

ctrlX OS

Operating System for ctrlX CORE Control Devices 02VRS

Copyright

© Bosch Rexroth AG 2025

All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

Disclaimer

The specified data only serves to describe the product. As our products are constantly being further developed, no statements concerning a certain condition or suitability for a certain application can be derived from our information. The information given does not release the user from the obligation of own judgment and verification. Please note that our products are subject to a natural process of wear and aging.

DOK-XCORE*-XCR***V02**-AP04-EN-P

DC-AE/PAX, PAG-SW, PAD-SYS (TD, MN, MS/MePe)

Table of contents

1	About this documentation	15
1.1	ctrlX OS – Licensing notes	19
2	Important directions on use	19
2.1	Intended use	19
2.1.1	Introduction	19
2.1.2	Areas of use and application	20
2.2	Unintended use	20
3	Safety instructions	20
4	System overview	21
4.1	ctrlX CORE – Control system	21
4.2	ctrlX CORE Virtual – Virtual control	22
4.3	ctrlX I/O – Module kit	22
4.4	ctrlX OS – Operating system	23
4.5	ctrlX OS— system concept	24
4.6	ctrlX OS – Web-based user interface	24
4.7	ctrlX OS – Data Layer	25
4.8	ctrlX WORKS	26
4.9	ctrlX I/O Engineering	27
4.10	ctrlX PLC Engineering	28
4.11	ctrlX DRIVE Engineering	29
4.12	ctrlX SAFETY Engineering	30
5	Managing ctrlX apps	31
5.1	Apps – Basics	31
5.2	Installing apps on the ctrlX device from the ctrlX Store	32
5.3	Installing apps from the engineering PC on the ctrlX device	33
5.4	Uploading apps to the app memory of the ctrlX device	33
5.5	Installing apps on the ctrlX device from the device-internal app storage	34
5.6	Updating apps	35
5.7	Changing app version	37
5.8	Disabling apps	37
5.9	Enabling apps	38
5.10	Downloading apps from the ctrlX device	38
5.11	Reinstalling apps	39
5.12	Uninstalling apps	39
5.13	Deleting apps from app storage	40
5.14	ctrlX device – Enabling the Service mode	40
5.15	ctrlX device – Enabling the Operating mode	40
5.16	Release cycle	41
5.16.1	Release cycle of apps and engineering software	41
5.17	Creating own apps	41
5.17.1	Introduction and overview	41

6	Working with the device	42
6.1	Users and rights	42
6.1.1	Introduction	42
6.1.2	User management – Functional principle	42
6.1.3	Users, groups, permissions, sessions	43
6.1.4	State upon delivery/default user	43
6.1.5	Logging users into the control	44
6.1.6	Login banner	44
6.1.7	Changing the user password	44
6.1.8	Password forgotten	44
6.1.9	Creating a new user	45
6.1.10	Creating a system user	45
6.1.11	Configuring the user management	46
6.1.12	Conventions for login and session monitoring	46
6.1.13	Username and password conventions	46
6.2	Network connection	47
6.2.1	General information on network interfaces	47
	Important notes for use	47
6.2.2	Configuring IP settings	48
6.2.3	Automatic/manual IP configuration	48
6.2.4	Defining routes	48
6.2.5	Bridging interfaces	49
6.2.6	Option for dynamic determination of the maximum package size (PMTUD)	49
6.2.7	Limiting network bandwidth	50
6.2.8	Link local	50
6.2.9	Connecting to WiFi network	50
6.2.10	Setting up WLAN hotspot	51
6.2.11	Adding and configuring VLAN interface	51
6.2.12	Configuring security settings for wired interfaces	52
6.2.13	EoE configuration (Ethernet over EtherCAT)	52
6.3	ctrlX licenses	53
6.3.1	Licenses – Overview	53
6.4	Managing certificates and keys	63
6.4.1	Introduction	63
	Certificates	64
	Keys	64
	“Generate certificate signing request”	65
6.5	Working with configurations	66
6.5.1	Configurations – Introduction	66
6.5.2	Working with configurations using the control interface	67
	Transferring active app data to the engineering PC	67
	Loading app data from the engineering PC	67
	Resetting active app data	68
	Viewing active app data on file level	69
	Saving active app data to the archive of the ctrlX device	69

	Uploading from the engineering PC to the archive of the ctrlX device	70
	Loading configuration from the archive to the ctrlX device	71
	Comparing configurations	71
	Transferring configuration from archive to engineering PC	72
	Viewing configuration in the archive	72
	Copying configuration to the archive	73
	Deleting configuration in the archive	74
	Renaming configuration in archive	74
	Displaying information on a configuration in the archive	74
	Editing description of a configuration	75
	Managing configurations	75
	Backing up active app data	77
	Downloading a configuration	77
	Loading app data	77
	ViewingLoading report	78
6.5.3	Working with configurations	78
6.6	Back up and restore	79
6.6.1	Backup – Introduction	79
6.6.2	Backup – Creating from ctrlX device	79
	Backup – Options when creating	80
6.6.3	Backup – Applying to ctrlX device	81
6.6.4	Backup – Showing report	82
6.6.5	Backup – Showing history	83
6.6.6	Backup – Restrictions	83
	Plain text password	83
	Improved encryption with SHA-256	83
	Process architecture	83
	App Hardware Support	84
	Licenses	84
	System user	84
	Configuration files	84
6.6.7	Backup – content and structure	84
	Backup – Structure of the backup file "ctrlx-setup.json"	84
	Backup – file content	86
	Backup – Certificates and keys	87
	Backup – Active configuration and configuration archive	88
	Backup – Connectivity	90
	Backup – Date and time	99
	Backup – Hosts	99
	Backup – User management(IdentityManagement)	99

	Backup – License information	101
	Backup – Package management (apps)	102
	Backup – Deleting apps	104
	Backup – Proxy	104
	Backup – SSH	104
	Backup – System	104
	Backup – System information	105
	Backup – Type plate information	105
6.6.8	Backup – Manual modification	105
6.6.9	Backup – Diagnostics	105
6.6.10	Backup – Command line	105
	Applying backup to a device	105
	Creating a backup from a device	106
	Checking a backup file	109
6.6.11	Backup – Automatic application upon system startup	109
	Prerequisites	110
	One-time backup application	111
	Autorestore authorizations	112
	Diagnostics	113
	LED display	113
	Device Portal backup	113
6.6.12	Backup – Extensions to the backup file format	113
	Version release XCR-V-0114	113
	Version release XCR-V-0116	114
	Version release XCR-V-0118	114
	Version release XCR-V-0120	114
	Version release XCR-V-0202	115
	Version release XCR-V-0204	115
	Version release XCR-V-0206	115
6.6.13	Backup – Dealing with changes	115
	Extensions to the backup file format	115
	New Setup app functionality in "old" system environment	116
	Updating the Setup app	116
6.6.14	Backup – HowTo's	116
	Backing up device settings before XCR-V-0116	116
6.6.15	Backup – FAQ	117
6.7	Hardware reset in case of ctrlX CORE and ctrlX DRIVEplus with integrated ctrlX CORE	118
6.7.1	Hardware reset in case of ctrlX CORE X3, ctrlX COREplus X3 and ctrlX DRIVEplus	118
6.7.2	Hardware reset in case of ctrlX COREplus X7	120
6.8	Setting up ctrlX OS	122
6.8.1	Device identification	122
	ctrlX Device ID	122
	Schema	122

	Schema validation	122
	ctrlX Device certificate	122
	Public Key Infrastructure (PKI)	123
6.8.2	Requesting a device certificate for your device	124
	Generating a device key	124
	Generating certificate signing request	126
	Requesting the device certificate	127
	Uploading the device certificate	128
	Automation	128
6.8.3	Licensing a ctrlX OS device	128
	Secure hardware anchor for licensing	128
	Registering a new device for licensing	128
	Manual licensing	128
	Trial licenses for development	129
6.8.4	Enabling Secure Boot	129
	Prerequisites	129
	Default ctrlX OS vs. customized ctrlX OS	129
	Secure Boot for default ctrlX OS	129
	Steps to own Secure Boot for customized ctrlX OS	129
	Managing your Secure Boot keys	130
	Signing EFI binaries for Secure Boot with a customized ctrlX OS	130
	Deploying your Secure Boot certificates to device NVRAM	131
	Target state for your device	131
	Revoking keys and hashes on your device	131
	UEFI password protection	131
6.8.5	Image conversion	132
	Precondition	132
	Conversion	132
7	Diagnostics	132
7.1	Logbook – Overview	132
7.2	Logbook – Exporting	133
7.3	Remote Logging – Configuring external server	136
7.3.1	Operating principle	136
7.3.2	System requirements	136
7.3.3	Server reference sources	136
7.3.4	Configuring the external server in the ctrlX device	136
7.3.5	Evaluation of server log messages	137
7.3.6	Backing up and restoring the configuration	137
7.3.7	Encrypted transmission	137
7.4	LED states	137
7.5	Creating system report in ctrlX OS	138
7.6	Control monitoring	139

8	Device interfaces	141
8.1	Interface description	141
8.2	Storage media	142
8.2.1	SD cards and USB support	142
8.2.2	Mounting external storage medium	143
8.2.3	Formatting external storage medium	144
8.2.4	Removing external storage medium safely	145
9	Real-time processing	145
9.1	System Handler	145
9.1.1	Introduction and overview	145
9.1.2	State machine of the System Handler	146
	Operating states	147
	Transitions	148
	Special case Safe Mode	148
9.1.3	State of the ctrlX OS Runtime components	148
	Transitions	148
9.1.4	Multicore support of the PLC app	149
9.1.5	LED diagnostic code	149
9.1.6	Ensuring a safe machine state after a critical error	150
9.1.7	Detecting a reboot loop of the Automation Core app	150
9.1.8	Safe mode	151
9.2	ctrlX CORE Scheduler	151
9.2.1	Introduction and overview	151
9.2.2	Tasks und callables	153
	Task response	153
	Task properties	154
	Task “Priority”	156
	Callables	157
	Program configuration mode	166
	Time measurements	168
	Task status	170
9.2.3	Administration	171
	Status	171
	Trigger source	173
	Configuring the startup	173
	Backing up and restoring the configuration	174
	Resetting the task configuration	174
	Selecting an initial CPU core	175
	Mode “Debug”	175
	Strict mode	175
	Scheduler load	176
9.3	Exception handling	176
9.3.1	Introduction and overview	176
9.3.2	Storage location of the log file	177
9.3.3	Structure of the log file	177
9.3.4	Diagnostics	178

9.4	Hardware watchdog	178
9.5	Script parser/interpreter	179
9.5.1	Introduction and overview	179
9.5.2	User interface via Data Layer	180
9.5.3	Initialization script	181
9.5.4	Search paths for Python modules	181
9.5.5	Python functions for Data Layer access	182
	Function (nodes) = datalayer.browse	
	(<path>)	182
	Function <value> = datalayer.read (<path>)	182
	Function datalayer.write(<path>, <value>) .	182
	Function datalayer.create(<path>, <value>)	183
	Function datalayer.remove(<path>)	183
	Function datalayer.read_json(<path>)	183
	Function datalayer.write_json(<path>, <value>)	183
	Function datalayer.create_json(<path>, <value>)	183
	Further information	184
10	ctrlX Data Layer	184
10.1	ctrlX OS Data Layer - Central data interface of ctrlX OS	184
10.1.1	Data storage and data access	184
10.1.2	Scope of data	184
10.1.3	Further information	184
10.2	Data exchange via Data Layer	184
10.3	Addresses	185
10.4	Data types	186
10.4.1	Introduction and overview	186
10.4.2	Simple data types	190
10.4.3	Complex data types	191
	Data Layer - FlatC	191
10.4.4	Data type "Variant"	192
10.5	Node structure	194
10.5.1	Introduction and overview	194
10.5.2	Metadata	194
10.5.3	Metadata database (MDDB)	197
	Introduction and overview	197
	CSV structure	198
	MDDB compiler	200
10.6	Data transport NRT	200
10.7	Accessing data (NRT)	201
10.7.1	Introduction and overview	201
10.7.2	Synchronous methods	201
10.7.3	Asynchronous methods	202

10.8	Providing data (NRT)	204
10.8.1	Introduction and overview	204
10.8.2	Simple wildcard "*"	204
10.8.3	Double wildcard "***"	205
10.8.4	Responsibility of nodes	205
10.8.5	Example of node responsibility	205
10.8.6	Special features for browse requests	206
10.8.7	Node methods	206
10.8.8	Response of a provider	207
10.8.9	Returning errors	207
10.8.10	Synchronous response	208
10.8.11	Asynchronous response	208
10.9	Exchanging data RT	208
10.9.1	Introduction and overview	208
10.9.2	Types of memory ranges	209
10.9.3	Structure of a memory range	210
10.9.4	Revision of a memory range	212
10.9.5	Owner interface	212
10.9.6	User interface	213
10.9.7	State machine of the user	213
10.9.8	NRT access to RT data	214
10.9.9	Special features of the memory type SharedRetain	215
10.9.10	RT MemoryType trigger	216
10.10	NRT data exchange between controls	218
10.10.1	Introduction and overview	218
10.10.2	Configuration	219
10.10.3	Access to data of another control	220
10.11	Data Layer - Global settings	220
10.12	ctrlX Data Layer Rest interface	221
10.12.1	Introduction and overview	221
10.12.2	Protocol mapping	221
10.12.3	Access rights	222
10.12.4	Content mapping	223
10.12.5	Troubleshooting	223
10.13	Subscriptions	224
10.13.1	Introduction and overview	224
10.13.2	Subscription - Rule set	225
10.13.3	Subscription - Notifications	229
10.13.4	Subscription - Global settings	229
10.13.5	Active subscriptions	230
10.13.6	Subscription - Language binding	230
10.13.7	Subscription - Client examples	230
10.13.8	Subscription - Web interface	232
10.13.9	Subscription - Lossless subscription to real-time data	233
10.14	Compliance Checker	234
10.14.1	Introduction and overview	234

10.15	Security	236
10.15.1	Introduction and overview	236
10.15.2	Scope check	238
10.15.3	Method <i>check-operations</i>	239
10.16	Description of the nodes	240
11	ctrlX diagnostic system	240
11.1	Introduction into the diagnostic system	240
11.1.1	Messages, warnings and errors	240
11.1.2	Main diagnostics and detailed diagnostics	240
11.1.3	Information on the use of diagnostics	241
11.1.4	Logbook	241
11.1.5	Further information	242
11.2	Using the diagnostic system	242
11.3	Error categories, log types and (error) deletion	242
11.4	Use cases	244
11.4.1	General information	244
11.4.2	Runtime diagnostics	245
11.4.3	Event diagnostics	246
11.5	Diagnostic handling summary	246
11.6	Examples	248
11.6.1	Examples of possible diagnostic combinations	248
11.6.2	Examples of unpermitted/impossible diagnostic combinations	248
11.7	Structure and use of customer, OEM and partner diagnostics	249
11.8	Structure of the diagnostic number	251
11.8.1	General information	251
11.8.2	Main diagnostics	252
11.9	Connection to the Data Layer RT (real time)	255
11.10	User interface via Data Layer	256
11.11	Registering and unregistering main and detailed diagnostics	256
11.11.1	General information	256
11.11.2	Structure of the JSON file	256
11.11.3	Options to register and unregister main and detailed diagnostics	258
11.12	Log structure (message format) <i>journald</i> and its metadata	259
11.12.1	Message format	259
11.12.2	Metadata	261
12	How to	262
12.1	Cleaning up ctrlX OS and updating to a new software release (backup)	262
12.1.1	Scenario 1: Clean up device and update software	263
12.1.2	Scenario 2 (Clone device)	264
12.1.3	Scenario 3: (apply external data carrier to device)	266
13	User interface	268
13.1	Navigation	268
13.1.1	ctrlX OS web interface – Overview	268

13.2	Windows	270
13.2.1	Window – “Home”	270
13.2.2	Window – Logbook	271
13.2.3	Window – Remote Logging	272
13.2.4	Window – “Manage app data”	273
	Windows – Manage app data > File view	277
	Windows – Manage app data > Configuration comparison	278
13.2.5	Window – “Apps”	279
13.2.6	Window – “Licenses”	284
	Tab – “Licenses”	284
	Tab – “Sources”	288
13.2.7	Window – “Data Layer”	288
13.2.8	Window – “Date & Time”	295
13.2.9	Window – “System Performance”	296
13.2.10	Window – “System Information”	297
13.2.11	Window – “Scheduler”	297
13.2.12	Window – “Backup & Restore”	298
13.2.13	Window – “Storage”	299
13.2.14	Window – “Firmware Update”	299
13.2.15	Window – “Reset Device Data”	300
13.2.16	Window – “Certificates & Keys”	301
13.2.17	Window – “PKIs”	302
13.2.18	Window – “Users & Permissions”	304
13.2.19	Window – “Users”	305
13.2.20	Window – “Groups”	306
13.2.21	Window – “Permissions”	306
13.2.22	Window – “Sessions”	307
13.2.23	Window – “Hosts”	308
13.2.24	Window – “Network Diagnostics”	309
13.2.25	Window – “Network discovery”	310
13.2.26	Window – “Network interfaces”	312
13.2.27	Window – “Proxy”	313
13.2.28	Window – “SSH”	313
13.2.29	Window – “Open Source Components”	314
13.2.30	Written offer for source code	315
13.2.31	Window – “About ctrlX OS”	316
13.3	Tabs	316
13.3.1	Tab – “Threads”	316
13.3.2	Tab – “Alarms”	317
13.3.3	Tab – “History”	320
13.3.4	Tab – “Resources”	325
13.3.5	Tab – “Performance”	326

