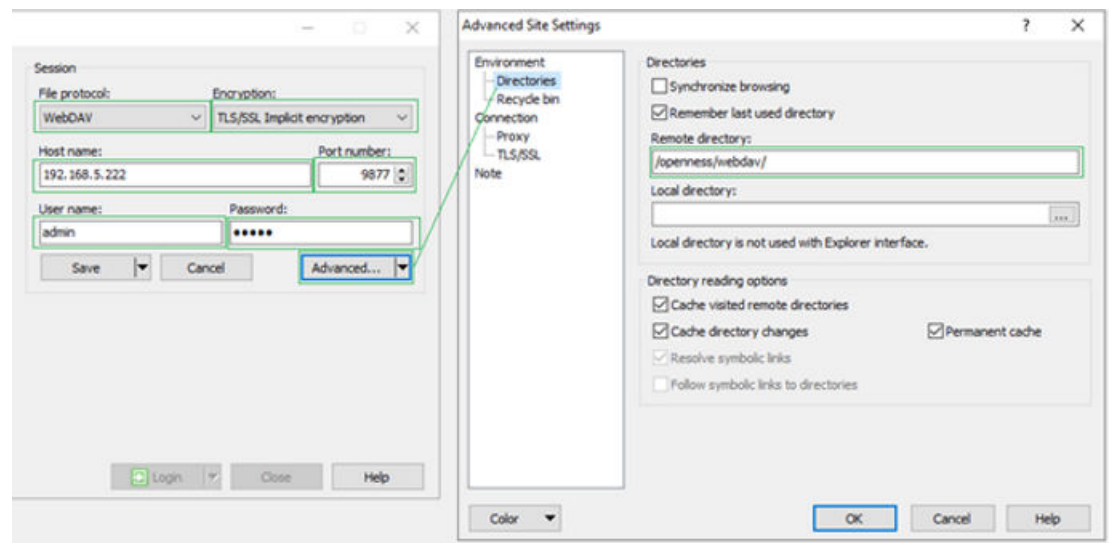
To import the CA certificate of the BFC gateway into your Linux distribution, you can use the description provided under one of the following links:

- <https://linux.die.net/man/8/mount.davfs>
- <https://sleeplessbeastie.eu/2017/09/04/how-to-mount-webdav-share/>
- <https://manpages.debian.org/unstable/davfs2/mount.davfs.8.en.html>
- <http://manpages.ubuntu.com/manpages/impish/man8/mount.davfs.8.html>
- <https://manpages.ubuntu.com/manpages/impish/man8/update-ca-certificates.8.html>

11.5 Using the WebDAV interface with WinSCP

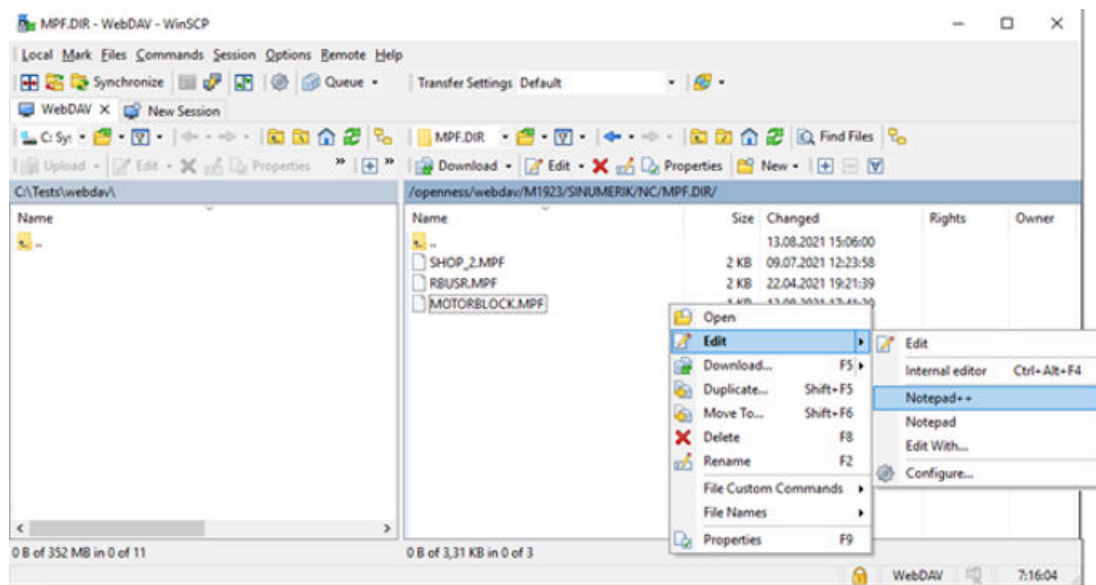
The WinSCP (<https://winscp.net>) tool supports various network protocols, including the WebDAV protocol. Using WinSCP, it is possible to access the WebDAV interface of the BFC gateway, and execute file operations on the connected machines. To do this, at least one directory with a READ authorization level must be configured on the required machine in the BFC gateway (see Configuring access rights (Page 467)).

The settings necessary to establish a WebDAV connection to the BFC gateway using WinSCP are subsequently listed:



The directories and files of a WebDAV data source are shown in the following example:

11.6 Directory structure on SINUMERIK machines



You can execute various file operations via the shortcut menu.

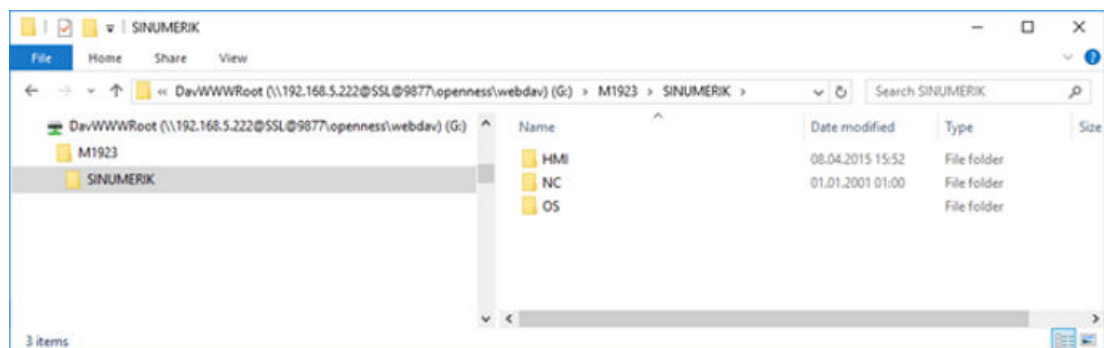
11.6 Directory structure on SINUMERIK machines

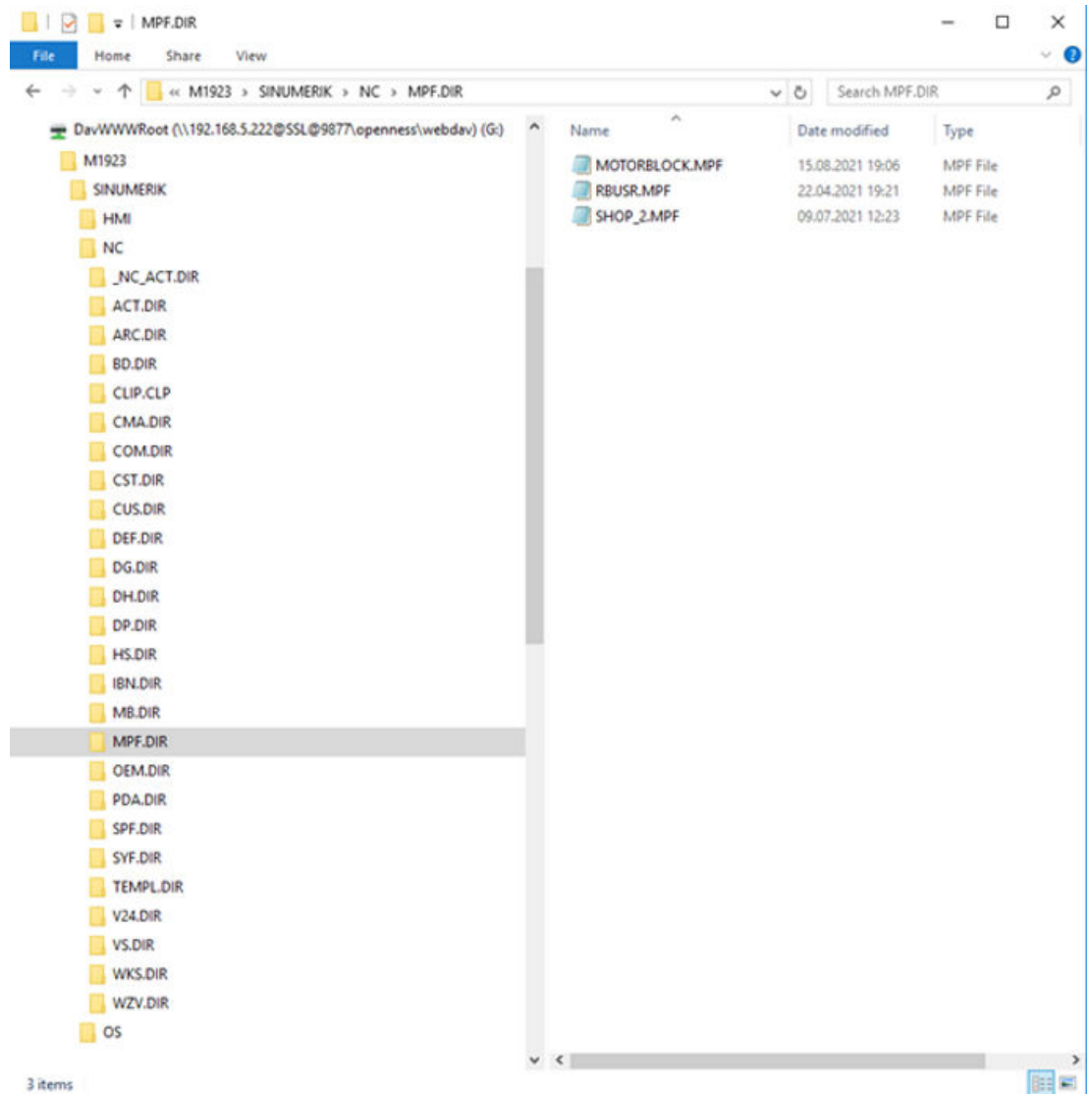
Machines equipped with a SINUMERIK control, which are connected to the BFC gateway via the BFC client, offer a standard view of the machine files via the BFC file functions.

The machine-specific root directory includes three virtual directories, which map the various file categories:

- NC: All NC files (NC programs, etc.) are mapped in this directory.
- OS: All files of the operating components are mapped in this directory. For a window system, all partitions are mapped here. For a Linux system, the content of the CF card is mapped here.
- HMI: Specific files for the HMI operating software are mapped in this directory (SINUMERIK Advanced or SINUMERIK Operate).

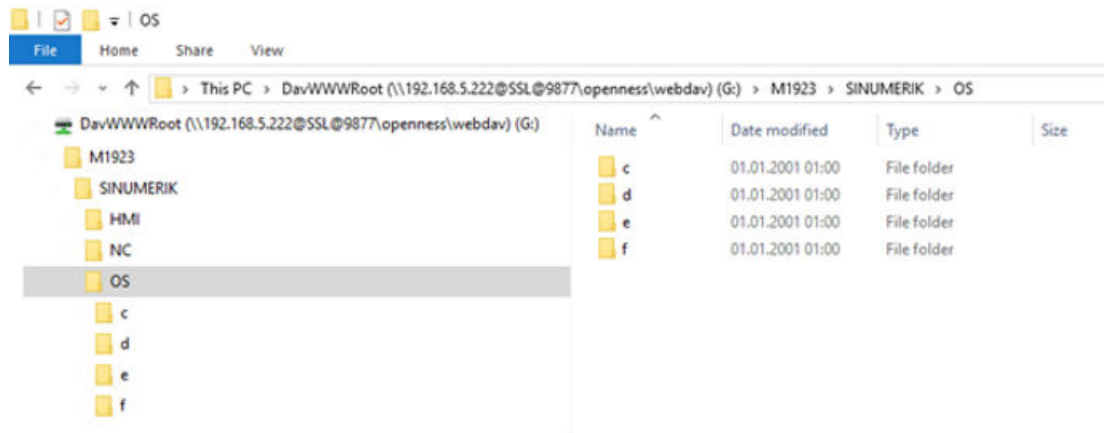
Example: Machine-specific root directory



Example: Part program - directory of the NC**Note**

NC files are accessed via the WebDAV interface, independent of which protection level is active on the SINUMERIK control.

Example: Mapping the Windows partitions as subdirectories of the virtual OS directory



Special feature of the HMI-Advanced operating software

For NC files, the SINUMERIK operating software "HMI-Advanced" makes a distinction between the "Loaded" and "Unloaded" states. This is mapped in HMI-Advanced in the corresponding "Loaded" column. For access operations via the WebDAV interface, it cannot be identified whether a file is "loaded".



In conjunction with this, the following rules apply:

- If an existing NC file is overwritten via the WebDAV interface, then this does not change either the "Loaded" or "Unloaded" state. The existing file is overwritten in the state in which it currently is.
- If a new NC file is created on a machine, then this is realized depending on the configuration entry shown below in the BFC client INI file in the "Loaded" or "Unloaded" state. Die BFC client INI file is located in the BFC installation directory on the control involved.

```
; -----  
; New NC programs can be created either in the NC memory or on the HMI hard drive.  
; This setting defines the used location. Allowed values: HMI NC  
; default: HMI  
; -----  
CreateNewFilesOn = HMI
```

- Parameter to create new NC files in state "Unloaded" (data memory in the Windows file system of the control):
CreateNewFilesOn = HMI
- Parameter to create new NC files in state "Loaded" (NC data memory):
CreateNewFilesOn = NC

Configuring the SSA gateway

12.1 Overview

To connect a SINUMERIK controller to SSA via the BFC gateway, you carry out the following steps once:

- Creating the asset type "bfc_ssa_sinumerik" (Page 481)
- Connecting a new machine to SSA (Page 484)
- Configuring the BFC client data acquisition (Page 495)
- Creating and saving a machine identity (Page 499)

12.2 Requirement

The following preconditions must be fulfilled:

- The BFC client is connected to the BFC gateway.
- MindSphere Tenant is available and set up with MindSphere Application SSA activated.

Note

MindSphere

Configuration activities in MindSphere are not part of this documentation.

For more information about the MindSphere Application SSA, please follow this link (<https://support.industry.siemens.com/cs/ww/en/sc/5369>).

- Middleware "SSA Service", "Scriptlogic" and "machinestatus-sinumerik" are active. More information on this topic can be found in Chapter: Check status of the middleware (Page 479).
- The SSA functionality is available for all SINUMERIK control systems (pl, sl, 828 and ONE).

12.3 Check status of the middleware

Proceed as follows to check whether middleware "machinestatus-sinumerik" is active.

Procedure

1. Establish a connection through WinSCP or PuTTY.
2. Navigate to the "bfc/docker/scriptlogic/samples/logicconfigurations" folder.
3. Open file "machinestatus-sinumerik.json".
4. Change value "active" from "false" to "true".

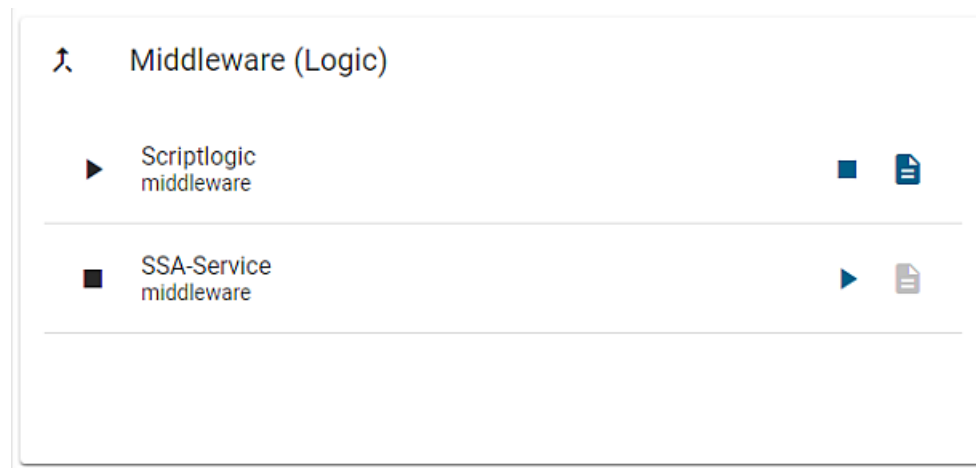
To check whether the middleware "SSA Service" and "Scriptlogic" are active, proceed as follows.

Note

In the development of script logic, the source code produced must be tested exhaustively and verified before productive use.

Procedure

1. Open the user interface of the BFC gateway.
2. Select the "Commissioning" area.
 - In the area "Middleware (Logic)", you will find the entries "SSA-Service" and "Scriptlogic".
 - You can recognize the current status by the symbols to the left of "SSA Service" and "Scriptlogic":
 The symbol "Running" means "active".
 The symbol "Stopped" means "not active".



12.4 Creating aspects in MindSphere

The following aspects must be created in MindSphere.

Name	Aspect	Category	Type
AgentOnlineStatus	core.agentstatus	Dynamic	Defined
Alarms	core.sinumerikbasicalarms	Dynamic	Defined
CH1_BasicConfig	core.sinumerikbasicconfig	Dynamic	Defined
CH1_MachineStatus	core.sinumerikbasicmachinestatus	Dynamic	Defined
connectivityStatus	core.connectivitystatus	Static	Inherited
CSM_AX01	mmmdev.CSM_AX01	Dynamic	Defined
CSM_AX02	mmmdev.CSM_AX02	Dynamic	Defined

12.5 Creating the asset type "bfc_ssa_sinumerik"

Name	Aspect	Category	Type
CSM_General_Info	mmmdev.CSM_General_Info	Dynamic	Defined
CSM_SP01	mmmdev.CSM_SP01	Dynamic	Defined
MachineModel	core.sinumerikbasicmachinemodel	Dynamic	Defined
SINUMERIK_CSALARMREACTION	mmmdev.SINUMERIK_CSALARMREACTION	Dynamic	Defined
SINUMERIK_CSMACHINESTATUS	mmmdev.SINUMERIK_CSMACHINESTATUS	Dynamic	Defined
SINUMERIK_CSPROTECTIONLEVEL	mmmdev.SINUMERIK_CSPROTECTIONLEVEL	Dynamic	Defined
SINUMERIK_CSRAW	mmmdev.SINUMERIK_CSRAW	Dynamic	Defined
SINUMERIK_CSRESULTS	mmmdev.SINUMERIK_CSRESULTS	Dynamic	Defined
Startup	core.sinumerikbasicstartup	Dynamic	Defined
status	core.agentstatus	Dynamic	Inherited
ClosedAlarms	iqsm.ClosedAlarms	Dynamic	Defined

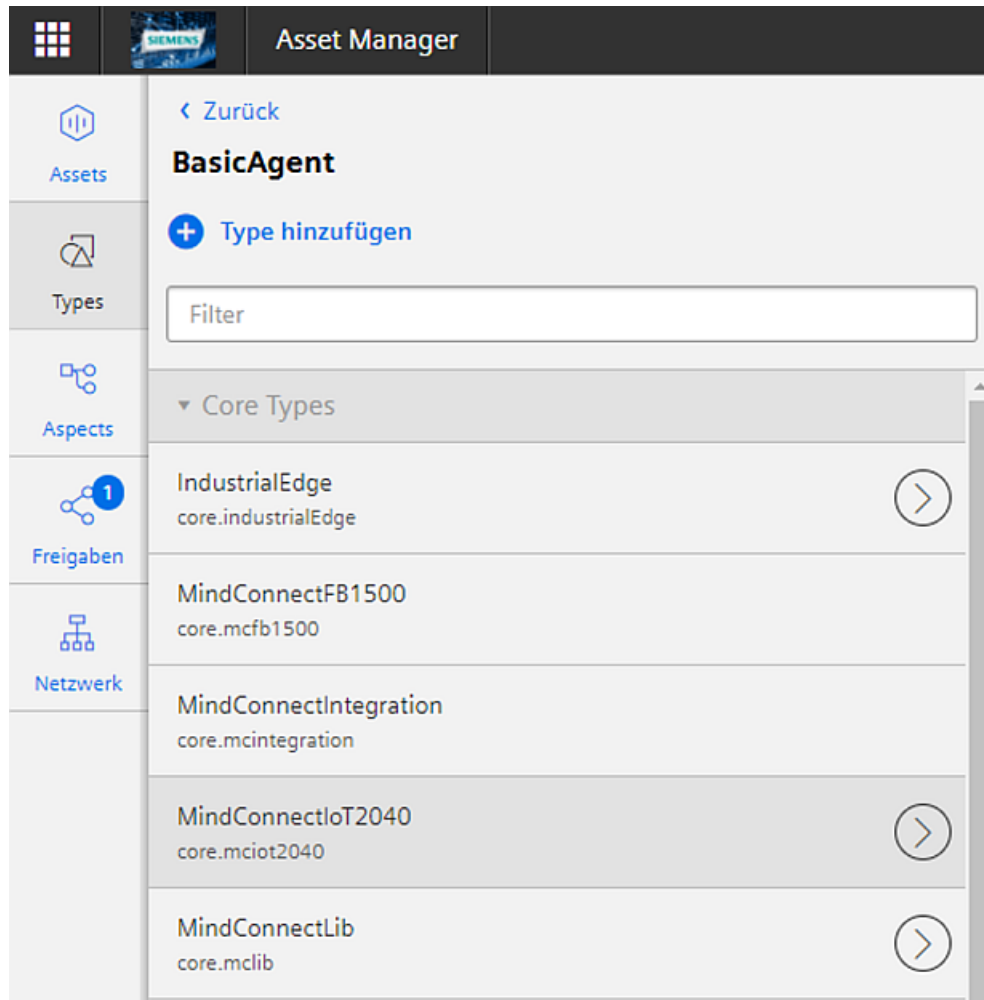
Further information can be found in the documentation for the MindSphere Application SSA (<https://support.industry.siemens.com/cs/ww/en/sc/5369>).

12.5 Creating the asset type "bfc_ssa_sinumerik"

If the asset type "bfc_ssa_sinumerik" does not exist in the Asset Manager of MindSphere, you have to create it.

Procedure

1. Open the Asset Manager in MindSphere.
 - Click on "Types" in the left-hand window area.
 - The "BasicAsset" window opens.
Navigate to the "BasicAgent" area.
Select the type "MindConnectLib" from the list in the right-hand window area.



2. The "MindConnectLib" window opens.
Click the "Add type" button.



3. The "Create type" window opens.
 - Fill in the input fields.
Enter the designation "bfc_ssa_sinumerik" in the "Name" input field.

4. Create the list of aspects according to the following overview.
 - Click on "Add Aspect"
 - Select the aspects as shown in the following figure.
You will find the complete overview of the aspects in the table following this description.
 - Click the "Save" button.

Name	Aspect	Category	Type
connectivityStatus	core.connectivitystatus	Static	Inherited
status	core.agentstatus	Dynamic	Inherited

Name	Aspect	Category	Type
AgentOnlineStatus	core.agentstatus	Dynamic	Defined
Alarms	core.sinumerikbasicalarms	Dynamic	Defined
CH1_BasicConfig	core.sinumerikbasicconfig	Dynamic	Defined
CH1_MachineStatus	core.sinumerikbasicmachinestatus	Dynamic	Defined
CSM_AX01	mmmdev.CSM_AX01	Dynamic	Defined
CSM_AX02	mmmdev.CSM_AX02	Dynamic	Defined
CSM_AX03	mmmdev.CSM_AX03	Dynamic	Defined
CSM_AX04	mmmdev.CSM_AX04	Dynamic	Defined
CSM_AX05	mmmdev.CSM_AX05	Dynamic	Defined
CSM_General_Info	mmmdev.CSM_General_Info	Dynamic	Defined
CSM_SP01	mmmdev.CSM_SP01	Dynamic	Defined
MachineModel	core.sinumerikbasicmachinemodel	Dynamic	Defined
SINUMERIK_CSALARMREACTION	mmmdev.SINUMERIK_CSALARMREACTION	Dynamic	Defined
SINUMERIK_CSMACHINESTATUS	mmmdev.SINUMERIK_CSMACHINESTATUS	Dynamic	Defined
SINUMERIK_CSPROTECTIONLEVEL	mmmdev.SINUMERIK_CSPROTECTIONLEVEL	Dynamic	Defined
SINUMERIK_CSRAW	mmmdev.SINUMERIK_CSRAW	Dynamic	Defined
SINUMERIK_CSRESULTS	mmmdev.SINUMERIK_CSRESULTS	Dynamic	Defined
ClosedAlarms	iqsm.ClosedAlarms	Dynamic	Defined

12.6 Connecting a new machine to SSA

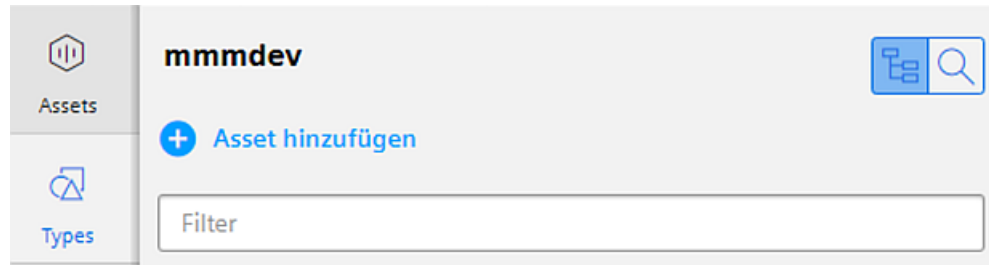
To connect a machine to MindSphere via BFC, perform the following steps:

- Creating a new asset of the type "bfc_ssa_sinumerik" (Page 485)
- Generating connection information of the assets (Page 488)
- Creating a MindSphere gateway for SSA (Page 489)

12.6.1 Creating a new asset of the type "bfc_ssa_sinumerik"

Procedure

1. Open the Asset Manager in MindSphere.
 - Click on "Assets" in the left-hand window area.
 - In the right-hand window area, navigate to the desired location in the asset hierarchy.
2. Click the "Add asset" button.



3. Select the asset type "bfc_ssa_sinumerik".



4. Assign a name, e.g. Machine 1, to the new asset.

Neues Asset

— Allgemein

Type ID:
mmmdev.bfc_ssa_sinumerik
Der ausgewählte Type kann nicht geändert werden

Name: *
my-machine-name

Beschreibung:
Beschreibung
255 Zeichen übrig

+ Ort

+ Variablen

5. All required aspects are listed.
Click the "Save" button.

Asset Manager

Assets
Types
Aspects
Freigaben
Netzwerk

Aspects-Liste

Name	Aspect	Kategorie
> AgentOnlineStatus	core.agentstatus	Dynamisch
> Alarms	core.sinumerikbasicalarms	Dynamisch
> CH1_BasicConfig	core.sinumerikbasicconfig	Dynamisch
> CH1_MachineStatus	core.sinumerikbasicmachinestatus	Dynamisch
> connectivityStatus	core.connectivitystatus	Statisch
> CSM_AX01	mmddev.CSM_AX01	Dynamisch
> CSM_AX02	mmddev.CSM_AX02	Dynamisch
> CSM_AX03	mmddev.CSM_AX03	Dynamisch
> CSM_AX04	mmddev.CSM_AX04	Dynamisch
> CSM_AX05	mmddev.CSM_AX05	Dynamisch
> CSM_General_Info	mmddev.CSM_General_Info	Dynamisch
> CSM_SP01	mmddev.CSM_SP01	Dynamisch
> MachineModel	core.sinumerikbasicmachinemodel	Dynamisch
> SINUMERIK_CSALARMREACTION	mmddev.SINUMERIK_CSALARMREACTION	Dynamisch
> SINUMERIK_CSMACHINESTATUS	mmddev.SINUMERIK_CSMACHINESTATUS	Dynamisch
> SINUMERIK_CSPROTECTIONLEVEL	mmddev.SINUMERIK_CSPROTECTIONLEVEL	Dynamisch
> SINUMERIK_CSRAW	mmddev.SINUMERIK_CSRAW	Dynamisch
> SINUMERIK_CSRESULTS	mmddev.SINUMERIK_CSRESULTS	Dynamisch
> status	core.agentstatus	Dynamisch

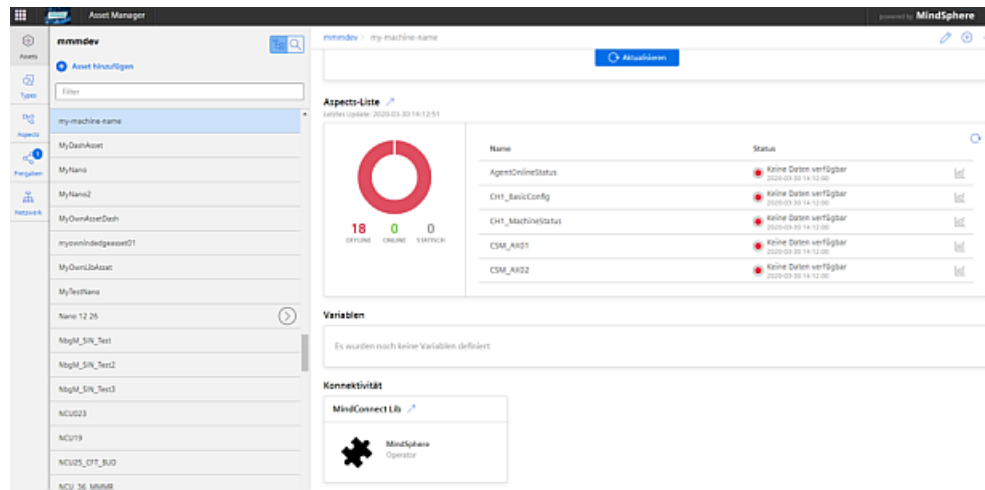
* Erforderliches Eingabefeld

Speichern Abbrechen

12.6.2 Generating connection information of the assets

Procedure

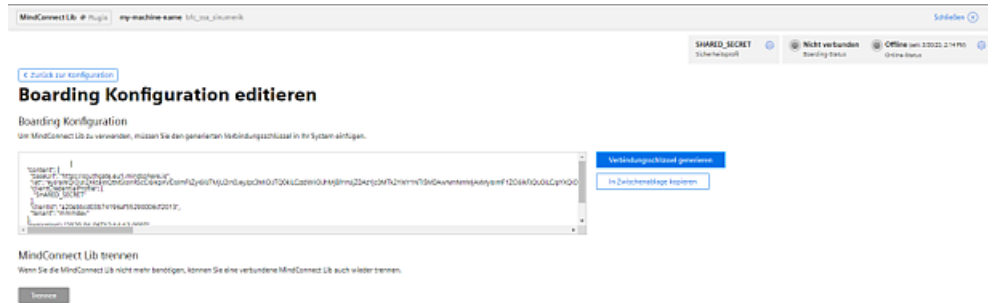
1. Open the Asset Manager in MindSphere.
 - Click on "Assets" in the left-hand window area.
 - In the middle area of the window, navigate to the BFC asset you have just created.
 - Click on the arrow in the lower right window area "Connectivity".