13.3.6	Tab – “Processes”	326
13.3.7	Tab – “Overview”	327
13.3.8	Tab – “Configuration”	328
13.3.9	Tab – “Startup”	331
13.3.10	Tab – “Debug”	332
13.3.11	Tab – “Certificates”	332
13.3.12	Tab – “Keys”	334
13.3.13	Tab – “Revocation lists”	337
13.3.14	Tab – “PKCS12 containers”	338
13.3.15	Tab – “Certificate enrollments”	339
13.3.16	Tab – “Restore”	340
13.3.17	Tab – “Back up”	341
13.4	Dialogs	341
13.4.1	Dialog – “Add VLAN interface”	341
13.4.2	Dialog – “Add a bandwidth”	342
13.4.3	Dialog – “Network settings”	342
13.4.4	Dialog – “Select columns” (Logbook)	343
13.4.5	Dialog – “Settings” (Logbook)	344
13.4.6	Dialog – “DCHP settings”	345
13.4.7	Dialog – “Certificate renewal”	345
13.4.8	Dialog – “Add remote logging server”	347
13.4.9	Dialog – “Settings”	348
13.4.10	Dialog – “Add user”	348
13.4.11	Dialog – “Change password”	349
13.4.12	Dialog – “Password policies”	349
13.4.13	Dialog – “Session policies”	351
13.4.14	Dialog – Login banner	352
13.4.15	Dialog – “Remote authentication”	353
13.4.16	Dialog – “Easy service access”	355
13.4.17	Dialog – “Generate key”	356
13.4.18	Dialog – “Create PKCS12 container”	357
13.4.19	Dialog – “Scheduler settings”	357
13.4.20	Dialog – “Add task”	358
13.4.21	Dialog – “Delete task”	359
13.4.22	Dialog – “Add callable to <taskname>”	359
13.4.23	Dialog – “Delete callable”	361
13.4.24	Dialog – “Watchlist”	362
13.5	Editors	363
13.5.1	Editor – “Remote Logging”	363
13.5.2	Editor – Permissions	365
13.5.3	Editor – Groups	366
13.5.4	Editor – Users	367
13.5.5	Editor – Certificates & Keys	368

13.5.6	Editor – “Network interfaces”	369
13.5.7	Sidebar – “schedMain” (Parameter)	373
13.5.8	Sidebar – “Task: <name>” (Task parameter)	373
13.5.9	Sidebar – “<name_callable>” (Callable parameter)	374
13.6	Widgets	374
13.6.1	Widget – “Hardware”	374
13.6.2	Widget – “Typeplate”	375
13.6.3	Widget – “General”	375
14	Related documentation	376
14.1	Overview	376
14.2	ctrlX AUTOMATION	377
14.3	ctrlX WORKS	377
14.4	ctrlX OS	378
14.5	ctrlX OS apps	378
15	Service and support	383
16	Index	385

1 About this documentation

Editions of this documentation

Edition	Release date	Note
01	2024-01	First edition ctrlX CORE version XCR-V-0202
02	2024-03	Edition ctrlX CORE version XCR-V-0204 New content: • ↗ Control monitoring • ↗ Window - "Firmware Update" • ↗ Window - "System Performance" • ↗ Tab - "Alarms" • ↗ Tab - "History" • ↗ Tab - "Revocation lists" • ↗ Tab - "Threads" • ↗ Dialog - "Settings" (Logbook) • ↗ Dialog - "Select columns" (Logbook) • ↗ Widget - "Hardware" • ↗ Backup - Restrictions • ↗ Working with configurations • ↗ Release cycle of apps and engineering software • ↗ Data Layer - FlatC • ↗ Data Layer - Global settings Revised content: • ↗ ctrlX CORE Scheduler • ↗ Exception handling • ↗ Hardware watchdog • ↗ Licenses - Overview • ↗ Creating a new user • ↗ Introduction • ↗ Logbook - Overview • ↗ Logbook - Exporting • ↗ Remote Logging - Configuring external server • ↗ Backup - Creating from ctrlX device • ↗ Backup - Options when creating • ↗ Backup - Applying to ctrlX device • ↗ Backup - file content • ↗ Backup - Structure of the backup file "ctrlx-setup.json" • ↗ Backup - Connectivity • ↗ Backup - Proxy • ↗ Backup - System • ↗ Backup - Command line • ↗ Backup - Automatic application upon system startup • ↗ Backup - HowTo's • ↗ Window - "Apps"

Edition	Release date	Note
		• ➔ Window – “PKIs” • ➔ Window – “Reset Device Data” • ➔ Window – “System Information” • ➔ Tab – “Certificates” • ➔ Tab – “Resources” • ➔ Tab – “Processes” • ➔ Dialog – “Certificate renewal” • ➔ Dialog – “Add user” • ➔ Dialog – “Password policies” • ➔ Editor – Users • ➔ Editor – Certificates & Keys • ➔ Widget – “General” • ➔ Widget – “Typeplate” • ➔ NRT access to RT data • ➔ Subscription - Rule set • ➔ Subscription - Global settings • ➔ Subscription - Client examples • ➔ Subscription - Lossless subscription to real-time data • ➔ Error categories, log types and (error) deletion
03	2024-06	Edition ctrlX CORE version XCR-V-0206 Reorganization and optimization of chapter structures New content: • ➔ System overview • ➔ Program configuration mode • ➔ LED states • ➔ Setting up ctrlX OS devices • ➔ Limiting network bandwidth • ➔ Option for dynamic determination of the maximum package size (PMTUD) • ➔ Tab – “PKCS12 containers” • ➔ Tab – “Certificate enrollments” • ➔ Dialog – “Create PKCS12 container” • ➔ Dialog – “Network settings” • ➔ Dialog – “Add a bandwidth” • ➔ Comparing configurations

Edition	Release date	Note
		Revised content: • ➔ Licenses – Overview • ➔ Introduction • ➔ Window – “Home” • ➔ Window – “Licenses” • ➔ Window – “Date & Time” • ➔ Window – “Apps” • ➔ Window – “Permissions” • ➔ Window – “PKIs” • ➔ Tab – “Licenses” • ➔ Tab – “Configuration” • ➔ Tab – “Certificates” • ➔ Tab – “Keys” • ➔ Tab – “History” • ➔ Editor – Certificates & Keys • ➔ Remote Logging – Configuring external server • ➔ Window – Remote Logging • ➔ Dialog – “Add remote logging server” • ➔ Editor – “Remote Logging” • ➔ Window – “Network interfaces” • ➔ General information on network interfaces • ➔ Adding and configuring VLAN interface • ➔ Bridging interfaces • ➔ Window – “System Information” • ➔ Widget – “General” • ➔ Window – “System Performance” • ➔ Tab – “Resources” • ➔ Window – “Manage app data” • ➔ Loading app data from the engineering PC • ➔ Resetting active app data • ➔ Chapter Saving active app data to the archive of the ctrlX device on page 69 • ➔ Uploading from the engineering PC to the archive of the ctrlX device • ➔ Loading configuration from the archive to the ctrlX device • ➔ Viewing configuration in the archive • ➔ Displaying information on a configuration in the archive • ➔ Editing description of a configuration • ➔ Managing configurations • ➔ Backing up active app data Important functional change Version XCR-V-0206 replaces the previous backup and restore function with the setup functionality while retaining the name "Backup and restore".

Edition	Release date	Note
		Adaptation of the documentation: • Removal of the previous contents of "Backup and restore". • Adaptation of the previous contents of "Setup" to "Backup" or "Backup and restore". • Change to the document structure/chapter "Setup" removed and replaced with adapted content. Adapted content of "Backup and restore" (formerly "Setup"): • ➔ Window – “Backup & Restore” • ➔ Tab – “Back up” • ➔ Tab – “Restore” • ➔ Backup – Introduction • ➔ Backup – Creating from ctrlX device • ➔ Backup – Options when creating • ➔ Backup – Applying to ctrlX device • ➔ Backup – Showing report • ➔ Backup – Restrictions • ➔ Backup – Showing history • ➔ Backup – Structure of the backup file "ctrlx-setup.json" • ➔ Backup – file content • ➔ Backup – Active configuration and configuration archive • ➔ Backup – Connectivity • ➔ Backup – Date and time • ➔ Backup – Hosts • ➔ Backup – User management (IdentityManagement) • ➔ Backup – License information • ➔ Backup – Package management (apps) • ➔ Backup – Deleting apps • ➔ Backup – Proxy • ➔ Backup – SSH • ➔ Backup – System • ➔ Backup – System information • ➔ Backup – Type plate information • ➔ Backup – Manual modification • ➔ Backup – Diagnostics • ➔ Chapter 6.6.10 Backup – Command line on page 105 • ➔ Backup – Automatic application upon system startup • ➔ Backup – HowTo's • ➔ Backup – Extensions to the backup file format • ➔ Backup – Dealing with changes • ➔ Backup – FAQ

Edition	Release date	Note
04	2025-10	New content: ➔ ctrlX OS Data Layer - Data storage and data access ➔ ctrlX OS Data Layer - Scope of data ➔ ctrlX OS Data Layer - Further information

1.1 ctrlX OS – Licensing notes

Open Source Components

A list of the open source components is available in ctrlX WORKS and in the ctrlX OS interface in the side navigation in the menu item “About” below the button “Open Source Components”, see web documentations:

➔ Window – About

➔ Window – Open Source Components

2 Important directions on use

2.1 Intended use

2.1.1 Introduction

Rexroth products are developed and manufactured to the state-of-the-art.

The products are tested prior to delivery to ensure operational safety and reliability.

⚠ WARNING	Personal injury and damage to property due to incorrect use of products! The products may only be used as intended. Failure to use the products as intended may cause situations resulting in property damage and personal injury.
NOTICE	Damages resulting from unintended use Rexroth As the manufacturer does not assume any warranty, liability or compensatory claims for damages resulting from unintended use of the products. The user alone shall bear the risks of an unintended use of the products. Before using Rexroth products, make sure that all the prerequisites for an intended use of the products are met: – Personnel that in any way, shape or form uses Rexroth products must first read and understand the relevant safety instructions and be familiar with their intended use – Leave hardware products in their original state, i.e., do not make any structural modifications. It is not permitted to decompile software products or alter source codes – Do not install damaged or defective products or commission them – It has to be ensured that the products have been installed as described in the relevant documentation

2.1.2 Areas of use and application

Products of the ctrlX series are suitable for Motion/Logic applications.

NOTICE	Products of the ctrlX series may only be used with the accessories, mounting parts, and other components specified in this documentation. Components that are not expressly mentioned must neither be attached nor connected. The same applies to cables and lines. Only to be operated with the hardware component configurations and combinations expressly specified and with the software and firmware specified in the corresponding documentations and functional descriptions.
---------------	---

Products of the ctrlX series are suitable for single-axis as well as for multi-axis drive and control tasks. Device types with different equipment and interfaces are available for using the system in specific applications.

Typical areas of application:

- Building automation
- IoT and Security Gateway or Device
- Handling & Robotic

Controls of the ctrlX CORE series may only be operated under the mounting and installation conditions, in the position of normal use and under the ambient conditions (temperature, degree of protection, humidity, EMC, etc.) specified in the related documentations.

2.2 Unintended use

"Unintended use" refers to using the ctrlX products outside of the above-mentioned areas of application or under operating conditions and technical data other than described and specified in the documentation.

ctrlX products must not be used if they are exposed to following conditions:

- Operating conditions that do not meet the specified ambient conditions.
Operation under water, under extreme temperature fluctuations or under extreme maximum temperatures is prohibited
- Applications that have not been expressly authorized by Rexroth

3 Safety instructions

The Safety instructions contained in the available application documentation feature specific signal words (DANGER, WARNING, CAUTION or NOTICE) and, where required, a safety alert symbol (in accordance with ANSI Z535.6-2006).

The signal word is meant to draw the reader's attention to the safety instruction and identifies the hazard severity.

The safety alert symbol (a triangle with an exclamation point), which precedes the signal words DANGER, WARNING and CAUTION, is used to alert the reader to personal injury hazards.

The Safety instructions in this documentation are designed as follows:

▲ DANGER	In case of non-compliance with this safety instruction, death or serious injury will occur.
▲ WARNING	In case of non-compliance with this safety instruction, death or serious injury could occur.
▲ CAUTION	In case of non-compliance with this safety instruction, minor or moderate injury could occur.
NOTICE	In case of non-compliance with this safety instruction, property damage could occur.

4 System overview

ctrlX AUTOMATION is the Rexroth automation system that pushes the boundaries and sets new standards in the world of automation with its ctrlX CORE control system and the operating system ctrlX OS.

The following chapters provide an insight into the core elements of ctrlX AUTOMATION:

- Control hardware:
 - ➔ [ctrlX CORE – Control system](#)
 - ➔ [ctrlX CORE Virtual – Virtual control](#)
 - ➔ [ctrlX I/O – Module kit](#)
- Operating system:
 - ➔ [ctrlX OS – Operating system](#)
 - ➔ [ctrlX OS— system concept](#)
 - ➔ [ctrlX OS – Web-based user interface](#)
 - ➔ [ctrlX OS – Data Layer](#)
- Software tools:
 - ➔ [ctrlX WORKS](#)
 - ➔ [ctrlX I/O Engineering](#)
 - ➔ [ctrlX PLC Engineering](#)
 - ➔ [ctrlX DRIVE Engineering](#)
 - ➔ [ctrlX SAFETY Engineering](#)

ctrlX AUTOMATION COMMUNITY

Keep up-to-date and find out the latest news and interesting facts, see: ➔ [ctrlX COMMUNITY](#)

Training opportunities at Rexroth

Benefit from the knowledge of our experts and visit our training program, see: ➔ [Rexroth Training Academy](#)

4.1 ctrlX CORE – Control system

The ctrlX CORE control system enables a forward-looking approach to the implementation of automation projects.

The decisive factor is the ctrlX OS operating system, which, with its modular, app-based system concept, ensures maximum flexibility and freedom on all ctrlX CORE controls.

Extend the functionality of your ctrlX CORE control by installing apps, for example from the range available in the ctrlX STORE, see: ➔ [ctrlX COMMUNITY](#)

For further information on the modular app system concept, go to: ➔ [ctrlX OS— system concept](#)

Features

- Real-time capable Linux operating system ctrlX OS
- A variety of apps for scalability of the functional scope of the software
- Multicore CPU performance and optional interfaces for hardware scalability
- High level of cyber security
- Simplified and transparent device management through remote administration via the ctrlX Device Portal
- Durable industrial hardware design



Fig. 1: ctrlX device variants CORE X3 /COREplus X3 / CORE X7

Further information

Visit the following pages to find out more about the ctrlX CORE device family:

- ➔ [ctrlX CORE – Product page on the web](#)
- ➔ [ctrlX CORE – Configurator](#)
- ➔ [ctrlX CORE – Quick Start Guide as download](#)
- ➔ [ctrlX CORE – Product documentation](#)
- ➔ [ctrlX AUTOMATION – "How to" articles](#)
- ➔ [ctrlX AUTOMATION – Community](#)

4.2 ctrlX CORE Virtual – Virtual control

"ctrlX CORE Virtual" is a virtual control to test functions and configurations on the engineering PC (without control hardware).

Using the ctrlX CORE virtual is free of charge and provided via the ctrlX WORKS installation on the engineering PC, see: ➔ [ctrlX WORKS](#)

Features

- Emulation of a ctrlX CORE control
- Installing and operating apps (restrictions due to non-existent hardware have to be taken into account)
- Offline configuration
- Suitable for training and testing purposes.
- Not permitted for productive use.
- No hard realtime.
- Limited runtime of up to 4 hours. A restart is then required.

For more information on setting up and using the ctrlX CORE virtual, see:

➔ [Web documentation](#)

4.3 ctrlX I/O – Module kit

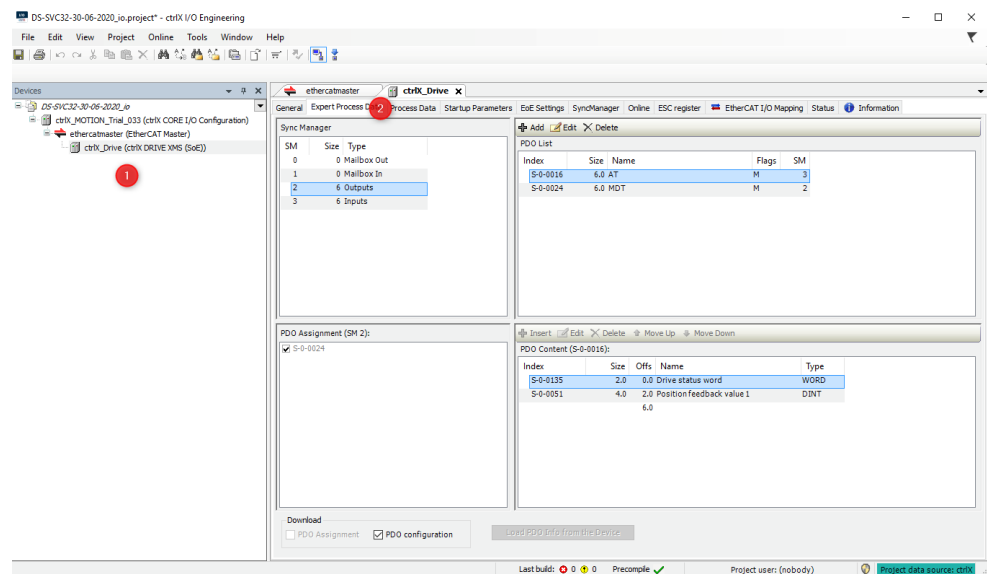
ctrlX I/O extends the automation system ctrlX AUTOMATION with additional interfaces to the field and control level. Bosch Rexroth relies uncompromisingly on standardized, industry- and manufacturer-neutral interfaces. The design is compact, modular and robust - significant savings are possible on the top-hat rail: Up to 50 % less space in the control cabinet, installation and wiring are possible without tools.

Among other things, the flexible, high-performance solution impresses with its openness to the EtherCAT ecosystem, flexible machine topologies and different communication standards. Web-based configuration and diagnostics, Plug & Run and integrated monitoring functions take engineering to a new level.

ctrlX I/O is perfectly matched to ctrlX CORE and the mapping in the ctrlX Data Layer enables easy I/O access for all apps or applications.

Features

- 50 % less space required in the control cabinet
- Flexible and unlimited topologies
- EtherCAT up to the clamp
- Simple, fast commissioning
- Perfectly matched to the ctrlX OS/ctrlX CORE system
- Option to integrate your own modules
- Future-proof



Further information

- ➔ [ctrlX I/O – Product page on the web](#)
- ➔ [ctrlX I/O – Product documentation](#)

4.4 ctrlX OS – Operating system

ctrlX OS is the Linux-based operating system used on all ctrlX CORE control devices from Bosch Rexroth.