2. The "Configure MindConnect Lib" window opens.
 - Select the option field "SHARED_SECRET" for the secure data connection
 - Click the "Save" button.



3. The "Edit boarding configuration" window opens.
Click the "Generate connection key" button.



4. Copy the generated connection key to the clipboard.

12.6.3 Creating a MindSphere gateway for SSA

Perform the following steps to create a MindSphere gateway for SSA:

- Step 1: "Select gateway type"
- Step 2: "Define basic configuration"
- Step 3: "Define dataset configuration"
- Step 4: "Define alarm configuration"
- Step 5: "Advanced configuration"

Requirement

The "Commissioning" area is open.

Parameters

Parameter	Description
① Select gateway type	
Select target Store...*	Selection of the gateway
② Define basic configuration	
Type gateway name...*	Gateway name
Type gateway description...	Gateway description
Paste MindSphere connection info...*	Connection information Entry of the MindSphere connection string in JSON format Remark: Paste the copied connection key from the clipboard.
Type proxy address if needed...	Enter the proxy address if a proxy is required for Internet access
③ Define dataset configuration / Optional	

12.6 Connecting a new machine to SSA

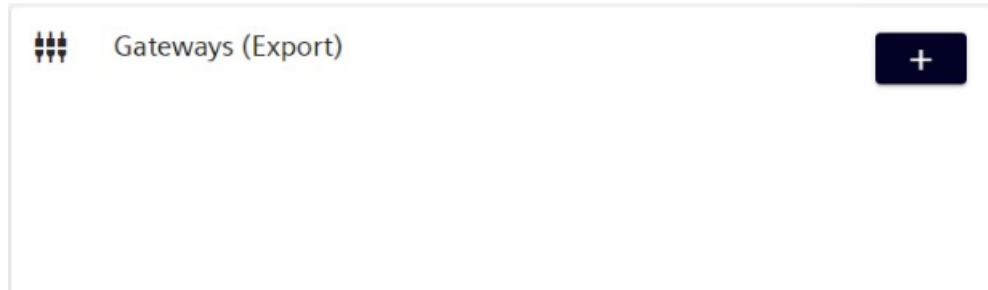
Parameter	Description
Pick a client ID	Client ID of the client, from which data is to be sent to MindSphere. Remark: All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
Pick a dataset	Select the client data set that should be sent to MindSphere. Remark: All data sets of the selected client ID are displayed
- OR -	
All dataset	All client data is sent to MindSphere
④ Define alarm configuration / Optional	
Pick a client ID	All IDs of the clients, which were created under this "Plant hierarchy", are displayed.
⑤ Advanced configuration / Optional	
Remark: These fields do not require any entries, and are only populated by the hotline in the case of service.	
Type image path...*	Path to MindSphere docker image
Type username to access the image path...	Username to access the image path
Type password to access the image path...	Password to access the image path
Set message TTL (time to live) in milliseconds...*	AMQP lifetime of the data
Set queue expiration in milliseconds...*	AMQP queue execution time
Set max queue size...*	AMQP maximum queue size
Set max queue size bytes...*	AMQP maximum queue size in bytes
Select queue mode...*	AMQP queue mode
Set prefetch count...*	AMQP prefetch counter
Type alarm topic*	MQTT topic alarm event
Type new reading topic*	MQTT topic new dataset
Environment variable 1	Environment variable to store the data from "CurrentAlarms" as BIG STRING in MindSphere
Environment variable 2	Environment variable to store the data from "MachineModel" as BIG STRING in MindSphere
Environment variable 3	Note that in the variable "IOTGATEWAY_OVERRIDE" the file topic is changed to the client ID of the connected machine.
Type new environment variable*	New environment variable for the image

*: Obligatory data

The character sequence to be used for the various entries is provided in the input windows.

Procedure

1. In the "Gateway (Export)" area, click on "+" to add a new gateway.



2. Step 1: "Select gateway type"
 - From the drop-down list, select entry "MindSphere".
 - Click on "Next".



- Populate the fields to perform the basic configuration.
Paste the copied connection key from the clipboard into the "Paste MindSphere connection info" input field.
- Click on "Next".

5. Step 4: "Define alarm configuration" / Optional
With this step, you define which alarms are to be sent to MindSphere.

- Select the alarms.

Note**Several entries**

- Click on "+" to add an additional entry.
-

- Click on "Next".

Add new gateway "my-machine-to-ssa"

The screenshot shows a configuration wizard for adding a new gateway named "my-machine-to-ssa". The wizard consists of five steps:

- 1 Select gateway type
- 2 Define basic configuration
- 3 Define dataset configuration
Optional
- 4 Define alarm configuration
Optional
- 5 Advanced configuration
Optional

Step 4 is currently active. It contains a text input field labeled "Pick a client ID" with the value "harald" entered. Below the input field is a blue "Next" button. To the right of the input field is a trash icon. At the bottom right of the wizard are two buttons: a lock icon and a close icon (X).

6. Step 5: "Advanced configuration" / Optional

- Add the environment variables:

```
MINDSPHERE_BIG_STRING_Alarms_CurrentAlarms=TRUE
MINDSPHERE_BIG_STRING_MachineModel_Data=TRUE
IOTGATEWAY_OVERRIDE={"iotClientConfig":{"fileTopics":
["equipments.*.services.<Machine Name>.files.*"]}}
MINDSPHERE_TIMESTAMP_ClosedAlarms_ctEndTime=TRUE
MINDSPHERE_TIMESTAMP_ClosedAlarms_ctStartTime=TRUE
MINDSPHERE_TIMESTAMP_ClosedAlarms_stEndTime=TRUE
MINDSPHERE_TIMESTAMP_ClosedAlarms_stStartTime=TRUE
```

Replace the exemplary designation "harald" with the ID of your machine.

- Click on "Save" to save the MindSphere gateway.

Add new gateway "my-machine-to-ssa"

5 Advanced configuration
Optional

Type image path... *

apc-docker-registry.dlinkdns.com/apc-koe/mts/iotserver/gateways/mindsphere.v1.0.0

Type username to access the image path...

Type password to access the image path...

Set message TTL (time to live) in milliseconds... *

7884000000

Set queue expiration in milliseconds... *

604800000

Set max queue size... *

15000000

Set max queue size bytes... *

5000000000

Save Cancel

Add new gateway "my-machine-to-ssa"

lazy

Set prefetch count... *

1000

Type alarm topic... e.g. "*"*. {clientId} * events'

..harald.*.events

Type new reading topic... e.g. "*"*. {clientId} *.record."

+

Environment variable 1

MINDSPHERE_BIG_STRING_Alarms_CurrentAlarms=TRUE

Environment variable 2

MINDSPHERE_BIG_STRING_MachineModel_Data=TRUE

Environment variable 3

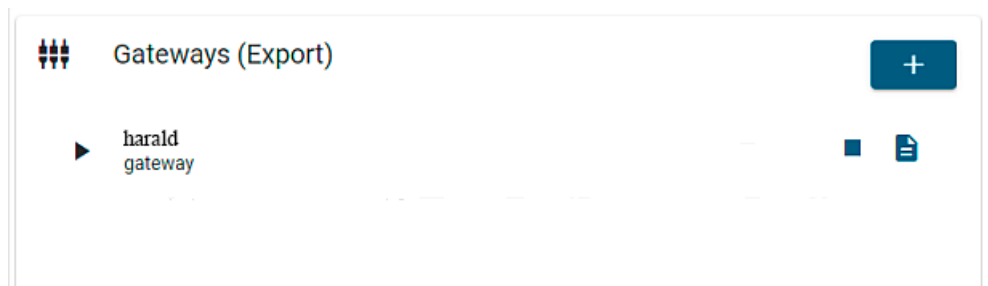
IOTGATEWAY_OVERRIDE=["iotClientConfig"(["fileTopics"(["equipments.*.services.harald.files.""])]]

Type new environment variable...

+

Save Cancel

7. The MindSphere gateway was successfully created, and is shown in the overview in the "Gateways (Export)" area.



12.7 Configuring the BFC client data acquisition

You can use the BFC client to record the following "datasets":

- CH1_BasicConfig
- SINUMERIK_CSRAW
- CSM_General_Info
- CSM_AX01
- CSM_AX02
- CSM_AX03
- CSM_AX04
- CSM_AX05
- CSM_SP01
- SINUMERIK_CSALARMREACTION

Procedure

Configure the variables shown in the following tables.

Dataset "CH1_BasicConfig"

Parameter:

- Dataset name: CH1_BasicConfig
- Reading mode: On Change
- Debounce time in milliseconds: 2000

Datapoint name	Datapoint address	Data type
Feedoverride	/Channel/State/feedRateIpoOvr[u1]	float
NCProgram	/Channel/ProgramPointer/progName[u1,1]	string
NCProgramStatus	/Channel/State/progStatus[u1]	int
NrOfAlarms	/Nck/State/numAlarms[u1]	int
OpMode	/Bag/State/opMode[u1]	int
Spindleoverride	/Nck/Spindle/speedOvr[u1,1]	float
StopCond	/Channel/State/stopCond[u1]	int
ProtectionLevel	/Nck/Configuration/accessLevel	int
AgentOnlineStatus	/Channel/State/progStatus[u1]	bool

Dataset "SINUMERIK_CSRAW"

Parameter:

- Dataset name: SINUMERIK_CSRAW
- Reading mode: Interval
- interval time in milliseconds: 5000

Datapoint name	Datapoint address	Data type
NCKAlive	/Nck/State/nckAliveAndWell	float
PowerOnTime	/Nck/ChannelDiagnose/poweronTime[1]	float
PrioAlarm	/Nck/Top/Prioalarm/textIndex[1]	float
SetupTime	/Nck/Channel/Diagnose/setupTime	float
timeshift_offset	/SharedVar/AutoCounter	float
timeshift_offset	/SharedVar/AutoCounter	float

Dataset "CSM_General_Info"

Parameter:

- Dataset name: CSM_General_Info
- Reading mode: Interval
- interval time in milliseconds: 30000

Datapoint name	Datapoint address	Data type
NUM_AXES_IN_SYSTEm	/Nck/ChannelDiagnose/dpAxisCfgNumAxes	float
NC_CPU_Ready	/Plc/DataBlock/Bit[c10,104.7]	float
Variable_Group1	/Plc/DataBlock/Byte[c10,108]	float
Variable_Group2	/Plc/DataBlock/Byte[c10,109]	float

Dataset "CSM_AX01"

Parameter:

- Dataset name: CSM_AX01
- Reading mode: Interval
- interval time in milliseconds: 30000

Datapoint name	Datapoint address	Data type
AX01_Drives_Status	/Drive/Data/DriveControl[u1,2]	float
AX01_Motor_Temp	/Drive/Data/DriveControl[u1,35]	float
AX01_ImpulseEnable_PLc	/Channel/MachineAxis/impulseEnable[u1,1]	float
AX01_ControlConfirmActive_NC	/Channel/MachineAxis/contrConfirmActive[u1,1]	float
AX01_Variable_Group	/Plc/DataBlock/Byte[c31,93]	float

Dataset "CSM_AX02"

Parameter:

- Dataset name: CSM_AX02
- Reading mode: Interval
- interval time in milliseconds: 30000

Datapoint name	Datapoint address	Data type
AX02_Drives_Status	/Drive/Data/DriveControl[u2,2]	float
AX02_Motor_Temp	/Drive/Data/DriveControl[u2,35]	float
AX02_ImpulseEnable_PLc	/Channel/MachineAxis/impulseEnable[u1,2]	float
AX02_ControlConfirmActive_NC	/Channel/MaschineAxis/contrConfirmActive[u1,2]	float
AX02_Variable_Group	/Plc/DataBlock/Byte[c32,93]	float

Dataset "CSM_AX03"

Parameter:

- Dataset name: CSM_AX03
- Reading mode: Interval
- interval time in milliseconds: 30000

Datapoint name	Datapoint address	Data type
AX03_Drives_Status	/Drive/Data/DriveControl[u3,2]	float
AX03_Motor_Temp	/Drive/Data/DriveControl[u3,35]	float
AX03_ImpulseEnable_PLC	/Channel/MachineAxis/impulseEnable[u1,3]	float
AX03_ControlConfirmActive_NC	/Channel/MaschineAxis/contrConfirmActive[u1,3]	float
AX03_Variable_Group	/Plc/DataBlock/Byte[c33,93]	float

Dataset "CSM_AX04"

Parameter:

- Dataset name: CSM_AX04
- Reading mode: Interval
- interval time in milliseconds: 30000

Datapoint name	Datapoint address	Data type
AX04_Drives_Status	/Drive/Data/DriveControl[u4,2]	float
AX04_Motor_Temp	/Drive/Data/DriveControl[u4,35]	float
AX04_ImpulseEnable_PLC	/Channel/MachineAxis/impulseEnable[u1,4]	float
AX04_ControlConfirmActive_NC	/Channel/MaschineAxis/contrConfirmActive[u1,4]	float
AX04_Variable_Group	/Plc/DataBlock/Byte[c34,93]	float

Dataset "CSM_AX05"

Parameter:

- Dataset name: CSM_AX05
- Reading mode: Interval
- interval time in milliseconds: 30000

Datapoint name	Datapoint address	Data type
AX05_Drives_Status	/Drive/Data/DriveControl[u5,2]	float
AX05_Motor_Temp	/Drive/Data/DriveControl[u5,35]	float
AX05_ImpulseEnable_PLC	/Channel/MachineAxis/impulseEnable[u1,5]	float
AX05_ControlConfirmActive_NC	/Channel/MaschineAxis/contrConfirmActive[u1,5]	float
AX05_Variable_Group	/Plc/DataBlock/Byte[c35,93]	float

Dataset "CSM_SP01"

Parameter:

- Dataset name: CSM_SP01
- Reading mode: Interval
- interval time in milliseconds: 30000

Datapoint name	Datapoint address	Data type
SP01_Drives_Status	/Drive/Data/DriveControl[u6,2]	float
SP01_Motor_Temp	/Drive/Data/DriveControl[u6,35]	float
SP01_ImpulseEnable_PLC	/Channel/MachineAxis/impulseEnable[u1,6]	float
SP01_ControlConfirmActive_NC	/Channel/MaschineAxis/contrConfirmActive[u1,6]	float
SP01_Variable_Group	/Plc/DataBlock/Byte[c36,93]	float

Dataset "SINUMERIK_CSALARMREACTION"

Parameter:

- Dataset name: SINUMERIK_CSALARMREACTION
- Reading mode: On Change
- Debounce time in milliseconds: 200

Datapoint name	Datapoint address	Data type
SINUMERIK_CSALARMREACTION	/Channel/State/acAlarmStat[u1,1]	float

12.8 Creating and saving a machine identity

You can store important information such as machine information and address information for each machine using an identSnapshot.xml.

The creation of an identSnapshot.xml only applies to SINUMERIK Operate.

Additional information can be found in the "SINUMERIK 840D sl, SINUMERIK Operate Commissioning Manual", Chapter: "Diagnostics and service" > "Machine identity", by following this link (<https://support.industry.siemens.com/cs/ww/de/view/109777907>).

identSnapshot.xml for HMI Advanced

The IdentSnapshot file is not present on the machine for HMI Advanced and is therefore generated automatically by the BFC gateway.

The fields under the category "Machine" are populated with the configuration data of the BFC client.

The IdentSnapShot file is generated automatically every four weeks (28 days) by default. Under "Advanced Configuration" of the clients, a different value can be set. While the value "0" deactivates the automatic function, leaving the field blank resets it to the default value.

The following IdentSnapshot file is an example of the available data for HMI Advanced machines:

```
<?xml version="1.0" encoding="UTF-8"?>
<Config date="2020-07-06 09:11:23" version="2.6">
  <Config_Identity>
    <Machine>
      <No>Drilling1</No>
      <Name>Drill Machine</Name>
      <Type/>
    </Machine>
    <Addresses/>
  </Config_Identity>
  <AlmPgiErt version="1.0">
    <AlmPgiErtLicenses>
      <AlmPgiErtProductId>SINUMERIK 840D pl</AlmPgiErtProductId>
      <AlmPgiErtHwId>2fd613a3-363e-40cd-8c85-e46dc556bed5</AlmPgiErtHwId>
      <AlmPgiErtMachineId>kai</AlmPgiErtMachineId>
      <AlmPgiErtLicKey/>
    </AlmPgiErtLicenses>
  </AlmPgiErt>
  <Versions>
    <Component>
      <Name>SINUMERIK 840D plC</Name>
    </Component>
    <Component>
      <Name>Windows</Name>
      <Version>MS-Windows NT (TM) Workstation 4.0 (Build 1381:
ServicePack 6a)</Version>
    </Component>
    <Component>
      <Name>BASESOFTWARE</Name>
      <Version>V07.03.02 (Build a)</Version>
    </Component>
    <Component>
      <Name>HMI-Advanced</Name>
      <Version>06.03.23.00</Version>
    </Component>
    <Component>
      <Name>Systemsoftware NCU</Name>
      <Version>51.28.00</Version>
    </Component>
    <Component>
      <Name>Hardware</Name>
    </Component>
    <Component>
      <Name>PCU50</Name>
      <Version>V2</Version>
    </Component>
  </Versions>
</Config>
```


identSnapshot.xml for HMI Operate

The IdentSnapShot file is generated automatically every four weeks (28 days) by default. Under "Advanced Configuration" of the clients, a different value can be set. While the value "0" deactivates the automatic function, leaving the field bank resets it to the default value.

Configuring the AMP gateway

The BFC Gateway offers the option of transferring selected files to the "SINUMERIK Integrate AMP 4.1" product. The "AMP gateway" must be set-up to transfer the data.

Specialist know-how must be available when it comes to handling AMP 4.1 for the configuration. Training courses provide the appropriate detailed know-how when it comes to configuring the necessary settings. Contact your regional Siemens sales partner for more detailed information about training courses.

Information about commissioning and operation is subsequently provided.

Note

Versions when connecting to AMP

Machines that are already connected to AMP as AMP gateway using another technology, must be evaluated on a project-for-project basis.

In cases such as this, contact your regional Siemens sales partner.

13.1 System requirements relating to the AMP server

The following requirements apply:

- **AMC server**

An installed AMP server is required to use the gateway.

The minimum version required depends on whether http or https is used for communication:

http:	From AMP Version 2.12 (SINUMERIK Integrate Version 4.1 SP8)
https:	From AMP Version 2.13 (SINUMERIK Integrate Version 4.1 SP10)

- **Clients**

Up to 40 machines can be connected to an AMP server via the BFC gateway. An additional AMP server is required if more machines are to be connected.

The higher the data frequency for each machine, the fewer the machines that can be used. For each application, carefully check whether the AMP server with the particular hardware is sufficient.

- **https operation**

For https operation, a valid SSL certificate must be installed in the AMP server and linked to the application (httpServer.dll of the AMP server).

The relevant AppID is: {e3924f26-00cf-4041-8291-6faa14d8bcea}

The corresponding issuer certificate must be known in the BFC gateway.

The following command can be used for the binding:

```
netsh http add sslcert ipport=0.0.0.0:8443
certhash=b5f2ea5c3f5b045a8ee4db64e8bc25b20e9e1bca
appid={e3924f26-00cf-4041-8291-6faa14d8bcea}
```

The `certhash` is the certificate thumbprint, and can be read from its properties.

For self-signed certificates, the `certstorename=Root` must be added.

- **Issuer certificate**

For the configuration, the issuer certificate (CA certificate) must be made available to the BFC gateway using parameter `gatewayCert`.

13.2 Configuring the AMP server

13.2.1 General settings in the AMC server

The configuration steps described here must be performed in the AMP server, not in the BFC gateway.

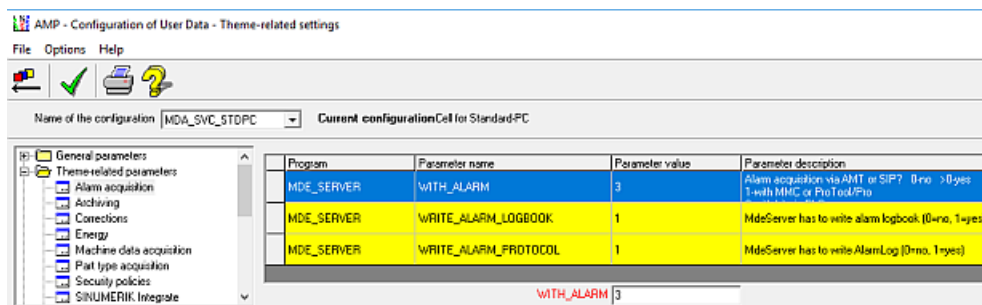
Note

It is important that you always consult the AMP product documentation.

13.2.1.1 Setting to process alarms

Make the following settings so that the AMP server can process alarms that have been transferred to it.

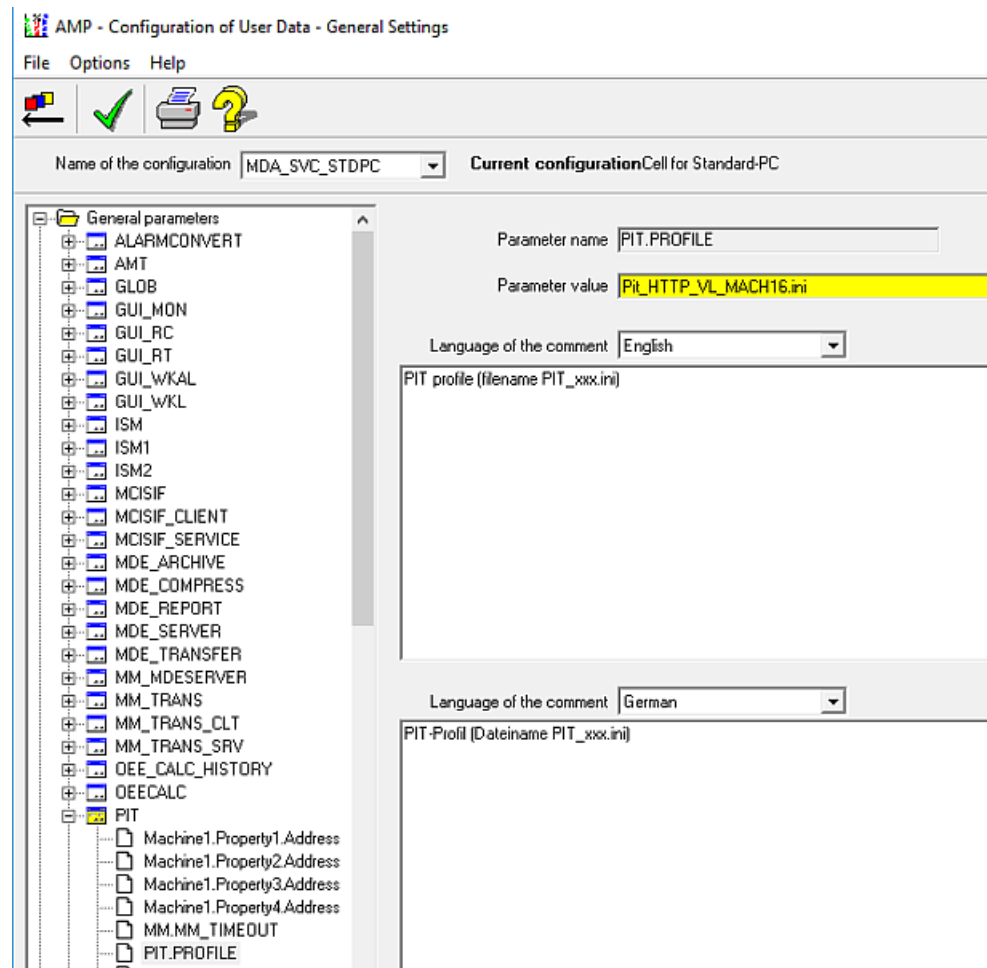
1. In "Customizing (User)" open the entry "General Parameters" > "Theme related parameters" > "Alarm acquisition".
2. Set parameter "WITH_ALARM" to the value "3".



13.2.1.2 Configuring the http interface

The http interface is configured in the PIT module.

- Under "General parameters" – PIT – "PIT.PROFILE", define the configuration file for the data acquisition.
Additional information about the configuring file example is provided in Chapter: Adapting Pit.ini (Page 505).



13.2.1.3 Adapting Pit.ini

Configuration file "Pit.ini", used in the AMP server, may have to be adapted so that the configured machines can be accessed via the http interface.

More information relating to this is provided in the documentation of the AMP.

As default setting, "Pit.ini" is located in the AMP installation directory:

C:\Siemens\SINUMERIK_Integrate\MDA

Sections

The following sections are required, and if necessary, must be supplemented in the existing configuration file:

```

[HTTP]
# polling cycle time in SecondsWatchDelay=5
WinTrace=1
MaxTraceLines=80

# 0=no interface protocol;
# 1=write files to trace all data from interface;
WithProtocoll=0

ByteOrder="Motorola"
OutputAllErrors=0

# max. number of events in VL buffer per machine
VlBuffersize=1000

# TCP/IP port receiving http telegrams
Port=3560

# prog id of http server com object to receive telegrams from VL
ServerProgID="Siemens.MDA.HttpServer"

# timespan for connection monitoring (data or keep alive) in sec
KeepAliveOutOfTime=60

# variable name of keep alive in message from http server
KeepAliveVar=KeepAlive

# UTC or local time from VL
WithUTC=0

# name of main http connection which is used for machine on and
connection monitoring
MainConnection="HTTP_MDA_MAIN"

# timespan for shutdown threads in msec
ThreadShutdownWaitingTime=5000

# online = 1 / offline mode = 0    (default offline)
OnlineMode=0

#ErrorValue for MDA_SIGNAL in online mode will be sent at
connection error.
# the value 800000000 means Bit_32 and is highest possible bit.
ErrorValue=800000000

```

Configuring a machine as http machine

If you configure an individual machine as http machine, additional adaptations are required; see the following example.

The machine in this example is machine 2.

```
##### Machine 2 #####

[Machine2]
Interface=HTTP
Server=WinMDE
Name=(,,)
WinTrace=0
MaxTraceLines=40

##### System #####

[Machine2.Property1]
Address=MachineOn
Type=Word
Name=MachineOn
InitValue=-2
AcquisitionType=0x1
LookForDifferences=1

[Machine2.Property2]
Address=MachineOnline
Type=Word
Name=MachineOnline
InitValue=-2
AcquisitionType=0x1
LookForDifferences=1

[Machine2.Property3]
Address=KeepAlive
Type=Word
Name=KeepAlive
InitValue=-2
AcquisitionType=0x1
LookForDifferences=0

[Machine2.Property4]
Address=Data_Lost
Type=Word
Name=Data_Lost
InitValue=-2
AcquisitionType=0x1
LookForDifferences=1

##### MDC variables (input) #####

[Machine2.Property5]
Address=MDA_SIGNAL
Type=Binary
ConversionType=VarLogic
InitValue=00000000
#ErrorValue will be sent at connection error.
# the value 00010000 means Bit_16 and is highest possible bit.
#ErrorValue=00010000
Name=VL_AggrStatus_1
AcquisitionType=0x1
```

```

### machine counter 1 to 3 ###

[Machine2.Property6]
Address=MDA_COUNTER_1
Type=Word
Name=CycleCounter_1
InitValue=0
AcquisitionType=0x1
LookForDifferences=1
[Machine2.Property7]
Address=MDA_COUNTER_2
Type=Word
Name=CycleCounter_2
InitValue=0
AcquisitionType=0x1
LookForDifferences=1

[Machine2.Property8]
Address=MDA_COUNTER_3
Type=Word
Name=CycleCounter_3
InitValue=0
AcquisitionType=0x1
LookForDifferences=1

##### PDC variables (input) #####
##### Alarm state detection #####

# NC+PLC+MMC Alarm
[Machine2.Property9]
Address=NrOfAlarm
Type=Word
Name=NrOfAlarm
InitValue=-1
AcquisitionType=0x1

##### UDC variables (input) #####

[Machine2.Property10]
Address=MDA_UDC_1
Type=Binary
Name=FeedOverride
InitValue=0
LookForDifferences=0
ConversionType=VarLogicUDC

[Machine2.Property11]
Address=MDA_UDC_2
Type=Binary
Name=SpindleOverride
InitValue=0
LookForDifferences=0
ConversionType=VarLogicUDC

### manual status from UI ###

```



```
[Machine2.Property12]
Address=MDA_MAN_STATUS
Type=String
InitValue=#

Name=MDA_MAN_STATUS
AcquisitionType=0x1
LookForDifferences=1

##### Production order properties #####

[Machine2.Property13]
Address=PDC_Key
Type=String
InitValue=#####
Name=PDC-Key
AcquisitionType=0x2
LookForDifferences=0

[Machine2.Property14]
Address=CycleCounterDelta_1
Type=DWord
InitValue=0
Name=CycleCounterDelta_1
AcquisitionType=0x2

[Machine2.Property15]
Address=CycleCounterDelta_2
Type=Dword
InitValue=0
Name=CycleCounterDelta_2
AcquisitionType=0x2

[Machine2.Property16]
Address=CycleCounterDelta_3
Type=Dword
InitValue=0
Name=CycleCounterDelta_3
AcquisitionType=0x2

[Machine2.Property17]
Address=CycleTimeDelta_1
Type=Dword
InitValue=0
Name=CycleTimeDelta_1
AcquisitionType=0x2

[Machine2.Property18]
Address=CycleTimeDelta_2
Type=Dword
InitValue=0
Name=CycleTimeDelta_2
AcquisitionType=0x2

##### System methods #####
```

```

[Machine2.Event1]
Source=(MachineOn,0)
Method1=(SetStateAutomatic,3)
Parameter1=($Aggregat1,$OFFLINE,1)

[Machine2.Event2]
Source=(MachineOn,1)
Method1=(SetStateAutomatic,3)
Parameter1=($Aggregat1,$OFFLINE,0)

[Machine2.Event3]
Source=(MachineOnline,0)
Method1=(SetStateAutomatic,3)
Parameter1=($Aggregat1,$CO_OFF,1)

[Machine2.Event4]
Source=(MachineOnline,1)
Method1=(SetStateAutomatic,3)
Parameter1=($Aggregat1,$CO_OFF,0)

[Machine2.Event5]
Source=(KeepAlive,)
Method1=(SetStateAutomatic,3)
Parameter1=($Aggregat1,$KEEPALIVE,1)

[Machine2.Event6]
Source=(Data_Lost,#,)
Method1=(SetStateAutomatic,3)
Parameter1=($Aggregat1,$DATA_LOST,$Source.New)

##### Status methods #####

[Machine2.Event7]
Source=(VL_AggrStatus_1,)
Method1=(SetStateAutomatic,4)
Parameter1=($Aggregat1,AggrStatus_1,$Source.Old,$Source.New)

##### Counter methods #####

[Machine2.Event8]
Source=(CycleCounter_1,#,0)
Method1=(SetCounterValue,3)
Parameter1=($Aggregat1,CycleCounter_1,$Source.New)

[Machine2.Event9]
Source=(CycleCounter_2,#,0)
Method1=(SetCounterValue,3)
Parameter1=($Aggregat1,CycleCounter_2,$Source.New)

[Machine2.Event10]
Source=(CycleCounter_3,#,0)
Method1=(SetCounterValue,3)
Parameter1=($Aggregat1,CycleCounter_3,$Source.New)

##### Production order methods #####

[Machine2.Event11]
Source=(PDC-Key,#,#####)

```

```
Method1=(SetProductionData,15)
Parameter1=($Aggregat1,Status,0,PDC-Key,$PDC-
Key,CycleCounterDelta_1,$CycleCounterDelta_1,CycleCounterDelta_2,$Cy
cleCounterDelta_2,CycleCounterDelta_3,$CycleCounterDelta_3,CycleTime
Delta_1,$CycleTimeDelta_1,CycleTimeDelta_2,$CycleTimeDelta_2)

##### Alarm methods #####

[Machine2.Event12]
Name=ALARM
Source=(NrOfAlarm,>,0)
Method1=(SetStateAutomatic,3)
Parameter1=($Aggregat1,Alarm,1)

[Machine2.Event13]
Name=ALARM
Source=(NrOfAlarm,0)
Method1=(SetStateAutomatic,3)
Parameter1=($Aggregat1,Alarm,0)

##### User Data methods #####

[Machine2.Event14]
Source=(FeedOverride,#,00)
Method1=(SetUserData,3)
Parameter1=($Aggregat1,UDC.FeedOverride,$Source.New)

[Machine2.Event15]
Source=(SpindleOverride,#,00)
Method1=(SetUserData,3)
Parameter1=($Aggregat1,UDC.SpindleOverride,$Source.New)

##### Manual Status methods #####

[Machine2.Event16]

Source=(MDA_MAN_STATUS,)
Method1=(SetStateManual,3)
Parameter1=($Aggregat1,$Source.New,1)
```

Configuring additional machines as http machines

If you configure additional machines as http machines, then only a shortened configuration is required.

In the previous example, machine 2 was configured as http machine, which means that parameter `MakeAs=2` can be used.

```
[Machine4]
Interface=HTTP
Server=WinMDE
Name=(,,)
WinTrace=0
MaxTraceLines=40
MakeAs=2
```

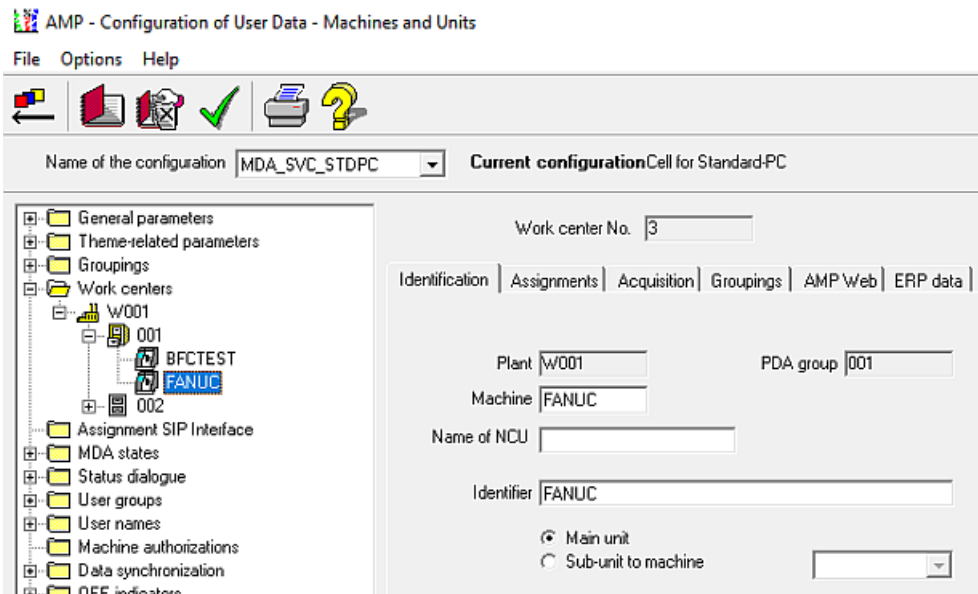
13.2.2 Configuring the machine in the AMP server

The data must be uniquely assigned to the particular machine.

To do this, you must configure the machine in the AMP for data acquisition via the http interface.

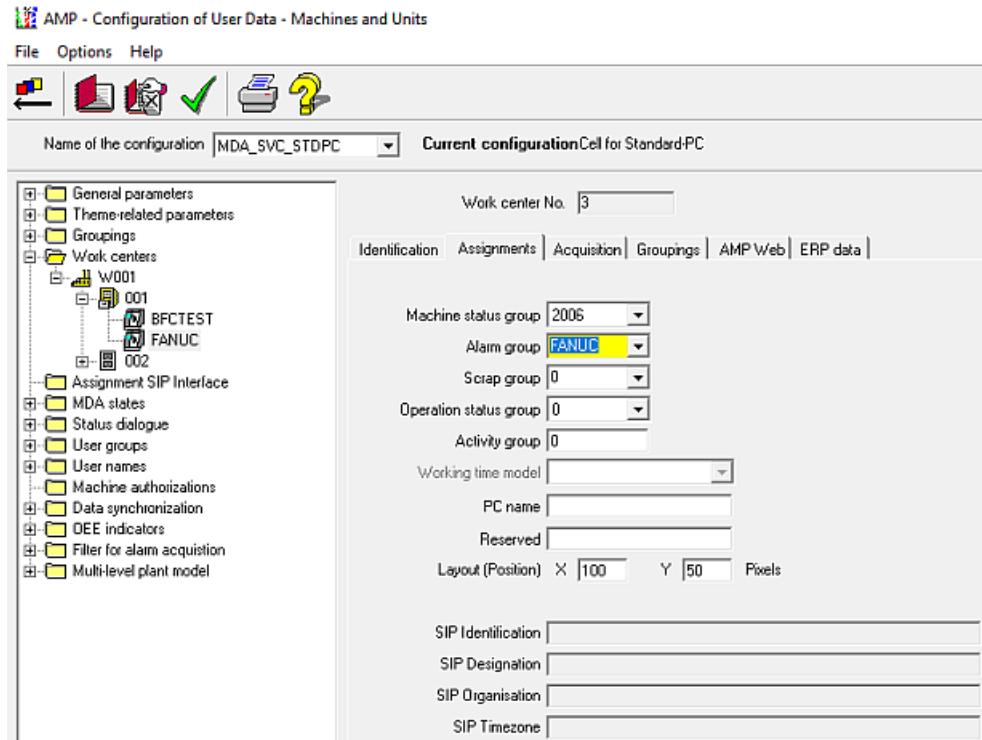
"Identification" tab

The configuration under the "Identification" tab is identical for all machines in AMP.



"Assignments" tab

Under the "Assignments" tab you can also select the machine status group "2006" and an alarm group specifically created for the machine; e.g. FANUC.



"Acquisition" tab

Under the "Acquisition" tab you configure the assignments of the acquisition level (PIT).

The value for the cluster is obtained from "(number of already configured machines) + 1". Configure the acquisition type for "Offline", as the buffered data with timestamp originate from the BFC gateway.

Note

Acquisition type: "Online"

For some projects, the acquisition type must be changed to "Online".

In this case no buffering takes place.

- Deactivate buffering in the configuration of the gateway.
- Activate the "Passthrough" mode so that the signals are routed directly to AMP.

More information can be found in Chapter: Configuration parameters of an AMP gateway (Page 515).

AMP - Configuration of User Data - Machines and Units

File Options Help

Name of the configuration: MDA_SVC_STDPC Current configuration: Cell for Standard-PC

Work center No. 3

Identification | Assignments | Acquisition | Groupings | AMP Web | ERP data

Type of acquisition: Offline

Type of automatic acquisition: MDC+PDC acquisition

Data block:

Cluster: 2

Position: 1

Cycle time: 60

Cycle factor: 1

Max. no. of operations: 1

Max. no. of persons: 1

Max. no. of planned operations: 0

tolerance range counter overflow:

"Groupings" tab

Under the "Groupings" tab, populate field "AMP light PC Name".

The value of the field has the structure "[IDENTIFIER]#[NUMBER]". A unique number must be used.

AMP - Configuration of User Data - Machines and Units

File Options Help

Name of the configuration: MDA_SVC_STDPC Current configuration: Cell for Standard-PC

Work center No. 3

Identification | Assignments | Acquisition | Groupings | AMP Web | ERP data

Area:

Group:

EV partner: ISM1

AMP light partner:

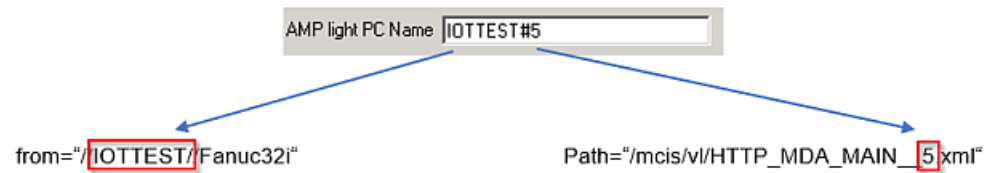
AMP light PC name: IOTTEST#5

Cost center:

Manufacturer:

Machine type:

You require the identifier "IOTTEST" used in this example, and the number "5" in the configuration of the AMP gateway.



The address in the AMP gateway configuration is formed as follows:
/mcis/vl/HTTP_MDA_MAIN_{AMP-Light-Number}.xml

13.3 Configuring the AMP gateway

Create an AMP gateway for each machine.

Note

One AMP gateway for every client

Each AMP gateway transfers the alarms and signals of precisely one client to AMP.

If several clients are to send data to AMP, then for each of these clients a dedicated AMP gateway must be set up.

13.3.1 Configuration parameters of an AMP gateway

Parameter

Basic parameters

The following basic configuration parameters are supported:

Parameter	Data type	Description
AMP URI ¹⁾	string	Address of the AMP server
AMP port	int	Port of the AMP server
AMP path ²⁾	string	Path of the AMP server for receiving data
AMP light name ^{1) 2)}	string	Parameterized AMP light name
AMP machine name	string	Name of the machine

¹⁾ These parameters are mandatory

²⁾ These parameters include the unique identifiers, which were also assigned in parameter "AMP light PC Name" of the machine in AMP

Advanced parameters

The following advanced configuration parameters are supported:

Parameter	Data type	Description
AMP keep alive interval	int	Send interval for the ConnectionCheck telegrams in seconds
AMP timeout	int	Timeout in seconds
AMP sending period	int	New signals are not sent directly to AMP. Instead, new signals are buffered for a time according to the number of seconds set here and sent together in a block. Remark: The "AMP sending period" parameter must be at least twice as large as "AMP timeout". If not, "AMP sending period" is automatically adapted. A corresponding message appears in the log.
Disable AMP buffering	bool	If buffering is to be disabled completely (type of acquisition "online"), set this parameter to "true"
AMP buffer	int	Maximum number of signals that should be buffered Remark: The parameter "AMP buffer" must be greater than the value "prefetch count" of the gateway. You find "Set prefetch count" under "Advanced configuration" in the gateway settings.
AMP CA certificate for https connections	string	The CA certificate used for https connections. ("\\n" substitutes line breaks)
Enable AMP passthrough mode	bool	If signals are to be sent directly to the AMP server without consideration of "AMP sending period", set this parameter to "true"(type of acquisition "online").

Requirement for encrypted communication via https

Carefully observe the following if communication is to be established via https:

- In "AMP URI" the prefix "https://" must be placed in front of the address.
- The "AMP Port" must be set to "8443".
For https, another port is not supported.
- The certificate must match the entry "AMP URI".
- If an IP address is configured, then the certificate must be created for the IP address.
- If a fully qualified DNS name is used, for example "https://ampserver.example.com", then carefully observe the following:
 - The certificate must be created for the fully qualified DNS names.
 - The BFC gateway must be able to resolve the DNS names.

- The AMP gateway must know the issuer certificate (CA certificate) of the AMP server certificate.
- The "AMP CA certificate for https connections" parameter contains as value the certificate as a string in the format:
-----BEGIN CERTIFICATE-----\n
MIIDdTCCA12gAwIBAgILBAAAAAABFUtaW5QwDQYJKoZIhvcNAQEFBQAwVzELMAkGA1UEBhMCQkUx\n
...
hm4qxFYxldBniYUr+WymXUadDKqC5JlR3XC32lY9YeRq4VzW9v493kHMB65jUr9TU/
Qr6cf9tveC\n
X4XSQRjbgbMEHMUfpIBvFSDJ3gyICh3WZlXi/EjJKSZp4A==\n
-----END CERTIFICATE-----\n

13.3.2 Data selection and topics of the AMP gateway

Observe the following notes to ensure that the gateway responds to all relevant alarms and signals of a device and client:

- The topics define which data is transferred from the AMP gateway to the AMP server.
- In the following values, the name of the client in the BFC gateway is used as "deviceId".
- The following topic is defined in field "alarm topic" under "Advanced configuration":
.. {deviceId} *.events
 - Example of a FANUC client with the name "Fanuc01":
*. *.Fanuc01. *.events
- The following "reading topics" are configured:
 - *. *. {deviceId} *.record.AMP_SIGNAL
 - *. *. {deviceId} *.record.AMP_CYCLEDATA
 - *. *. {deviceId} *.record.AMP_PARTCOUNT

Example for machine with deviceId "fanuc01"

Type alarm topic... e.g. *. *. {deviceId} *.events
*. *.fanuc01. *.events

Reading topic 1
*. *.fanuc01. *.record.AMP_SIGNAL

Reading topic 2
*. *.fanuc01. *.record.AMP_CYCLEDATA

Reading topic 3
*. *.fanuc01. *.record.AMP_PARTCOUNT

Note

Data processing

For all control systems or devices, data must be processed specifically for the project, based on the applicable customer requirements.

For FANUC control systems an example is provided to show how data is processed (script logic).

13.3.3 Environment variables of the AMP gateway

To avoid data loss in buffer operation, for an AMP gateway, you must set the environment variable "USE_SINGLE_QUEUE" to "TRUE".

Procedure

1. Set the environment variable "USE_SINGLE_QUEUE" to "TRUE" as follows:
"Advanced Configuration" > Field "Environment variable":

Environment variable 1

USE_SINGLE_QUEUE=TRUE

2. Click "Save".
3. The gateway is created and is ready for operation.



13.4 Script logic to link FANUC clients to AMP

The FANUC client of the BFC gateway only reads raw signals from the control system.

However, AMP requires preprocessed signals.

The BFC gateway is supplied with a standard script logic to generate preprocessed data from raw signals.

The default script logic provided assumes that the FANUC client supplies data set "main" with the following data points:

Identifier	Address	Type
NC program status	/focas/cnc/statinfo/run	int
Operation mode	/focas/cnc/statinfo/aut	int
Feed-rate override	/focas/pmc/rdpmcrng/G/Byte[1,12]	int
Current program	/focas/cnc/rddynamic2/prgnum	int
Spindle speed	/focas/cnc/rddynamic2/acts	float
Status-of-Emergency	/focas/cnc/statinfo/emergency	int
Spindle-Override	/focas/pmc/rdpmcrng/G/Byte[1,30]	int
Sequence number	/focas/cnc/rddynamic2/seqnum	int
Motion	/focas/cnc/statinfo/motion	int
Cycletime-ms	/focas/cnc/rdparam/dword[0,6757]	int
Cycletime-min	/focas/cnc/rdparam/dword[0,6758]	int
Partcounter	/focas/cnc/rdparam/dword[0,6712]	int

The default script logic must be started for each FANUC client.

- Use WinSCP or PuTTY to access the computer file system, which the BFC gateway executes as service.

13.4.1 Creating a script configuration for a client

Note

In the development of script logic, the source code produced must be tested exhaustively and verified before productive use.

Procedure

- For each FANUC control system, create a copy of configuration file "ampfanuc.json" in directory: "bfc/docker/scriptlogic/samples/logicconfigurations".

Note

Client name

We recommend that the client name is used as part of the file name.

For the copied configuration file "ampfanuc.json", "oem_1234.json" is used as an example in the following.

- You can create several script configurations for script "ampfanuc.js", which can be run completely independent of one another.
Example:
"bfc/docker/scriptlogic/samples/logicconfigurations/oem_1234.json"
- Adapt the value "id"; in the example shown below: oem_1234.json

13.4 Script logic to link FANUC clients to AMP

- Change the attribute value from "active" to "true".
- Assign the machine name in the "connectionStateTopics" and "readingSetTopics"; in the example shown below: "oem1234"

Note

Assigning machine names

Accept the machine name that is entered in area "Clients" of the BFC gateway.

Example for file "oem_1234.json"

```
{
  "id": "oem_1234.json",
  "logicName": "ampfanuc.js",
  "active": true,
  "streamConfig": {
    "type": "mqtt",
    "config": {
      "alarmTopics": [],
      "brokerURI": "tcp://broker:1883",
      "clientId": "",
      "username": "iotserver",
      "password": "b940f8cd-0000-1111-2222-2e8c6960305a",
      "connectionStateTopics": ["connectivity/+/oem1234/+/devicestate"],
      "readingSetTopics": ["connectivity/+/oem1234/+/record/main"],
      "qos": 0
    }
  }
}
```

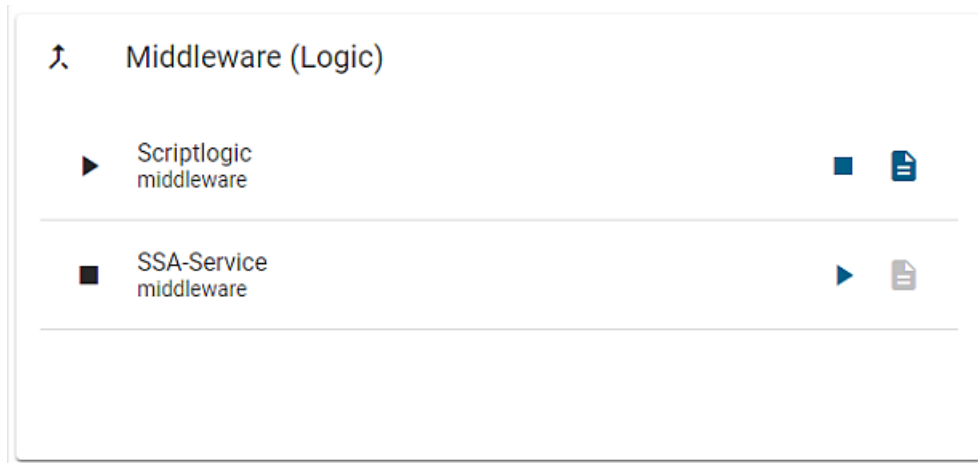
13.4.2 Restarting the script logic

Procedure

The script logic of the BFC gateway does not automatically identify a modified/changed parameterization.

As a consequence, proceed as follows:

1. Open the user interface of the BFC gateway.
2. Open the "Commissioning" area.
In the area "Middleware (Logic)" you will find the entry "Scriptlogic".
 - Click on icon "Start middleware" to restart the script logic.
 - Click on icon "Stop middleware" to stop the script logic.



BFC apps

14.1 IBase app

The IBase app partially automates the "IBase-Service" in the BFC Gateway environment.

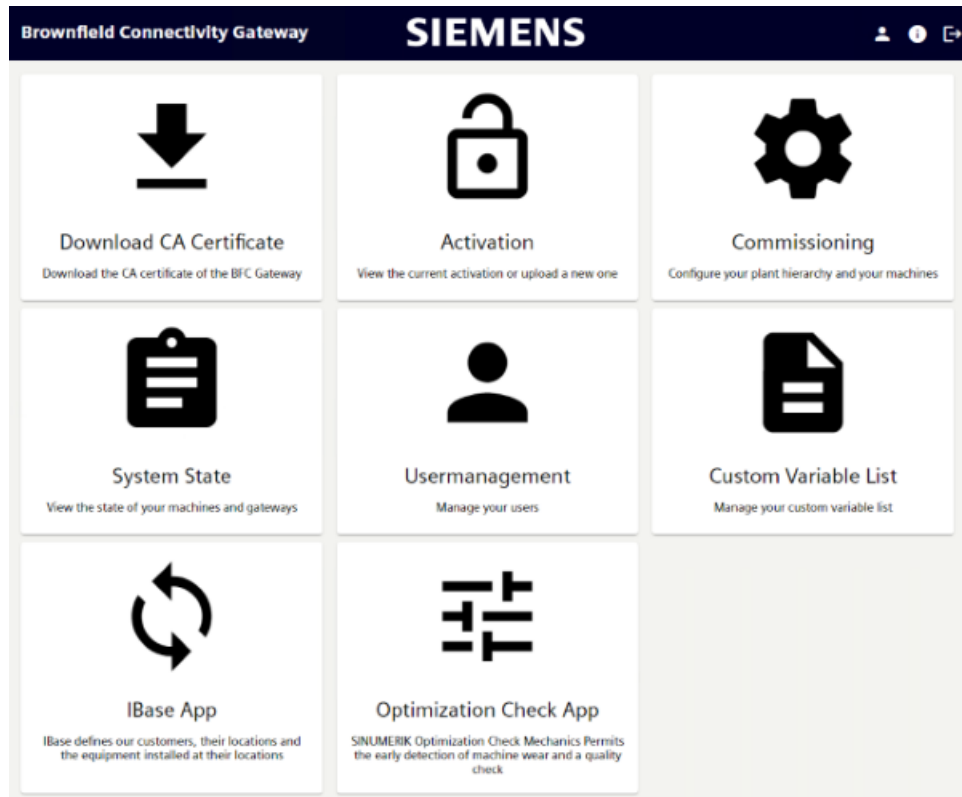
A registration offers you the following advantages:

- Extended liability for defects period for selected products without any additional cost to you
- Faster access to product support that is tailored to your needs and for plant downtimes.
- Weak points can be avoided, combined with asset optimization and obsolescence management to protect your investments and minimize risk

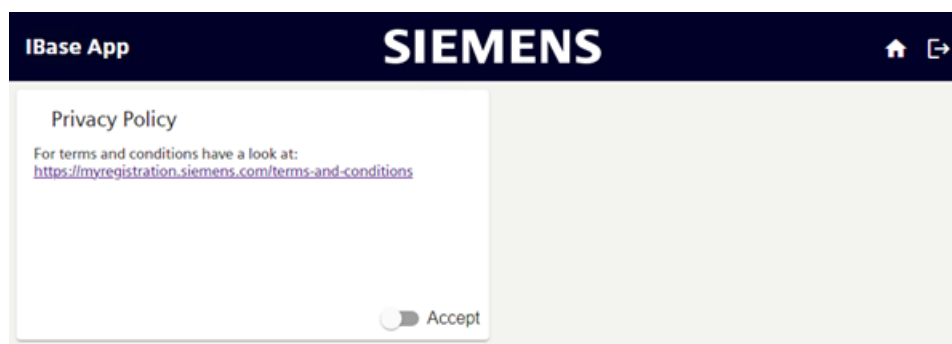
14.1.1 Using the IBase app

Procedure

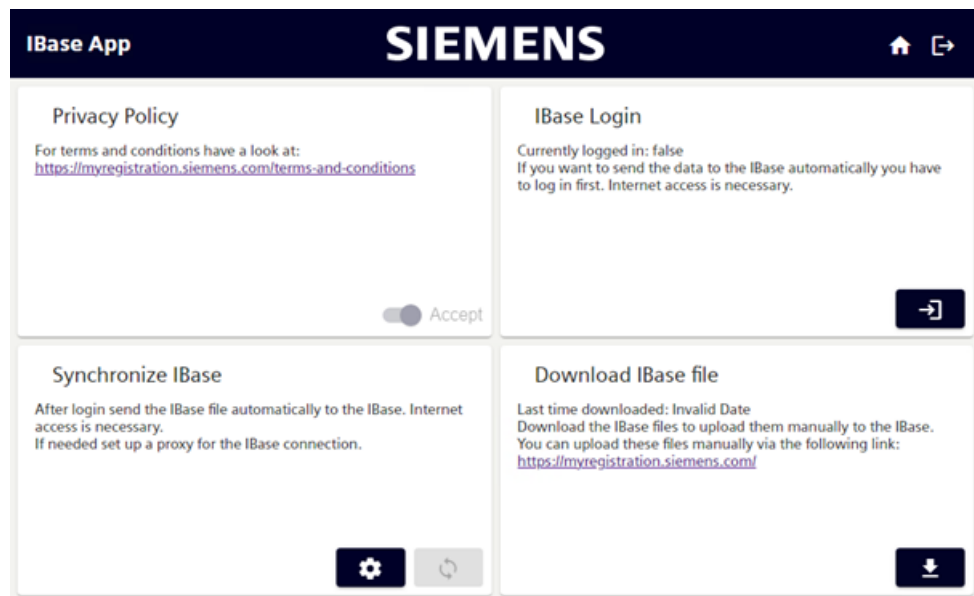
1. Click on the "IBase App" tile on the landing page.
A website opens.