In addition, ctrlX OS is also available as separate solution for operation on compatible third-party devices for the industrial environment.

The basic concept of ctrlX OS is an open, real-time-capable system that enables a completely new system concept thanks to its modular, app-based approach.

Similar to a smart device, you can decide which additional functions to implement by installing apps on the ctrlX control device. The following chapter provides information on the app-based system concept.

With ctrlX OS, ctrlX CORE controls can be used as Motion controls, PLC controls or even as IoT or Security Gateways. This depends on the application.

Features

- Consistently open standards
- App-based system concept
- Real-time capable Linux operating system
- Web-based engineering
- Standardized data communication over ctrlX Data Layer
- Comprehensive IoT connection

Further information

- ➔ [ctrlX OS – Product page on the web](#)
- ➔ [ctrlX OS – Product documentation](#)

4.5 ctrlX OS— system concept

Thanks to the modular and open system concept of ctrlX OS, the functional scope of ctrlX CORE devices can be adapted to the requirements of the application by installing additional apps, see: ➔ [ctrlX COMMUNITY](#)

The installed apps are independent software packages managed in the ctrlX OS user interface, see: ➔ [Window – “Apps”](#).

The apps communicate with each other mainly via the ctrlX Data Layer which acts as a central nervous system and via which real-time and non-real-time data can be exchanged, see: ➔ [ctrlX OS – Data Layer](#).

Further information

- ➔ [Apps – Basics](#)
- ➔ [Licenses – Overview](#)

Developing own apps for ctrlX OS

Use the Software Development Kit (SDK) for ctrlX AUTOMATION to develop your own apps for use on ctrlX CORE devices.

Find out more about this topic in the "How to" section of the ctrlX AUTOMATION website, see: ➔ [SDK for ctrlX AUTOMATION](#).

4.6 ctrlX OS – Web-based user interface

The ctrlX OS operating system has a web-based user interface as the central interface between the user and the ctrlX CORE device.

The user interface can be accessed via common web browsers such as Firefox, Google Chrome or Windows Edge using the IP address of the ctrlX device.

The user interface supports the app-based system concept of the ctrlX OS operating system, in which newly installed apps with their own interfaces integrate seamlessly into the ctrlX OS user interface.

Features

- Overview on and operation of the installed apps
- Device status/logbook
- Configuring device settings
- Commanding operating state
- App management/installing, updating and uninstalling apps
- Calling programming tools, e.g. PLC Engineering/I/O Engineering
- ...

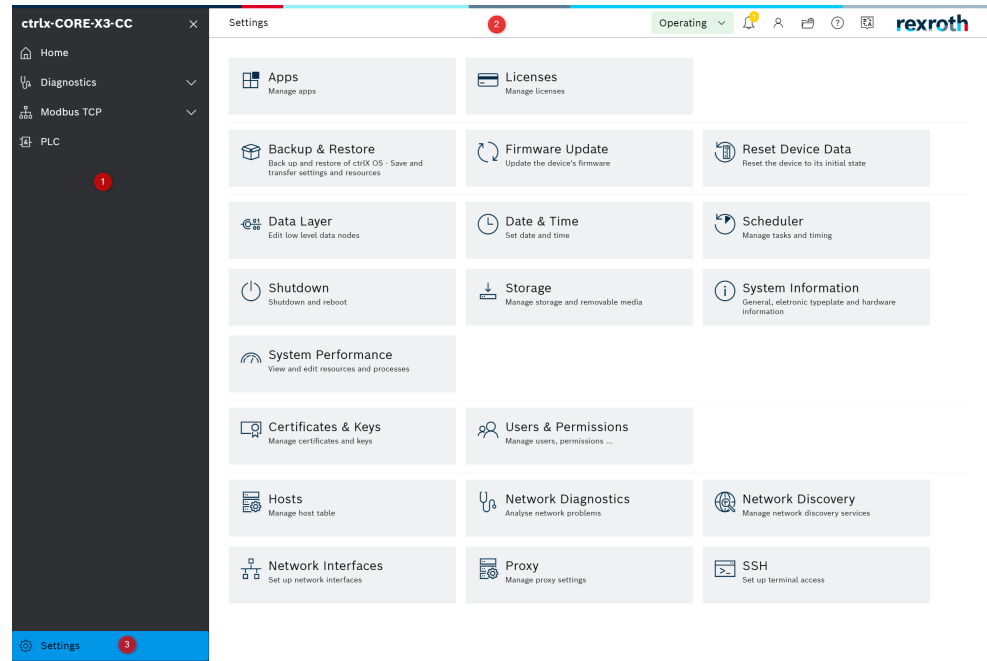


Fig. 2: Example: ctrlX OS user interface – Window Settings

- 1 Navigation bar
- 2 Header
- 3 “Settings” window

Further information

➔ [ctrlX OS web interface – Overview](#)

4.7 ctrlX OS – Data Layer

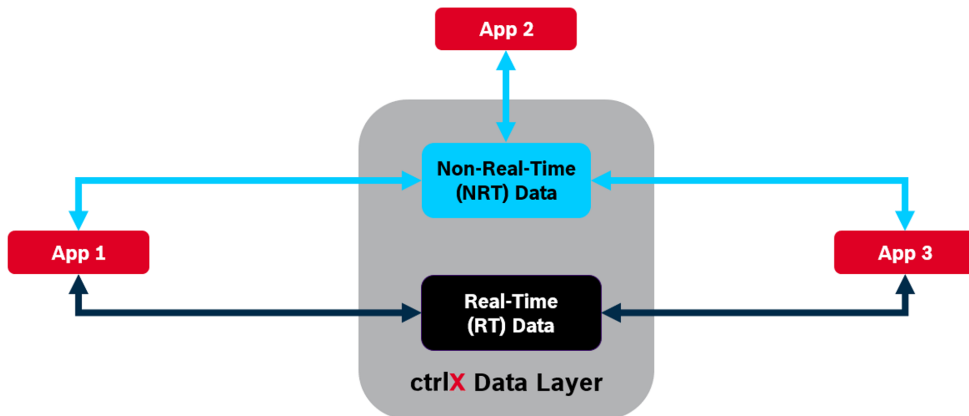
The Data Layer is like the central nervous system of the ctrlX OS system by enabling and ensuring a secure communication and data exchange between the apps and the system.

Via the Data Layer apps can exchange real-time data (e.g. process communication) and non-real-time data (e.g. configuration data) with each other also via OPC UA.

The Data Layer belongs to the basis systems of ctrlX OS and provides the following options:

- Apps can provide their own data in the ctrlX Data Layer in app-specific node structures so that data can be used by other apps.
- Apps can use the ctrlX Data Layer to access data provided by other apps.

These options are available for configuration information, which has normally no real-time requirements, and process information that has to be exchanged in real time. For example, an I/O app such as the EtherCAT Master App and the PLC App can communicate via the Data Layer.



Further information

- ➔ [Data Layer FAQ in the web](#)
- ➔ [Data exchange via Data Layer](#)

4.8 ctrlX WORKS

ctrlX WORKS is a free software toolbox for your engineering PC.

Features

- **Device overview**
Display of all ctrlX control devices accessible in the network.
Call the control devices directly from the list (call of the web user interface/login). The same applies to virtual controls operated locally on the engineering PC.
Pin preferred controls to display devices permanently displayed in the overview even without network connection (cannot be opened without network connection).
- **Virtual control**
Create virtual controls in ctrlX WORKS, e.g. to test your projects without hardware, see: ➔ [Web documentation](#).
- **Software tools**
If you installed additional ctrlX engineering tools on your engineering PC, e.g. PLC Engineering or I/O Engineering, the tools can be started directly from the overview.
- **App build environment**
Add an app build environment to develop your own for ctrlX OS, see: ➔ [How to use SDK](#).
- **Settings**
The following configurations can be made in the settings, see: ➔ [Web documentation](#).

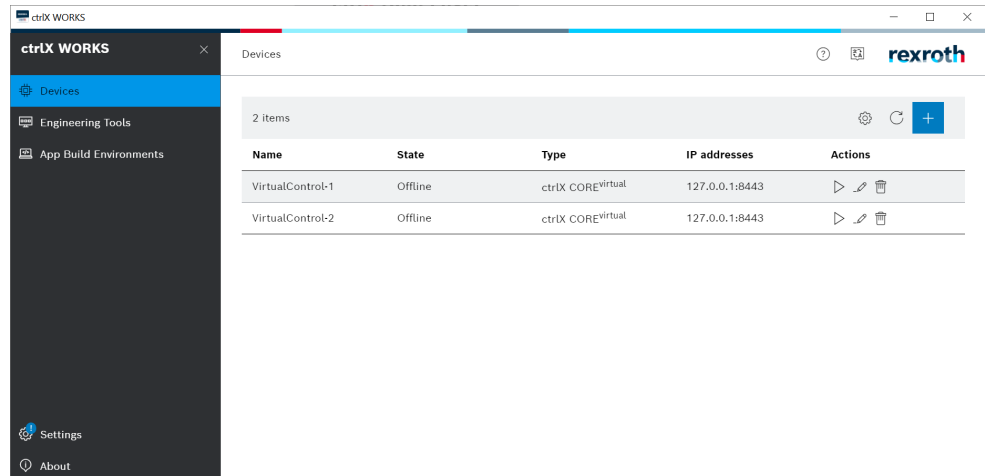


Fig. 3: Example: Device overview with two virtual controls

Installing on the your PC

ctrlX WORKS is provided as stand-alone installation package.

At the time of publication of this documentation, the installation package can only be downloaded by registered customers in the Rexroth Collaboration Room, see: [Web link](#).

For further information on the enrollment, see: [Web link](#).

Further information

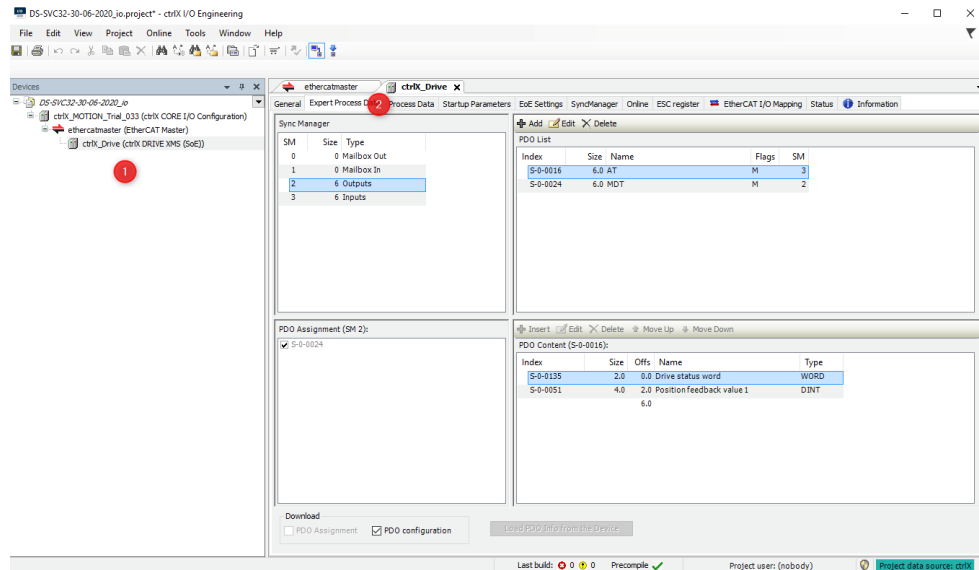
[ctrlX WORKS – Product documentation](#)

4.9 ctrlX I/O Engineering

ctrlX I/O Engineering is a free software tool for use on the engineering PC, used for the EtherCAT bus configuration for the ctrlX control device.

Features

- Configuring or reading out the EtherCAT configuration
- Downloading the EtherCAT configuration to the EtherCAT Master app via ctrlX Data Layer
- EtherCAT diagnostics
- Project synchronization between engineering PC and ctrlX device
- Extendable with Codesys Professional Edition



- 1 Device tree/EtherCAT Master with subordinate bus devices
- 2 Workspace/generic device editor with device-specific configuration options

Installing on the your PC

ctrlX I/O Engineering is provided as stand-alone installation package.

At the time of publication of this documentation, the installation package can only be downloaded by registered customers in the Rexroth Collaboration Room, see: [Web link](#).

For further information on the enrollment, see: [Web link](#).

Further information

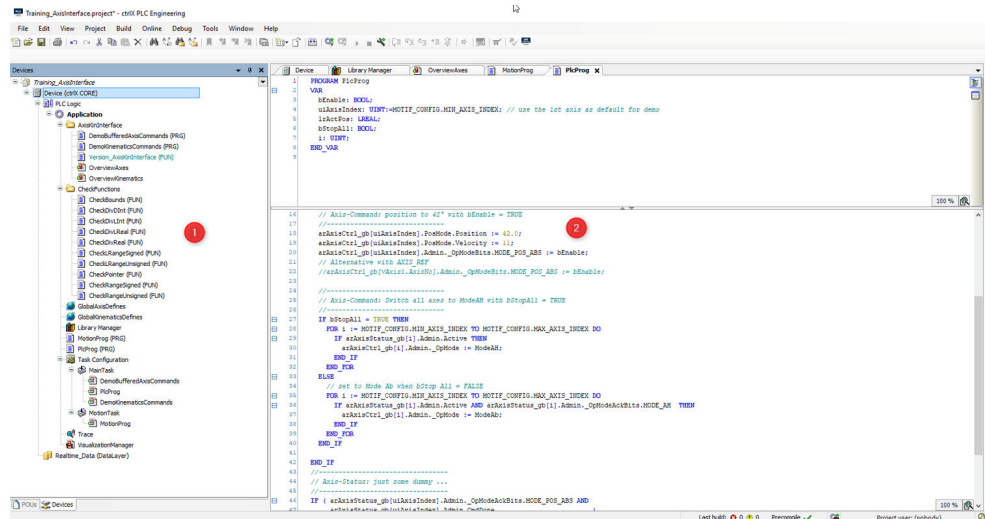
[ctrlX I/O Engineering – Product documentation](#)

4.10 ctrlX PLC Engineering

ctrlX PLC Engineering is a free software tool for use on the engineering PC to create PLC applications for the ctrlX control device.

Features

- Classic PLC programming according to IEC-61131
- Project synchronization between engineering PC and ctrlX device
- System-comprehensive data exchange via access functions with ctrlX Data Layer
- Extendable with Codesys add-ons → available via the [ctrlX STORE](#)
 - PROFINET I/O Controller/Device
 - Ethernet IP Scanner/Adapter
 - EtherCAT Master
 - IO-Link
 - IIoT libraries SL add-on
 - Codesys Professional Developer Edition



- 1 Device tree/control node and application
- 2 Workspace/programming editor

Installing on the your PC

ctrlX PLC Engineering is provided as stand-alone installation package.

At the time of publication of this documentation, the installation package can only be downloaded by registered customers in the Rexroth Collaboration Room, see: [Web link](#).

For further information on the enrollment, see: [Web link](#).

Further information

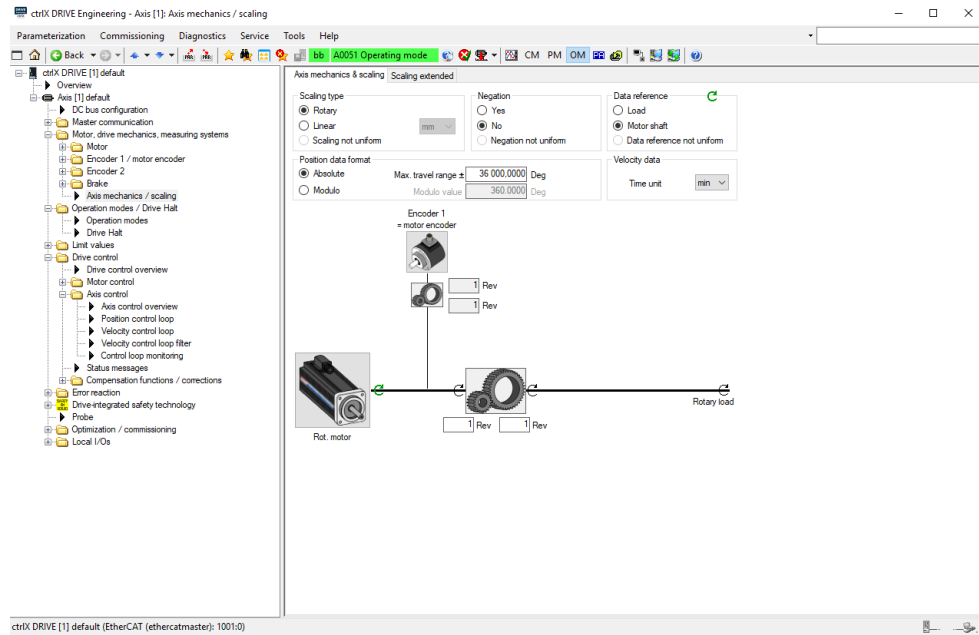
[ctrlX PLC Engineering – Product documentation](#)

4.11 ctrlX DRIVE Engineering

ctrlX DRIVE Engineering is a free software tool for use on the engineering PC. ctrlX DRIVE Engineering ensures a quick and easy commissioning of the ctrlX DRIVE servo drives with graphical dialogs and guided commissioning wizards. Numerous diagnostic functions support drive optimization for optimum application results.

Features

- Step-by-step commissioning
- Simplified initial commissioning
- LogAnalyzer
- and much more...



Installing on the your PC

ctrlX DRIVE Engineering is provided as stand-alone installation package.

At the time of publication of this documentation, the installation package can only be downloaded by registered customers in the Rexroth Collaboration Room, see: [Web link](#).

For further information on the enrollment, see: [Web link](#).

Further information

[ctrlX DRIVE Engineering – Product documentation](#)

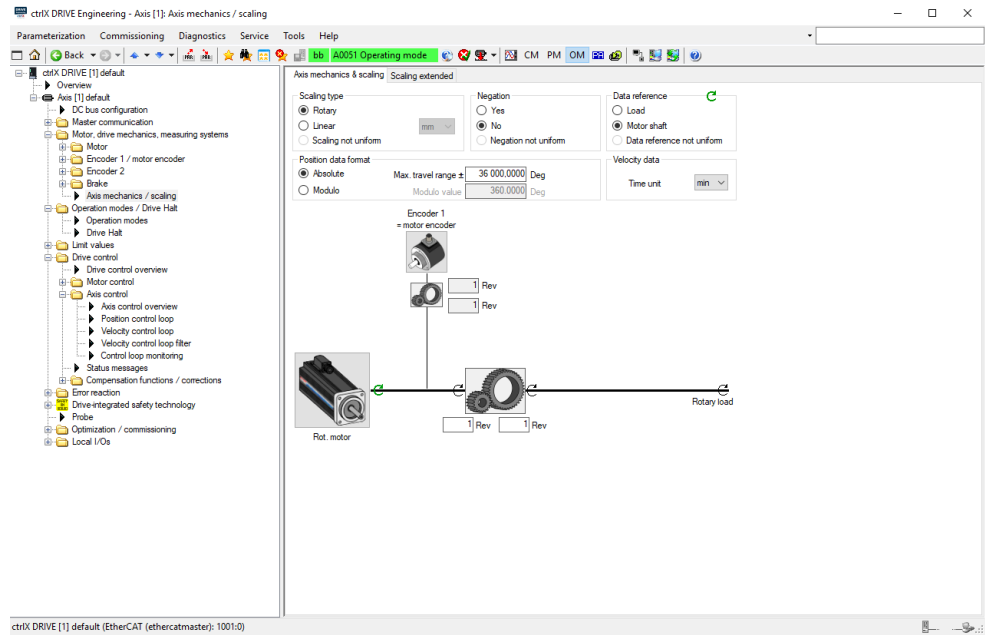
4.12 ctrlX SAFETY Engineering

ctrlX SAFETY Engineering is a free software tool for use on the engineering PC and enables intuitive and fast configuration of the ctrlX SAFETY controls with a graphical user interface.

Numerous download, upload and diagnostic functions support debugging and validation based on the security requirements of the users.

Features

- Project/configuration tree
- Multi-level editor view (device, wiring, function)
- MainTask and Fast Channel function context
- Modularization support for functional schemes
- Intuitive graphical function block editor
- Function block library
- Device interface - Error/diagnostic logbook
- Device interface - Signal status
- and much more...



Installing on the your PC

ctrlX SAFETY Engineering is provided as stand-alone installation package.

At the time of publication of this documentation, the installation package can only be downloaded by registered customers in the Rexroth Collaboration Room, see: [Web link](#).

For further information on the enrollment, see: [Web link](#).

Further information

[ctrlX SAFETY Engineering – Product documentation](#)

5 Managing ctrlX apps

5.1 Apps – Basics

Upon delivery, only system apps to operate the control are installed on a ctrlX device.

As for a “smart device”, the functionality of the control can be extended by installing apps and the requirement of the automation project can be modified.

ctrlX Automation apps

For an overview on the available ctrlX apps, refer to:

[Licenses – Overview](#).

Apps can be downloaded from the ctrlX Store:

[ctrlX Store in the web](#)

Useful web links

[HOW-TO section](#)

[ctrlX AUTOMATION FORUM](#)

[ctrlX AUTOMATION Community](#)

App management on the ctrlX device

The apps are managed on the ctrlX device in the “Apps” window in the web interface of the ctrlX device, see:

➔ [Window – “Apps”](#)

Actions to manage apps

- ➔ [Uploading apps to the app memory of the ctrlX device](#)
- ➔ [Installing apps on the ctrlX device from the device-internal app storage](#)
- ➔ [Installing apps from the engineering PC on the ctrlX device](#)
- ➔ [Installing apps on the ctrlX device from the ctrlX Store](#)
- ➔ [Updating apps](#)
- ➔ [Changing app version](#)
- ➔ [Disabling apps](#)
- ➔ [Enabling apps](#)
- ➔ [Downloading apps from the ctrlX device](#)
- ➔ [Uninstalling apps](#)
- ➔ [Reinstalling apps](#)
- ➔ [Deleting apps from app storage](#)
- ➔ [ctrlX device – Enabling the Service mode](#)
- ➔ [ctrlX device – Enabling the Operating mode](#)

"Automation app" Term definition

Installable functional software extensions are called “app” (application) in the ctrlX OS world. Technically, ctrlX OS apps are divided into two categories “apps” and “Automation apps”.

“Automation apps” are functional extensions to operate centralized automation tasks such as the PLC runtime environment, the drive control and the field bus communication.

All further functional extensions are provided from common apps.

Available ctrlX OS Automation apps:

- **PLC App**
App to operate the PLC runtime environment
- **Motion App**
App to configure and control drives, axes and kinematics
- **EtherCAT Master App**
App to operate the EtherCAT field bus

5.2 Installing apps on the ctrlX device from the ctrlX Store

If the ctrlX device is connected to the internet and connected to the ctrlX Store, apps can directly be installed on the ctrlX device from the ➔ [ctrlX Store](#). The connection status to the ctrlX Store is shown in the ➔ [Window – “Apps”](#) in the command bar:

-  Connection to the ctrlX Store active
-  No connection to the ctrlX Store

If the connection is active, available apps are shown as tiles in the app overview below the category “Available apps”. For the installation steps, refer to the following description:

➔ [Installing apps on the ctrlX device from the device-internal app storage](#)

5.3 Installing apps from the engineering PC on the ctrlX device

Apps can directly be transferred from the engineering PC to the ctrlX device and installed. In this case, the engineering PC is like an installation data source, e.g. if the ctrlX device and the ctrlX Store cannot be connected (web download).

However, the respective app is already saved on the engineering PC, see: [↗ Apps in ctrlX Store](#)

Single or multiple apps can thus be selected for installation on the engineering PC. The installation sequence is automatically specified for multiple apps.

NOTICE

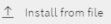
Prerequisite for the following action:

Working at apps and app settings is only possible in the "Service" mode.

Editing real-time data on the ctrlX device is stopped in "Service" mode.

Prior to enabling the "Service" mode, the ctrlX device has to be in a safe state!

Procedure

1.  In the ctrlX OS web interface, navigate to the window "*Settings → Apps*", see:
[↗ Window – "Apps"](#)
2.  Switch the ctrlX device to the "Service", see:
[↗ ctrlX device – Enabling the Service mode](#)
3.  Click on the  button
➔ A file selection window opens
4.  Select the app to be installed in the file selection window and confirm the selection
➔ The required app data is uploaded to the ctrlX device first
➔ After uploading, there is an automatic check for dependencies to other apps on the ctrlX device. If dependencies exist, they are displayed in the installation dialog with the respective app and its required software version (lowest/highest version). The dependencies can also be displayed in the app details after installation, see "Extended app overview" in the chapter [↗ Window – "Apps"](#)
5.  Confirm the installation dialog with "Install"
➔ The app is installed on the ctrlX device and then shown in the app overview.
6.  When the app is installed, app settings have usually to be modified.
After the modifications, the ctrlX device can be switched again to the "Operating" mode, see:
[↗ ctrlX device – Enabling the Operating mode](#)

5.4 Uploading apps to the app memory of the ctrlX device

Apps can be saved in the device-internal app storage on the ctrlX device without installation. This is advantageous for example if the ctrlX device and the ctrlX Store cannot be connected and if apps cannot be downloaded to the ctrlX device via web download. A "stock" of apps can directly be saved on the ctrlX device and installed when required, see:

➔ Installing apps on the ctrlX device from the device-internal app storage



The app storage uses the internal memory resources of the ctrlX device and can be enabled or disabled, see ➔ [Dialog – “Settings”](#).

In order not to unnecessarily increase the load of the memory resources of the ctrlX device, it is recommended to delete unused apps from the app storage.



To load apps to the app storage, download the respective app to the engineering PC before, see: ➔ [Apps in ctrlX Store](#)

Procedure

1. ➔ In the ctrlX OS web interface, navigate to the window “*Settings → Apps*”, see:
➔ [Window – “Apps”](#)
2. ➔ In the command bar, click on the  **App storage** button.
➔ The app overview changes and shows the content of the app storage on the ctrlX device.
3. ➔ In the command bar, click on the  button.
➔ A file selection dialog opens.
4. ➔ For the previously downloaded app, navigate in the dialog to the storage location on the engineering PC.
5. ➔ Select the app to be uploaded to the app memory of the ctrlX device and confirm the dialog.
➔ App data is uploaded to the app storage of the ctrlX device and shown in the app overview after completion.
6. ➔ To close the view, select the  button.
➔ The uploaded app is now saved in the app storage of the device and can also be installed on the ctrlX device, see:
➔ [Installing apps on the ctrlX device from the device-internal app storage](#)

5.5 Installing apps on the ctrlX device from the device-internal app storage

Apps can be saved in the device-internal app storage on the ctrlX device without an installation being executed. This option allows to save apps before on the ctrlX device. It required, they can also be installed on the ctrlX device. Possible use cases: Optional device functions or different machine variants that require additional apps to be operated. The prerequisite is that the required apps were previously uploaded to the app memory of the ctrlX device, see:

➔ Uploading apps to the app memory of the ctrlX device



The app storage uses the device-internal memory resources of the ctrlX device and can be enabled or disabled, see ➔ [Dialog – “Settings”](#).

In order not to unnecessarily increase the load of the memory resources of the ctrlX device, it is recommended to delete unused apps from the app storage.

NOTICE

Prerequisite for the following action:

Working at apps and app settings is only possible in the “Service” mode.

Editing real-time data on the ctrlX device is stopped in the “Service” mode.

Prior to enabling the “Service” mode, the ctrlX device has to be in a safe state!

Procedure

1. In the ctrlX OS web interface, navigate to the window “*Settings → Apps*”, see:
[↪ Window – “Apps”](#)
2. Switch the ctrlX device to the “Service” mode, see:
[↪ ctrlX device – Enabling the Service mode](#)
3. In the app overview, navigate to the category “Available apps”.
 ➔ In this category, all apps saved in the app storage on the ctrlX device and all apps provided via the ctrlX Store are shown, see:
[↪ Installing apps on the ctrlX device from the ctrlX Store](#)
4. Search for the app to be installed and click on **Install**.
 ➔ If multiple app versions are provided for installation, the available versions as well as the app source are shown in the list. In this case, select the app from the list to start the installation.
 If only one app version is provided, the installation starts directly.
 After the installation, the app is shown in the app overview, category “Installed apps”.
5. When the app is installed, app settings have usually to be modified.
 After the modifications, the ctrlX device can be switched again to the “Operating” mode, see:
[↪ ctrlX device – Enabling the Operating mode](#)

5.6 Updating apps

Updating an app is similar to the installation procedure. If an app update is available, the “Update” button is highlighted in green on the right next to the app tile. To start the app update, click on the button.



If multiple app versions are available for an "update", a version list is displayed. Select the version to be updated. App online updates of ctrlX CORE devices and ctrlX OS devices are limited to 30 update windows in the first year and 12 update windows per device per year in subsequent years. An update window starts with the first download of an app from the ctrlX Online Store and ends after 12 hours. During the 12 hours of the current update window, any number of apps can be updated on the respective device.

Loading app update to the ctrlX device

Updates can be loaded to the ctrlX device as follows:

- Update installation from an engineering PC directly to the ctrlX device, see:
➔ [Installing apps from the engineering PC on the ctrlX device](#)
- Update upload to the app storage of the ctrlX device and manual installation start, see:
➔ [Uploading apps to the app memory of the ctrlX device](#)
➔ [Installing apps on the ctrlX device from the device-internal app storage](#)
- Update via the ctrlX Store, see:
➔ [Installing apps on the ctrlX device from the ctrlX Store](#)

Action – Update all apps

The “Update all apps” action checks whether updates are available for the installed apps. If updates were found, these are shown in a list. The list shows the respective apps and the current as well as the new app version number. Select the apps to be updated from the list or exclude them from the update.

NOTICE	Prerequisite for the following action: Working at apps and app settings is only possible in the “Service” mode. Editing real-time data on the ctrlX device is stopped in “Service” mode. Prior to enabling the “Service” mode, the ctrlX device has to be in a safe state!
---------------	---

Procedure

1. ➔ In the ctrlX OS web interface, navigate to the window “*Settings → Apps*”, see:
➔ [Window – “Apps”](#)
2. ➔ Switch the ctrlX device to the “Service” mode, see:
➔ [ctrlX device – Enabling the Service mode](#)
3. ➔ In the command bar, click on the  button.
Note: In smaller browser windows, the button can also be in the extended option [:].
➔ The system checks whether app updates are available and shows them in a list.
4. ➔ If updates were found, check for the updates to be executed in the list and click on “Update”.
➔ The update is started.

Notes:

- Some app updates require a ctrlX device restart. In this case, the progress is displayed during the update notifying you that the ctrlX device may not be switched off. The ctrlX device is automatically restarted after the update. Log into the device again to continue with further actions.
- When updating apps that do not require an update, a notification on the successful update is shown. Confirm the dialog to continue with further actions.

After the update, the ctrlX device can be switched again to the “Operating” mode, see:

➔ [ctrlX device – Enabling the Operating mode](#)

5.7 Changing app version

If multiple versions of the installed app are available, it can be changed to a later or older version.

NOTICE

Prerequisite for the following action:

Working at apps and app settings is only possible in the “Service” mode.

Editing real-time data on the ctrlX device is stopped in the “Service” mode.

Prior to enabling the “Service” mode, the ctrlX device has to be in a safe state!

Procedure

1. In the ctrlX OS web interface, navigate to the window “*Settings → Apps*”, see:
[↪ Window – “Apps”](#)
2. Switch the ctrlX device to the “Service” mode, see:
[↪ ctrlX device – Enabling the Service mode](#)
3. In the app overview, navigate to the app with the version to be changed.
4. Click on the app tile.
➔ The extended app overview is opened with additional actions.
5. Click on the button “Change version”.
➔ All available app versions are shown.
6. Select the version and confirm the update notification.
➔ The app version is installed on the ctrlX device.
The ctrlX device can now be switched again to the “Operating” mode, see:
[↪ ctrlX device – Enabling the Operating mode](#)

5.8 Disabling apps

Installed apps can be disabled. Disabling the app does not uninstall or delete it from the ctrlX device. The app can be enabled again and updated at any time. Previously generated data is not lost.

NOTICE

Prerequisite for the following action:

Working at apps and app settings is only possible in the “Service” mode.

Editing real-time data on the ctrlX device is stopped in the “Service” mode.

Prior to enabling the “Service” mode, the ctrlX device has to be in a safe state!

Procedure

1. In the ctrlX OS web interface, navigate to the window “*Settings → Apps*”, see:
[↪ Window – “Apps”](#)
2. Switch the ctrlX device to the “Service” mode, see:
[↪ ctrlX device – Enabling the Service mode](#)
3. In the app overview, navigate to the app to be disabled.

4. ➤ Click on the app tile.
 - ➔ The extended app overview is opened with additional actions.
5. ➤ Click on the button “Disable”
 - ➔ The app is disabled.

The ctrlX device can now be switched again to the “Operating” mode, see:

➔ [ctrlX device – Enabling the Operating mode](#)

5.9 Enabling apps

NOTICE

Prerequisite for the following action:

Working at apps and app settings is only possible in the “Service” mode.

Editing real-time data on the ctrlX device is stopped in the “Service” mode.

Prior to enabling the “Service” mode, the ctrlX device has to be in a safe state!

Procedure

1. ➤ In the ctrlX OS web interface, navigate to the window “*Settings → Apps*”, see:
 - ➔ [Window – “Apps”](#)
2. ➤ Switch the ctrlX device to the “Service” mode, see:
 - ➔ [ctrlX device – Enabling the Service mode](#)
3. ➤ In the app overview, navigate to the app to be enabled.
4. ➤ Click on the app tile.
 - ➔ The extended app overview is opened with additional actions.
5. ➤ Click on the button “Enable”
 - ➔ The app is enabled.

The ctrlX device can now be switched again to the “Operating” mode, see:

➔ [ctrlX device – Enabling the Operating mode](#)

5.10 Downloading apps from the ctrlX device

Installed apps can be downloaded from the ctrlX device on an engineering, e.g. to load the app to other ctrlX devices. After the download, the app file is in the “Downloads” directory on the engineering PC.

Procedure

1. ➤ In the ctrlX OS web interface, navigate to the window “*Settings → Apps*”, see:
 - ➔ [Window – “Apps”](#)
2. ➤ In the app overview, navigate to the app to be downloaded to the engineering PC.
3. ➤ Click on the app tile.
 - ➔ The extended app overview is opened with additional actions.
4. ➤ Click on the button “Download”
 - ➔ The app is downloaded and saved in the “Download” directory on the engineering PC.

5.11 Reinstalling apps

NOTICE

Prerequisite for the following action:

Working at apps and app settings is only possible in the “Service” mode.

Editing real-time data on the ctrlX device is stopped in the “Service” mode.

Prior to enabling the “Service” mode, the ctrlX device has to be in a safe state!

Procedure

1. In the ctrlX OS web interface, navigate to the window “*Settings → Apps*”, see:
[↪ Window – “Apps”](#)
2. Switch the ctrlX device to the “Service”, see:
[↪ ctrlX device – Enabling the Service mode](#)
3. In the app overview, navigate to the app to be reinstalled.
4. Click on the app tile.
 ➔ The extended app overview is opened with additional actions.
5. Click on the button “Reinstall”
 ➔ The app is reinstalled.
 The ctrlX device can now be switched again to the “Operating” mode, see:
[↪ ctrlX device – Enabling the Operating mode](#)

5.12 Uninstalling apps

NOTICE

Prerequisite for the following action:

Working at apps and app settings is only possible in the “Service” mode.

Editing real-time data on the ctrlX device is stopped in the “Service” mode.

Prior to enabling the “Service” mode, the ctrlX device has to be in a safe state!

Procedure

1. In the ctrlX OS web interface, navigate to the window “*Settings → Apps*”, see:
[↪ Window – “Apps”](#)
2. Switch the ctrlX device to the “Service” mode, see:
[↪ ctrlX device – Enabling the Service mode](#)
3. In the app overview, navigate to the app to be uninstalled.
4. Click on the button “Uninstall” on the right next to the app tile.
 ➔ A confirmation prompt opens.
5. Acknowledge the confirmation prompt.
 ➔ The app is uninstalled.
 The ctrlX device can now be switched again to the “Operating” mode, see:
[↪ ctrlX device – Enabling the Operating mode](#)

5.13 Deleting apps from app storage



Apps in the app storage (not installed on the ctrlX device) can be deleted. Deletion reduces the memory requirement on the ctrlX device.