2. Click the link.
 - Read the "Terms and Conditions".
 - Accept the terms and conditions with the "Accept" button.



3. To send the data from the BFC Gateway to IBase, choose one of the following options:
 - "Download IBase file" tile to send the data manually
 - "IBase Login" and "Synchronize IBase" tiles to send the data semiautomatically



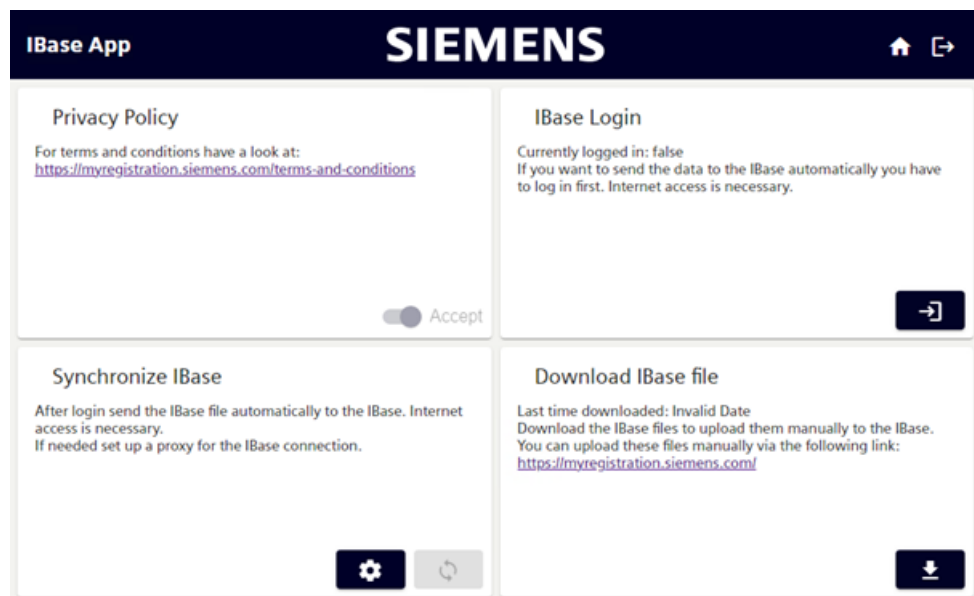
14.1.2 Manually sending data to IBase

Requirement

You have installed and opened the IBase app.

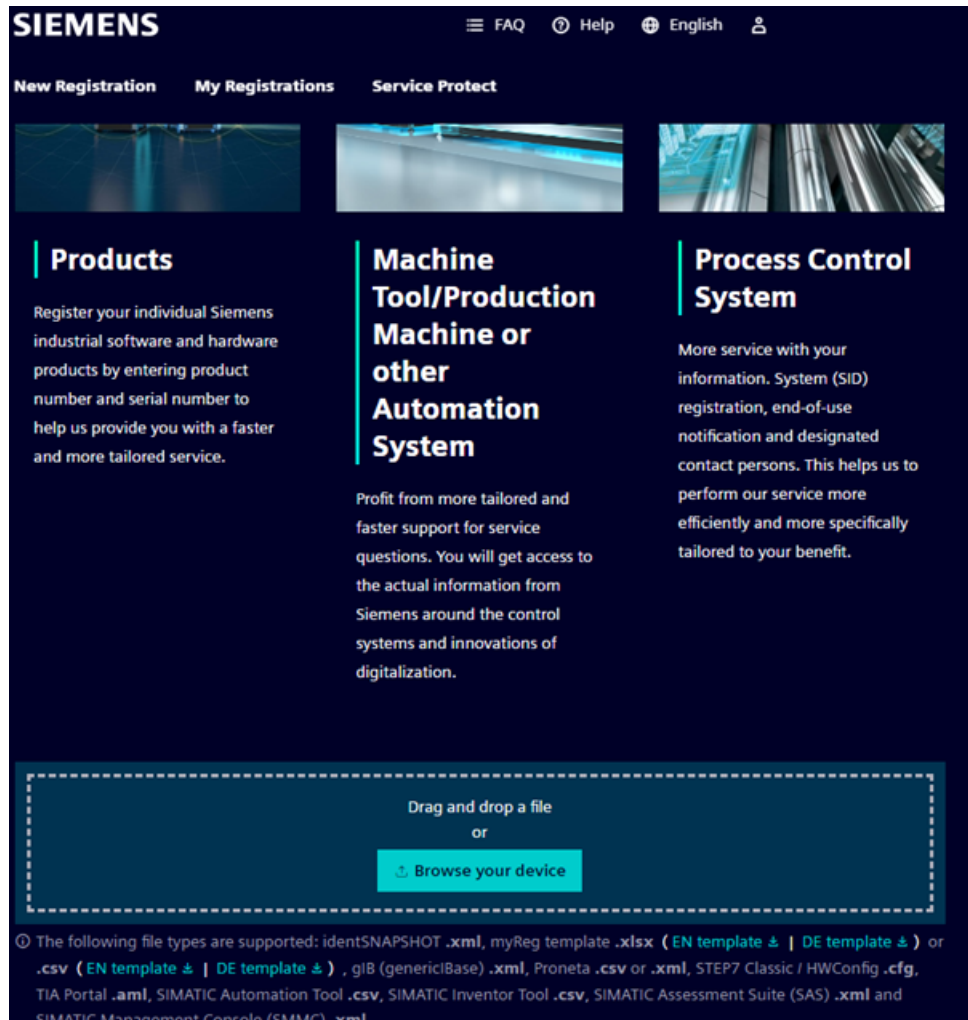
Procedure

1. Navigate to the "Download IBase file" tile.



2. Click the "Download" button to download an XML file.

3. Click the link on the tile.
4. Log on securely, for example, with the PKI key.
A website opens.
5. Navigate to the lower window area.
Upload the XML file that you downloaded in step 2.



6. Follow the intuitive menu-based operation.

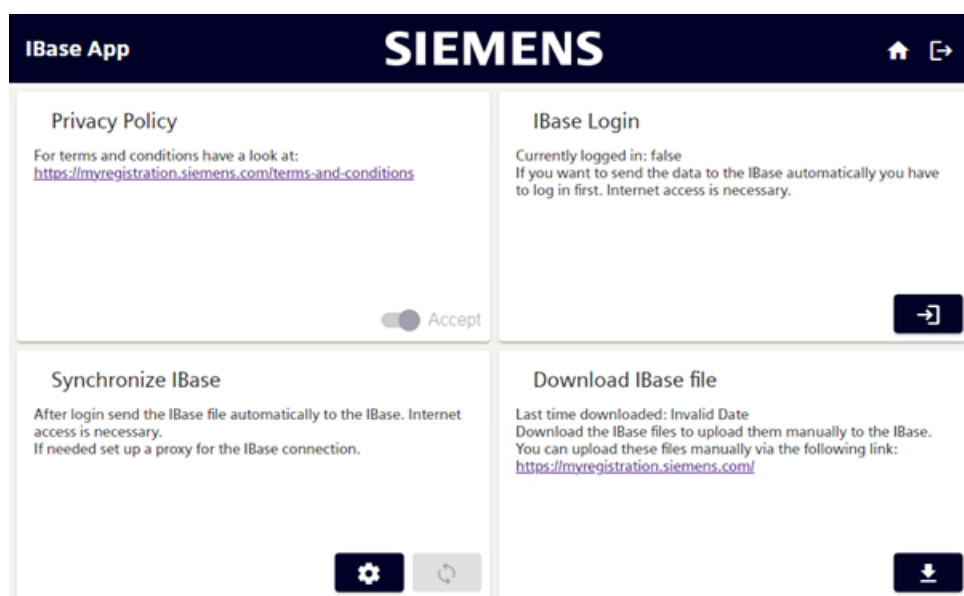
14.1.3 Sending data semiautomatically to IBase

Requirement

You have installed and opened the IBase app.

Procedure

1. Navigate to the "IBase Login" tile.
2. Click on "Login".
3. Log on securely, for example, with the PKI key.
The login is valid for 12 hours.
4. Navigate to the "Synchronize IBase" tile.
5. Set up a proxy server if necessary.
6. Click the "Synchronize" button to send the BFC Gateway data to IBase.



14.2 Optimization Check app

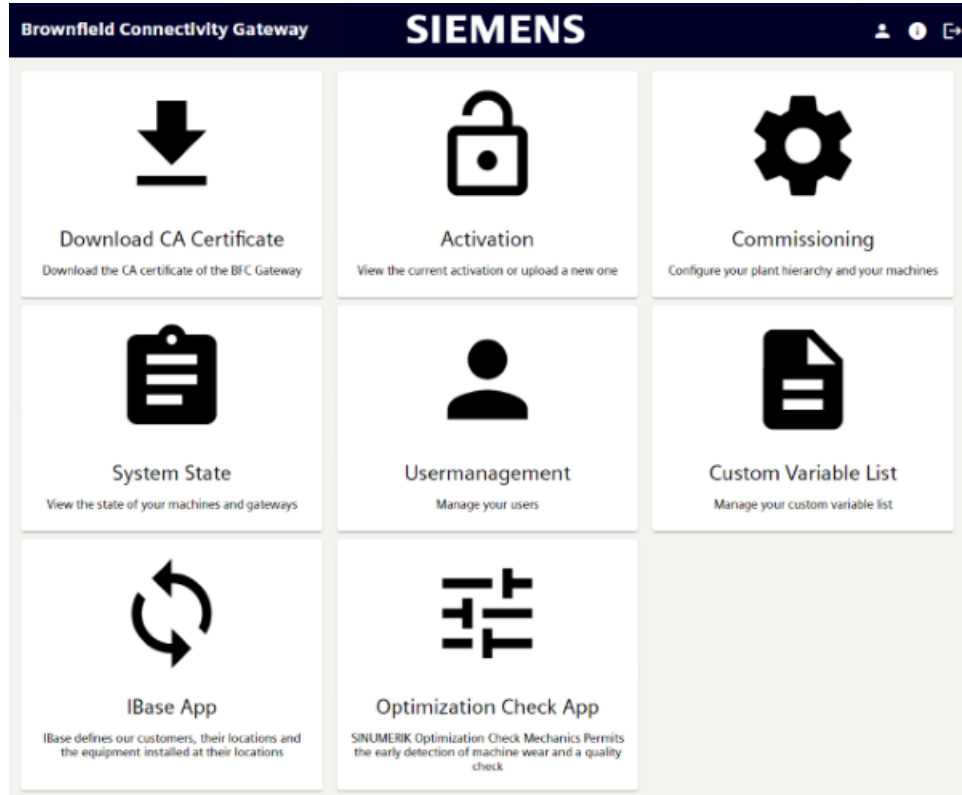
The Optimization Check app semiautomates the "Optimization Check" service in the environment of the BFC gateway.

14.2.1 Storing licenses

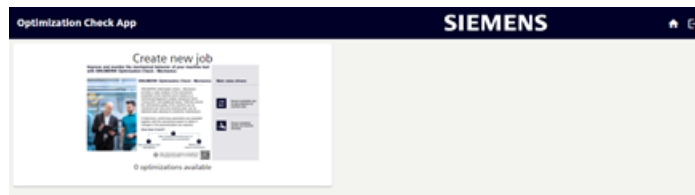
Licenses must be saved in order to use the Optimization Check app.

Procedure

1. Click on the "Optimization Check App" tile on the landing page.
A website opens.

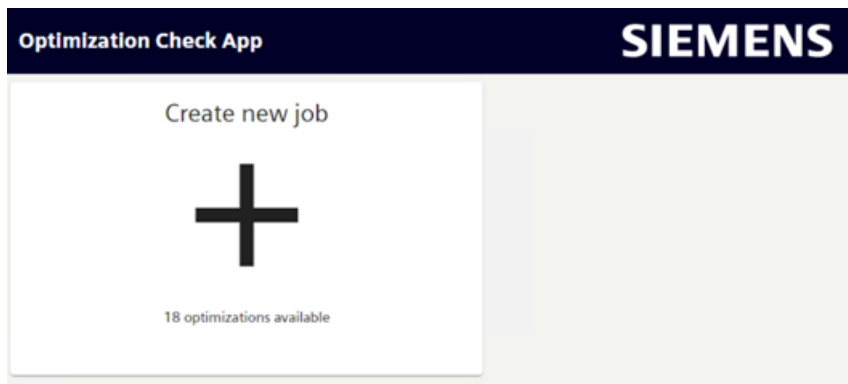


2. Double-click the image on the "Create new job" tile.
The image is enlarged to full-screen size.



3. Click the link above the image.
You are redirected to the Siemens Industry Mall.

4. Purchase the licenses in the Siemens Industry Mall.
5. Install the acquired licenses in the ActivationUI of the BFC gateway.
 - Licenses for the Optimization Check are permanently reserved in the system after completion.
 - The number of available licenses is displayed on the user interface of the Optimization Check app.



14.2.2 Creating a log

Procedure

1. Open the following link (<https://support.industry.siemens.com/cs/de/en/sc/5417>). At the end of the web page, you will find a description of how to create the archive. A manual is provided there.
2. Create the archive by following the description.

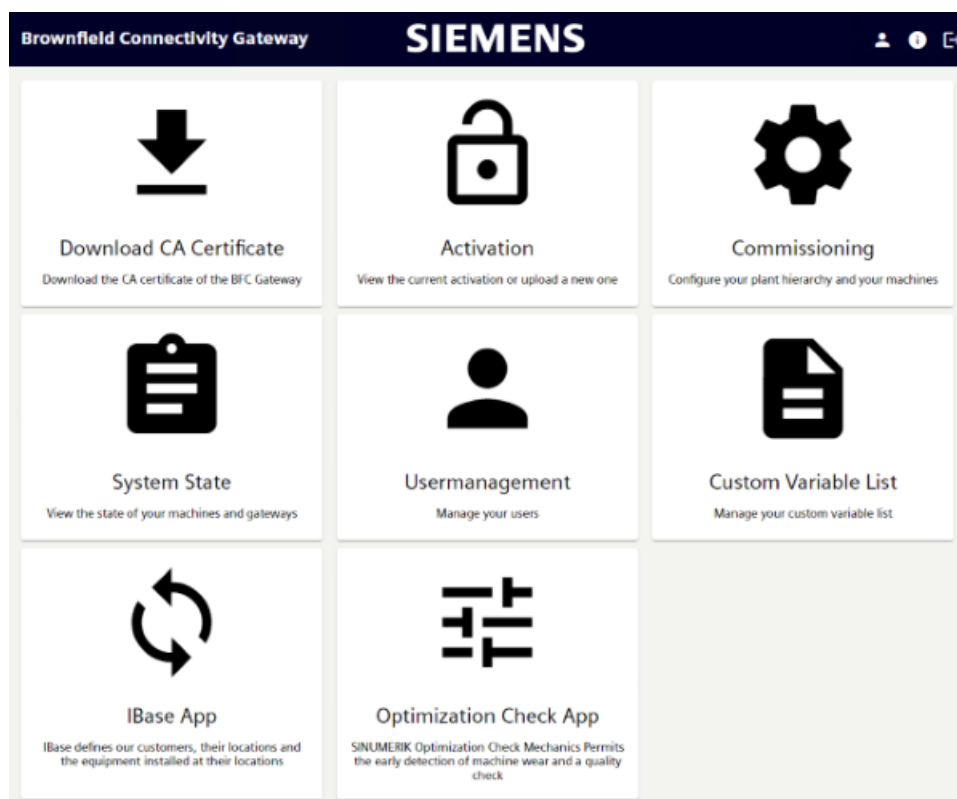
14.2.3 Using the Optimization Check app

Requirement

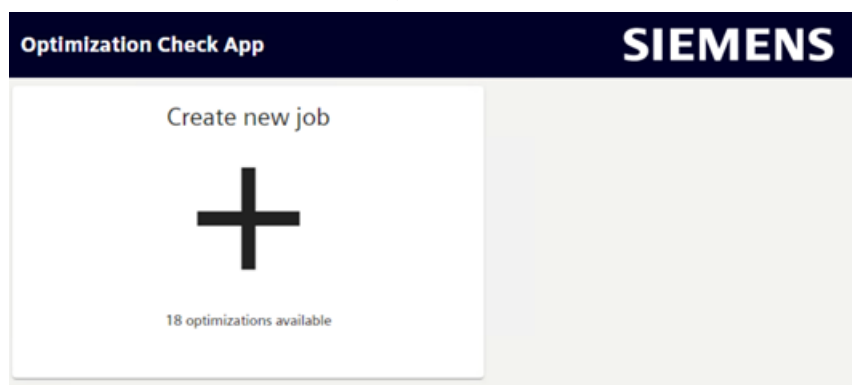
- You have acquired and stored licenses for the Optimization Check on the system.
- You have created the archive on the machine.

Procedure

1. Click on the "Optimization Check App" tile on the landing page.



2. Click the "+" icon to create a new job.



3. Choose your settings in the dialog box:

- Machine
- Archive
- AST files
- Version file

The "Create a new job" window with "new" status opens. After a few minutes, "new" status changes to "metadata-created" status.

The "gear wheel" button is displayed.

Create new job

Select machine...
LolaLinux ▼

Select archive...
200312ND_M25002.arc ▼

Select ast files...
AST_AX1_X1_SPG2016071408309.xml, ast_history_SPG2016071408309.xml ▼

Select version data file...
Lola+SPG2016071408309_version.txt ▼





4. Click the "gear wheel" button.
A dialog box opens.

5. Select the relevant axes.
Set the soft limits.
Save.

Choose Axis

☒ X1	min -40	max 40
☐ Y1	min -100	max 100
☐ Z1	min -100	max 100
☐ SP1	min 0	max 359
☐ X2	min -100	max 100
☐ Y2	min -100	max 100

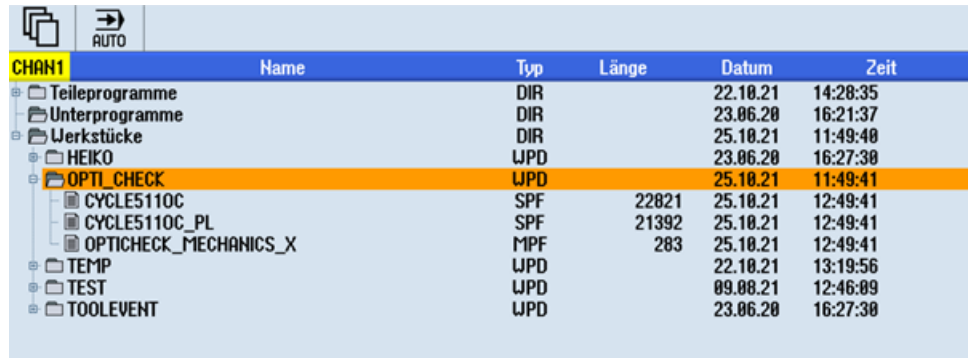



Result

Once you have saved, the status changes to "metadata-ready". In the background, the relevant part programs and trace configurations are stored on the machine. After this, the status changes to "programs-created" and you can record the trace session with the relevant programs.

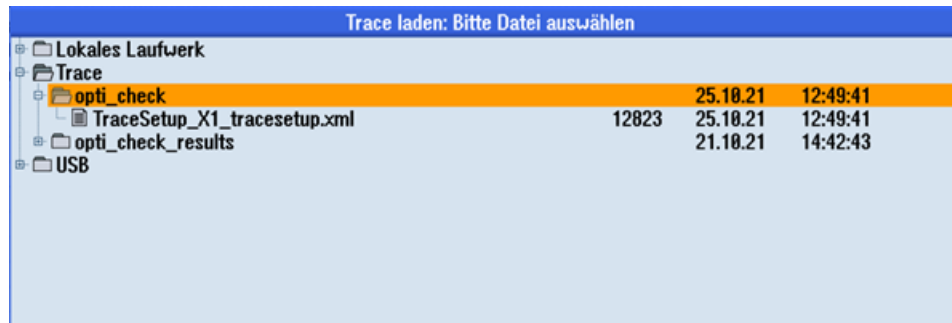
You will find the part programs and trace configurations here:

- The part programs are located under "Workpieces" in the folder "OPTI_CHECK".



CHAN1	Name	Typ	Länge	Datum	Zeit
Teileprogramme		DIR		22.10.21	14:28:35
Unterprogramme		DIR		23.06.20	16:21:37
Werkstücke		DIR		25.10.21	11:49:40
HEIKO		UPD		23.06.20	16:27:30
OPTI_CHECK		UPD		25.10.21	11:49:41
CYCLE5110C		SPF	22821	25.10.21	12:49:41
CYCLE5110C_PL		SPF	21392	25.10.21	12:49:41
OPTICHECK_MECHANICS_X		MPF	283	25.10.21	12:49:41
TEMP		UPD		22.10.21	13:19:56
TEST		UPD		09.08.21	12:46:09
TOOLEVENT		UPD		23.06.20	16:27:30

- The trace configurations are located in the "Trace" folder in the "opti_check" directory.



Trace laden: Bitte Datei auswählen				
Lokales Laufwerk				
Trace				
opti_check			25.10.21	12:49:41
TraceSetup_X1_tracesetup.xml		12823	25.10.21	12:49:41
opti_check_results			21.10.21	14:42:43
USB				

14.2.4 Recording a trace session

Requirement

The part programs and trace configurations are stored on the machine.

Procedure

- Select the program and load the associated trace configuration.
The programs and trace configurations are created according to the following pattern:
 - Programs: OPTICHECK_MECHANICS_<axis_name>.MPF
 - Trace configurations: TraceSetup_<axis_name>_tracesetup.xml
- When the trace configuration is loaded, first start the trace session and then the program.
- When recording has ended, save the trace session, including the stored data under the suggested name.
- Once the data for all selected axes have been recorded, the status of the job changes to "programs-finished" in the Optimization Check app.
- Download the data as a ZIP archive with the "Download" button.



Appendix

A.1 List of abbreviations

Admin	Administrator (user role)
BFC	Brownfield Connectivity
CIP	Common Industrial Protocol
CNC	Computerized Numerical Control
COM	Communication
CP	Communications Processor
CPU	Central Processing Unit
Curl	Client URL Request Library
DIR	Directory
EIP	Ethernet IP
FAQ	Frequently Asked Questions
GUD	Global User Data
h	Hour
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IB	Commissioning engineer (user role)
ID	Identification number
IE	Internet Explorer
IoT	Internet of Things
IPC	Industrial PC
MB	Megabyte
MLFB	Machine-Readable Product Code
MQTT	Message Queuing Telemetry Transport
MMM	Manage MyMachines
MCP	Machine Control Panel
NC	Numerical Control
NCU	Numerical Control Unit, NC hardware unit
OEE	Overall equipment efficiency
OEM	Original Equipment Manufacturer
OP	Operator Panel
OSS	Open Source Software
PC	Personal Computer
PCU	PC Unit
PDU	Protocol Data Unit
PLC	Programmable Logic Controller
RIP	REST Input Protocol

SI	SINUMERIK Integrate
SK	Softkey
PLC	Programmable Logic Controller
SSA	SINUMERIK Service Assistance
SW	Software
TSAP	Transport Service Access Point
URL	Uniform Resource Locator
UTC	Universal Time Coordinated

A.2 Checking the connection to a Heidenhain control

Tool "TNCcmd.exe" can be used to check the connection to a Heidenhain control.

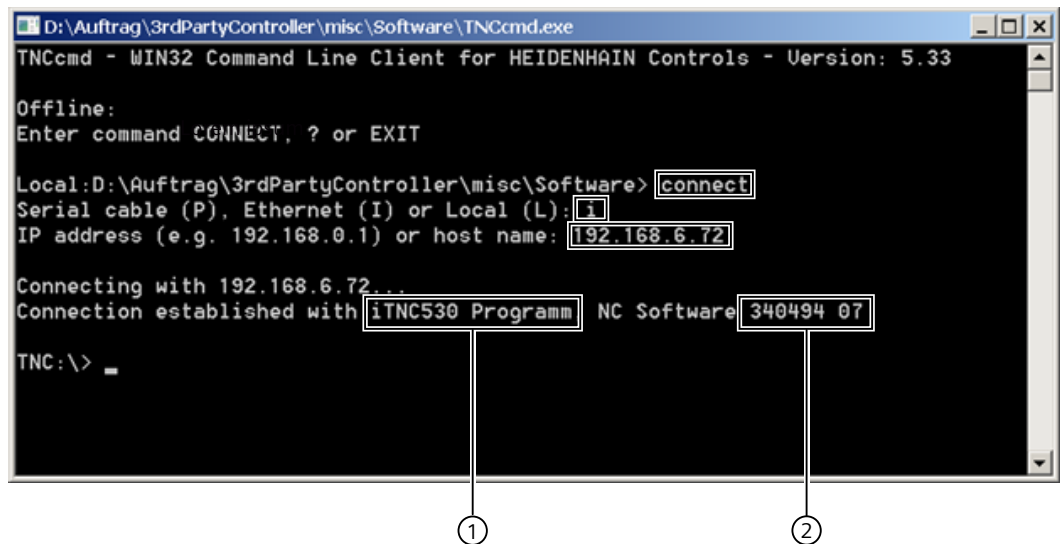
Note

Contact the hotline to obtain the tool.

Procedure

1. Start the "TNCcmd.exe" tool.
2. Enter the following commands and/or parameters one after the other.
 - Command: "connect"
 - Parameter: "i"
 - Parameter: "<IP address of the control>"

If the connection is successful, a message, information about the type of control and the version are output.



The screenshot shows a Windows command prompt window titled "D:\Auftrag\3rdPartyController\misc\Software\TNCcmd.exe". The text inside the window is as follows:

```
TNCcmd - WIN32 Command Line Client for HEIDENHAIN Controls - Version: 5.33

Offline:
Enter command CONNECT, ? or EXIT

Local:D:\Auftrag\3rdPartyController\misc\Software> connect
Serial cable (P), Ethernet (I) or Local (L): i
IP address (e.g. 192.168.0.1) or host name: 192.168.6.72

Connecting with 192.168.6.72...
Connection established with iTNC530 Programm NC Software 340494 07

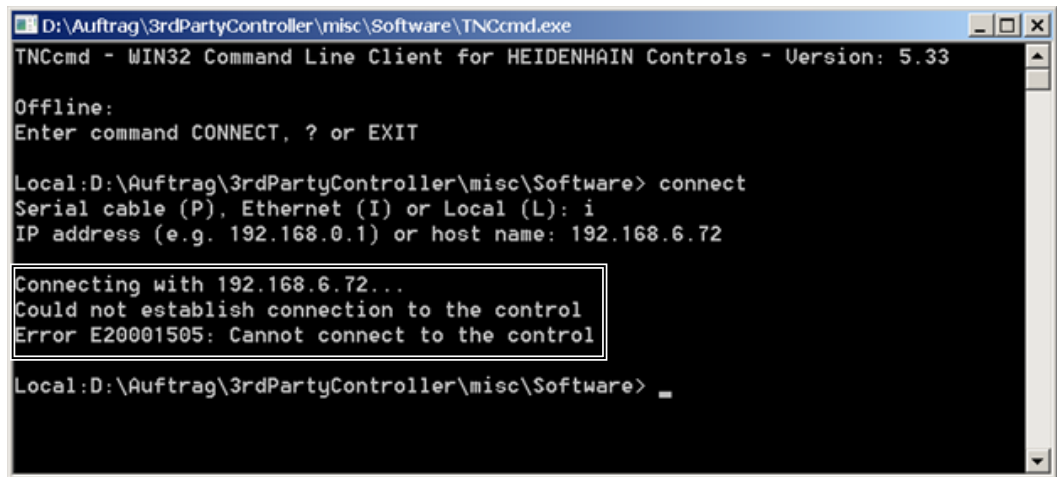
TNC:\> _
```

Two callout boxes with numbers 1 and 2 are present. Callout 1 points to the text "iTNC530 Programm" and callout 2 points to the text "340494 07".

- ① Control type
- ② Version

3. Resolve any errors that may have occurred.

– Cannot connect to the control



```
D:\Auftrag\3rdPartyController\misc\Software\TNCcmd.exe
TNCcmd - WIN32 Command Line Client for HEIDENHAIN Controls - Version: 5.33

Offline:
Enter command CONNECT, ? or EXIT

Local:D:\Auftrag\3rdPartyController\misc\Software> connect
Serial cable (P), Ethernet (I) or Local (L): i
IP address (e.g. 192.168.0.1) or host name: 192.168.6.72

Connecting with 192.168.6.72...
Could not establish connection to the control
Error E20001505: Cannot connect to the control

Local:D:\Auftrag\3rdPartyController\misc\Software> _
```

Troubleshooting:

Ensure that the Heidenhain machine in the customer network is ready and can be accessed.

– Access privilege not granted

```

D:\Auftrag\3rdPartyController\misc\Software\TNCcmd.exe
TNCcmd - WIN32 Command Line Client for HEIDENHAIN Controls - Version: 5.33

Offline:
Enter command CONNECT, ? or EXIT

Local:D:\Auftrag\3rdPartyController\misc\Software> connect
Serial cable (P), Ethernet (I) or Local (L): i
IP address (e.g. 192.168.0.1) or host name: 192.168.6.72

Connecting with 192.168.6.72...
Error E20001717: Access privilege not granted

Local:D:\Auftrag\3rdPartyController\misc\Software>

```

Troubleshooting:

In the machine or MOD operating mode, press the "External Access On/Off" softkey.

If the "External Access On/Off" softkey is not available, the entry

"REMOTE.LOCKSOFTKEYVISIBLE = YES" must be available in the OEM.SYS configuration file. The entry must not be commented out by a semicolon.

With newer NC software versions from 34049X-03 or higher, you can prohibit access by certain clients.

To do this, entry "REMOTE.PERMISSION" must be present in the TNC.SYS. This entry contains a list of IP addresses or host names of clients for which remote access is permitted. Add the IP address or the host name of the BFC as shown in the following example.

Example: REMOTE.PERMISSION = PC123;192.168.0.92

An SE Linux firewall is integrated for newer NC software versions from 60642x. If access does not function, then it is possible that a rule is active that is blocking access.

Machine builders (OEMs) have the option of restricting access to areas or variables.

Contact the machine manufacturer of your control.