Procedure

1. In the ctrlX OS web interface, navigate to the window “*Settings → Apps*”, see:
↪ [Window – “Apps”](#)
2. In the command bar, click on the  **App storage** button.
➔ The app overview changes and shows the content of the app storage on the ctrlX device.
3. One or multiple apps can be changed at the same time. Enable the checkboxes of the respective apps and click on the button .
4. Acknowledge the confirmation prompt
➔ The selected apps are deleted.

5.14 ctrlX device – Enabling the Service mode

Editing real-time data on the ctrlX device is stopped in the “Service” mode. Working and modifying apps and their settings on the ctrlX device is only possible in the “Service” mode.

NOTICE

Prerequisite for the following action:

Prior to enabling the “Service” mode, the ctrlX device or the system has to be in a safe state!

Procedure

1. In the ctrlX OS web interface, navigate to the window “*Settings → Apps*”, see:
↪ [Window – “Apps”](#)
2. Ensure that the device in a safe state!
3. To switch the ctrlX device to the “Service” mode, click on the button  **Service mode**.
➔ Switching to the “Service” mode can take a moment.

5.15 ctrlX device – Enabling the Operating mode

Editing real-time data on the ctrlX device is started in the “Operating” mode. Working and modifying apps and their settings on the ctrlX device is locked in the “Operating” mode.

Procedure

1. In the ctrlX OS web interface, navigate to the window “*Settings → Apps*”, see:
↪ [Window – “Apps”](#)
2. To switch the ctrlX device to the “Operating” mode, click on the button  **Operation mode**.
➔ Switching to the “Operating” mode can take a moment.

5.16 Release cycle

5.16.1 Release cycle of apps and engineering software

As for smart devices, the functional scope of ctrlX CORE controls can be extended with apps and modified. Thus, the functionality corresponds to the requirements of the application.

Upon delivery, a ctrlX CORE control only includes apps required for the basic operation.

Bosch Rexroth releases new versions of apps and software at regular intervals. Thus, customers and developers can access new functionalities regularly.

Release cycle of apps and engineering software

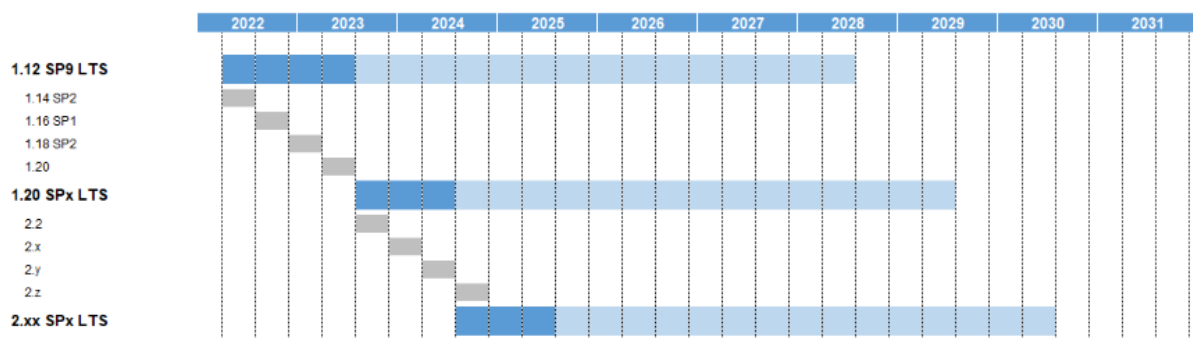


Fig. 4: Release cycle

Long-term support releases

LTS or "Long Term Support" releases are published once a year at the end of July. LTS versions were validated in the field and are recommended for most serial applications. They are maintained over several years as service packs (SPx) with bug fixes and security updates.

LTS versions are available via the ctrlX Store. Only a myRexroth user account is required for download. LTS releases are also available in the Device Portal.

Publications for early users

Releases for early users are published every 4 months between LTS versions. They provide the latest functions and are used for field validation. These releases are supported for 4 months and are not bugfixed or updated after this period. They are available for download until the functionality is integrated into the next LTS release.

The latest releases for early users are stored in an area with restricted access.

Security updates

Security updates are provided at regular intervals with the releases of the LTS versions and as part of service packs. In versions for early users, security updates are published when a vulnerability poses a high risk.

5.17 Creating own apps

5.17.1 Introduction and overview

Prerequisites to create own snaps

The Linux system required to create Ubuntu Core Snaps is Ubuntu 22.04, either server or desktop.

The ctrlX WORKS App Build Environment (with Ubuntu Server 22.04) is recommended.

Furthermore, internet access is required to install other software packages later on.

Depending on the programming language, additional packages might be required.

ctrlX OS supports the processor architectures amd64 and arm64. Thus, snaps should be provided for both architectures.

Further information

- ➔ [APP ZONE, SDK FOR CTRLX AUTOMATION](#)
- README.md:
Documentation on the setup of a tool chain and general documentations on each example
- ➔ <https://snapcraft.io/docs>:
Information on how to create a snap
- ➔ [GitHub](#):
Documentation and example projects on own snaps



To create real-time applications that integrate into the real-time task system of ctrlX OS (ctrlX CORE Scheduler), a special SDK is available on request. Please contact your account manager at Bosch Rexroth.

6 Working with the device

6.1 Users and rights

6.1.1 Introduction

The user and rights management ensures that the protected device functionality may only be used by authorized personnel and only with the rights specified in the respective right profile.

Further information

- ➔ [Chapter 6.1.2 User management – Functional principle on page 42](#)
- ➔ [Chapter 6.1.3 Users, groups, permissions, sessions on page 43](#)
- ➔ [Chapter 6.1.4 State upon delivery/default user on page 43](#)
- ➔ [Chapter 6.1.5 Logging users into the control on page 44](#)
- ➔ [Chapter 6.1.7 Changing the user password on page 44](#)
- ➔ [Chapter 6.1.8 Password forgotten on page 44](#)
- ➔ [Chapter 6.1.11 Configuring the user management on page 46](#)

6.1.2 User management – Functional principle

The user management is device-based. That is:

- All user management data (users, groups, permissions, etc.) is saved on the device
- Authentication/authorization as well as access restrictions are integrated into the device firmware and executed directly on the device

The user management of the control is configured using the WebClient. The following results:

- The engineering PC has to be connected to the device
- User data is directly retrieved from the device or written to the device
- There is never more user data in the memory of the PC than required for the current configuration

- No user data remains on the hard disk of the PC
- If user rights were changed while the user was logged in, the user has to log off and log in again to apply the changed permissions.

6.1.3 Users, groups, permissions, sessions

Users

Users are persons that can operate certain protected functions after logging into the device. On devices with a configured user management, a list of users who can log into the device with username and password is available.

Refer to [Chapter 13.2.19 Window – “Users” on page 305](#)

Groups

A group represents a selection of users with the same permissions with regard to a certain protected functional scope (e.g. engineering functions).

Refer to [Chapter 13.2.20 Window – “Groups” on page 306](#)

Permissions

A right allows the use of a certain protected function of the device. Permissions are assigned directly to groups as well as to users.



The permission profile of a user results from the sum of permissions of all groups the user is assigned to and its individual permissions.

Refer to [Chapter 13.2.21 Window – “Permissions” on page 306](#)



“Full access” permission

- The “Full access” right cannot be assigned to any group
- The “Full access” right can only be assigned to other users by users that already have the “Full access” right
- The properties of users that have the “Full access” right can only be changed by users that have the “Full access” right

Sessions

For each established connection to the control, a session with ID, creation timestamp and further configurable parameters are assigned to each user.

Refer to [Chapter 6.1.12 Conventions for login and session monitoring on page 46](#)

6.1.4 State upon delivery/default user

Upon delivery, the following default user is configured on the control:

Username	Password
boschrexroth	boschrexroth

The default user configured upon delivery has the permission to create and to configure more users.

NOTICE

Attention: The user “boschrexroth” has all permissions (i.e. the user is administrator)

The password of the default user has to be changed after the first login.

After commissioning the device, it is recommended to design a suitable user structure (including admin) and to disable the default user.

6.1.5 Logging users into the control

To log into the control, go to the ctrlX WORKS interface or use the web browser and enter the control IP address [“http://<ip address of the control>/login”]. The login screen of the control opens in both cases. Log in with the respective user name and password.

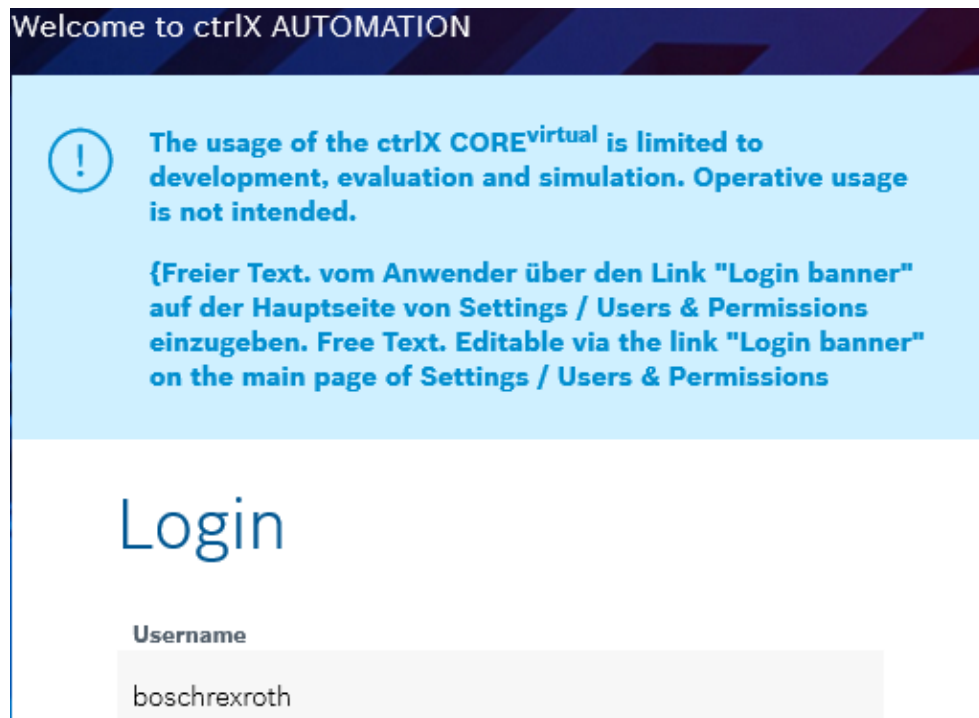
The logged in user is shown in the side navigation of the ctrlX OS web interface next to the icon .

Initial commissioning

Use the login data of the default user upon the initial commissioning, refer to [Chapter 6.1.4 State upon delivery/default user on page 43](#).

6.1.6 Login banner

A free text can be displayed in the banner of the “Login” dialog. It can be entered and changed in an editor, see [Chapter 13.4.14 Dialog – Login banner on page 352](#).



6.1.7 Changing the user password

To change the password of the logged in user, go to the “Change password” dialog, see [Chapter 13.4.11 Dialog – “Change password” on page 349](#).

6.1.8 Password forgotten

If the password is lost, a user has to specify a new password for this user in the user management. The user specifying the new password requires the permission “Manage users and groups”.



The user should change the newly specified password immediately if possible. It is recommended to use a password management software to minimize the effort caused by forgotten passwords.

If the password of the user with the permission “Manage users and groups” is lost, a hardware reset of the control is required!

After a hardware reset of the control, all users and passwords have to be assigned again!

NOTICE

In case of a hardware reset, the PLC project and the Motion configuration are also deleted on the control!
Note the information given in the online help.

6.1.9 Creating a new user

Proceed with the following steps:

1. In the side navigation, navigate to the window “*Settings → Users & Permissions → Users*”
2. In the window “Users”, click on the  button
 - ➔ The “Add user” dialog opens
3. Enter a username, select a password policy and enter a corresponding password. Then confirm the dialog with “Add user”
 - ➔ The new user is created and shown in the “Users” window in the table

Further information

- ➔ [Chapter 13.4.10 Dialog – “Add user” on page 348](#)
- ➔ [Chapter 13.4.12 Dialog – “Password policies” on page 349](#)

6.1.10 Creating a system user

General information

A “System User” is a specific, extended account for developers with “super user” access to the operating system resources.

A system user has to be requested individually for each ctrlX CORE device and then created on each device.

Contact the Bosch Rexroth Service for the required “Assertion” file providing them with the following information:

- Device serial number
- Desired user name
- Desired initial password
- Desired expiry date

Prerequisites

- If there is already a user with the same name on the device, creating a system user is denied and an error message is issued
- Different system users with different names can be created
- System users have an “expiry date” that can be selected individually

Characteristics of the system user

- Administrator rights on the ctrlX CORE device
- Member of the “sshuser” group

Creating a system user

To create a system user on the device, the generated file “Assertion” has to be loaded to the ctrlX CORE device. Proceed with the following steps:

1. In the side navigation, navigate to the window “*Settings → Users & Permissions → Users*”

2. ➤ In the window “Users”, click on the button 
 - ➔ The file explorer opens
3. ➤ Select the “Assertion” file in the file explorer and press “OK” to start the upload to the device
 - ➔ The system user is created on the device and shown in the “Users” window in the table: “<Username> (system user)”

Further information

- ➔ [Chapter 13.2.19 Window – “Users” on page 305](#)

6.1.11 Configuring the user management



To edit users, groups and rights, the user has to have the right “Manage users, groups and permissions”.

Upon control delivery, the user has the right “boschrexroth”.



“Full access” permission

- The “Full access” right cannot be assigned to any group
- The “Full access” right can only be assigned to other users by users that already have the “Full access” right
- The properties of users that have the “Full access” right can only be changed by users that have the “Full access” right

Opening the user management

To open the user management, there has to be a connection to the control, as device-specific information on users, groups and rights is stored on the control and not in the project. Open the main site of the user management via the side navigation of the ctrlX OS web interface, see ➔ [Chapter 13.2.18 Window – “Users & Permissions” on page 304](#).

6.1.12 Conventions for login and session monitoring

The failed attempts are monitored when logging into the control. The duration and the number of sessions per user and the activity during the session are monitored. A user with the right "Manage users, groups and permissions" can specify the session conventions in the “Session Policies” dialog, refer to ➔ [Chapter 13.4.13 Dialog – “Session policies” on page 351](#).

The following can be configured:

- Logout after a certain time (in minutes) of inactivity
- Logout after a certain time (in minutes) without further condition
- Maximum number of sessions per user
- Maximum number of subsequent failed logins without waiting time
- Waiting time (in seconds) after a specified number of subsequent logins (default value: 300)

Further information

- ➔ [Chapter 6.1.1 Introduction on page 42](#)
- ➔ [Chapter 6.1.3 Users, groups, permissions, sessions on page 43](#)

6.1.13 Username and password conventions

It is monitored in the input fields of the user management whether the conventions for names and passwords are met. A user with the right "Manage users, groups and permissions" can specify the password conventions in the “Password Policies” dialog, refer to ➔ [Chapter 13.4.12 Dialog – “Password policies” on page 349](#).

The following can be configured:

- Minimum password length (default value is 12)
- Minimum number of lower case letters (default value is 1)
- Minimum number of upper case letters (default value is 1)
- Minimum number of digits (default value is 1)
- Minimum number of special characters
- Username may/must not be part of the password
- The password history can be used to determine the number (10 max.) of the unique new passwords that have to be used by the user before using the old password again. Default value is 5, the function is disabled with the value 0,

Spaces in the password are valid.

The conventions for user and group names are set as follows:

- The name has to be unique
- Minimum length = 1
- Maximum length = 32
- First character = Small letters (a-z) and underscore (_)
- More characters = Small letters (a-z), numbers (0-9) and special characters (- _) in the name.
- A special character (\$) is valid as last character in the name
- Spaces in the name are invalid

When validating the valid conventions, a note is displayed in the input window

Further information

- ➔ [Chapter 6.1.1 Introduction on page 42](#)
- ➔ [Chapter 6.1.11 Configuring the user management on page 46](#)

6.2 Network connection

6.2.1 General information on network interfaces

The network settings of the ctrlX device are managed in the “Network interfaces” window, see:

➔ [Window – “Network interfaces”](#)

The available or configured network interfaces are shown as tiles in the window.

Click on one of the tiles to open an editor to configure the interface settings, see:

➔ [Editor – “Network interfaces”](#)

Important notes for use

Supporting jumbo frames

Note that the XF51 interface on ctrlX CORE devices does not support jumbo frames.

To use jumbo frames, either use the XF10 interface or a USB Ethernet adapter that supports jumbo frames.

Note that this problem can also occur on other hardware platforms. Therefore, check the specifications and functions of your hardware to ensure that jumbo frames are supported if you require this function.



Jumbo frames

Jumbo frames are a network technology that enables larger data packages to be transmitted over a network. Data packages in Ethernet networks are normally transmitted with a size of 1500 bytes. However, jumbo frames allow data packages with a size of up to 9000 bytes to be transmitted. It is important to note that all devices in the network have to support jumbo frames to use them effectively. Otherwise, the data packages are reduced to the standard size of 1500 bytes, which can result in inefficient transmission.

Jumbo frames are used in various application areas, for example:

- Video surveillance systems
- Data storage and data transfer
- Virtualization
- Cluster computing

Jumbo frames improve the transmission of high-resolution video data, increase efficiency when transferring large files and optimize network performance in virtualized environments and cluster computing systems.

6.2.2 Configuring IP settings

The settings for IPv4 and IPv6 can be made in the editor of the respective interface, see:

➔ [Editor](#) – “Network interfaces”

Only the settings for DHCP and IP addresses are available in the default view in the editor.

Click on “Show advanced settings” for further settings like DNS server, gateway, routes and link local (only for IPv4).

6.2.3 Automatic/manual IP configuration

Upon delivery of the ctrlX device, interfaces are already preconfigured and set to obtain IP settings automatically from a DHCP server.

In addition to the automatic configuration, the IP can also be configured manually, e.g. to define additional, static IP addresses with an own DNS server, gateway and routes.

1. ➔ For manual IP configuration, open the editor of the relevant interface, see:

➔ [Editor](#) – “Network interfaces”

2. ➔ Go to tab IPv4 or IPv6

3. ➔ Click on [⚙]

➔ The dialog to set the DHCP configuration opens, see:

➔ [Dialog](#) – “DCHP settings”

4. ➔ Make the settings and confirm the dialog

6.2.4 Defining routes

Select “Routing” to route the communication to the set network segments using a defined gateway, see:

➔ [Editor](#) – “Network interfaces”

If routing entries exist, the respective IP address configurations (IPv4/IPv6) should already be present. Otherwise, make the following entries in the editor for routing:

- Target(address and subnet mask)
- Gateway
- Metric



If routes are defined for an IP address range (IPv4 or IPv6), no gateway should be entered into the Ipv4 or Ipv6 settings, as routing conflicts can be caused.

6.2.5 Bridging interfaces

NOTICE

Bridging interfaces can cause a connection loss to the control!

Thus, before saving the changes, check the IP settings of the bridge to ensure that it is still reachable after the modification.

To bridge the interfaces, proceed as follows:

1. Open the “Network interfaces” window, refer to:
→ [Window – “Network interfaces”](#)
2. In the header of the window, click on [+] and select “Add Bridge” from the submenu.
 - ➔ A new network bridge is created in the system and displayed in the window.
 - ➔ Creating a network bridge may take a moment.

Configuring network bridges

After creating a new network bridge, the network interfaces to be bridged have to be specified.

To configure bridges, proceed as follows:

1. Select the bridge in the “Network interfaces” window.
 - ➔ The network interface editor is displayed on the side.
2. In the editor, go to the “Bridge” tab.
 - ➔ The available network interfaces are displayed.
3. Select the network interfaces to be bridged.
4. Save the configuration.
 - ➔ Note that the connection to the ctrlX device can be temporarily lost when setting up the network bridge.
 - ➔ This operation can take a few minutes.
 - ➔ If the connection to the ctrlX device cannot be set up again automatically or by reloading (F5) the browser page, try to access the device via the default IP: **192.168.2.1**
 - ➔ Once set up, the bridge status is displayed via the bridge icon (connected/not connected).

6.2.6 Option for dynamic determination of the maximum package size (PMTUD)

The Path MTU Discovery (PMTUD) option dynamically determines the maximum package size that a network path can process.

When this option is enabled, the sender starts with an assumed MTU, adapts based on feedback (ICMP "Fragmentation Needed" messages) and iterates until it finds the optimal size to minimize data fragmentation for an efficient data transmission.

To enable Path MTU Discovery, proceed as follows:

1. Open the “Network interfaces” window, refer to:
→ [Window – “Network interfaces”](#)

2. ➤ To open the network settings, go to the [⚙️] button in the header of the window.
➔ The “Network settings” dialog opens, see:
➔ [Dialog – “Network settings”](#)
3. ➤ Open the “Path MTU Discovery” tab.
4. ➤ Enable the “Enable Path MTU Discovery” option and save the setting.
➔ The option is enabled.

6.2.7 Limiting network bandwidth

The data flow in a network can be regulated by limiting the network bandwidth to prevent network overload and to ensure a fair distribution of bandwidth among users or applications.

The following instructions describe how to set up a network bandwidth limit in the web interface of the ctrlX device.

1. ➤ Open the “Network interfaces” window, refer to:
➔ [Window – “Network interfaces”](#)
2. ➤ To open the network settings, go to the [⚙️] button in the header of the window.
➔ The “Network settings” dialog opens, see:
➔ [Dialog – “Network settings”](#)
3. ➤ Click on the [⊕] button in the “Bandwidth management” tab.
➔ The “Add a bandwidth” dialog opens, see:
➔ [Dialog – “Add a bandwidth”](#)
4. ➤ Enter the required details and values into the dialog and select the network interfaces to which the bandwidth limitation is to be assigned.
5. ➤ Confirm the dialog
➔ The bandwidth limitation is set up and displayed in the “Network settings” dialog.
6. ➤ Save the network setting.

6.2.8 Link local

This option is only available for IPv4 and enables an additional IP address in the range 169.254.0.0/16.

The address is assigned automatically and should only be used for ad-hoc and isolated networks.

6.2.9 Connecting to WiFi network

If a supported WiFi adapter is connected to the ctrlX device (e.g. via USB), the WiFi interface of the adapter is displayed in the “Network interfaces” window. The WiFi interface is configured in the “WLAN” tab of the interface editor. Click on the WiFi interface to open it.

It cannot be scanned for networks in the current version. The network data has to be entered manually.

To add a WiFi network, click on the + button. A dialog opens to enter network access data.

Supported encryption methods:

- No encryption
- WPA2/EAP-PEAP

- WPA/WPA2 Personal
- WPA/WPA2 Enterprise (TLS)

Note the following: ➔ [Configuring security settings for wired interfaces](#)

Important notes and restrictions:

- Radio networks are not supported if the special character "/" is used in the network name
- The control always connects to the strongest network!
To connect a control to a specific network, delete all entered networks before
- If the ctrlX device is connected to a WiFi network, it is shown on the tile of the WiFi interface and in the "WLAN" tab

6.2.10 Setting up WLAN hotspot

If a supported WLAN adapter is connected to the ctrlX device (e.g. via USB), the WLAN interface of the adapter is displayed in the "Network interfaces" window.

In the "WLAN" tab of the interface editor, the mode can be switched to "Hotspot" and access data can be configured.

Currently, only the WPA/WPA2 Personal security mode is supported.



6.2.11 Adding and configuring VLAN interface

VLAN is an acronym for "Virtual Local Area Network", a method of dividing a physical network into multiple virtual networks for the purpose of isolating data traffic, improving network security and optimizing network performance. VLANs are used in large networks to logically group devices, even if they are geographically dispersed.

To add a VLAN interface, proceed as follows:

- Open the "Network interfaces" window, refer to:
➔ [Window](#) - "Network interfaces"
- In the header of the window click on [+] and select "Add VLAN interface" from the submenu.
➔ The "Add VLAN interface" dialog opens.
- Select the parent interface and enter a VLAN ID.
- To close the dialog, click on the "Close" button.
➔ The newly added VLAN interface is shown in the "VLAN" category



The "VLAN ID" is used as a digital identifier that divides network traffic into different virtual networks within the same physical infrastructure. It is important that the network infrastructure (switches, routers, etc.) supports VLAN IDs, but the individual end devices do not have to be VLAN-capable as long as they are connected via compatible network hardware.

Configuring VLAN network priorities

The VLAN interface allows to map the network priorities of sent or received data. In general, there are two types of network priorities:

- **Socket priorities**

Used to prioritize data within the operating system (Linux, Ubuntu Core 22): see here:

➔ <https://man7.org/linux/man-pages/man7/socket.7.html>

- **VLAN priorities**

Used to prioritize data within a virtual network (VLAN) based on the priorities in the VLAN tag (PCP field).

To configure a network priority for the VLAN interface, proceed as follows:

1. In the “Network interfaces” window, click on the VLAN whose network priorities are to be configured, see:

➔ [Window – “Network interfaces”](#)

➔ The network interface editor is shown.

2. In the editor, open the “VLAN” tab in the editor.

3. Select one of the following network priorities from the drop-down bar:

- **Ingress**

For incoming data, the VLAN priorities are mapped 1:1 to the socket priorities (ingress-qos-map 0:0 1:1 2:2 3:3 4:4 5:5 6:6 7:7).

- **Egress**

For outgoing data, the socket priorities are mapped 1:1 to the VLAN priorities (egress-qos-map 0:0 1:1 2:2 3:3 4:4 5:5 6:6 7:7).

- **Ingress and Egress**

Combination of the "Ingress" and "Egress" options.

- **None**

The network priorities are not assigned among each other and thus there is no prioritization of data. All data is treated equally.

4. Save the setting.

➔ The network priority is applied to the VLAN interface.

6.2.12 Configuring security settings for wired interfaces



For WLAN interfaces, the security settings are made in the individual WLAN networks.

Enable the “IEEE 802.1X / EAP-TLS security” option in the “Security” tab of the interface editor to ensure that the ctrlX allows only authenticated connections to an interface, see:

➔ [Window – “Network interfaces”](#)

Only IEEE 802.1X with the authentication method EAP-TLS is currently supported as access control. When setting up the access protection, a form is displayed to enter the required access data. Certificates and keys have to be selected for example in the respective selection fields. If the respective certificates or keys were not uploaded to the ctrlX device, click on the "Upload" button to the right of the respective selection fields.

Also refer to [Certificates – Introduction](#)

6.2.13 EoE configuration (Ethernet over EtherCAT)



The following section is only relevant if the EtherCAT Master is used together with the EoE communication.

The EoE communication uses a virtual network adapter on the ctrlX CORE device. The system adds the network adapter automatically to the controls if the EoE communication is enabled in the EtherCAT master configuration of the ctrlX I/O Engineering tool.

The settings of the network adapter are only made in ctrlX I/O Engineering!



In the ctrlX OS web interface, the EoE network adapter is shown in the “Network interfaces”. Its name is „eoe0“. The advanced address settings may not be changed in IPv4 and IPv6!

6.3 ctrlX licenses

6.3.1 Licenses – Overview



The installation and use of ctrlX apps are subject to license.

For more information on the topic "Licensing", refer to the "HOW-TO" section of the [ctrlX Automation website](#).

As for smart devices, the functional scope of ctrlX CORE controls can be extended with apps and modified. Thus, the functionality corresponds to the requirements of the application.

Upon delivery, a ctrlX CORE control only includes apps required for the basic operation.

In the window "Licenses", all licenses of the apps already loaded on the ctrlX CORE are shown in the overview list. These are those apps upon delivery that were already purchased when ordering the ctrlX CORE with the device, see [Chapter 13.2.6 Window – "Licenses" on page 284](#).

App responses when the license is missing/expired

When started, check apps that require a license whether there is a matching license. Licenses can be added or removed at runtime (e.g. by expiry after a certain time). The license status of the device is thus not static and can be changed dynamically. Thus, the apps query the licenses cyclically. The query interval depends on the license status. An app will query the status more frequently if no license is available.

It is generally checked for the license when started and at runtime of the app. It is tried to start a functionality via the app website, but no license is available for this functionality. The user is thus redirected to a site informing on the missing license. A link allows to directly go to the specific application in the App Store and to purchase the license. Another option is to open the "Licenses" site using a second link. The licenses available on the device are shown.

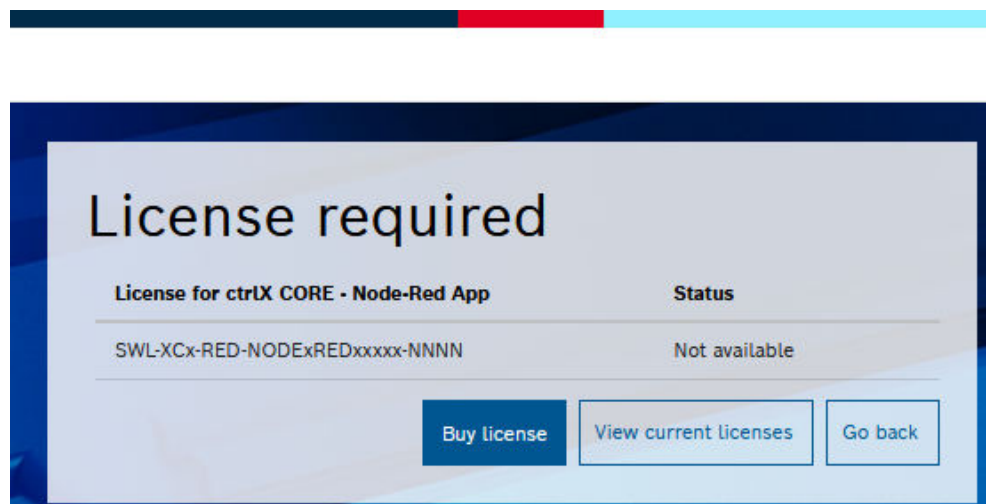


Fig. 5: License required

If the license expires while an app is running, this is detected upon the cyclic license request. The app is not stopped. If a license is missing, the responses of apps are different.

Some examples:

- Firewall app: Critical security problems can be caused if a license is not available anymore when the firewall configuration is disabled. The configuration thus remains active. However, the Firewall app cannot be opened anymore to view or modify the configuration for example. When opening the site, the user is redirected to the information site
- VPN Client app: If there is no license, the user is redirected when opening the site. It is not possible to start new connections or to create a configuration. Existing VPN connections are not closed.
- Motion app: The app can be opened. Axes or kinematics cannot be created. The following message is entered into the logbook:
090A0004 "Waiting for motion license"
"Waiting for motion basic license"
0C560257 "AddInfo: Lic: SWL-XC*-MOT-STD MOTIONxxx-NNN; vers: 1.0
If the "Motion basic license" is available, more than four axes can be configured. When it is tried to switch the Motion to RUN, the message "Motion switching failed" is issued. The following message is entered into the logbook:
091F0050 "License for current configuration missing"
"Configuration invalid – License missing"
0C56020F "AddInfo: Axes allowed: xx; Axes configured: yy
Switching the Motion to RUN is impeded.

This chapter provides an overview on the available functions, apps and licenses.

Available ctrlX apps and licenses

Name	Description	License of the type code	Part number
ctrlX OS License - EtherCAT Master Basic	This app allows to use the master interface for the real-time Ethernet system EtherCAT.	SWL-XC*-ECM-ETHER-CATMAS**-BANN	R911400508
ctrlX OS License - PLC Basic (02VRS+)	The PLC app is based on the high-performance PLC runtime system of CoDeSys V3 and extends it with comprehensive technology functions for a faster engineering. The basic license for the PLC app can be functionally extended via in-app licenses.	SWL-XC*-PLC-PLC*XOS*****-BANN	R911421866
ctrlX OS License - Motion Standard 4 Axes	Basic Motion functions for single-axis positioning for up to 30 axes per controller. Up to four interpolating axes per kinematics. The functional scope can be extended with optional technology packages for robotics and handling or systems with axis synchronization requirements.	SWL-XC*-MOT-STD MOTION****-NNNN (for 4 axes)	R911400503
ctrlX OS License - Motion Standard 10 Axes		SWL-XC*-MOT-STD MOTION**10-NNNN (for 10 axes)	R911414460

Name	Description	License of the type code	Part number
ctrlX OS License - OPC UA Server Standard	The OPC UA Server app provides a standardized machine-to-machine (M2M) communication. Consistent services allow to access all device data such as PLC, Motion Control and system information.	SWL-XC*-UAS-OPCUA-SERVER**-NNNN	R911397816
ctrlX OS License - OPC UA Client	The OPC UA Client app provides a standardized machine-to-machine (M2M) communication. Use the OPC UA Client to connect the OPC UA Server. The interface is standardized. Thus, the OPC UA Client can access every OPC UA Server and retrieve the provided data.	SWL-XC*-UAC-OPCUA-CLIENT**-NNNN	R911401476
ctrlX OS License - Firewall	The Firewall app can be used to restrict the communication between the machine and the network individually and thus to protect unintended access from viruses and Trojans. The user-friendly configuration of rules allows very fast changes without programming knowledge.	SWL-XC*-FRW-FIRE-WALL****-NNNN	R911400505
ctrlX OS License - VPN Client	The VPN Client app allows to encrypt communication channels for a secure remote access and a secure data transfer from and to a machine. The basic scope provides the remote access and the encryption of up to two channels.	SWL-XC*-VPN-VPNCLIENT****-NNNN	R911400506
ctrlX OS License - Python Runtime	The Python Runtime app provides the runtime environment to execute the Python code on a ctrlX CORE device. The Python Runtime app can load any Python scripts to the ctrlX CORE device.	SWL-XC*-PYR-PYTHON-RUNTIME-NNNN	R911404540
ctrlX OS License - Oscilloscope	The Oscilloscope app cyclically records and represents ctrlX Data Layer signals.	SWL-XC*-OSC-STDOSCI*****-NNNN	R911406327
ctrlX OS License - Node-RED	Node-RED is a flow-based programming application to “cable” hardware devices, APIs and online services. The Node-RED app is the Node-RED Framework adapted for the Bosch Rexroth ctrlX AUTOMATION.	SWL-XC*-RED-NODE*RED*****-NNNN	R911399092

Name	Description	License of the type code	Part number
ctrlX OS License - IDE Textual Code (Integrated Development Environment)	The IDE app provides different engineering environments (coding, debugging, etc.) used to create program sequences for the ctrlX CORE. Textual Coding is included in the basic scope of the app and provides a web-based engineering environment with a native ctrlX CORE connection to create Python scripts or to edit files in the active control configuration. The IDE app can be extended functionally via the in-app license Visual Coding .	SWL-XC*-IDE-TEXTUAL-CODE**-NNNN	R911409788
ctrlX OS License - 3D Viewer	The 3D Viewer app can be used for a 3D kinematic simulation on ctrlX CORE.	SWL-XC*-3DV-3DVIEWER*****-NNNN	R911413191
ctrlX OS License - PROFINET Device	The PROFINET Device app enables the "device" interface for the real-time Ethernet system PROFINET.	SWL-XC*-PND-PROFINETDEV**-NNNN	R911412495
ctrlX OS License - Telegraf	Use the Telegraf app to collect, process and aggregate information such as metrics, events and logs and to write them to the target systems, such as the InfluxDB.	SWL-XC*-TSA-TELEGRAF*****-NNNN	R911416258
ctrlX OS License - Modbus TCP	The Modbus TCP app allows to connect the ctrlX device to third-party devices using Modbus TCP. Data can be read and written. Data of the Modbus devices are available for all apps on the ctrlX CORE via the real-time data layer. The Modbus TCP app supports the following functions: • Communication with the Modbus network • Configuration of the Modbus device • Management of the Modbus device	SWL-XC*-MBT-MOD-BUSTCP****-NNNN	R911413192