A.3 Checking the connection to the MTConnect control

Procedure

1. Before starting the installation, carefully ensure that the MTConnect interface of the machine can be accessed in the customer network.
2. For each machine, determine the data made available from MTConnect agent.
To do this, call up the following URLs once from a PC in the customer network using a web browser:
 - <http://<MTConnect-Agent-IP>/probe>
 - <http://<MTConnect-Agent-IP>/current>
3. Save the displayed result (XML) for diagnostics.

A.4 Check connection to FANUC control system

Use the "Fanuc Focas Tester" to check the connection to a FANUC control system.

We strongly recommend checking the accessibility of the FANUC control system in the network **before** commissioning.

Note

"Fanuc Focas Tester" tool

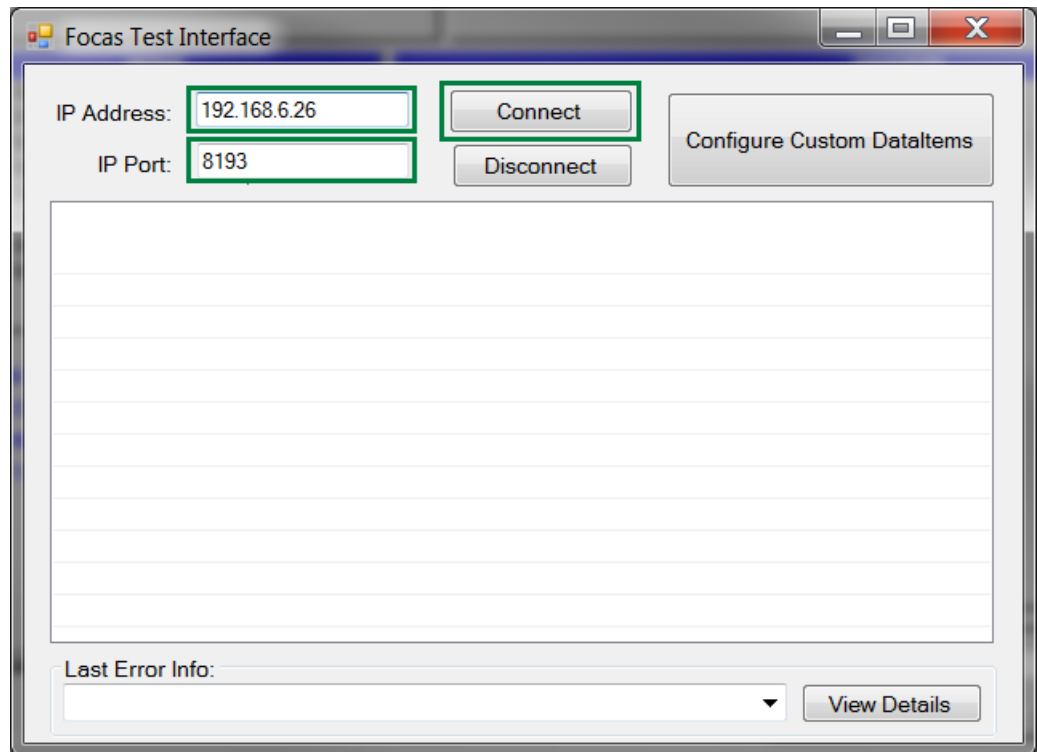
The tool is not included with the software, and can be purchased under this address (<http://www.memexoe.com/support/downloads/>).

Parameters

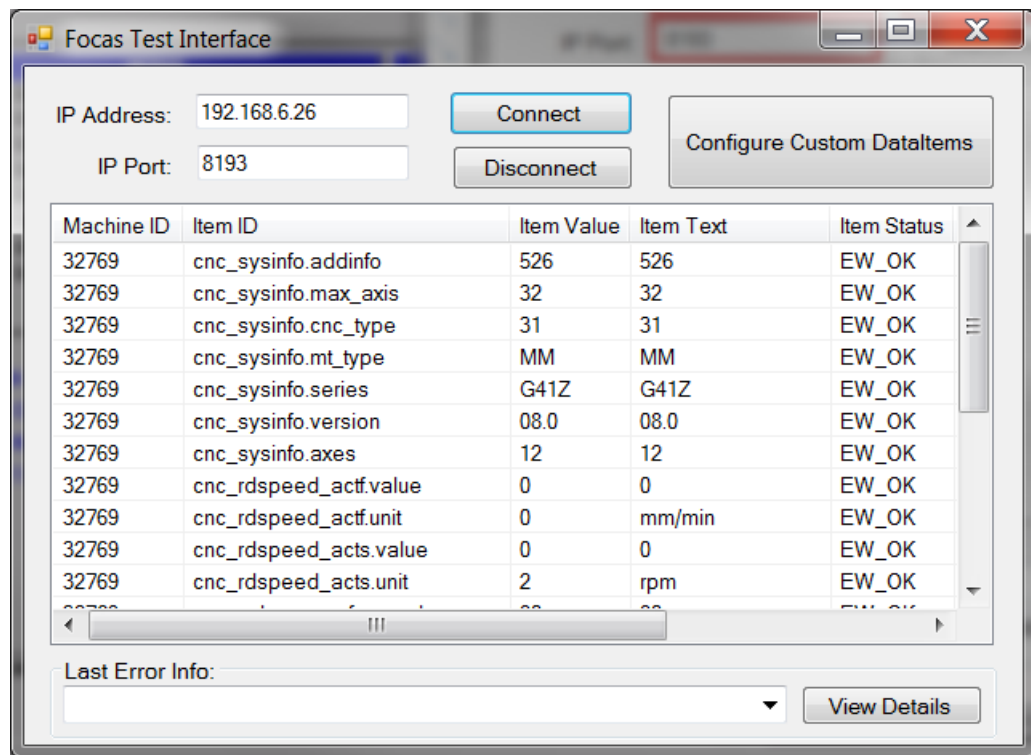
Parameter	Description
IP address	IP address of the Fanuc control system
Port	Port of the Fanuc control system Remark: The port only has to be adapted if this was explicitly communicated.

Procedure

1. Start the "Fanuc Focas Tester" tool.
2. The input window opens.
 - Enter the addresses.
 - Click on "Connect" to check the connection to the FANUC control system.



3. The result of the connection check is shown as follows:
 - Successful connection to the FANUC control system:
Information about the FANUC control system is provided in the lower section of the input window.



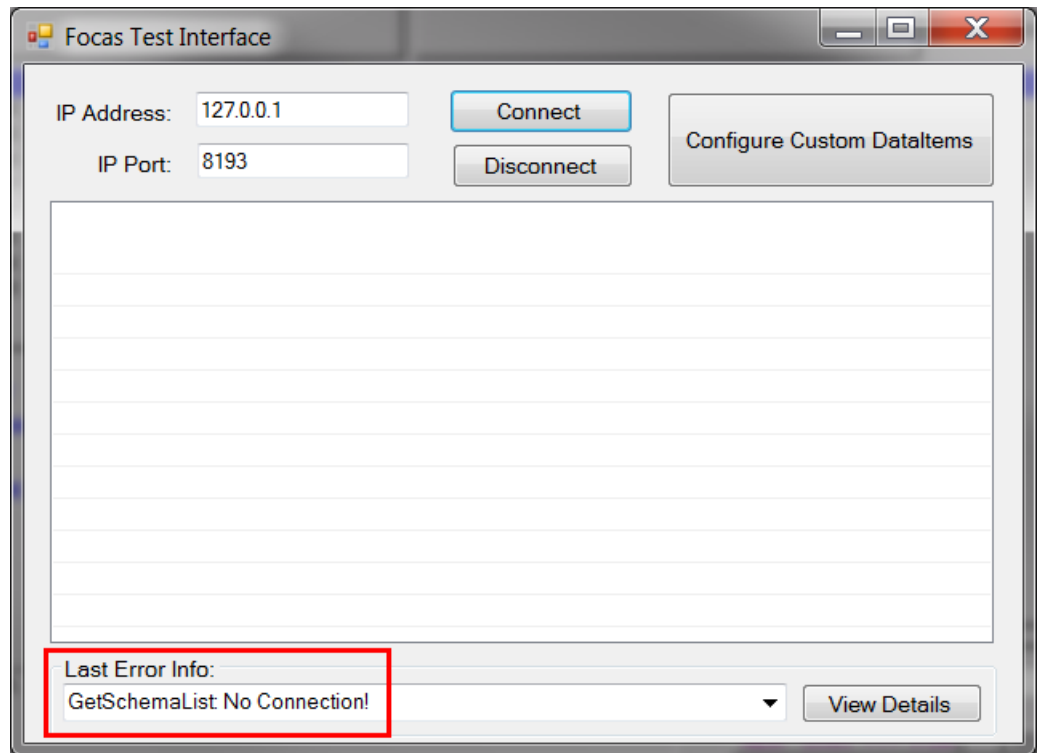
- OR -

- No connection to the FANUC control system:
The error message is displayed.

Note

Error message

In the case of an error message, check whether the FANUC control system can be reached in the network using a ping command.



A.5 API calls of the FOCAS interface

API calls of the FOCAS interface

The BFC driver for machines equipped with a FANUC control system support the API calls of the FOCAS interface listed in the following table.

API call (FOCAS v4.14)	Supported FANUC devices	Description
M = Machining / T = Turning / LC = Loader / P = Punch press / L = Laser, W = Wire		
cnc_allclibhdl3	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L, W), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Internal: Allocates the library handle and connects to CNC that has the specified IP address or the Host Name.
cnc_freelibhdl	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L, W), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Internal: Frees the library handle which was used by the Data window library.
cnc_sysinfo	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L, W), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Reads system information such as type of CNC system, Machining(M) or Turning(T), series and version of CNC system software and number of the controlled axes.

API call (FOCAS v4.14)	Supported FANUC devices	Description
M = Machining / T = Turning / LC = Loader / P = Punch press / L = Laser, W = Wire		
cnc_statinfo	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L, W), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Reads the status information of CNC like mode, operation status, status of axis movement, ...
cnc_setpath	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Selects the path number which is the target path in the multi-path system or system with loader control.
cnc_rdpdf_subdirn	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Reads the number of subfolders/files under the specified folder.
cnc_rdpdf_subdir	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Reads the information of subfolder under the specified folder.
cnc_rdpdf_alldir	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Reads the file information under the specified folder.
cnc_getfigure	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L, W), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Internal: Reads the maximum valid figures and the number of decimal places under the input/output unit, which is related to various data of CNC.
cnc_rddynamic2	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L), 18i (M, T, P), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Reads various data which changes at CNC operation at a time. (like axis position)
cnc_rdparam	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L, W), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Reads the parameter specified by "number" and "axis" (only for the parameter with axis).
cnc_diagnoss	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L, W), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Internal: Reads the diagnosis specified by "number" and "axis" (only for the diagnosis with axis).
cnc_rdset	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L), 18i (M, T, P), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Internal: Reads the setting data specified by "number" and "axis" (only for the setting data with axis).
cnc_rdsetnum	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L, W), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Internal: Reads minimum, maximum, total number of the CNC setting data.
cnc_rdaxisdata	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Reads various data relating servo axis/spindle axis that is got by cnc_rdposition, cnc_rdspeed, cnc_rdsvmeter, cnc_rdspmeter, cnc_rdhndintrpt, with supporting "extended axis name".
cnc_rdaxisname	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L), 18i (M, T, P), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Reads the axis names from 1st axis to the specified axis number.
cnc_rdspdlname	0i-B/C (M, T), 0i-D/F (M, T), 15i (M), 16i-A/B (M, T, P, L), 18i (M, T, P), 21i (M, T), 30i/31i/32i-A (M, T), 30i/31i/32i/35i-B (M, T, L), Mate i-D	Reads the spindle names from 1st spindle to the specified spindle number.

API call (FOCAS v4.14)	Supported FANUC devices	Description
M = Machining / T = Turning / LC = Loader / P = Punch press / L = Laser, W = Wire		
cnc_alarm2	0i-B/C (M, T, P), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L), 18i (M, T, P), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Reads the alarm status of CNC.
cnc_rdalmmsg	0i-B/C (M, T, P), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L), 18i (M, T, P), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Reads the currently arising CNC alarm messages.
cnc_rdalmmsg2	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Reads the currently arising CNC alarm messages. (version for FOCAS 2)
pmc_rdpmcrrng	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L, W), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Reads the PMC data of the specified PMC address/range.
pmc_get_number_of_pmc	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Reads the number of existing PMC paths (i.e. number of unit).
pmc_select_pmc_unit	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Selects PMC that will be the target of other PMC FOCAS2 function by the unit type.
pmc_rdalmmsg	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L, W), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Reads the alarm messages from PMC.
pmc_get_pmc_unit_types	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Reads the PMC system information that shows what kind of PMC units compose the multi-path PMC system.
pmc_wrpmcrrng	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L, W), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Writes the PMC data of the specified PMC address/range.
cnc_wrparam	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L, W), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Writes the parameter specified by "number" and "axis" (only for the parameter with axis).
cnc_wrset	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L), 18i (M, T, P), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Internal: Writes the setting data specified by "number" and "axis" (only for the setting data with axis).
cnc_wrtimer	15i (M)	Sets cutting time, cycle time, and other timer data of CNC.
cnc_rdtimer	15i (M)	Gets cutting time, cycle time, and other timer data of CNC.
cnc_upstart4	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Notifies the start of uploading NC data.
cnc_upload4	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Read NC data registered on the memory in CNC.
cnc_upend4	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Notifies the end of uploading NC data.

API call (FOCAS v4.14)	Supported FANUC devices	Description
M = Machining / T = Turning / LC = Loader / P = Punch press / L = Laser, W = Wire		
cnc_dwnstart4	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Notifies the start of downloading NC data.
cnc_download4	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Output NC data to be registered.
cnc_dwnend4	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Notifies the end of downloading NC data.
cnc_pdf_del	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Deletes the folder or file under the specified folder.
cnc_pdf_rdmain	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Reads the file information that is select currently as the main program.
cnc_exeprgname	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Reads full path name of the program which is being currently executed in CNC.
cnc_exeprgname2	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Reads full path name of the program which is being currently executed in CNC.
cnc_pdf_rename	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Renames the folder or file under the specified folder.
cnc_pdf_add	0i-D/F (M, T, P), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Motion i-A	Internal: Creates the folder or file under the specified folder.
cnc_modal	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L, W), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Reads the modal information of CNC.
cnc_rdtofs	0i-B/C (M, T), 0i-D/F (M, T, P), 15i (M), 16i-A/B (M, T, P, L, W), 18i (M, T, P, W), 21i (M, T), 30i/31i/32i-A (M, T, W), 30i/31i/32i/35i-B (M, T, P, L, W), Mate i-D, Mate i-H, Motion i-A	Reads the tool offset value specified by "number", "type".

A.6 AMP gateway

A.6.1 AMP gateway signals

The gateway supports the following signals.

Signal	Function	Type	Comment
MDA_SIGNAL	Machine state	String	hex format with leading zeros ("00000041" -> bit 0 and bit 7 set)
MDA_COUNTER_1	Machine counter good unit number absolute	Int	-
MDA_COUNTER_2	Machine counter reject unit number absolute	Int	-

Signal	Function	Type	Comment
MDA_COUNTER_3	Energy unit counter or machine counter post working absolute	Int	-
MDA_MAN_STATUS	Manual machine state	String	-
PDC_Key	Part type or task ID	String	-
CycleCounterDelta_1	Part counter good unit number incremental	Int	Must be sent using the PDC_KEY
CycleCounterDelta_2	Part counter reject unit number incremental	Int	Must be sent using the PDC_KEY
CycleCounterDelta_3	Part counter post working number incremental	Int	Must be sent using the PDC_KEY
CycleTimeDelta_1	Part type referred target cycle time	Int	Must be sent using the PDC_KEY
CycleTimeDelta_2	Part type referred actual cycle time	Int	Must be sent using the PDC_KEY

Note

Signals

The signals are case sensitive.

The signals must be made available in this form.

Additional information on the relevant topic names is provided in Chapter: Data selection and topics of the AMP gateway (Page 517).

A.6.2 Script logic example: AMP_SIGNAL set

Send a set with MDA_SIGNAL "00000001" (production):

```
var nowDate = new Date();
var setInfoSignal {
    "Source": {
        "ProjectID": "iotserver",
        "DeviceType": "software",
        "ApplicationType": "scriptlogic",
        "ClientID": "testclient"
    },
    "Set": {
        "Time": nowDate.toISOString(),
        "Name": "AMP_SIGNAL"
    }
};
var v = [{
    "Name": "MDA_SIGNAL",
    "Type": "string",
    "Value": "00000001"
}];
```

```
setInfoSignal.Set["Data"] = v;  
publishReadingSet(setInfoSignal);
```

A.6.3 Script logic example: AMP_PARTCOUNT set

Send a set with MDA_COUNTER_1 and value "714":

```
var nowDate = new Date();  
var setInfoCounter1 = {  
    "Source": {  
        "ProjectID": "iotserver",  
        "DeviceType": "software",  
        "ApplicationType": "scriptlogic",  
        "ClientID": "testclient"  
    },  
    "Set": {  
        "Time": nowDate.toISOString(),  
        "Name": "AMP_PARTCOUNT"  
    }  
};  
var vCount = [{  
    "Name": "MDA_COUNTER_1",  
    "Type": "int",  
    "Value": 714  
}];  
setInfoCounter1.Set["Data"] = vCount;  
publishReadingSet(setInfoCounter1);
```

A.6.4 Script logic example: AMP_CYCLEDATA set

Send a set with PDC_KEY, CycleCounterDelta_1, CycleCounterDelta_2, CycleCounterDelta_3, CycleTimeDelta_1 and CycleTimeDelta_2:

```
var nowDate = new Date();  
var setInfoCycleData = {  
    "Source": {  
        "ProjectID": "iotserver",  
        "DeviceType": "software",  
        "ApplicationType": "scriptlogic",  
        "ClientID": "testclient"  
    },  
    "Set": {  
        "Time": nowDate.toISOString(),  
        "Name": "AMP_CYCLEDATA"  
    }  
};  
var vCycleTime = [{  
    "Name": "PDC_Key",  
    "Type": "string",  
    "Value": "Part 4711"  
},
```



```

{
  "Name": "CycleCounterDelta_1",
  "Type": "int",
  "Value": 2
},
{
  "Name": "CycleCounterDelta_2",
  "Type": "int",
  "Value": 0
},
{
  "Name": "CycleCounterDelta_3",
  "Type": "int",
  "Value": 0
},
{
  "Name": "CycleTimeDelta_1",
  "Type": "int",
  "Value": 0
} ,
{
  "Name": "CycleTimeDelta_2",
  "Type": "int",
  "Value": 200
}
];

setInfoCycleData.Set["Data"] = vCycleTime;
publishReadingSet(setInfoCycleData);

```

A.6.5 Alarms in the AMP gateway

The following attributes are supported for an alarm in the gateway.

Attribute	Description
id	Alarm ID (alarm number)
source	Source of the alarm
text	Alarm text
category	Alarm category
instance	Within an alarm source, an alarm can occur a multiple number of times at the same time. The instance ID is used to make a differentiation.
priority	Alarm priority
state	Alarm state
timestamp	Alarm time stamp

For an alarm, these parameters are sent to the AMC server. If not all of them can be supplied, then some of these parameters remain empty.

Additional information on the relevant topic names for alarms is provided in Chapter: Data selection and topics of the AMP gateway (Page 517).

A.6.6 AMP gateway logging

Each send operation in the AMP server is listed in the gateway log. You can see the complete telegram, and clearly understand which signals and alarms were sent.

Example of a connection check telegram

```
INFO: 2019/11/15 10:12:41.013524 Module[main] - ID[]
- Message[Sending message <Message from="//IOTTEST/Unknown"
seq="5" time="2019-11-15T10:12:41Z"><ConnectionCheckRequest></
ConnectionCheckRequest></Message> ...]
```

Example of a signal telegram

```
<<Module[main] - ID[] - Message[Sending
message <Message from="//IOTTEST/Unknown" seq="12718"
time="2019-10-28T08:10:54Z"><VLRecordData name="HTTP_MDA_MAIN"
time="2019-10-28T08:10:51Z" change="4268"><Variable
name="MDA_SIGNAL" type="string" value="00000004" changed="1"></
Variable></VLRecordData><VLRecordData name="HTTP_MDA_MAIN"
time="2019-10-28T08:10:51Z" change="4269"><Variable
name="MDA_COUNTER_1" type="string" value="535" changed="1"></
Variable></VLRecordData><VLRecordData name="HTTP_MDA_MAIN"
time="2019-10-28T08:10:51Z" change="4270"><Variable name="PDC_Key"
type="string" value="Unnamed_Part" changed="1"></Variable><Variable
name="CycleCounterDelta_1" type="string" value="1"
changed="1"></Variable><Variable name="CycleCounterDelta_2"
type="string" value="0" changed="1"></Variable><Variable
name="CycleCounterDelta_3" type="string" value="0" changed="1"></
Variable><Variable name="CycleTimeDelta_1" type="string"
value="0" changed="1"></Variable><Variable name="CycleTimeDelta_2"
type="string" value="10" changed="1"></Variable></VLRecordData></
Message> ...]>>
```

Example of an alarm telegram

```
<<Module[main] - ID[] - Message[Sending message
<Message from="//IOTTEST/Unknown" chan="alarm" seq="12728"
time="2019-10-28T08:13:38Z"><VLAlarmList name="HTTP_MDA_MAIN"
time="2019-10-28T08:13:38Z" change="1"><Alarm id="500"
source="fanuc-sim" text="(Y)+ OVERTRAVEL ( SOFT 1 )" category=""
instance="1234" p1="" p2="" p3="" p4="" p5="" p6="" p7="" p8=""
priority="0" state="" timestamp="2019-10-28T08:13:38Z"></Alarm></
VLAlarmList></Message> ...]>>
```

For errors, additional information can be found, for example if the AMP server does not respond, or the telegram was incorrectly acknowledged.

A.7 BFC client data points

A.7.1 Overview

The control system addresses supported by the BFC client are described below. These addresses can be used for reading / writing variables from a SINUMERIK control system.

More information on the configuration of the reading processes can be found in this Chapter: Creating a BFC client (Page 97)

You can find more information on the write processes in Chapter: Write access to variables (Page 368).

Because the writing of variables on a SINUMERIK control system represents a security risk, the address of a variable to be written to must be released explicitly for this purpose. More information can be found in Chapter: Configuring the release list for write accesses (Page 368).

Variables in the SINUMERIK control system

The following data areas of the SINUMERIK control system are frequently used for write access:

- R parameters
- GUDs
- Machine data
- PLC variables

A.7.2 NC variables

The following table lists examples for addressing SINUMERIK variables.

Addresses of SINUMERIK variables	Description
/Bag/State/opMode[u1]	Active mode [type:Integer]
/Nck/ChannelDiagnose/poweronTime[1]	Time since last normal boot (in minutes) [type:Integer]
/Nck/State/sysTimeSinceStartup[u1]	System run time in seconds since NCK ramp-up [type:Float]
/Nck/Configuration/accessLevel	Level of the access rights currently set [type:Integer]
/Nck/State/numAlarms[u1]	Number of pending general alarms [type:Integer]
/Channel/State/progStatus[u1]	Program status [type:Integer]
/Channel/State/stopCond[u1]	Number of the NC stop state [type:Integer]
/Channel/state/allAxesStopped	Code specifying whether axes are in exact stop [type:Bool]
/Channel/GeometricAxis/feedRateOvr[u1,1]	Feedrate override for first channel and first axis in percent [type:Float]
/Channel/Spindle/speedOvr[u1,1]	Spindle override in percent [type:Float]
/Channel/State/actParts[u1]	Number of workpieces produced Corresponds to NC variable \$AC_ACTUAL_PARTS [type:integer]
/Channel/State/actToolIdent[u1]	Identifier of active tool [type:String]
/Channel/ProgramInfo/blockNoStr[u1]	Block number [type:String]

Addresses of SINUMERIK variables	Description
/Channel/ProgramPointer/progName[u1,1]	Name of the active NC program [type:String]
/Channel/ProgramInfo/workPName[u1]	Name of the active workpiece [type:String]
/Channel/State/cmdDwellTime[u1,1]	Programmed dwell time [type:Float]
/Channel/State/G0Mode[u1,1]	EN: G00 is active [type:Integer]
/Channel/MachineAxis/cmdContrPos[u1,1]	Position, setpoint of the first axis [type:Float]
/Channel/GeometricAxis/actToolBasePos[u1,1]	Position, setpoint of the first axis [type:Float]
/Channel/MachineAxis/vaCurr[u1,1]	Drive actual current value [type:Float]
/Channel/MachineAxis/vaTorque[u1,1]	Drive torque setpoint [type:Float]
/Channel/MachineAxis/vaPower[u1,1]	Active drive power [type:Float]
/Channel/Spindle/actSpeed[u1,1]	Spindle speed, actual value [type:Float]
/Channel/SelectedFunctions/Mval[u1]	Value of the M-function [type:Integer]
/Channel/Parameter/R[u1,10]	R parameter R10 [type:Float]
/Channel/MachineAxis/aaLoad[u1,1]	Drive utilization in % [type:Float]*)
/Channel/MachineAxis/aaTorque[u1,1]	Drive torque setpoint in Nm [type:Float]*)
/Channel/MachineAxis/aaCurr[u1,1]	Axis current actual value in A [type:Float]*)

*) For the acquisition of these drive variables, the machine data "MD36730 \$MA_DRIVE_SIGNAL_TRACKING" must be set. For this purpose, you need the additional license "Analysis of internal drive values":

→ SINUMERIK 840D pl : 6FC5251-0AB17-0AA0

→ SINUMERIK 828D: 6FC5800-0AS53-0YB0

→ SINUMERIK 840D sl : 6FC5800-0AM41-0YB0

More information and the complete list of the available OPI addresses are available in:

- List Manual: SINUMERIK 840D sl, NC Variables and Interface Signals (<https://support.industry.siemens.com/cs/ww/en/view/109769139>)
- List Manual: SINUMERIK 840D, NC Variables and Interface Signals (<https://support.industry.siemens.com/cs/document/109738472>)

A.7.3 PLC tags

The following syntax forms are supported for access to the PLC:

- HMI-Advanced syntax
- Operate S7 syntax

Both syntax forms are equally suitable for controls based on HMI-Advanced and for controls based on SINUMERIK Operate.

If necessary, a required conversion is performed by the BFC client.

The following table shows examples of access operations to a PLC data block.

HMI-Advanced syntax	S7 syntax	BFC type
/Plc/DataBlock/Bit[c300,40.1]	DB300.DB40.1	bool
/Plc/DataBlock/Bit[c300,40.1]	DB300.DBX40.1	bool
/Plc/DataBlock/Byte[c300,40]	DB300.DBB40	int

HMI-Advanced syntax	S7 syntax	BFC type
/Plc/DataBlock/Word[c300,20]	DB300.DBW40	int
/Plc/DataBlock/Word[c300,>20]	DB300.DBW20	int
/Plc/DataBlock/DoubleWord[c300,10]	DB300.DBD40	int
/Plc/DataBlock/DoubleWord[c300,>10]	DB300.DBD10	int
/Plc/DataBlock/Integer[c300,20]	DB300.DBW40:INT	int
/Plc/DataBlock/Integer[c300,>20]	DB300.DBW20:INT	int
/Plc/DataBlock/DoubleInteger[c300,10]	DB300.DBD40:DINT	int
/Plc/DataBlock/DoubleInteger[c300,>10]	DB300.DBD10:DINT	int
/Plc/DataBlock/Float[c300,10]	DB300.DBD40:REAL	float
/Plc/DataBlock/Float[c300,>10]	DB300.DBD10:REAL	float
/Plc/DataBlock/String[c300,64]	DB300.DBB64:STRING	string ¹⁾
/Plc/DataBlock/Char[c300,66,#8]	DB300.DBB66:CHAR[8]	string ²⁾

¹⁾ Access to a S7 string structure in the DB300 as from byte offset 64. To write to an S7 string, the S7 string structure must already be initialized with regard to the maximum length (byte 0).

²⁾ Access to an 8-character long string of ASCII characters in the DB300 as from byte offset 66.

The addressing of strings in the form of /Plc/DataBlock/String[c300,64] and /Plc/DataBlock/Char[c300,66,#8] is not supported natively by HMI-Advanced. It is available only in the BFC client and was introduced to enable a uniform procedure between the generations of control systems with HMI-Advanced and SINUMERIK Operate.

A.7.4 Machine data

Machine data of the control is accessed via an address that comprises a prefix and the name of the machine data.

The prefix is specific for the particular machine data type.

Table A-1 Possible prefixes

Machine data type	Prefix
General machine data	"/INC/_N_NC_TEA_ACX"
Channel machine data	"/INC/_N_CH_TEA_ACX"
Axis machine data	"/INC/_N_AX_TEA_ACX"
General setting machine data	"/INC/_N_NC_SEA_ACX"
Channel setting machine data	"/INC/_N_CH_SEA_ACX"
Axis machine data	"/INC/_N_AX_SEA_ACX"

Example: General machine data

General machine data			
10050	\$MN_SYSCLOCK_CYCLE_TIME	0.002 s	po
10061	\$MN_POSCTRL_CYCLE_TIME	0.002 s	po

Address	Value	Type
/NC/_N_NC_TEA_ACX/\$MN_SYSCLOCK_CYCLE_TIME	0.002	float
/NC/_N_NC_TEA_ACX/\$MN_POSCTRL_CYCLE_TIME	0.002	float

General machine data				
10000[0]	\$MN_AXCONF_MACHAX_NAME_TAB	X1		po
10000[1]	\$MN_AXCONF_MACHAX_NAME_TAB	Y1		po
10000[2]	\$MN_AXCONF_MACHAX_NAME_TAB	Z1		po

Address	Value	Type
/NC/_N_NC_TEA_ACX/\$MN_AXCONF_MACHAX_NAME_TAB[1]	X1	string
/NC/_N_NC_TEA_ACX/\$MN_AXCONF_MACHAX_NAME_TAB[2]	Y1	string
/NC/_N_NC_TEA_ACX/\$MN_AXCONF_MACHAX_NAME_TAB[3]	Z1	string

When accessing indexed machine data, the address must be extended by the index (...[<index>]).

Example: Channel machine data

Channel machine data				
20000	\$MC_CHAN_NAME	CHAN1		po
20050[0]	\$MC_AXCONF_GEOAX_ASSIGN_TAB	1		po
20050[1]	\$MC_AXCONF_GEOAX_ASSIGN_TAB	2		po

Address	Value	Type
/NC/_N_CH_TEA_ACX/\$MC_CHAN_NAME[1]	CHAN1	string
/NC/_N_CH_TEA_ACX/\$MC_AXCONF_GEOAX_ASSIGN_TAB[1,1]	1	int
/NC/_N_CH_TEA_ACX/\$MC_AXCONF_GEOAX_ASSIGN_TAB[1,2]	2	int

When accessing channel-specific machine data, the address must be extended by the channel number (...[<Channel>]). For indexed machine data, a second index must also be used (...[<Channel>,<Index>]).

When accessing axis machine data, the address must be extended by the axis number (...[<axis>]). For indexed machine data, a second index must also be used (...[<axis>,<index>]).

The addresses described here are applicable for all SINUMERIK controls.

A.7.5 Global User Data (GUD)

The SINUMERIK control system allows global variables to be defined which can then be read and written to in an NC program.

Read and write access to these variables is also possible via the BFC client.

Overview

The GUD variables can be defined NC-globally or channel-specifically.

Various data types are available, for example: INT, REAL, STRING.

The GUD variables are defined in a DEF file on the SINUMERIK control system.

More information

More information on defining user variables (DEF) can be found in:

- SINUMERIK 840D sl Programming Manual, NC programming (<https://support.industry.siemens.com/cs/document/109767448/sinumerik-840d-sl-nc-programming?dti=0&lc=en-WW>)

More information on defining user data can be found in:

- SINUMERIK 840D/840Di/810D Programming Manual, Work preparation (<https://support.industry.siemens.com/cs/document/108681744/sinumerik-840d-840di-810d-programming-guide-advanced?dti=0&lc=en-WW>)

More information on defining and activating user variables can be found in:

- SINUMERIK 840D sl/828D Operating Manual, Turning (<https://support.industry.siemens.com/cs/mdm/48057241?c=21169303051&lc=en-WW>)

More information on user data/user variables (GUD, PUD, LUD) can be found in:

- SINUMERIK 840D/840Di/810D Operating Manual, HMI-Advanced (<https://support.industry.siemens.com/cs/document/109301956/sinumerik-840d-840di-810d-hmi-advanced?dti=0&lc=en-WW>)

DEF file

The DEF files are stored in the following operating area: "Commissioning" > "System data" > "NC data" > "Definitions".

In the operating area you can display existing DEF files or create new DEF files. If you have to define new GUDs, we recommend that you create a new DEF file to be used on all affected machines, e.g. UGUD.DEF or GUD4.DEF.

File names	Description
MGUD.DEF	Definitions for global machine manufacturer data
UGUD.DEF	Definitions for global user data
GUD4.DEF	User-definable data
GUD8.DEF, GUD9.DEF	User-definable data

Under SINUMERIK Operate, the current values of the GUDs are displayed in the "Parameters" > "User variables" operating area.

Addressing of the GUD variables

The defined name of the variable with a prefix that is dependent on the DEF file is used for addressing the GUD variables.

Furthermore, the prefix depends on whether the variable was defined NC-globally or channel-specifically.

List of address prefixes

/NC/_N_NC_GD1_ACX	NC gud data
/NC/_N_NC_GD2_ACX	NC gud data MGUD.DEF
/NC/_N_NC_GD3_ACX	NC gud data UGUD.DEF
/NC/_N_NC_GD4_ACX	NC gud data GUD4.DEF
/NC/_N_NC_GD5_ACX	NC gud data
/NC/_N_NC_GD6_ACX	NC gud data
/NC/_N_NC_GD7_ACX	NC gud data
/NC/_N_NC_GD8_ACX	NC gud data
/NC/_N_NC_GD9_ACX	NC gud data

/NC/_N_CH_GD1_ACX	channel gud data
/NC/_N_CH_GD2_ACX	channel gud data MGUD.DEF
/NC/_N_CH_GD3_ACX	channel gud data UGUD.DEF
/NC/_N_CH_GD4_ACX	channel gud data GUD4.DEF
/NC/_N_CH_GD5_ACX	channel gud data
/NC/_N_CH_GD6_ACX	channel gud data
/NC/_N_CH_GD7_ACX	channel gud data
/NC/_N_CH_GD8_ACX	channel gud data
/NC/_N_CH_GD9_ACX	channel gud data

Example

Definition and addressing of variables from GUD4.DEF

GUD4.DEF

```

DEF NCK INT NCK_INT_VAR
DEF NCK REAL NCK_REAL_VAR
DEF NCK INT NCK_INT_ARR[3]
DEF NCK STRING[10] NCK_STR_VAR
DEF NCK STRING[3] NCK_STR_ARR[2]
DEF CHAN INT CH_INT_VAR
DEF CHAN REAL CH_REAL_VAR
DEF CHAN INT CH_INT_ARR[2]
DEF CHAN STRING[10] CH_STR_VAR
DEF CHAN STRING[3] CH_STR_ARR[2]
DEF CHAN INT CH_INT_ARR2[5,3]
M30

```

To avoid naming conflicts, use a prefix for new variables, such as MY_VAR1.

Field index and channel number

When accessing arrays, you also use the array index.

Please note the following:

- The first element of an array is addressed in the address with index 1.
- In an NC program, the first element is addressed with index 0.

Address	BFC type
/NC/_N_NC_GD4_ACX/NCK_INT_VAR	int
/NC/_N_NC_GD4_ACX/NCK_REAL_VAR	float
/NC/_N_NC_GD4_ACX/NCK_INT_ARR[1]	int
/NC/_N_NC_GD4_ACX/NCK_STR_VAR	string
/NC/_N_NC_GD4_ACX/NCK_STR_ARR[2]	string

For access to channel-specific variables you use the channel number with uX.

Address	BFC type
/NC/_N_CH_GD4_ACX/CH_INT_VAR[u1]	int
/NC/_N_CH_GD4_ACX/CH_REAL_VAR[u1]	float
/NC/_N_CH_GD4_ACX/CH_INT_ARR[u1,1]	int
/NC/_N_CH_GD4_ACX/CH_STR_VAR[u1]	string
/NC/_N_CH_GD4_ACX/CH_STR_ARR[u1,2]	string

A.7.6 Drive parameters

In the following the access to the drive components of these control series is described:

Control series	Drive
SINUMERIK Powerline	SIMODRIVE
SINUMERIK Solutionline	SINAMICS

A subset of the drive parameters "aaLoad", "aaTorque", "aaCurr" can be read as NC variables in both control series. More information can be found in the annex: NC variables (Page 551).

In the case of the SINUMERIK 840D sl controls based on the SINUMERIK Operate operating software, the drive parameters can be read using the "High-frequency data acquisition" function. More information can be found in Chapter: Configuring high-frequency data acquisition (Page 340).

SINUMERIK 840D SolutionLine with SINUMERIK Operate operating software

The drive parameters are accessed via the address: `/DriveData/...`

Note**Access to `/DriveData/...`**

For access to the address `/DriveData/...`, the BFC client uses internally the MD service of the SINUMERIK Operate operating software.

These addresses are not directly available via the CAP service. The drive parameters can only be read cyclically. The "OnChange" reading mode is not supported.

The MD service is available from SINUMERIK Operate Version 4.5 and higher. For control systems with the SINUMERIK Operate version <4.5, accesses via the `/DriveData/...` address are ignored. A corresponding log message is generated in the logfile of the BFC client.

The following table describes the structure of the addresses for accessing the drive parameters.

Address	Description
<code>/DriveData/DriveControl[u<axis>,<param>]</code>	Access to a single parameter
<code>/DriveData/DriveControl[u<axis>,<param>,<index>]</code>	Access to a single parameter of a parameter array

You can display the drive parameters in the SINUMERIK Operate operating software. To do this, navigate to the operating area "Commissioning" > "Drive parameters".

The structure of the address is explained using the following examples.

- Access to the motor temperature of the first drive axis is made with the following BTT address: `/DriveData/DriveControl[u1,35]`

510101 ↓ 0 <no text available>

Drive parameters		AX1 X1/SERVO_3.3:2
r26	DC link voltage smoothed	319.6 U
r27	Absolute actual current smoothed	0.04 Aeff
r28	Modulation depth smoothed	1.3 %
r29	Current actual value field-generating smoothed	0.00 Aeff
r30	Current actual value torque-generating smooth...	0.04 Aeff
r31	Actual torque smoothed	0.02 Nm
r32	Active power actual value smoothed	0.00 kJ
r33	Torque utilization smoothed	2.9 %
r34	Motor utilization thermal	0 %
r35	Motor temperature	30.8 °C
r36	Power unit overload I2t	0.0 %
r37[0]	Power unit temperatures:Inverter maximum va...	26 °C
r37[1]	Power unit temperatures:Depletion layer maxi...	28 °C
r37[2]	Power unit temperatures:Rectifier maximum v...	-200 °C
r37[3]	Power unit temperatures:Air intake	-200 °C
r37[4]	Power unit temperatures:Interior of power unit	48 °C
r37[5]	Power unit temperatures:Inverter 1	26 °C
r37[6]	Power unit temperatures:Inverter 2	-200 °C

Motor temperature

	General SD	Channel SD	Axis SD	Display MD

- If the drive parameter is an array, the address is extended by the additional index: /DriveData/DriveControl[u1,39,0].

SINUMERIK 840D SolutionLine with SINUMERIK HMI Advanced operating software

Address	Description
/acx/drv_dc_tea<parm>[u<axis>]	Access to a single parameter
/acx/drv_dc_tea<parm>[u<axis>, <index>]	Access to a single parameter of a parameter array

Example for accessing the motor temperature of the first drive axis:

```
/acx/drv dc tea/35[u1]
```

Drive parameter		DP3.SLAVE3.SERVO_3.3:2 [2] AX1:X1		Drive -
r2	Drive operating display	[0] Operation - eve...		Direct selection
p10	Drive commissioning parameter filter	[0] Ready		
p15	Macro drive object	0		
r20	Speed setpoint smoothed	0.0	rpm	
r21	Actual speed smoothed	0.0	rpm	Save / Reset (po)
r22	Speed actual value rpm smoothed	0.0	rpm	
r24	Output frequency smoothed	0.0	Hz	
r25	Output voltage smoothed	0.6	V	
r26	DC link voltage smoothed	323.5	V	Search
r27	Absolute actual current smoothed	0.00	A	
r28	Modulation depth smoothed	0.3	%	
r29	Current actual value field-generating smoothed	-0.00	A	
r30	Current actual value torque-generating smoothed	0.00	A	Continue search
r31	Actual torque smoothed	0.00	Nm	
r32	Active power actual value smoothed	0.00	kW	
r33	Torque utilization smoothed	0.1	%	
r34	Motor utilization	0	%	Display options
r35	Motor temperature	31.4	°C	
r36	Power unit overload I2t	0.0	%	
r37[0]	Power unit temperatures: Inverter, maximum value	27	°C	

SINUMERIK Powerline with the Sinumerik HMI Advanced operating software

For the SINUMERIK Powerline controls, access to the drive parameters is via the following address: DriveHsa/...

The table shows examples of the available drive parameters.

Drive parameters	Description
/DriveHsa/State/actualCurrent[u1]	Smoothed actual value current
/DriveHsa/State/actualSpeed[u1]	Actual speed
/DriveHsa/State/motorTemperature[u1]	Motor temperature
/DriveHsa/State/load[u1]	Load

More information can be found in the List Manual SINUMERIK 840D/840Di/810D, SIMODRIVE 611 digital (<https://support.industry.siemens.com/cs/ww/de/view/109738472>).

A.8 FANUC client data points (reading)

A.8.1 cnc_rdtimer

/focas/cnc/rdtimer/minute[1]

/focas/cnc/rdtimer/msec[1]

Reads the Operating time (minute or msec) [type: Integer]

/focas/cnc/rdtimer/minute[2]

/focas/cnc/rdtimer/msec[2]

Reads the Cutting time (minute or msec) [type: Integer]

/focas/cnc/rdtimer/minute[3]
/focas/cnc/rdtimer/msec[3]
Reads the Cycle time (minute or msec) [type: Integer]

A.8.2 cnc_sysinfo

/focas/cnc/sysinfo/addinfo
Additional information [type: Integer]

/focas/cnc/sysinfo/max_axis
Max. controlled axes [type: Integer]

/focas/cnc/sysinfo/cnc_type
Type of CNC [type: String]

/focas/cnc/sysinfo/mt_type
Type of M/T [type: String]

/focas/cnc/sysinfo/series
Series number [type: String]

/focas/cnc/sysinfo/version
Version number [type: String]

/focas/cnc/sysinfo/axes
Current controlled axes [type: String]

A.8.3 cnc_statinfo

Data points for the 15-type series

/focas/cnc/statinfo/aut
AUTOMATIC mode selection [type: Integer]

/focas/cnc/statinfo/manual
MANUAL mode selection [type: Integer]

/focas/cnc/statinfo/run
Status of automatic operation [type: Integer]

/focas/cnc/statinfo/edit
Status of program editing [type: Integer]

/focas/cnc/statinfo/motion
Status of axis movement,dwell [type: Integer]

/focas/cnc/statinfo/mstb
Status of M,S,T,B function [type: Integer]

/focas/cnc/statinfo/emergency
Status of emergency [type: Integer]

/focas/cnc/statinfo/write
Status of writing backed up memory [type: Integer]

/focas/cnc/statinfo/labelskip
Status of label skip [type: Integer]

/focas/cnc/statinfo/alarm
Status of alarm [type: Integer]

/focas/cnc/statinfo/warning
Status of warning [type: Integer]

/focas/cnc/statinfo/battery
Status of battery [type: Integer]

Data points for new control systems

/focas/cnc/statinfo/hdck
Status of manual handle re-trace [type: Integer]

/focas/cnc/statinfo/tmmode
T/M mode selection [type: Integer]

/focas/cnc/statinfo/aut
AUTOMATIC/MANUAL mode selection [type: Integer]

/focas/cnc/statinfo/run
Status of automatic operation [type: Integer]

/focas/cnc/statinfo/motion
Status of axis movement,dwell [type: Integer]

/focas/cnc/statinfo/mstb
Status of M,S,T,B function [type: Integer]

/focas/cnc/statinfo/emergency
Status of emergency [type: Integer]

/focas/cnc/statinfo/alarm
Status of alarm [type: Integer]

/focas/cnc/statinfo/edit
Status of program editing [type: Integer]

A.8.4 cnc_rddynamic2

/focas/cnc/rddynamic2/axis
axis number (of this call only) [type: Integer]

/focas/cnc/rddynamic2/alarm
alarm status [type: Integer]

/focas/cnc/rddynamic2/prgnum
current program number [type: Integer]

/focas/cnc/rddynamic2/prgmnum
main program number [type: Integer]

/focas/cnc/rddynamic2/seqnum
current sequence number [type: Integer]

```

/focas/cnc/rddynamic2/actf
actual feedrate [type: Float]

/focas/cnc/rddynamic2/acts
actual spindle speed [type: Float]

/focas/cnc/rddynamic2/absolute[n]
absolute pos (n is the number of the axis) [type: Float]

/focas/cnc/rddynamic2/machine[n]
machine pos (n is the number of the axis) [type: Float]

/focas/cnc/rddynamic2/relative[n]
relative pos (n is the number of the axis) [type: Float]

/focas/cnc/rddynamic2/distance[n]
distance to go (n is the number of the axis) [type: Float]

```

A.8.5 cnc_rdpdf_subdirn

```

/focas/cnc/rdpdf_subdirn/files[path]
number of files under the specified folder (path) [type: Integer]

/focas/cnc/rdpdf_subdirn/folders[path]
number of subfolders under the specified folder (path) [type: Integer]

/focas/cnc/rdpdf_subdirn/[path]
number of files and subfolders under the specified folder (path) [type: Integer]

```

A.8.6 cnc_rdpdf_subdir

```

/focas/cnc/rdpdf_subdir/[path,n]
reads subdir information for dir [n] (first subdir "0", then "1" etc) in folder [path] [type: String]

```

A.8.7 cnc_rdpdf_alldir

```

/focas/cnc/rdpdf_alldir/data_kind[path,n]
reads the kind of data for item [n] (first item "0", then "1" etc) in folder [path] (0 if its a folder,
1 if its a file) [type: Integer]

/focas/cnc/rdpdf_alldir/time[path,n]
reads the last edited time of item [n] (first item "0", then "1" etc) in folder [path] (its only
valid for files) [type: String]

/focas/cnc/rdpdf_alldir/size[path,n]
reads size of item [n] (first item "0", then "1" etc) in folder [path] (the size is in bytes and only
valid for files) [type: Integer]

/focas/cnc/rdpdf_alldir/attr[path,n]
reads attributes of item [n] (first item "0", then "1" etc) in folder [path] [type: Integer]

```



```
/focas/cnc/rdpdf_alldir/d_f[path,n]
reads name of item [n] (first item "0", then "1" etc) in folder [path] (foldername or filename)
[type: String]

/focas/cnc/rdpdf_alldir/comment[path,n]
reads comment of item [n] (first item "0", then "1" etc) in folder [path] (its only valid for files)
[type: String]

/focas/cnc/rdpdf_alldir/o_time[path,n]
reads the process timestamp of item [n] (first item "0", then "1" etc) in folder [path] (its only
valid for files) [type: String]
```

A.8.8 cnc_rdparam

"axis = 0" is no axis and "axis = 1...m" is the number of the axis

```
/focas/cnc/rdparam/bit[axis,param]
reads the parameter with number [param] as bit (in this form the complete byte is returned)
[type: Integer]

/focas/cnc/rdparam/bit0[axis,param]
reads the parameter with number [param] as bit (returns the specified bit of the parameter
as bool. Bit 0 to 7 is possible.) [type: Bool]

/focas/cnc/rdparam/byte[axis,param]
reads the parameter with number [param] as byte [type: Integer]

/focas/cnc/rdparam/word[axis,param]
reads the parameter with number [param] as word [type: Integer]

/focas/cnc/rdparam/dword[axis,param]
reads the parameter with number [param] as dword [type: Integer]

/focas/cnc/rdparam/real[axis,param]
reads the parameter with number [param] as real [type: Float]
```

A.8.9 cnc_diagnoss

"axis = 0" is no axis and "axis = 1...m" is the number of the axis

```
/focas/cnc/diagnoss/bit[axis,param]
reads diagnosis data with number [param] as bit (in this form the complete byte is returned)
[type: Integer]

/focas/cnc/diagnoss/bit0[axis,param]
reads diagnosis data with number [param] as bit (returns the specified bit of the parameter
as bool. Bit 0 to 7 is possible.) [type: Bool]

/focas/cnc/diagnoss/byte[axis,param]
reads diagnosis data with number [param] as byte [type: Integer]

/focas/cnc/diagnoss/word[axis,param]
reads diagnosis data with number [param] as word [type: Integer]
```

```
/focas/cnc/diagnoss/dword[axis,param]
reads diagnosis data with number [param] as dword [type: Integer]

/focas/cnc/diagnoss/real[axis,param]
reads diagnosis data with number [param] as real [type: Float]
```

A.8.10 cnc_rdset

"axis = 0" is no axis and "axis = 1...m" is the number of the axis

```
/focas/cnc/rdset/bit[axis,param]
reads setting data with number [param] as bit (in this form the complete byte is returned)
[type: Integer]

/focas/cnc/rdset/bit0[axis,param]
reads setting data with number [param] as bit (returns the specified bit of the parameter as
bool. Bit 0 to 7 is possible.) [type: Bool]

/focas/cnc/rdset/byte[axis,param]
reads setting data with number [param] as byte [type: Integer]

/focas/cnc/rdset/word[axis,param]
reads setting data with number [param] as word [type: Integer]

/focas/cnc/rdset/dword[axis,param]
reads setting data with number [param] as dword [type: Integer]

/focas/cnc/rdset/real[axis,param]
reads setting data with number [param] as real [type: Float]
```

A.8.11 cnc_rdsetnum

```
/focas/cnc/rdsetnum/set_min
minimum number of setting data [type: Integer]

/focas/cnc/rdsetnum/set_max
maximum number of setting data [type: Integer]

/focas/cnc/rdsetnum/total_no
total number of setting data [type: Integer]
```

A.8.12 cnc_rdaxisdata

```
/focas/cnc/rdaxisdata/[data class]/[data type]/[axis]
Reads various data relating to servo/axis/spindle, that is defined by the data class, data type and
axis (not always an axis is required) [type: Integer or Float]
Example: "/focas/cnc/rdaxisdata/1/3/1" (reads distance-to-go for axis 1)

Additional information about the description of data class and data type is provided, for
example, in the Internet under this address (https://www.inventcom.net/fanuc-focas-library/position/cnc\_rdaxisdata).
```

A.8.13 cnc_rdaxisname

/focas/cnc/rdaxisname/name[number]
reads axis name of axis number [number] [type: String]

/focas/cnc/rdaxisname/suff[number]
reads axis subscript of axis number [number] [type: String]

A.8.14 cnc_rdpdlname

/focas/cnc/rdspdlname/name[number]
reads spindle name of spindle number [number] [type: String]

/focas/cnc/rdspdlname/suff1[number]
reads spindle subscript 1 (spindle name 1) of spindle number [number] [type: String]

/focas/cnc/rdspdlname/suff2[number]
reads spindle subscript 2 (spindle name 2) of spindle number [number] [type: String]

/focas/cnc/rdspdlname/suff3[number]
reads spindle subscript 3 of spindle number [number] [type: String]

A.8.15 cnc_alarm2

/focas/pmc/get_number_of_pmc
Reads the number of existing PMC paths [type: Integer]

A.8.16 pmc_get_number_of_pmc

/focas/pmc/get_number_of_pmc
Reads the number of existing PMC paths [type: Integer]

A.8.17 pmc_rdmcrng

/focas/pmc/rdpmcrng/[type]/byte[pmcNr,address]
Reads the PMC data of Type [type] as Byte of PMC Nr. [pmcNr] at address [address] [type: Integer]
Example: "/focas/pmc/rdpmcrng/G/byte[1.30]"

/focas/pmc/rdpmcrng/[type]/bit[pmcNr,address]
Reads the PMC data of Type [type] as Bit of PMC Nr. [pmcNr] at address [address] (in this form the complete byte is returned) [type: Integer]

/focas/pmc/rdpmcrng/[type]/bit0[pmcNr,address]
Reads the PMC data of Type [type] as Bit of PMC No. [pmcNr] at address [address] (returns the specified bit of the parameter as bool. Bit 0 to 7 is possible.) [type: Bool]

/focas/pmc/rdpmcrng/[type]/word[pmcNr,address]
Reads the PMC data of Type [type] as Word of PMC Nr. [pmcNr] at address [address] [type: Integer]

```
/focas/pmc/rdpmcrng/[type]/long[pmcNr,address]
```

Reads the PMC data of Type [type] as Long of PMC Nr. [pmcNr] at address [address] [type: Integer]

```
/focas/pmc/rdpmcrng/[type]/float[pmcNr,address]
```

Reads the PMC data of Type [type] as Float of PMC Nr. [pmcNr] at address [address] [type: Float]

```
/focas/pmc/rdpmcrng/[type]/double[pmcNr,address]
```

Reads the PMC data of Type [type] as Double of PMC Nr. [pmcNr] at address [address] [type: Float]

A.8.18 **cnc_exeprgname**

```
/focas/cnc/exeprgname/name
```

Reads the program name being executed [type: String]

```
/focas/cnc/exeprgname/o_num
```

Reads the program number being executed [type: int]

A.8.19 **tooldata**

toolnumber

```
/focas/tooldata/tool/actual
```

Reads the actual tool number [type: int]

```
/focas/tooldata/tool/next
```

Reads the next tool number [type: int]

```
/focas/tooldata/tool/afternext
```

Reads the tool number following after next [type: int]

radius offset

```
/focas/tooldata/radiusoffset/actual
```

Reads the actual radius offset [type: int]

```
/focas/tooldata/radiusoffset/next
```

Reads the next radius offset [type: int]

```
/focas/tooldata/radiusoffset/afternext
```

Reads the radius offset following after next [type: int]

length offset

```
/focas/tooldata/lengthoffset/actual
```

Reads the actual length offset [type: int]

```
/focas/tooldata/lengthoffset/next
```

Reads the next length offset [type: int]

```
/focas/tooldata/lengthoffset/afternext
```

Reads the length offset following after next [type: int]

radius wear

/focas/tooldata/radiuswear/actual

Reads the actual radius wear [type: int]

/focas/tooldata/radiuswear/next

Reads the next radius wear [type: int]

/focas/tooldata/radiuswear/afternext

Reads the radius wear following after next [type: int]

length wear

/focas/tooldata/lengthwear/actual

Reads the actual length wear [type: int]

/focas/tooldata/lengthwear/next

Reads the next length wear [type: int]

/focas/tooldata/lengthwear/afternext

Reads the length wear following after next [type: int]

A.9 FANUC client data points (writing)

A.9.1 cnc_wrparam

"axis = 0" is no axis and "axis = 1...m" is the number of the axis

/focas/cnc/wrparam/byte[axis,param]

Writes the parameter with number [param] as byte [type: Integer]

/focas/cnc/wrparam/word[axis,param]

Writes the parameter with number [param] as word [type: Integer]

/focas/cnc/wrparam/dword[axis,param]

Writes the parameter with number [param] as dword [type: Integer]

/focas/cnc/wrparam/real[axis,param]

Writes the parameter with number [param] as real [type: Float]

A.9.2 cnc_wrset

"axis = 0" is no axis and "axis = 1...m" is the number of the axis

/focas/cnc/wrset/byte[axis,param]

Writes setting data with number [param] as byte [type: Integer]

/focas/cnc/wrset/word[axis,param]

Writes setting data with number [param] as word [type: Integer]

/focas/cnc/wrset/dword[axis,param]

Writes setting data with number [param] as dword [type: Integer]

`/focas/cnc/wrset/real[axis,param]`
Writes setting data with number [param] as real [type: Float]

A.9.3 **pmc_wrpncrng**

`/focas/pmc/wrpncrng/[type]/byte[pmcNr,address]`
Writes the PMC data of Type [type] as Byte of PMC Nr. [pmcNr] at address [address] [type: Integer]
Example: `"/focas/pmc/wrpncrng/G/byte[1,30]"`

`/focas/pmc/wrpncrng/[type]/bit[pmcNr,address]`
Writes the PMC data of Type [type] as Bit of PMC Nr. [pmcNr] at address [address] (in this form the complete byte is returned) [type: Integer]

`/focas/pmc/wrpncrng/[type]/bit0[pmcNr,address]`
Writes the PMC data of Type [type] as Bit of PMC Nr. [pmcNr] at address [address] (returns the specified bit of the parameter as bool. Bit 0 to 7 is possible.) [type: Bool]

`/focas/pmc/wrpncrng/[type]/word[pmcNr,address]`
Writes the PMC data of Type [type] as Word of PMC Nr. [pmcNr] at address [address] [type: Integer]

`/focas/pmc/wrpncrng/[type]/long[pmcNr,address]`
Writes the PMC data of Type [type] as Long of PMC Nr. [pmcNr] at address [address] [type: Integer]

`/focas/pmc/wrpncrng/[type]/float[pmcNr,address]`
Writes the PMC data of Type [type] as Float of PMC Nr. [pmcNr] at address [address] [type: Float]

`/focas/pmc/wrpncrng/[type]/double[pmcNr,address]`
Writes the PMC data of Type [type] as Double of PMC Nr. [pmcNr] at address [address] [type: Float]

A.9.4 **cnc_wrtimer**

`/focas/cnc/wrtimer/minute[1]`
`/focas/cnc/wrtimer/msec[1]`
Writes the Operating time (minute or msec) [type: Integer]

`/focas/cnc/wrtimer/minute[2]`
`/focas/cnc/wrtimer/msec[2]`
Writes the Cutting time (minute or msec) [type: Integer]

`/focas/cnc/wrtimer/minute[3]`
`/focas/cnc/wrtimer/msec[3]`
Writes the Cycle time (minute or msec) [type: Integer]

A.10 S7 client

A.10.1 Addressing in the SIMATIC PLC

The S7 client enables the reading of inputs, outputs, memories and data blocks.

The addressing parameter follows the addressing in SIMATIC STEP 7 and SINUMERIK PLC diagnostics.

Addressing inputs, outputs, memories

Start of addressing

The addresses always start as follows:

- The addresses of inputs start with "I" or "Q".
- The addresses of outputs start with "A" or "Q".
- The addresses of memories start with "M".