Name	Description	License of the type code	Part number
ctrlX OS License - Container Engine	The ctrlX Container Engine app allows to use the Docker® technology to independently provide applications in different languages in separate containers. Operating principle: Docker® can pack an application and its dependencies as image into a virtual container that can be executed on devices with different operating systems such as Linux or Windows. The ctrlX Container Engine app hosts these applications on the ctrlX device and allows to communicate via defined interfaces. The Container Engine app was created by Rexroth using the source code provided by Docker, Inc. and it is not supported or published by Docker, Inc. For detailed information on the Docker® technology, refer to: ↗ https://www.docker.com/	SWL-XC*-DOE-DOCKER-ENGINE*-NNNN	R911409943
ctrlX OS License - OPC UA Pub/Sub	The ctrlX OPC UA Pub/Sub app allows a standardized, manufacturer-specific C2C communication between control devices or the communication between ctrlX devices and MES or data acquisition systems. Operating principle: Use the OPC UA Pub/Sub app to publish or write subscribed data from the ctrlX DataLayer of the ctrlX devices in real-time. Thus, it can be exchanged between different devices supporting the OPC UA Pub/Sub app.	SWL-XC*-UAP-OPCUA-PUBSUB**-NNNN	R911416700
ctrlX OS License - InfluxDB	The ctrlX InfluxDB App allows to install and operate a time-based database system to a ctrlX CORE device with an external memory. Use the InfluxDB App to filter, visualize and analyze time series from different data sources via queries.	SWL-XC*-IDB-INFLUX*DB****-NNNN	R911417120

Name	Description	License of the type code	Part number
ctrlX OS License - Service Indicator	The ctrlX Service Indicator App enables the calculation and display of the failure probability of a drive controller and motor. Using QR codes on Rexroth components, which can be scanned with the Digital Service Assistant smartphone app, it can directly be communicated with the Bosch Rexroth Service.	SWL-XC*-SIN-SIN*****-BANN	R911416259
ctrlX OS License - IoT Dashboard	The IoT Dashboard app allows the visualization of data in dynamic interactive dashboards. Features: • Panels with heatmaps, histograms, graphs and geo maps • Notifications; easy to create and to configure • Transformations; allow renaming, summarizing, combining or performing calculations across all data streams • Remarks to comment diagrams with comprehensive events from various data sources • Panel editor; provides an easy option to configure, customize and explore panels The IoT Dashboard app is based on the open source software Grafana. Grafana™ is a registered trademark/owner of Grafana Labs, which is not associated with the ctrlX IoT Dashboard app and does not support it.	SWL-XC*-GDB-GRANA*DB**-NNNN	R911418579
ctrlX OS License - Key Value Database	The Key Value Database app allows to create, use and store any data in the ctrlX Data Layer in customized persistent nodes. Example: Use the "create" or "write" methods, new Data Layer addresses can be created in existing Data Layer structures, for example in existing app nodes. Data in the new data layer addresses can be updated, deleted and retrieved by other apps. Data is saved and retained even after a ctrlX device restart.	SWL-XC*-KVD-KEY-VALUE*DB**-NNNN	R911418610

Name	Description	License of the type code	Part number
ctrlX OS License - G-Code Runtime	The G-Code Runtime app provides the runtime environment to execute CNC programs in G-code (DIN 66025-1 / ISO 6983-1) on ctrlX CORE devices.	SWL-XC*-GCO-GCOR- UNTIME*-BANN	R911420238
ctrlX OS License - Model Connect	Model Connect Basic (Education) Target Runtime for model-based generated control code from Simulink®. License for non-commercial use.	SWL-XC*-MOC-SIMU- LINK*EDU*-NNNN	R911421261
	Model Connect Basic Target Runtime for model-based generated control code from Simulink®. Standard license with limited functionality: limited inputs/only "free-running" (not synchronized with Scheduler / IOs...)	SWL-XC*-MOC-SIMU- LINK*****-BANN	R911421259
	Model Connect Performance Target Runtime for model-based generated control code from Simulink®. High-end license with unlimited functionality: Support for unlimited inputs/bundle mechanism (real-time synchronization with ctrlX Scheduler).	SWL-XC*-MOC-SIMU- LINK*****-PRNN	R911421260

Name	Description	License of the type code	Part number
ctrlX OS license - System Overview	For a top-level application overview, the System Overview app (SOV) provides a general system topology of the connected components and their basic status/ diagnostic information. Installing the System Overview app adds two additional screens to the diagnostic menu of the ctrlX CORE web user interface: • "System overview" (tree structure) Provides quick diagnostics and a system overview: - Complete overview on all connected components. - Considering the ctrlX configuration, e.g. assignment of axes to the kinematics. • "System information" (list style) Providing the following information: - Hardware and software components used. - Additional information such as type codes, part numbers, revision numbers. The app also provides the following functions: • Export of system information as file for error analysis and compatibility check. • Links to other apps for further diagnostics, maintenance and commissioning.	Not subject to license / installation free of charge	R911420592
ctrlX IPC License - Cam Designer	Only for use on an engineering PC Tool for easy and convenient creation of electronic cams. It is used to configure FlexProfiles. An event editor and a formula editor, as well as a large number of motion laws are included.	SWL-PC*-MOT-CAMDE-SIGNER**-NNNN	R911421308

Name	Description	License of the type code	Part number
ctrlX OS License - Display	Support of the display port on the ctrlX CORE hardware (ctrlX COREplus X5 / X7) or other ctrlX OS third party devices. • Display of the ctrlX OS web UI interface on a display. • Configuration of the ctrlX OS via the web UI. • Configuration of a web URL, displayed in kiosk mode.	Not subject to license / installation free of charge	
ctrlX OS License - Terminal	“Terminal” is an app for advanced users and developers that can be used to communicate with the operating system via a command line interface (SSH access). • SSH access to the ctrlX OS operating system. • Access to system Linux logs. • Reboot the device. • Access when Device Admin is hanging.	SWL-XC*-TER-TERMINAL*****-NNNN	R911425351

Available ctrlX in-app licenses

In-app licenses can enable the options of an already installed app.

Name	Description	License of the type code	Part number
Runtime in-app option: Data Layer Access NRT	The option " Data Layer Access NRT " allows to integrate own individual apps on the ctrlX CORE device. The extension allows to access the ctrlX Data Layer of the control to exchange data in non-real-time using the most common software protocols, e.g. REST HTTPS, OPC UA.	SWL-XC*-RUN-DLAC-CESSNRT**-NNNN	R911401462
ctrlX OS License - Motion Cartesian (add-on)	The option " Cartesian Kinematics " extends the Motion app and allows to use the functionality for Cartesian handling applications.	SWL-XC*-MOT-CARTESIAN****-NNNN	R911400509
ctrlX OS License - Motion additional Axis (5-pack / add-on)	Five more axes can be added with the option " 5 Additional Axes ".	SWL-XC*-MOT-AXIS*****-NNS5	R911413186
ctrlX OS License - Motion Electronic Gear (add-on)	Motion functionality extended by an electronic gear with axis synchronization.	SWL-XC*-MOT-ELEC-GEAR*****-NNNN	R911417992

Name	Description	License of the type code	Part number
ctrlX OS License - Motion Delta & Scara Kinematics (add-on)	This option extends the "Basis" Motion functionality by additional motion functions for applications with level 2 kinematics (e.g. Delta and Scara kinematics): • coordinated path motions • Jog motion for the entire kinematics	SWL-XC*-MOT-KIN*LEVEL2***-NNNN	R911421232
ctrlX OS License - Motion Articulated Kinematics (add-on)	This option extends the "Basis" Motion functionality by additional motion functions for applications with level 3 kinematics (e.g. articulated kinematics): • coordinated path motions • Jog motion for the entire kinematics	SWL-XC*-MOT-KIN*LEVEL3***-NNNN	R911421233
ctrlX OS License - Motion Cam (add-on)	This option extends the "Basis" Motion functionality by cams.	SWL-XC*-MOT-CAM*****-NNNN	R911417993
ctrlX OS License - Motion FlexProfile (add-on)	Extension of the Motion by Flex-Profile with event- and time-controlled cams, which can be used to implement complex motion sequences.	SWL-XC*-MOT-FLEX-PROFILE**-NNNN	R911417994
ctrlX OS License - IDE Visual Code (add-on)	Visual Coding provides a programming framework to create programs based on visual elements, Python Script and Java Script.	SWL-XC*-IDE-VISUAL-CODE***-NNNN	R911409789
ctrlX OS License - PLC Standard (02VRS+ / add-on)	The ctrlX PLC Standard license entitles for the use of the IEC 61131 PLC runtime environment with support of 1x CPU core, multiple PLC tasks and 1 kByte I/O data on the ctrlX CORE and on other ctrlX OS devices.	SWL-XC*-PLC-PLC*XOS*****-NNNN	R911421867
ctrlX OS License - PLC Advanced (02VRS+ / add-on)	The ctrlX PLC Advanced license entitles for the use of the IEC 61131 PLC runtime environment with support of multiple CPU cores, multiple PLC tasks and 2 kByte I/O data on the ctrlX CORE and on other ctrlX OS devices.	SWL-XC*-PLC-PLC*XOS*****-ADNN	R911421868
ctrlX OS License - PLC Performance (02VRS+ / add-on)	The ctrlX PLC Performance license entitles for the use of the IEC 61131 PLC runtime environment with support of multiple CPU cores, multiple PLC tasks and a maximum of I/O data on the ctrlX CORE and on other ctrlX OS devices.	SWL-XC*-PLC-PLC*XOS*****-PRNN	R911421869

6.4 Managing certificates and keys

6.4.1 Introduction

Overview

The Certificates & Keys functionality is the interface for the user to manage client certificates, server certificates, certificate revocation lists and keys for the services available on the device and to load them to the device or delete them from there.

The revocation lists can be used by apps to check certificates.

Use the "Certificate Management" protocol to automatically renew the validity of certificates loaded to the ctrlX OS device before they become invalid. The connections to the respective PKI servers can be specified in the certificate management, see ➔ [Chapter 13.2.17 Window – “PKIs” on page 302](#).

Mittels Simple Certificate Enrollment Protocol (SCEP) can be used to request certificate enrollments from PKIs that support SCEP using private keys generated on the device.

The following apps provided on the ctrlX CORE upon delivery have the certificates and/or keys included:

- Web server
- Remote logging
- SSH

After installing additional apps, they are automatically shown on the overview site of the certificate and & key management.



Users to whom read/write permissions were assigned for an app can implicitly receive permissions for the certificate and keystore of this app if the app supports this function, see ➔ <https://boschrexroth.github.io/ctrlx-automation-sdk/package-assets.html#certificate-stores>

Special features of the web server certificates

To ensure a secure connection to the web server via the IPv4 address, the self-signed certificate "webserver_cert.pem" is provided on the device with the respective private key "webserver_key.pem" upon delivery of the ctrlX CORE. They are shown in the tabs "Certificates" and "Keys" of the web server.



This certificate and the respective private key may neither be renamed nor deleted. If either this certificate or the key is deleted or renamed, they are newly generated upon the ctrlX CORE restart.

The self-signed certificate "webserver_custom_cert.pem" with the respective private key "webserver_custom_key.pem" is also stored on the ctrlX CORE. These are required to set up a connection to the device via the "Common name", e.g. mydevice.company.com (CN = mydevice.company.com).

The certificate and the key are also newly generated if they are renamed or deleted. It is possible to replace them by an individual certificate with the respective key if the device is running in an infrastructure with a certificate authority (CA).



The file names of the individual certificate and the key have to have their original names.

Exchange of web server key and certificate:

To obtain an individually signed certificate provided by a CA, generate a "Certificate signing request" using the available private key ("webserver_custom_key.pem") and send it to the CA. Another option to generate a new key is "Generate key" and using it to create the "Certificate signing request". The generated key may not be "HSM-secured".



Then load the individually signed certificate received from the CA to the ctrlX OS device. The original certificate and the original key can be deleted. Rename the file name of the new individual certificate to its original name ("webserver_custom_cert.pem") before or after its upload to the control. This also applies to a newly generated key ("webserver_custom_key.pem").

Important:

The ctrlX OS device may not be switched off between deleting the original certificate and the key and renaming the key and the certificate.

Further information

- ➔ [Chapter 13.2.16 Window – “Certificates & Keys” on page 301](#)
- ➔ [Chapter 13.5.5 Editor – Certificates & Keys on page 368](#)

Certificates

Categories

Certificates are managed in the following categories:

- “CA certificate”
Certificate given by the certificate authority
- “Own”
Self-created (server) certificates
- “Trusted”
Received (client) certificates classified as trusted
- “Rejected”
(Client) certificates classified as not trusted. After the check, they can be classified as trusted and managed in the “Trusted” category

Further information

- ➔ [Chapter 6.4.1 Introduction on page 63](#)
- ➔ [Chapter 13.2.16 Window – “Certificates & Keys” on page 301](#)
- ➔ [Chapter 13.5.5 Editor – Certificates & Keys on page 368](#)

Keys

Keys are managed with the following information:

Type:

- “Public”
Public key to encrypt data for the publisher of the key or to check a signature
- “Private”
Private key to decrypt data encrypted with an own public key, to generate digital signatures or for authentication purposes

Category:

- “Own”
Self-created keys
- “Trusted”
Included trusted keys
- “Rejected”
Included keys which are not trusted anymore

Status:

- “Available”
Status of the key is “Available”. It can be used
- “Generating”

- The key is currently generated on the ctrlX OS device
- “Encrypted”
 - The key is encrypted and on the ctrlX OS device. It cannot be used. The key can be decrypted on the ctrlX OS device after entering the respective username and password (action "decrypt"). It is then decrypted on the device and can be used

Further information

- ➔ [Chapter 6.4.1 Introduction on page 63](#)
- ➔ [Chapter 13.2.16 Window – “Certificates & Keys” on page 301](#)
- ➔ [Chapter 13.5.5 Editor – Certificates & Keys on page 368](#)

“Generate certificate signing request”

Creating the certificate signing request

To obtain a signed certificate issued by a Certificate Authority (CA), e.g. for a web service, a certificate signing request (CSR) is required. ctrlX OS provides a corresponding mechanism.

Open the window of the respective service you require a signed certificate file for under ctrlX OS side navigation “*Settings → Certificates & Keys*”.

In the “Keys” tab, by using a key of the “type” “Private”, click on the icon  “Generate certificate signing request” in the “Actions” column of the key and the dialog to enter the required information opens.



A certification signing request cannot be created for keys of the type "Private" created with the algorithm ED25519.

Once all required information has been entered into the “Generate certificate signing request” dialog, the button  is enabled. Click on the interface and the certification request is generated. A dialog with information and possible actions for the created CSR opens.

The CRS can be downloaded to the engineering PC or to the clipboard via the icons in the “Actions” column in “.pem” and “.der” format. During the download to the clipboard, the “.pem” format is used.

Downloading and copying to the clipboard is also possible at a later point in the “Show certificate signing requests” dialog of the key. The CSR can also be deleted by ctrlX OS device via this dialog.

Overview on the created certification requests

The CSRs created using a key of “type” “Private” are displayed in a dialog in the “Actions” column of the key when clicking on the icon  “Show certificate signing requests”. Use this dialog to download the certification request to the engineering PC, to copy it to the clipboard or to delete the request on ctrlX OS device.

Further procedure

Transfer the certificate signature request downloaded by ctrlX OS device as “.pem” or “.der” file to the selected certification authority. The authority will then provide you with the signed certification file for the service.

Go to ctrlX OS side navigation “*Settings → Certificates & Keys*” to download the certificate to the ctrlX OS device.

In the command bar, click on the “Certificates” tab in the window of the respective service on the  “Upload certificate file” icon. In the opened Explorer window of the engineering PC, navigate to the storage location of the certificate file and select the file for upload.

Further information

- ➔ [Chapter 6.4.1 Introduction on page 63](#)
- ➔ [Chapter 13.2.16 Window – “Certificates & Keys” on page 301](#)
- ➔ [Chapter 13.5.5 Editor – Certificates & Keys on page 368](#)

6.5 Working with configurations

6.5.1 Configurations – Introduction

The system provides a file storage to manage configurations of the apps. Apps can access this storage to load and change settings. In addition to the active configuration, archived configurations can be stored on the ctrlX device. To manage the file storages, go to the window “Manage app data”, see:

➔ [Window – “Manage app data”](#)

Configurations are typically used to manage project engineering information of the automation solution.

The parent backup functionality can be used to manage all ctrlX device settings. A backup contains the device settings and the installed apps in addition to the configurations described here, see:

➔ [Backup – Introduction](#)

Use cases:

- Loading a configuration
- Archiving the active configuration
- Backing up a configuration on an engineering PC
- Uploading a configuration from the engineering PC to the ctrlX device (restoration)

Required user rights

Loading a configuration seriously changes the functionality of the apps. Configurations can contain content that is worth protecting. For this purpose, there are permissions to view configurations "View configurations" and to manage configurations "Manage configurations", also refer to:

➔ [Window – “Users & Permissions”](#)

Note on further access options

The ctrlX device implements the WebDAV protocol, a standard interface that allows to access configuration files. This allows file access via WinSCP for example. For more information, go to the ➔ [CtrlX Software Development Kit website](#).

Working with configurations using the control interface

The window “Manage app data” is divided into two sections:

- **“Active” section**
The “Active” section shows settings currently loaded on the ctrlX device.
- **“Archive” section**
The “Archive” section contains settings as configuration files.

Working with configurations using the REST interface

ctrlX devices are provided with a "REST" interface allowing an external access to the configuration data of the ctrlX device.