Subsequent addressing for "Bit"

For bit addresses, the byte offset > a dot > the bit to be addressed follow directly.

Note

Bit assignment

- Bit 0 = LSB (Least Significant Bit)
 - Bit 7 = MSB (Most Significant Bit)
-

Examples:

- E0.3: Bit 3 in input byte 0
- Q4.7: Bit 7 in output byte 4
- M1.0: Bit 0 in memory byte 1

Subsequent addressing for "Byte, Word, DWord"

When addressing Byte, Word and DWord, "B", "W" or "D" follow.

Examples:

- EB0: Input byte 0 as unsigned integer
- AW0: Output word 0, consisting of output bytes 0 and 1 as unsigned integer
- MD4: Memory double word 4, consisting of byte 4, 5, 6, 7 as unsigned integer
- MD8:REAL: Memory double word 8 contains a floating point number

Byte order

The byte order is in "Big Endian format".

Multibyte numbers start with the most significant byte value.

Examples:

- MW2 consists of MB2 ($\cdot 2^8$) and MB3.
- MD2 consists of MB2 ($\cdot 2^{24}$), MB3 ($\cdot 2^{16}$), MB4 ($\cdot 2^8$) and MB5.
- ...

To read several bytes with one access operation, byte arrays can also be read:

- ":BYTE[i]" follows the start address.
"i" must be replaced by the number of bytes.

Examples:

- EB3:BYTE[3] : Read 3 bytes from input byte address 3
- AB2:BYTE[5] : Read 5 bytes from output byte address 2
- MB4:BYTE[4] : Read 4 bytes from memory byte address 4

Addressing data in data blocks

All data are addressed with byte offset. For this reason it is not possible to read optimized data blocks of the SIMATIC families S7-1200 and S7-1500.

All addresses begin with "DB". This is followed by the data block number.

Addressing for scalar data:

- DB10.DBX1.5: Read bit 5 in byte 1 of data block 10.
- DB10.DBB5: Read byte 5 of data block 10 as unsigned integer.
- DB10.DBW6: Read word from byte 6 of data block 10 as unsigned integer.
- DB10.DBD12: Read double word from byte 12 of data block 10 as unsigned integer.
- DB10.DBD16:REAL: Read double word from byte 16 of data block 10 as floating point number.

Addressing arrays and character strings

Examples:

- DB10.DBB32:BYTE[8]: Read 8 bytes from byte 32 of data block 10.
The read values are returned as string in the format "[0, 1, 2, 3, 4, 5, 123, 255]".
- DB10.DBB42:CHAR[5]: Read 5 bytes from byte 42 of data block 10 as a string of ASCII characters.
A 0 value is interpreted as the end of the character string.
- DB10.DBB64:STRING: Read S7 string structure from address 64 of data block 10.

A.10.2 Data types, accuracy and formatting for write operations

When writing, restrictions relating to the data type and checks are more stringent than when reading. If an element of a write job violates the subsequent guideline, then data is not written to the PLC.

If the PLC identifies an error when writing to an address, then the other values are still written to the PLC. The write operation is then partially executed.

Permissible data types and value ranges

S7 address	Data type on the data bus	Value range
Bit	Bool	true/false
Byte	Int	0 ... 255, without sign
Word (16 bit)	Int	0-65535, without sign
DWord (32 bit)	Int	0-4294967295, without sign
REAL (32 bit)	Int	with sign, IEEE754 floating-point number Note: Rounding deviations occur for numbers higher than ± 16777215 .
REAL (32 bit)	Float (64 bit)	IEEE754 floating-point number Note: Rounding deviations can occur between 64-bit floating-point numbers on the data bus and 32-bit floating-point numbers in the PLC.
Byte array (*:BYTE[n])	String	Formatting as JSON array of bytes in a decimal notation, e.g.: "[0,1,2,3]"
Character string (*:CHAR[n])	String	The memory requirement of the UTF-8 coded character string must not exceed the addressed memory area.
S7 string (*:STRING)	String	The memory requirement of the UTF-8 coded character string must not exceed the capacity of the addressed S7 string.

Addressing character strings and fields, filling write jobs

If less data is specified in the write job than the destination address permits, then the write job is filled with zero values, e.g.

- Destination address: "DB2.DBB0:BYTE[8]", newly assigned value: "[1,2,3,4]"
⇒ 8 bytes are written to the PLC: 1,2,3,4,0,0,0,0
- Destination address: "DB2.DBB8:CHAR[8]", newly assigned value: "ABCD"
⇒ 8 bytes are written to the PLC: "ABCD\0\0\0\0".

S7 strings (address component :STRING)

When writing to S7 strings, only the changed bytes are overwritten in the PLC. As a consequence, a write job with a short, new value can be successful; however, assigning a longer value is unsuccessful

Character coding in the character string (address component :CHAR[n]) and S7 strings (address component :STRING)

The character strings are interpreted as UTF-8, i.e. for multibyte characters, the number of assigned bytes is counted.

Example:

The euro character ("€", Unicode value: 0x20AC), for UTF-8, is coded in three bytes (0xE2 0x82 0xAC). The three bytes for this character are then assigned in the PLC memory.

A.11 Heidenhain client data points (reading)

A.11.1 Structure of the datapoints

The datapoints of the Heidenhain client are structured according to the following pattern:

/hh/[Funktionsname]/[Parameter]

- Function name
The function name is the name of the API function (Adontec Heidenhain Library).
- Parameter
Depending on the function, the parameter is the variable that is to be returned.
The parameter is not required for every datapoint. For some functions, an input parameter such as the axis number also follows in square brackets.

Note**Data type**

If a data type other than the intended one was specified during configuration of the datapoints, an automatic conversion takes place.

Not all Heidenhain machines support the datapoints listed in the following chapters.

A.11.2 GetRunInfo

Axis configuration

/hh/getruninfo/axes_config/count

The number of axis configuration entries [Data type: integer]

```
/hh/getruninfo/axes_config/axisid[x]  
The ID of axis no. X (beginning with 1) [Data type: integer]  
  
/hh/getruninfo/axes_config/axistype[x]  
The axis type of axis no. X (beginning with 1) [Data type: integer]  
  
/hh/getruninfo/axes_config/axistypecomment[x]  
Description of axis no. X (beginning with 1) [Data type: string]  
  
/hh/getruninfo/axes_config/axisname[x]  
The name of axis no. X (beginning with 1) [Data type: string]
```

Axis position

```
/hh/getruninfo/axes_position/count  
The number of axis configuration entries [Data type: integer]  
  
/hh/getruninfo/axes_position/axisid[x]  
The ID of axis no. X (beginning with 1) [Data type: integer]  
  
/hh/getruninfo/axes_position/axisname[x]  
The name of axis no. X (beginning with 1) [Data type: string]  
  
/hh/getruninfo/axes_position/isinch  
Actual position specification in inches (true/false) [Data type: bool]  
  
/hh/getruninfo/axes_position/position[x]  
Position of axis no. X (beginning with 1) [Data type: float]
```

Cutter location

```
/hh/getruninfo/cutter_location/count  
The number of cutter configuration entries [Data type: integer]  
  
/hh/getruninfo/cutter_location/coordinatename[x]  
The coordinate name of cutter no. X (beginning with 1) [Data type: string]  
  
/hh/getruninfo/cutter_location/isinch  
Actual position specification in inches (true/false) [Data type: bool]  
  
/hh/getruninfo/cutter_location/position[x]  
Position of cutter no. X (beginning with 1) [Data type: float]
```

Execution mode

```
/hh/getruninfo/execution_mode  
The current execution mode [Data type: string]
```

Execution point

```
/hh/getruninfo/execution_point/blocknr  
The current block number [Data type: integer]
```

Overrides information

`/hh/getruninfo/overrides_info/feed`
The current feed override as percentage [Data type: float]

`/hh/getruninfo/overrides_info/speed`
The current speed override as percentage [Data type: float]

`/hh/getruninfo/overrides_info/rapid`
The current rapid override as percentage [Data type: float]

Program status

`/hh/getruninfo/program_status`
The current program status [Data type: string]

DNC mode

`/hh/getruninfo/dnc_mode`
The current DNC mode [Data type: string]

NC uptime

`/hh/getruninfo/nc_uptime`
The NC operating time [Data type: integer]

Machine uptime

`/hh/getruninfo/machine_uptime`
The machine operating time [Data type: integer]

Machine running time

`/hh/getruninfo/machine_runningtime`
The machine running time [Data type: integer]

PLC time

`/hh/getruninfo/plc_timeX`
The PLC time X (X is in the range 1 to 12) [Data type: integer]

Tool

`/hh/getruninfo/tool/toolnr`
Tool number [Data type: integer]

`/hh/getruninfo/tool/toolindex`
Tool index [Data type: integer]

`/hh/getruninfo/tool/toolaxis`
Tool axis [Data type: integer]

```
/hh/getruninfo/tool/toollen
```

Tool length [Data type: float]

```
/hh/getruninfo/tool/toolrad
```

Tool radius [Data type: float]

A.11.3 GetMachineParameters

```
/hh/getmachineparameters/[Parameter]
```

Read machine parameters [Data type: depends on parameter]

Examples:

```
/hh/getmachineparameters/MP3515
```

```
/hh/getmachineparameters/3210
```

```
/hh/getmachineparameters/MP3210.0
```

A.11.4 DataGetValue

```
/hh/datagetvalue/[Identifier]
```

Read variables based on the symbolic name [Data type: depends on identifier]

Examples:

Read memory value from PLC memory:

```
/hh/datagetvalue/PLC/memory/M/4178
```

Read word value from PLC memory:

```
/hh/datagetvalue/PLC/memory/W/322
```

Read tool data:

```
/hh/datagetvalue/TABLE/TOOL/T/1/DOC
```

A.11.5 ReadMemory

Memory bytes

```
/hh/readmemory/byte/address
```

Read memory byte of address [Data type: bool]

Example:

Read memory byte at address 2080:

```
/hh/readmemory/byte/2080
```

Word values

```
/hh/readmemory/word/Adresse
```

Read word of address [Data type: integer]

A.11 Heidenhain client data points (reading)

Example:

Read word value at address 494:

```
/hh/readmemory/word/494
```

DWord values

```
/hh/readmemory/dword/Adresse
```

Read DWord of address [Data type: integer]

Example:

Read DWord value at address 322:

```
/hh/readmemory/dword/322
```

A.11.6 Examples of important Heidenhain datapoints

Program status

```
/hh/getruninfo/program_status
```

Possible values:

- STARTED(0)
- STOPPED(1)
- FINISHED(2)
- CANCELED(3)
- INTERRUPTED(4)
- ERROR(5)
- ERROR_CLEARED(6)
- IDLE(7)

Execution mode

```
/hh/getruninfo/execution_mode
```

Possible values:

- MANUAL(0)
- MDI(1)
- RPF(2)
- SINGLESTEP(3)
- AUTOMATIC(4)
- OTHER(5)

Axis in motion

/hh/datagetvalue/PLC/memory/W/1028

Possible values:

- 0: No axis in motion
- 1: X-axis in motion
- 2: Y-axis in motion
- ...

Emergency

/hh/datagetvalue/PLC/memory/M/4178

Possible values:

- false: Not active
- true: active

Overrides

Feed override:

/hh/getruninfo/overrides_info/feed

Speed override:

/hh/getruninfo/overrides_info/speed

Rapid override:

/hh/getruninfo/overrides_info/rapid

A.12 Heidenhain client data points (writing)**A.12.1 SetMachineParameters**

/hh/setmachineparameters/Parameter[write mode]

Write machine parameters.

Supported write modes:

- 0 = Persistent (default value)
- 1 = Temporary
- 2 = Persistent at the next reset

The default value is used if a write mode is not specified.

Valid examples:

/hh/setmachineparameters/MP3515

/hh/setmachineparameters/MP3515[1]

```
/hh/setmachineparameters/3210  
/hh/setmachineparameters/MP3210.0
```

A.12.2 DataSetValue

```
/hh/datasetvalue/Bezeichner[Format]
```

Write variables based on the symbolic names. [Data type: depends on the identifier and format]

Supported formats (always one letter):

- Logic (bit memory): M
- Byte: B
- Unsigned byte: U
- Short (word): W
- Long: L
- Float: F
- Double: D
- String: S
- Special Value: V

For addresses based on notation "PLC/memory/..", the format is automatically read from the address. For all other addresses, the format must be specified in square brackets.

Valid examples:

Write bit memory value from the PLC memory:

```
/hh/datasetvalue/PLC/memory/M/4178
```

Write word value from the PLC memory:

```
/hh/datasetvalue/PLC/memory/W/322
```

Read only one byte from the word value in the PLC memory:

```
/hh/datasetvalue/PLC/memory/W/322[B]
```

A.12.3 WriteMemory

Bit memory bytes:

```
/hh/writememory/byte/address
```

Write bit memory byte to address [data type: bool]

Word values:

```
/hh/writememory/word/address
```

Write word to address [data type: integer]

DWord values:

```
/hh/writememory/dword/address
```

Write DWord to address [data type: integer]

Valid examples:

Write bit memory byte to address 1:

/hh/writememory/byte/1

Write DWord value to address 322:

/hh/writememory/dword/322

A.13 Data points Beckhoff client

The datapoints of the Beckhoff client are structured according to the following pattern:

- {port}:{index}:{offset}
or
- {port}:{symbol}

At the beginning of the address you will always find the port range used. This can be defined either by a number or a name.

Port name	Port number
AMSPORT_LOGGER	100
AMSPORT_R0_RUNTIME	200
AMSPORT_R0_TRACE	290
AMSPORT_R0_IO	300
AMSPORT_R0_SPS	400
AMSPORT_R0_NC	500
AMSPORT_R0_ISG	550
AMSPORT_R0_PCS	600
AMSPORT_R0_PLC	801
AMSPORT_R0_PLC_RTS1	801
AMSPORT_R0_PLC_RTS2	811
AMSPORT_R0_PLC_RTS3	821
AMSPORT_R0_PLC_RTS4	831
AMSPORT_R0_PLC_TC3	851

The port specification is followed by a colon. This is followed by the addressing. This can be done either by the index and offset (separated by a colon) or using a symbolic name.

Note

- If a different data type than intended was specified when configuring the data points, an automatic conversion is attempted.
- Only NC parameters are standardized and can be viewed via the Beckhoff documentation (see <https://infosys.beckhoff.com/english.php?content=../content/1033/tcadsdevicenc2/81064794547748107.html&id=>).
- All other parameters are control system-specific.

A.14 EIP client

A.14.1 Addressing of data, general

The addresses consist of several individual parameters that are appended to each other with "&".

Address parameters

Address parameters	Description
name	Address identifier (see Addressing in the PCCC message format (Page 582))
elem_type	Data type
elem_count	Number of values to be read

Examples:

- name=N7:0
- name=MyTagName&elem_type=string
- elem_type=string&name=MyTagName

Examples 2 and 3 address the same data.

For simplicity, the parameter identifier "name=" can be omitted if its value is defined first. This is then added internally in the EIP client and can therefore appear in the log messages.

The following examples are equivalent to examples 1 and 2 above:

- N7:0
- MyTagName&elem_type=string

If the "elem_count" parameter is omitted, "elem_count=1" is implicitly inserted. This can appear in log messages.

A.14.2 Addressing in the PCCC message format

The addressing format presented here applies to the Rockwell and Allen Bradley control system families Micrologix, PLC/5 and SLC 500.

Basically, the address consists of the memory area, the file, and the index within the memory area or the file.

Examples

"I1:0" - the 16-bit number at index "0" of input memory "1":

I	1	:	0
Prefix/file type	File number	Separator	Index

"B3:11+" - the 16-bit number signed at index "11" in binary file "3":

B	3	:	11	+
Prefix/file type	File number	Separator	Index	Signed

"N7:12/15" - bit 15 of the 16-bit value at index "12" in number memory "7":

N	7	:	12	/	15
Prefix/file type	File number	Separator	Index	Bit selector	Bit index

File prefixes

The file prefixes determine the data type of the addressed data.

Therefore, no "elem_type" parameter can be specified for these addresses, since it is ignored.

Prefix	Designation	Data type	Access	Bit access ⁴⁾	Array access	Remark
A	Ascii	1 byte	Read/Write	-	Yes	1)
B	BIN	16-bit int	Read/Write	Read	Yes	-
C	Counter	Structure	Read	-	-	2)
D	BCD Number	16-bit int	Read/Write	Read	Yes	1) 2) 3)
F	Float	32-bit float	Read/Write	-	Yes	-
I	Input	16-bit int	Read	Read	Yes	-
L	Long Int	32-bit int	Read/Write	-	Yes	-
N	Number	16-bit int	Read/Write	Read	Yes	-
O	Output	16-bit int	Read/Write	Read	Yes	-
R	Register	Structure	Read	-	-	2)
S	Status	16-bit int	Read/Write	Read	Yes	-
ST	String	84 bytes	Read/Write	-	-	1)
T	Timer	Structure	Read	-	-	2)

1) This function is implemented but not verified by a system test.

2) The structures can be read. The meaning of the values is not documented.

3) The nibbles of the BCD number patterns are not converted to decimal numbers by the EIP client, i.e. the bit pattern of the nibble must be evaluated separately.

4) The bit offset is not evaluated at the control level, but in the EIP client. Only the complete numbers from the control system are always read and written.

To change a bit, the complete current number value must be read, changed, and written back. This is not done automatically to avoid edge effects with the running PLC user program.

File numbers

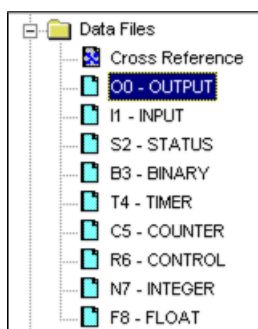
The existing files and file numbers depend on the control system.

Observe the documentation of the control system.

The following is an example of file numbers for a Micrologix 1100:

Data memory	Content
O0	Output
I1	Input
S2	Status
B3	Binary
T4	Timer
C5	Counter
R6	Control register
N7	Number
F8	Float

The file numbers of the example are displayed in the development environment as follows:



A.14.3 Interpretation of 16 and 32-bit numbers, signed or unsigned

The distinction whether a value is coded as an unsigned integer or signed integer is configured via an appended + character at the address.

Examples

N7:0 - 16 bit unsigned, value range: 0 to 65535

N7:0+ - 16 bit signed, value range: -32768 to 32767

L12:0 - 32 bit unsigned, value range: 0 to $2^{32}-1$

L12:0+ - 32 bit signed, value range: -2^{31} to $2^{31}-1$

A.14.4 Addressing in the Allen-Bradley CIP message format with tags

The addressing presented here applies to the Rockwell/Allen-Bradley control system families ControlLogix, CompactLogix, Micro850/870, the control system families Omron NJ and Omron NX, and compatible control systems.

The data are addressed as they are defined in the development environment of the PLC user program.

The tags can be defined at the system level as arrays with up to three dimensions.

Example:

MyTagForNumbers[2,2] - addresses the element at index 2,2 in a two-dimensional memory area named "MyTagForNumbers"

Tags on the program level can also be defined with up to three dimensions.

Example:

MyProgramName:MyVariable - addresses a single variable with the name "MyVariable" on the program level of the "MyProgramName" program.

Specification of the tag data type

The EIP client cannot determine the data type of the addressed data itself. Therefore, this must be specified via the "elem_type" parameter when addressing in BFC.

Permitted values for the "elem_type" parameter:

Value	Meaning	Derived from
lint	64 bit signed	Long Int
ulint	64 bit unsigned	Unsigned Long Int
dint	32 bit signed	Double Word Int
udint	32 bit unsigned	Unsigned Double Word Int
int	16 bit signed	Int
uint	16 bit unsigned	Unsigned Int
sint	8 bit signed	Short Int
usint	8 bit unsigned	Unsigned Short Int
bool	Boolean, 1 bit	-
real	32-bit floating-point number IEEE 754	-
lreal	64-bit floating-point number IEEE 754	Long Real
string	Character string	-

Examples of the configuration of addresses in BFC in relation to the above tags in the control system:

- MyTagForNumbers[2,2]&elem_type=udint
- Program:<prog>.<tagname>&elem_type=<type>

Note

Addressing of user-defined data structures is not supported.

A.14.5 Addressing arrays

Basically, only numbers can be read out as an array. Writing of arrays is not supported.

The described formats are used to address individual elements. To read an array of these elements starting from the start address, the number of elements must be appended to the address as a parameter.

The element count is transferred via the "elem_count" addressing parameter.

Examples:

- N7:10+&elem_count=10
- My3DArray[0,2,3]&elem_type=int&elem_count=10

For both examples, a one-dimensional array with 10 elements (here: 16-bit signed number) is returned from the defined start address.

A.14.6 Reading and writing very large 64-bit numbers

To avoid deviations due to rounding effects during serialization or deserialization of numbers in messages in JSON format, 64-bit numbers should be transmitted as strings.

In this case, conversion takes place directly in the EIP client. When reading, the numerical values are always supplied in decimal notation. When writing, the numbers can be encoded in the following formats:

- Decimal without sign (e.g.: 123)
- Decimal with sign (e.g.: -999)
- Hexadecimal with prefix "0x" (e.g.: 0xFFFFFFFFFFFFFFF)
- Octal with prefix "0o" or "0" (e.g.: 0o123456701234567)
- Binary with prefix "0b" (e.g.: 0b1111111110101010 = 0xFFAA)

A.15 Data points of the Omron client

The data points of the Omron client are structured according to the following pattern:

- {address}:{index}:{length}
or
- { address}:{ index,bit-number}

At the beginning of the address, you will always find the address range used. This must be defined using a name.

Address range	Description
ciobit	CIO area (bit)
wrbit	Work area (bit)
hrbit	Holding area (bit)
arbit	Auxiliary area (bit)

Address range	Description
cioword	CIO area (word)
wrword	Work area (word)
hrword	Holding area (word)
arword	Auxiliary area (word)
timercountercompletionflag	Counter completion flag
timercounterpv	Counter preset values
dmbit	Data area (bit)
dmword	Data area (word)
taskbit	Task flags (bit)
taskstatus	Task flags (status)
indexregisterpv	Index register preset values (bit)
dataregisterpv	Data register preset values (bit)
clockpulsesconditionflagsbit	Condition flags (bit)

The address range is followed by a colon. Then the addressing can be performed either via an index and a bit number (separated by a comma) or via an index and length (separated by a colon).

Note

- The data type defined in configuration of the data points influences how the data are read from the control.
- All parameters depend on the configuration of the control.

A.16 MQTT export gateway

A.16.1 Topics

The sent data is sent to the following topics:

- Data sets
`<baseTopic>/<clientID>/<application type>/dataset/<set name>`
 Example: bfc/my-client/iotclient/dataset/dataset1
- Data set collection
`<baseTopic>/<clientID>/<application type>/databatch/<set name>`
 Example: bfc/my-client/iotclient/databatch/dataset1
- Alarms
`baseTopic>/<clientID>/<application type>/alarmlist`
 Example: bfc/my-client/iotclient/alarmlist

- Alarm state changes
`<baseTopic>/<clientID>/<application type>/alarmstatechange`
 Example: `bfc/my-client/iotclient/alarmstatechange`
- HF data sets
`<baseTopic>/<clientID>/<application type>/hfdata/<set name>`
 Example: `bfc/my-client/iotclient/hfdata/hfdataset1`

The "baseTopic" can be defined via the gateway configuration.

For "clientID", "application type" and "set name", the configuration of the client is used.

A.16.2 Data format: data set

The data to be exported is sent to the configured MQTT broker in the following format:

Parameter	Description
client ID	The ID assigned during the client configuration
ownerID	Unique ID of the equipment under which the client was created
application type	Type of source application
dataset name	Name of the data record configured for the client.

```
{
  "source": {
    "applicationType": "<application type>",
    "clientID": "<client ID>",
    "ownerID": "<owner ID>"
  },
  "set": {
    "name": "<dataset name>",
    "timestamp": "<RFC3339 timestamp>",
    "datapoints": [
      {
        "name": "<variable1 name>",
        "type": "bool|string|int|float",
        "value": <value of variable1>
      },
      {
        "name": "<variable2 name>",
        "type": "bool|string|int|float",
        "value": <value of variable2>
      },
      .
      .
      .
      {
        "name": "<variableN name>",
        "type": "bool|string|int|float",
        "value": <value of variableN>
      }
    ]
  }
}
```



```
]
}
}
```

Example

```
{
  "source": {
    "clientID": "opc-ua-1",
    "applicationType": "opcuaclient",
    "ownerID": "4787325f-c636-412f-97a9-2698271f705c"
  },
  "set": {
    "timestamp": "2020-05-29T08:08:41.9308624Z",
    "name": "ds2",
    "datapoints": [
      {
        "name": "dp5",
        "value": 15452,
        "type": "int"
      },
      {
        "name": "dp4",
        "value": 5929000.39629511,
        "type": "float"
      },
      {
        "name": "dp3",
        "value": "one",
        "type": "string"
      },
      {
        "name": "ds2",
        "value": 59290004,
        "type": "int"
      },
      {
        "name": "dp1",
        "value": false,
        "type": "bool"
      }
    ]
  }
}
```

A.16.3 Data format: alarms

Alarms are sent to the configured MQTT broker in the following format:

Parameter	Description
client ID	The ID assigned during the client configuration
ownerID	Unique ID of the equipment under which the client was created
application type	Type of source application
additional information (data)	Additional information stored for the alarm on the client

```
{
  "source": {
    "applicationType": "<application type>",
    "clientID": "<client ID>",
    "ownerID": "<owner ID>"
  },
  "list": {
    "timestamp": "<RFC3339 timestamp>",
    "alarms": [
      {
        "id": "<alarm ID 1 / alarm number>",
        "message": "<alarm message>",
        "timestamp": "<RFC3339 alarm timestamp>",
        "data": { <JSON object with additional information (optional)> }
      },
      {
        "id": "<alarm ID 2 / alarm number>",
        "message": "<alarm message>",
        "timestamp": "<RFC3339 alarm timestamp>",
        "data": { <JSON object with additional information (optional)> }
      },
      .
      .
      .
      {
        "id": "<alarm ID N / alarm number>",
        "message": "<alarm message>",
        "timestamp": "<RFC3339 alarm timestamp>",
        "data": { <JSON object with additional information (optional)> }
      },
    ]
  }
}
```

Example

```
{
  "source":{
    "clientID":"mt-connect-test",
    "applicationType":"mtconnectclient"
  },
```

```

"list":{
  "timestamp":"2020-05-29T09:41:31.299565Z",
  "alarms":[
    {
      "number":"Cloadc",
      "message":"This is a test alarm",
      "state":"active",
      "timestamp":"2020-05-29T09:41:31.299565Z",
      "data":{
        "Name":"","
        "Source":"Cloadc"
      }
    }
  ]
}

```

A.16.4 Data format: alarm state changes

Alarm state changes are sent to the configured MQTT broker in the following format:

Parameter	Description
client ID	The ID assigned during the client configuration
ownerID	Unique ID of the equipment under which the client was created
application type	Type of source application
data (additional information)	Additional information stored for the alarm on the client

```

{
  "source": {
    "applicationType": "<application type>",
    "clientID": "<client ID>",
    "ownerID": "<owner ID>"
  },
  "alarm": {
    "id": "<alarm ID 1 / alarm number>",
    "message": "<alarm message>",
    "timestamp": "<RFC3339 alarm timestamp>",
    "state": "active",
    "data": { <JSON object with additional information (optional)> }
  }
}

```

Example

```

{
  "source":{
    "clientID":"mt-connect-test",
    "applicationType":"mtconnectclient"
  },

```

```

"alarm":{
  "id":"Cloadc",
  "message":"This is a test alarm",
  "timestamp":"2020-05-29T09:41:06.299565Z",
  "state":"active",
  "data":{
    "Name":"","
    "Source":"Cloadc"
  }
}
}
}

```

A.16.5 Data format: high frequency data

High frequency data is sent to the MQTT Broker in the following format:

Parameter	Description
client ID	The ID assigned during the client configuration
ownerID	Unique ID of the equipment under which the client was created
application type	Type of source application
hfDataInfo	Contains the following information: - Additional information on the recording of data Examples: context variables, recorded signals and associated signal types - Start time of recording

```

{
  "data":[
    {
      "timeOffset":<offset1>,
      "values":[
        <signal1 value at offset1>,
        <signal2 value at offset1>,
        .
        .
        .
        <signalM value at offset1>,
      ]
    },
    {
      "timeOffset":<offset2>,
      "values":[
        <signal1 value at offset2>,
        <signal2 value at offset2>,
        .
        .
        .
        <signalM value at offset2>,
      ]
    },
  ],
}

```

```

.
.
.
{
  "timeOffset":<offsetN>,
  "values":[
    <signal1 value at offsetN>,
    <signal2 value at offset1>,
    .
    .
    .
    <signalM value at offsetN>,
  ]
},
],
"hfDataInfo":{
  "blockNo":2,
  "config":"<hfdata config file name>",
  "context":{
    "<context var1>": <context var1 value>,
    "<context var2>": <context var2 value>,
    .
    .
    .
    "<context varK>": <context varK value>,
  },
  "id":"jfsrvuk271",
  "moreData":true,
  "signalNames":[
    "<signal1>",
    "<signal2>",
    .
    .
    .
    "<signalM>"
  ],
  "signalTypes":[
    "<type of signal1>",
    "<type of signal2>",
    .
    .
    .
    "<type of signalM>"
  ],
  "startTime":"<RFC3339 timestamp of start of recording>"
},
"source": {
  "applicationType": "<application type>",
  "clientID": "<client ID>",
  "ownerID": "<owner ID>"
},

```

```
"time":""<RFC3339 message timestamp>"
}
```

Example

```
{
  "data":[
    {
      "timeOffset":2.176,
      "values":[
        7.1,
        10.4
      ]
    },
    {
      "timeOffset":2.184,
      "values":[
        7.9,
        10.9
      ]
    },
    {
      "timeOffset":2.192,
      "values":[
        14.8,
        6.6
      ]
    },
    {
      "timeOffset":2.2,
      "values":[
        16.2,
        4.3
      ]
    },
    {
      "timeOffset":2.208,
      "values":[
        20.0,
        4.1
      ]
    }
  ],
  "hfDataInfo":{
    "blockNo":2,
    "config":"PMON1",
    "context":{
      "ActDNumber":0,
      "ActTNumber":0,
      "ActToolDuplo":0,
      "ActToolIdent":"0",
      "ActToolRTime":0,
      "Feedoverride":100,

```

```
"NCProgram": "_N_HFTEST_MPF",
"NCProgramLineNumber": 0,
"Spindleoverride": 100
},
"id": "jfsrvuk271",
"moreData": true,
"signalNames": [
"Channel_SEMA_vaIm1_u1_1",
"Channel_SEMA_vaIm1_u1_2"
],
"signalTypes": [
"float",
"float"
],
"startTime": "2020-08-07T14:29:52.518803+02:00"
},
"source": {
"applicationType": "iotclient",
"clientID": "Ellen",
"ownerID": ""
},
"time": "2020-08-07T14:29:56.916994+02:00"
}
```

A.17 OPC UA

A.17.1 Supported OPC UA profiles

OPC UA profile	Client/Server	Link to the OPC UA specification
Core Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/CoreFacet
Core 2017 Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/Core2017Facet
Base Server Behaviour Facet	Server	http://opcfoundation.org/UA-Profile/Server/Behaviour
Documentation Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/Documentation
Standard DataChange Subscription Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/StandardDataChangeSubscription
Standard DataChange Subscription 2017 Server	Server	http://opcfoundation.org/UA-Profile/Server/StandardDataChangeSubscription2017
Data Access Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/DataAccess
Standard Event Subscription Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/StandardEventSubscription
Address Space Notifier Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/AddressSpaceNotifier
A & C Base Condition Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACBaseCondition
A & C Refresh2 Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACRefresh2

OPC UA profile	Client/Server	Link to the OPC UA specification
A & C Address Space Instance Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACAddressSpaceInstance
A & C Alarm Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACAlarm
A & C Acknowledgeable Alarm Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACAckAlarm
Method Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/Methods
Node Management Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/NodeManagement
Historical Raw Data Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalRawData
Standard UA Server Profile	Server	http://opcfoundation.org/UA-Profile/Server/StandardUA
Standard 2017 UA Server Profile	Server	http://opcfoundation.org/UA-Profile/Server/StandardUA2017
Core Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/Core
Core 2017 Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/Core2017
Base Client Behaviour Facet	Client	http://opcfoundation.org/UA-Profile/Client/Behaviour
Entry Level Support 2015 Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/EntryLevelSupport2015
Documentation – Client	Client	http://opcfoundation.org/UA-Profile/Client/Documentation
Attribute Read Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/AttributeRead
DataChange Subscriber Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/DataChangeSubscriber
DataAccess Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/DataAccess
Event Subscriber Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/EventSubscriber
Base Event Processing Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/BaseEventProcessing
A & C Alarm Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/ACAlarm
Standard UA Client Profile	Client	http://opcfoundation.org/UA-Profile/Client/Standard
Standard UA Client 2017 Profile	Client	http://opcfoundation.org/UA-Profile/Client/Standard2017

A.18 MindSphere MMM Dashboard

You can create a gateway asset in MindSphere.

Procedure

1. Open the Asset Manager and create a "MindConnectLib"-based asset.

Assets / Details / Create asset

Add asset

General

Type ID:
nbgmsiot.BFCGateway
Selected type of asset cannot be changed

Name: *
BFC_machinename

Description:
Testrack
247 characters left

+ Location

+ Variables

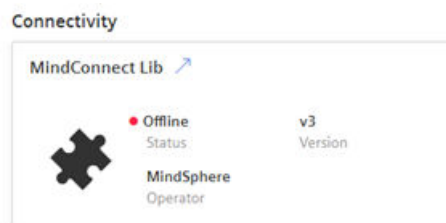
Static aspects

Name
> connectivityStatus

* Required input field

Save Cancel

2. Open the connectivity tile.



3. Create a SHARED_SECRET key-based security profile.

MindConnect Lib # Plug KDEF_BFC_Client_1 BFCGateway

[Close](#)

[Back to configuration](#)

Boarding configuration

Onboard your MindConnect Lib

In order to use MindConnect Lib, you have to insert the generated onboarding key into your system.

```
{
  "url": "https://gate.evt.mindsphere.io/",
  "url_auth": "https://auth.mindsphere.io/auth/realms/mindconnect",
  "client_id": "mindconnect",
  "client_secret": "SHARED_SECRET",
  "scope": "openid profile email",
  "grant_type": "password",
  "username": "admin@mindconnect.com",
  "password": "MindConnect123!",
  "expiration": "2021-08-05T10:08:47.000Z"
}
```

[Generate onboarding key](#)
[Copy to clipboard](#)

4. Create your own MindSphere export gateway for each client that is to be connected to MindSphere.

The screenshot shows the Apicurio Studio interface. The sidebar on the left contains three icons: a square with a circle (Clients), a cylinder (Middlewares), and a hexagon (Gateways). The main area is titled 'Clients (Import)' and contains a table with one entry:

Name	ID	Version
NCU BFC Client	02.12.00.07	

Below the table, there are three icons: an upward arrow, a pencil, and a vertical ellipsis. The bottom right corner features a dark blue button with a white plus sign and the text 'Add gateway'.

5. For "Select Gateway type", select "MindSphere".



Add new gateway

1 Select gateway type

Select target device

MindSphere

Next

6. Use the same gateway names as in MindSphere.

7. Enter the key generated in MindSphere.

[illegible]

8. Select the machine and the dataset to be uploaded.
 You can select between individual datasets and ****All datasets****.
 Datasets "CH1_BasicConfig" and "CH1_MachineStatus" are recommended for the "Manage MyMachines" apps.

CH1_BasicConfig	
☐ Feedoverride	DOUBLE
☐ NCProgramStatus	DOUBLE
☐ NrOfAlarms	DOUBLE
☐ Opmode	DOUBLE
☐ ProtectionLevel	DOUBLE
☐ Spindleoverride	DOUBLE
☐ StopCond	DOUBLE
CH1_MachineStatus	
☐ MachineStatus	DOUBLE

Note

The following variables are supported for FANUC controls:

- NCProgramStatus
- OpMode
- NCProgram
- Feedoverride
- Spindleoverride
- NrOfAlarms
- onlineStatus
- CurrentAlarms

For native client machines, dataset "CH1_MachineStatus" is formed in MindSphere. This can be mapped in the BFC gateway using script logic.

CH1_MachineStatus	Time of the machine status in the selected time period. The machine status is displayed numerically:
0	Unknown status
1	Production
2	Technical Disturbance
3	Organizational Disturbance
4	No Disturbance

1 Define dataset configuration
Optional

Pick a client ID: Pick a dataset:

9. Select the appropriate client if alarms are to be uploaded

4 Define alarm configuration
Optional

Pick a client ID:

10. If data is to be written from MindSphere into the machine, for the particular client, enter the corresponding variables and the type.

5 Define write configuration
Optional

Pick a client ID: Type address you want to write:

11.No adaptations are required in the "Advanced configuration".

12.Once the client has been saved, then it is automatically connected with MindSphere. The dataset is automatically created when a connection is successfully established.

Data point	Data point ID	Linked	Data type
Spindeloverride	03efb79c-5e79-4c40-82e1-5c...	✓	DOUBLE
poweronTime	38805cab-84f6-47c7-9795-0df...	✓	DOUBLE
operatingTime	e6e18cf2-0012-4138-951a-18...	✓	DOUBLE
cuttingTime	141d5022-2156-438d-93f0-8...	✓	DOUBLE
NCProgram	768b17aa-9f94-4b20-8af8-75...	✓	STRING
NCProgramStatus	45d2a7c6-ef3a-478f-a095-40...	✓	DOUBLE
Opmode	92061361-6f62-438a-b522-3...	✓	DOUBLE
StopCond	60864867-2f6f-4af5-b8b3-ff9...	✓	DOUBLE
Feedoverride	67020ecf-b716-4778-864d-f9...	✓	DOUBLE
NCAlarms	484856ba-3c6a-474f-894b-05...	✓	DOUBLE
Protectionlevel	acc03503-5d22-49c1-8866-f6...	✓	DOUBLE

13.Dependent on the variable names and data types used, you must link the data points with the variables in MindSphere.

A.19 MS SQL gateway

Fixing certificate problems with encrypted connection

Unknown certification authority:

If the MS SQL server certificate is issued by a certificate authority that is not known to the BFC gateway, you will receive the following error message:

"TLS Handshake failed: x509: **certificate signed by unknown authority**"

1. Import the CA certificate in Base64-encoded form into the list of trusted certificates in the "/etc/ssl/certs" directory.

Example of a Debian/Ubuntu command sequence:

```
cd /usr/local/share
sudo mkdir ca-certificates/extra
sudo cp /home/me/root.cert.pem ca-certificates/extra/root.cert.crt
sudo update-ca-certificates
```

2. The commands and directories may differ depending on the Linux distribution used. Refer to the documentation of the Linux distribution you are using.

Note the file extensions "pem" of the source file and "crt" of the target file.

Missing information in the server certificate:

If the IP address, server name used, or fully qualified server name is not included in the server certificate, you will receive the following error message:

"TLS Handshake failed: x509: **certificate is valid for** w2k22sql.example.com,
not w2k22sql"

- OR -

"TLS Handshake failed: x509: **certificate is valid for** w2k22sql.example.com,
not 172.16.12.56"

1. The parameter "hostNameInCertificate=w2k22sql.example.net" fixes the problem.

Preventing a failed connection to an MS SQL server

A connection to an MS SQL Server with dynamic port assignment or to an MS SQL Server with named instance fails if the Windows service "SQL Server Browser" is inactive or the port "1434/udp" is blocked by the Windows firewall.

1. Enable the Windows service "SQL Server Browser".

- OR -

Check that the port "1434/udp" is not blocked in the Windows firewall.

2. With dynamic port assignment of the SQL Server instance: Unlock the executable file "sqlservr.exe" in the Windows firewall in the inbound rules.

Example: %ProgramFiles%\Microsoft SQL
Server\MSSQL15.MSSQL2019\MSSQL\Binn\sqlservr.exe

A.20 HTTP script client

Publishing the online status

The `publishConnectionState()` function reports the client as online or offline in BFC.

Example:

```
publishConnectionState( true ); // online
publishConnectionState( false ); // offline
```

Publishing data

Publishing reading sets

The `publishReadingSet()` function can be used to publish the data. The sender information (the source attribute) is automatically added.

Return values: Success: `true`, error: `false`

Example:

```
var rs = {
  "Set": {
    "Time": new Date().toISOString(),
    "Name": "readingSetName",
    "Data": [{
      "name": "item1",
      "type": "int",
      "value": 42
    }, {
      "name": "item2",
      "type": "string",
      "value": "Hello World!"
    }
  ]
};
publishReadingSet(rs);
```

Publishing alarm lists

The `publishAlarmList()` function can be used to publish the active alarms. The sender information (the source attribute) is automatically added.

Return values: Success: `true`, error: `false`

Example:

```
var al = {
  "List": {
    "Alarms": [{
      "Message": "Emergency stop",
      "Number": "3000",
      "Timestamp": new Date().toISOString(), // start
      "State": "active"
    }
  ],
  "Timestamp": new Date().toISOString() // of list
};
publishAlarmList(al);
```

JavaScript for the HP 3D API

The predefined JavaScript is used to read data with the HP 3D printer API. The script is divided into several sections.

General auxiliary functions

- `addHeader`: adds a header element to a header object
- `addBasicAuthHeader`: adds a basic authentication header element to a header object
- `addBearerAuthHeader`: adds a bearer authentication header element to a header object
- `addTokenAuthHeader`: adds a token authentication header element to a header object
- `getXMLRootElement`: returns the root element of an XML structure by skipping declarations sections and empty strings
- `xml2object`: converts an XML document into a JavaScript object structure to simplify data access afterwards
- `newReadingSet`: creates a new, empty reading set structure
- `newItem`: creates an item structure for a reading set
- `newAlarmList`: creates a new, empty alarm list structure
- `newAlarmEntry`: creates a new alarm structure for the alarm list
- `updateChangedFlag`: compares the data of a reading set with its previous data and then adds the changed attribute

Class `hp3dAPI`

The `hp3dAPI` class encapsulates all communication to the HP SmartStream 3D Command Center and its device proxy. The class includes detection of available services, authentication, listing of devices, detection of available services of the device, loading of country-specific texts, and available services of the device. The status of communication to the device and HP Command Center is published in the BFC Gateway with connection state messages (online/offline).

The `hp3dAPI` class implements only communication and is used by the remaining code to retrieve data from the device.

Cyclically running handler classes for data retrieval, preparation and data publication

The conversion of HP 3D API data structures to BFC data structures is implemented in three handler classes. An instance of each handler class is created in the `startReadingSetWorkers` function and its "Handler()" method is executed cyclically.

- **printerAlarms class**
This handler collects information about active alarms/alerts of the device and publishes them as alarm list in the BFC Gateway. The handler is started at run-up when a reading set "Alarms" is configured. The read interval of the "Alarms" reading set determines the polling interval for the device. Any name/address/type configuration can be set as an item in the reading set. The alarm messages are delivered in the configured language (default: "en-US")
- **MachineInfo class**
This handler collects information about the device and publishes it as a reading set. The handler is started at run-up when a reading set "MachineInfo" is configured. The read interval of the "MachineInfo" reading set determines the polling interval for the device. Any name/address/type configuration can be set as an item in the reading set.
- **JobInfo class**
This handler collects information about the active print job and publishes it as a reading set. The handler is started at run-up when a reading set "JobInfo" is configured. The read interval of the "JobInfo" reading set determines the polling interval for the device. Any name/address/type configuration can be set as an item in the reading set.

Initialization routine

At the end of the JavaScript, the previously documented classes are created as an instance with the parameterization from the configuration interface.

The global object "config" contains the configuration parameters from the user interface.

The `init()` function is run cyclically until initialization is complete and the handlers are started. The `startReadingSetWorkers()` function starts the handlers based on the configuration as the last step of the initialization. Each handler object is stored in the global object "workers".

A.21 Further notes

Procedure for BFC client installation

A BFC client must be installed on a machine together with a contact person for the BFC Gateway. After completing the installation on the controller and the required reboot

- the log file must be checked
- the log file must be backed up

In the log file, check whether the client is entered in the "lobby" of the BFC Gateway. This can be recognized through the entry "MQTT broker connected, Host:=[] Port:=1883 Username:=lobby:guest ClientId:=", followed by a dynamically created Client ID.

The client installation is not complete until the client has been transferred from the lobby to the system configuration at the BFC Gateway.

Furthermore, together with the contact person for the BFC Gateway, it must be ensured that the machine is created in the BFC Gateway.

Make sure that the data connection between machine and gateway is working and check if data of the machine is visible. Only then is the commissioning of a machine considered complete. More information can be found in Chapter Installing a BFC client (Page 321).

Note

Installation without checking the correct connection to the BFC Gateway is not permitted and may result in additional costs.

Licenses for higher-level or external systems

Licenses for higher-level or external systems must be purchased additionally. Note this especially in the event that higher-level or external systems may also need to be licensed per asset / per device.

File transfer

Make sure that there is enough space on the target system before starting a file transfer.

Support for arrays

Currently, the datasets used do not support arrays. This approach is well suited for smaller sets of data points. If the quantity increases considerably, e.g. when reading a whole data block from a PLC, reading blocks or arrays of data points is supported.

Starting with version v1.8 of the BFC Gateway, the internal OPC UA server and the OPC UA clients support arrays within datasets. Since backward compatibility must be ensured, some special rules must be observed. Contact the hotline for support.

File transfer – ACT.DIR

Files in the `ACT.DIR` directory of a SINUMERIK control are generated dynamically by the NC software. A `ListDir` command does not display the contents of this folder.

However, direct access to files in `ACT.DIR` is possible.

Example: `.../NC/ACT.DIR/INITIAL.INI` or `.../NC/ACT.DIR/TO.DIR/TO_INI.INI`

Contact the hotline regarding the handling of direct access to files.

BFC Gateway Kubernetes installations

BFC Gateway Kubernetes installations cannot access the CA certificates of the host system. These CA certificates are required, among other things, when gateways are to establish encrypted connections to a higher-level system, e.g. to a MindSphere instance.

To use CA certificates in a BFC Gateway Kubernetes installation, a secret named `bfc-ca-bundle` must be created in the Kubernetes environment. The certificates can then be stored in this secret. The secret and the certificates are created with `kubectl`.

Example: `kubectl -n <Namespace der BFC Installation> create secret generic bfc-ca-bundle --from-file=/etc/ssl/certs/mycert1.crt --from-file=/homer/bfc/mindsphere.pem`

In this example, the secret is created with the name `bfc-ca-bundle`. The certificate is added to the bundle via the flag `from-file`. A separate `from-file` instruction must be created for each certificate that is to be added. In the example, two certificates are added to the bundle.

If a new service is now created or restarted via the ConfigUI, the stored certificates are automatically mounted in the container and can be used by it.

Certificates of a host can be generated under Linux with `openssl` and `sed`:

```
openssl s_client -connect www.mindsphere.io:443 2>/dev/null </dev/null | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > cert.pem
```

In this example, the certificate is downloaded from `www.mindsphere.io` and saved as `cert.pem`.

UTC time stamp

BFC Gateway modules use a time stamp in UTC format by default. This applies regardless of which time zone is set on the installation system. If a specific time zone is to be used, the environment variable `"TZ=[time zone]"` (e.g. `"TZ=Europe/Berlin"`) must be set for the respective module.

Limits for the BFC Gateway running under Kubernetes

For the BFC Gateway running under Kubernetes, limits are preconfigured for the CPU and memory usage for the individual pods. The preset value should be adapted depending on the number of clients and the requirements. The Kubernetes administrator must adapt the settings for `"cpu"` and `"memory"` under `"limits"` in the configuration of the individual pods. Contact the hotline for additional information.

Operating instruction SR=1-6588854929 (sampling rates)

This operating instruction applies when using the BFC client to acquire data on a SINUMERIK solutionline control under SINUMERIK Operate (PCU) or SIMUMERIK Operate embedded (TCU).

Make sure that only sampling rates of 200 ms or higher (slower) are used.

Check the configuration for each BFC client. If a value is not equal to 200 ms, make the following changes:

Requirement

The BFC Gateway is upgraded to hotfix release v1.9.1.

1. In the BFC configuration interface, switch to the "Define dataset configuration / ① New Dataset" configuration step of the corresponding BFC client. More information can be found in Chapter Creating a BFC client (Page 97).
2. Change the set value in the "Set debounce time in milliseconds" field to another value (e.g.: currently set value + 1).
3. Save the change.

Saving a changed sampling rate ensures that the Hotfix Release v1.9.1 changes take effect.

The following sampling rates are supported: 200 ms, 1,000 ms, 10,000 ms.

Values smaller than 200 ms cannot be entered in the user interface. If a different value is configured, the next lower value that is supported is sampled on the control. However, the feedback to the BFC Gateway takes place in the desired, configured cycle.

Examples:

- Configured value = 500 ms corresponds to a sampling rate of 200 ms + 300 ms waiting time until transmission to the BFC Gateway
- Configured value = 2,500 ms corresponds to a sampling rate of 1,000 ms + 1,500 ms waiting time until transmission to the BFC Gateway
- Configured value = 2,500 ms corresponds to a sampling rate of 1,000 ms + 1,500 ms waiting time until transmission to the BFC Gateway
- Configured value = 15,000 ms corresponds to a sampling rate of 10,000 ms + 5,000 ms waiting time until transmission to the BFC Gateway

A.22 Troubleshooting

Problem	Description	Remedy
The logs displayed in ConfigUI may not be up-to-date.	Service logs in docker come from all containers associated with the service (including cached logs).	By restarting (stopping and starting) the corresponding client, the logs can be updated.
Very slow system start of the BFC gateway	The system start of the BFC gateway takes a very long time.	On the BFC host system, make sure that the DNS servers entered in the system file <code>/etc/resolv.config</code> are reachable and have a reasonable "response time". You can check this on the BFC host system via the console as follows: <pre>host COMPUTERNAME</pre> Instead of <code>COMPUTERNAME</code>, you must enter a name of a system within your local network (for example, the computer name of a controller). - Result if the IP address of the computer <code>COMPUTERNAME</code> can be determined. The response of the system must arrive immediately: <pre>COMPUTERNAME has address 127.0.0.1 // (the IP is only an example)</pre> - Result if the IP address of the computer <code>testsystem99</code> cannot be determined. <pre>Host COMPUTERNAME not found: 2 (SERVFAIL)</pre> - If the system does not respond immediately, the customer's IT department must resolve this issue before the BFC gateway can be used.
InfluxDB gateway error message "InfluxDB retention policy could not be found"	If you specify an InfluxDB database as a target for the InfluxDB gateway, the error message "InfluxDB retention policy could not be found" appears in the gateway log messages.	The InfluxDB gateway always requires a database with a DEFAULT retention policy. This retention policy can be created either by setting "retention-autocreate = true" or by entering it manually.
Elasticsearch gateway error message "Error 429 (Too many requests)"	The error message "Error 429 (Too many requests)" appears in the log messages of the Elasticsearch gateway.	The "Prefetch Count" option in the "Advanced configuration" in the gateway settings in the ConfigUI must be set to a smaller value (default: 1000)
The following error appears in the log of the Heidenhain client: doHHLogIn() to area <PLCDEBUG> failed with Error <-2>. (TNC Error: <49> <The specific area does not exist or is protected.>)	For certain Heidenhain functions, access to the PLC area of the control is necessary. This access is exclusive. This means that only one client at a time can access this area. If another client already claims this area, the Heidenhain BFC client cannot execute certain functions and this results in the mentioned error message.	In the configuration of the client (under "Define advanced configuration"), the parameter "Do logout after every PLC-Area operation..." can be set to "true". Thus, at the expense of performance, the PLC area is released again with each corresponding function call. Thus, in the time between calls, another client can continue to read values. Remark: This does not completely avoid the error message, but it should occur less frequently.

Problem	Description	Remedy
Message 909020 "Directory could not be reliably determined" in connection with HMI PRO sl	When using the BFC client or other additionally installed add-on applications on SINUMERIK Operate controls together with the HMI PRO sl software, a message with the number 909020 is output by HMI PRO sl when starting the control. This is an HMI PRO sl group message that prompts the user to call the HMI PRO sl error display to view details. In connection with the BFC client, the following message occurs here: "Directory could not be reliably determined".	According to the HMI Pro hotline, the message 909020 only refers to the additionally installed application, which has no effect in relation to HMI PRO sl. This message can be acknowledged by the user.
If a BFC gateway service, e.g. the OPC UA server, is to be deployed, the message "exec user process caused: argument list too long" is shown.	If changes are made to the environment variables in the docker-compose.yml of a service, it may happen that the service cannot be started afterwards and is aborted with the message "exec user process caused: argument list too long". This can happen especially with the OPC UA server when editing the enable list for write operations. The reason for this is that the sum of all environment variables in the system may only contain a certain amount of characters. This limit depends on the operating system used.	The content of the environment variables must be shortened. With the OPC UA server, for example, variables can be removed from the enable list. If possible, the ConfigUI should be used to configure the services.
After a login attempt with a user without administrator rights, a user with administrator rights can no longer log in.	After unsuccessfully trying to log in with as a user without administrator rights, a user with administrator rights may no longer be able to log in. After such a login attempt, users are directed to a page where the tiles for BFC operation are missing.	The browser must be closed and restarted. Then the ConfigUI of the BFC can be controlled again and a user with administrator rights can log in.
After some operating time, the BFC gateway no longer starts reliably.	Under certain installation conditions, the message "No space left on device" may appear.	If a distinction is made between the home partition and the data partition when installing the operating system, make sure that the BFC gateway and all components are installed on a partition whose size meets to the system requirements.
The BFC gateway cannot be installed under Kubernetes K3S without access to the BFC container registry.	During Kubernetes installation, the images are downloaded from the BFC container registry. However, access data is required for this purpose.	If the installation of the gateway is performed locally on a K3S system, then no access to the container registry is required. Instead, the -k3s flag must be set during setup. Example: <pre>./setup.linux.x64 -u admin -p admin -kubernetes -k3s -kubeconfig '/etc/rancher/k3s/k3s.yaml' -release '1.9' -deploy</pre> The setup thus imports the supplied containers automatically.

Problem	Description	Remedy
The BFC gateway cannot be installed under Kubernetes without access data for the BFC container registry.	During Kubernetes installation, the images are downloaded from the BFC container registry. However, access data is required for this purpose.	The provided images can be imported into a customer-internal registry to perform the installation without access to the BFC container registry. Contact Siemens Support for this type of installation.
The ConfigUI cannot be opened on Kubernetes Red Hat OpenShift.	The web interface of the BFC gateway cannot be opened because the internal routing does not work.	Under Red Hat OpenShift, a configuration entry must be changed before performing the setup. Open the file under the path <code>/kubernetes/builds/1.9/ingress/entry-deployment.yaml</code> . Replace there line 60 (value: <code>kube-dns.kube-system.svc.cluster.local</code>) by the following line: <code>value: dns-default.openshift-dns.svc.cluster.local</code> Then save the file.
Some pods do not start after installation under Kubernetes Red Hat OpenShift.	Due to lack of permissions, pods are not allowed to perform write operations in their own file system.	To allow write operations for the corresponding pods, the administrator of the cluster must assign advanced rights to the BFC's ServiceAccount (<code>bfc-app-manager-service-account</code>). For this purpose, the following command must be executed in the OC CLI: <pre>oc adm policy add-scc-to-user nonroot -z bfc-app-manager-service-account</pre>
No more data arrives at the target system from the clients connected to the BFC gateway.	For example, the following message appears in the log file of a gateway. <code>NOT_FOUND - failed to perform operation on queue xxxxx' in vhost '/' due to timeout</code>	Delete the configured gateway in BFC and create it with a much higher time to live (TTL). The value should be at least 2,800,000,000 ms (this corresponds to a TTL of 30 days). The TTL can be configured in the "Advanced Settings" of a gateway.

A.23 Release Notes V1.11

System State

User interface "System State" allows versions of clients, middlewares and gateways to be updated.

ConfigUI

There is a new "Variable browser" function. This allows the variable list from the client to be displayed online (OPC UA and MTConnect) or offline.

BFC Gateway API

In addition to clients, middlewares and gateways can now also be managed via the API.

Heidenhain client

The Heidenhain client supports reading, writing, creating and deleting programs and directories for compatible devices using WebDAV.

Trigger Based Reading

- With the new read mode "OnTrigger", clients support the configuration of triggers within the ConfigUI.
- A template with a complex trigger is supplied for the Scriptlogic middleware.

FANUC client

- Adding API calls of the FOCAS interface
- Adding data points "cnc_exeprgname" and "tooldata"

BFC Gateway update

- Expansion of the update procedure (Docker and Kubernetes-based)

Improved product security

- The BFC client carries out more checks.
- The ConfigUI was improved through more confirmation prompts.

Glossary

Asset

For the BFC Gateway, an asset is each connected element that provides data. This can be a machine or an individual component.

Asset Manager

The Asset Manager is a MindSphere application. The assets of a machine are created and configured in the Asset Manager. The application is also used for the management of customers, users and shop floors.

Client

A client is a module in the BFC Gateway and reads data from a machine or a server that provides data. The data to be read is organized in data sets and data points.

A data set comprises one or several data points. A data point corresponds to a variable in a machine control, for example.

A client can contain several data sets.

Data points or variables

In the context of the BFC, datapoints or variables are all the values that can be acquired from the NC, PLC and from the drives, e.g. sampling rate, temperature, jerk. Users must define and configure these in the client-configuration as data points. Data points combined create one or several data sets. The captured values are then transferred as time series via gateways, e.g. for MindSphere. There are also preconfigured data sets, such as the SINUMERIK basic configuration and the machine availability. Details are documented in the corresponding chapters.

MindSphere - Industrial IT Siemens ecosystem

MindSphere is the cloud-based, open IoT operating system from Siemens that connects your products, plants, systems and machines, enabling you to harness the wealth of data generated by the Internet of Things (IoT) with advanced analytics.

MQTT

Message Queuing Telemetry Transport (MQTT) is an open messaging protocol for machine-to-machine communication (M2M); it facilitates the transfer of telemetric data in the form of messages between devices, in spite of long delay times or restricted networks. The appropriate devices range from sensors and actuators, mobile telephones, embedded systems in vehicles or laptops, up to fully developed computers.

SSA

MindSphere application "SINUMERIK Service Assistance"