It can be accessed as follows:

- Via the ctrlX OS - Solutions API, see also the help menu [?] in the header of the ctrlX OS web interface:
[?] *"API reference → ctrlX OS → Solutions API"*
- Via WebDAV, see: ➔ <https://boschrexroth.github.io/ctrlx-automation-sdk>

Related topics

- ➔ [Window – "Manage app data"](#)
- ➔ [Saving active app data to the archive of the ctrlX device](#)
- ➔ [Viewing active app data on file level](#)
- ➔ [ViewingLoading report](#)
- ➔ [Resetting active app data](#)
- ➔ [Uploading from the engineering PC to the archive of the ctrlX device](#)
- ➔ [Loading configuration from the archive to the ctrlX device](#)
- ➔ [Transferring configuration from archive to engineering PC](#)
- ➔ [Viewing configuration in the archive](#)
- ➔ [Editing description of a configuration](#)
- ➔ [Deleting configuration in the archive](#)

6.5.2 Working with configurations using the control interface

Transferring active app data to the engineering PC

While transferring to the engineering PC, active app data is first saved to the file storage on the ctrlX device, see also:

- ➔ [Backing up active app data](#)

The file storage is zipped into a ZIP archive and provided for download, see:

- ➔ [Downloading a configuration](#)

Use cases:

- To archive configurations ctrlX device-externally
- To transfer configurations to other ctrlX devices, see:
➔ [Loading app data from the engineering PC](#)

Handling instruction

1. ➔ In the ctrlX OS web interface, open the window "Manage app data" via the button [📁] in the header or via the navigation bar *"Home" → tile "Manage app data"*.
2. ➔ Click on the [...] button in the "Active" section to open the extended actions.
3. ➔ Click on the button [↓]

Related topics

- ➔ [Window – "Manage app data"](#)
- ➔ [Configurations – Introduction](#)

Loading app data from the engineering PC

Configurations saved on the engineering PC can be loaded to the apps of the ctrlX device and enabled.

Operating principle

While uploading to the ctrlX device, the configuration (ZIP file) is unzipped automatically. The included app data is then enabled on the ctrlX device.

- Uploading a ZIP archive.
- Loading apps with the uploaded files, see also: [↗ Loading app data](#)
- Refreshing the app data file storage.

Use cases:

- Transferring existing configurations to other ctrlX devices.
- Restoring a configuration on the ctrlX device.

Handling instruction

1.  In the ctrlX OS web interface, open the window “Manage app data” via the button  in the header or via the navigation bar “Home” → tile “Manage app data”.
2.  Click on the  button in the “Active” section to open the extended actions.
3.  Click on the button  Upload]
 - ➔ A standard dialog opens to select the ZIP archive from the file system of the engineering PC.
4.  Select the ZIP archive and confirm the dialog.
5.  Acknowledge the confirmation prompt.
Note: Acknowledging the confirmation prompt overwrites the active app data on the ctrlX device!
 - ➔ The configuration is uploaded, unzipped and enabled. If errors result while loading, a loading report with error causes is shown.



As an alternative to select the ZIP archive via the "Open" dialog, use drag & drop onto the “Active” area for the file.

Related topics

- [↗ Transferring active app data to the engineering PC](#)
- [↗ Window – “Manage app data”](#)
- [↗ Configurations – Introduction](#)

Resetting active app data

The app settings can be reset.

This action loads an empty configuration to the apps, also refer to:

[↗ Loading app data](#)

Use case:

- Restoring the initial state of the installed apps on the ctrlX device.

Handling instruction

1.  In the ctrlX OS web interface, open the window “Manage app data” via the button  in the header or via the navigation bar “Home” → tile “Manage app data”.
2.  Click on the  button in the “Active” section to open the extended actions.
3.  Click on the button  Reset] to load the empty configuration.

4. Acknowledge the confirmation prompt.

Note: Acknowledging the confirmation prompt overwrites the active app data on the ctrlX device!

- ➔ The empty configuration is enabled. If errors result while loading, a loading report with error causes is shown.

Related topics

➔ [Window – “Manage app data”](#)

➔ [Configurations – Introduction](#)

Viewing active app data on file level

The active app files can be viewed in the “Active” section of the “Manage app data” window, see:

➔ [Managing configurations](#)

Use case:

- Checking app data backed up to the ctrlX device.

Handling instruction

1. In the ctrlX OS web interface, open the window “Manage app data” via the button  in the header or via the navigation bar “Home” → *tile “Manage app data”*.

- ➔ Categories with current app data structured in subdirectories are shown in the section “Active” → “Content”.

2. There are two options to open and show the settings in the file view:

- Button  →  **File view**
All subdirectories of app data are displayed.
- Click on the respective category in the “Content” section
The specific file storage of an app is displayed.

Related topics

➔ [Window – “Manage app data”](#)

➔ [Configurations – Introduction](#)

Saving active app data to the archive of the ctrlX device

App data currently loaded on the ctrlX device can be backed up into the archive of the ctrlX device.

Operating principle

First, the active app data is backed up ("Synchronize app data" option), see also:

➔ [Backing up active app data](#)

The new or selected archive entry is created or updated based on the app data file storage.

Use cases:

- Backing up active app data to the archive of the ctrlX device
- Loading archived configurations to other ctrlX devices.

See:

- ➔ [Transferring configuration from archive to engineering PC](#)
- ➔ [Uploading from the engineering PC to the archive of the ctrlX device](#)
- ➔ [Loading configuration from the archive to the ctrlX device](#)

Handling instruction

1. ➔ In the ctrlX OS web interface, open the window “Manage app data” via the button  in the header or via the navigation bar “Home” → *tile “Manage app data”*.

➔ Categories with current app data structured in subdirectories are shown in the section “Active” → “Content”.

2. ➔ Click on the button  “Save to archive”] to save the app data to the archive of the ctrlX device.

➔ The dialog “Save to archive” opens.

3. ➔ Enter a name for the archive entry to be created into the dialog and confirm the dialog.

You can also select an already existing archive entry in the dialog. In this case, please note that the content of the original archive entry will be replaced while saving.

Related topics

➔ [Window – “Manage app data”](#)

➔ [Configurations – Introduction](#)

Uploading from the engineering PC to the archive of the ctrlX device

Configurations saved on the engineering PC can be uploaded to the archive of the ctrlX device.

Use cases:

- Transferring existing configurations to other ctrlX devices
- Restoring a configuration on the ctrlX device

Operating principle

While uploading to the ctrlX device, the configuration (ZIP file) is transferred to the device, unzipped automatically and then displayed in the “Archive” section of the “Manage app data” window, see:

➔ [Window – “Manage app data”](#)



Configurations in the archive of the ctrlX device do not affect the current settings of the apps of the ctrlX device.

Handling instruction

1. ➔ In the ctrlX OS web interface, open the window “Manage app data” via the button  in the header or via the navigation bar “Home” → *tile “Manage app data”*.

2. ➔ Click on the [ Uploads configuration] button in the “Archive” section.

➔ A dialog opens to select the ZIP archive from the file system of the engineering PC.

3. Select the ZIP archive and confirm the dialog.
 - ➔ The dialog “Upload configuration” opens.
In the dialog, select whether to overwrite an existing configuration or to create a new configuration with a new name.
 - Note:** While uploading to an existing configuration, note that the previously included settings are overwritten.
4. Select the option and confirm the dialog.
 - ➔ The configuration is loaded to the ctrlX device, unzipped and shown in the “Archive” section.



As an alternative to the selection of the ZIP archive via the "Open" dialog, use drag & drop onto the “Archive” area for the file.

Related topics

➔ [Configurations – Introduction](#)

Loading configuration from the archive to the ctrlX device

To enable an archived configuration on the ctrlX device, load the configuration to the ctrlX device, see:

➔ [Loading app data](#)

Use cases:

- Restoring app settings
- Loading modified archive

Handling instruction

1. In the ctrlX OS web interface, open the window “Manage app data” via the button in the header or via the navigation bar “Home” → *tile “Manage app data”*.
2. In the “Archive” section, navigate to the entry to be loaded to the ctrlX device and click on the button Loads the configuration].
 - ➔ A confirmation prompt opens.
3. Acknowledge the confirmation prompt.
 - ➔ The configuration is loaded to the ctrlX device and thus enabled automatically.



The archived configuration can also be loaded into the ctrlX device from the open file view.

Related topics

➔ [Loading app data from the engineering PC](#)

➔ [Window – “Manage app data”](#).

➔ [Configurations – Introduction](#)

Comparing configurations

The active configuration and configurations saved on the ctrlX device can be compared with each other.

The comparison provides a view in which the differences in the respective file directories are displayed.

To analyze the differences in detail, navigate to the source of the difference in the subdirectories and files.

Handling instruction

1. ➤ In the ctrlX OS web interface, open the window “Manage app data” via the button [📁] in the header or via the navigation bar “Home” → *tile “Manage app data”*.
2. ➤ Click on the [...] button in the “Active” section or in the “Archive” section to open the extended actions.
3. ➤ Click on the button [↔ Compare].
 - ➔ A selection box opens with all other configurations that can be used for comparison.
4. ➤ Select a configuration for comparison.
 - ➔ A window opens with the configuration comparison.



If you are in the file view of the configuration, you can also use the [↔] button in the header for a comparison for the configuration that is currently open.

Related topics

- ➔ [Configurations – Introduction](#)
- ➔ [Window – “Manage app data”](#)

Transferring configuration from archive to engineering PC

Saved configurations can be transferred to an engineering PC via download. The ctrlX device creates a ZIP archive that contains the configuration files, see:

- ➔ [Downloading a configuration](#)

Use cases:

- To archive configurations ctrlX device-externally
- To transfer configurations to other ctrlX devices, see:
 - ➔ [Uploading from the engineering PC to the archive of the ctrlX device](#)
 - ➔ [Loading configuration from the archive to the ctrlX device](#)

Handling instruction

1. ➤ In the ctrlX OS web interface, open the window “Manage app data” via the button [📁] in the header or via the navigation bar “Home” → *tile “Manage app data”*.
2. ➤ In the “Archive” section, navigate to the configuration to be transferred and click on the button [⬇ Downloads the configuration].
 - ➔ A dialog opens with options to create a .zip archive.

Related topics

- ➔ [Transferring active app data to the engineering PC](#)
- ➔ [Window – “Manage app data”](#)
- ➔ [Configurations – Introduction](#)

Viewing configuration in the archive

To view the app setting of archived configurations, go to the ctrlX CORE web interface, also refer to:

- ➔ [Managing configurations](#)

Use case:

- Checking app data before loading it to the ctrlX device.
- Comparing app data
- Editing app data

Handling instruction

1. In the ctrlX OS web interface, open the window “Manage app data” via the button  in the header or via the navigation bar “Home” → tile “Manage app data”.
2. In the “Archive” section, navigate to the configuration to be viewed.
3. Either click on the configuration in the list or open the extended options via the button  and then click on  File view].
 - ➔ A file view opens.

Related topics

- ➔ [Configurations – Introduction](#)
- ➔ [Window – “Manage app data”](#)

Copying configuration to the archive

Configurations saved in the archive of the ctrlX device can be copied within the archive. You can either overwrite an already existing archive entry or create a new archive entry.

Use cases:

- Creating a configuration as backup
- Duplicating a configuration as basis for a new configuration

Handling instruction

1. In the ctrlX OS web interface, open the window “Manage app data” via the button  in the header or via the navigation bar “Home” → tile “Manage app data”.
2. In the “Archive” section, navigate to the configuration to be copied and click on the  button to open the extended actions.
3. Click on the button  Copy].
 - ➔ The dialog “Copy configuration” opens.
The dialog contains two fields:
 - The name of the source is shown in the upper field.
 - The lower field is used to enter the target name and provides the following options:
 - **Create new configuration with new name**
Enter the name of the copy to be created to create a new configuration file.
 - **Overwrite existing configuration**
Click on the input field to open a selection list with already existing configurations. Select the configuration to be overwritten.
4. Enter the target configuration and confirm the dialog.
 - ➔ The copy is created and shown in the “Archive” section.



Use the “Description” field to better distinguish between the configurations, see:

➔ [Editing description of a configuration](#)

Related topics

➔ [Window – “Manage app data”](#)

➔ [Configurations – Introduction](#)

Deleting configuration in the archive

Downloading configurations from the archive of the ctrlX device.

Handling instruction

1. ➔ In the ctrlX OS web interface, open the window “Manage app data” via the button [📁] in the header or via the navigation bar “Home” → *tile “Manage app data”*.
2. ➔ In the “Archive” section, navigate to the configuration to be deleted and click on the [⋮] button to open the extended actions.
3. ➔ Click on the button [🗑️] Delete and acknowledge the confirmation prompt to delete the configuration.

Related topics

➔ [Window – “Manage app data”](#)

➔ [Configurations – Introduction](#)

Renaming configuration in archive

Configurations in the archive of the ctrlX device can be renamed.

Handling instruction

1. ➔ In the ctrlX OS web interface, open the window “Manage app data” via the button [📁] in the header or via the navigation bar “Home” → *tile “Manage app data”*.
2. ➔ In the “Archive” section, navigate to the configuration to be renamed and click on the [⋮] button to open the extended actions.
3. ➔ Click on the [✎] button in the extended options Edit].
➔ The “Edit configuration” dialog is shown.
4. ➔ Enter the name and save the entry.

Related topics

➔ [Window – “Manage app data”](#)

➔ [Configurations – Introduction](#)

Displaying information on a configuration in the archive

The following information on a configuration can be displayed in the archive of the ctrlX device:

- User who saved the configuration.
- Device on which the configuration was saved.
- App versions with which the configuration was saved.
- Currently installed app versions.

Handling instruction

1. In the ctrlX OS web interface, open the window “Manage app data” via the button  in the header or via the navigation bar “Home” → *tile “Manage app data”*.
2. In the “Archive” section, navigate to the configuration on which information should be displayed and click on the  button to open the extended actions.
3. To open the info, select the  button.

Related topics

- ➔ [Window – “Manage app data”](#)
- ➔ [Configurations – Introduction](#)

Editing description of a configuration

A configuration contains a freely definable description. This can be used for identification purposes. The description is displayed for the active configuration and for configurations in the archive.

Use cases:

- Describing main characteristics of a configuration, e.g. machine variant.
- To distinguish it from other configurations, e.g. by specifying a version.

Handling instruction

1. In the ctrlX OS web interface, open the window “Manage app data” via the button  in the header or via the navigation bar “Home” → *tile “Manage app data”*.
2. To open the advanced actions, click into the “Active” section or click on the  button for a configuration in the “Archive” area.
3. Open the editor via the  Edit button.
4. Open the description and save the input.

Related topics

- ➔ [Window – “Manage app data”](#)
- ➔ [Configurations – Introduction](#)

Managing configurations

You can view the files of a configuration in the archive and in the “Active” view.

Files in the archive can be changed.

Files in the “Active” section might be write-protected by the corresponding apps.

There are multiple options to reach the file view:

- In the “Active” section via  in the submenu  or via a content line
- In the “Archive” section via  in the submenu  or via an entry

The following actions are available in the view:

-  Loads the configuration to the ctrlX device (only for archived configurations).
-  Compares the configuration with other configurations.

When comparing, the differences between two configurations are displayed.

- [👁] Showing details in the list.
- [⬆] Uploading files or file content to the currently displayed directory.
- [⬆] Creating directory or file in the currently displayed directory.
- [🔄] Refreshing view.

Files and directories

Files and directories are displayed in a list. The list contains the following columns:

- “Name”
The name of the directory or the file.
- “Mode”
The Unix file mode (can be displayed via the [👁] button in the header).
- “Modified”
The date of the last file change.
- “Size”
The size of a file.
- “Actions”
The possible actions on the displayed element:
 - [⬇] Download.
 - [⋮] Further actions with the following buttons:
 - [📂] Open file.
 - [🔍] Compare file.
When comparing, the differences between two files or two directories are displayed.
 - [✏] Rename
 - [📄] Copy
 - [➡] Move
 - [🗑] Delete

Notes on file view

- Write-protected files in the active configuration are marked by a lock symbol [🔒] and can only be opened, compared and copied.
- To open the view on the subdirectory, click on a directory entry.
- A directory with the name “.” is displayed in the list to navigate to the parent directory via a click.
- To open the detailed view for files, click on a file. Text contents can be viewed and also edited if necessary.



Since app data represents the active app settings, modifying them is limited. Read-only files are marked with the symbol [🔒].

Initiate changes to these files via a loading process:

1. Save the current app data in a new archive entry, see: ➔ [Saving active app data to the archive of the ctrlX device](#)
2. Make the changes in the new archive entry.
3. Load the changed archive entry, see ➔ [Loading configuration from the archive to the ctrlX device](#)

Related topics

➔ [Configurations – Introduction](#)

➔ [Window – “Manage app data”](#)

Backing up active app data

Active app data is typically refreshed directly in the file storage when changed via the interface. For a few settings, e.g. retain data, this is not possible due to the performance. Therefore, when app data is saved, the apps are requested to refresh their storage. This operation is performed implicitly when backing up to the archive and downloading.

Related topics

- ➔ [Configurations – Introduction](#)
- ➔ [Window – “Manage app data”](#)

Downloading a configuration



A configuration is downloaded via a ZIP archive. A file name is suggested. The query of the storage location and the file name can be enabled in common web browsers. See Settings / Download in the respective browser.

Related topics

- ➔ [Configurations – Introduction](#)
- ➔ [Window – “Manage app data”](#)

Loading app data

The configuration state is restored when app data is loaded. This is the case when the following actions are performed:

- Restarting the ctrlX device
- Loading app data from archive
- Loading app data by uploading a configuration (ZIP archive)
- Resetting the app configuration

⚠ WARNING

Set the system to a safe state. Usually, the cyclic processing of a control process has to be paused.

- When loading a configuration, basic changes in app data can result, e.g.:
 - Exchange of the PLC project
 - Changes in the axis and kinematic definitions
- It is possible that the ctrlX device has to be switched to the operating state "Setup" if the changes require this state. The operating state can be switched using the status buttons in the header of the web interface.

Loading procedure

1. ➔ Based on the target configuration, apps decide whether the change is possible.
 - ➔ If an app cannot apply the target configuration, the process is aborted and a respective message is issued.
 - ➔ If aborted, the active app data remains unchanged.
2. ➔ The target configuration is loaded from the apps.
 - ➔ Errors while loading can be viewed via the loading report, see: [➔ Viewing Loading report](#)
 - ➔ The reached state of the apps is undefined in case of error!
3. ➔ The loaded settings are saved in the active app data.

Related topics

➔ [Configurations – Introduction](#)

➔ [Window – “Manage app data”](#)

Viewing Loading report

The last loading report can be viewed. The report informs on loading a configuration from the archive to the ctrlX device or uploading a configuration from the engineering PC.

Use case:

- To check whether all configuration data was loaded successfully to the ctrlX device

The Loading report contains the following information:

- Date and time of the loading
- Link to the window “Logbook”
- Successfully loaded data
- Data that could not be loaded



Loading report is created while loading.

If no configuration has yet been loaded to the ctrlX device, no Loading report exists.

Handling instruction

1. ➔ In the ctrlX OS web interface, open the window “Manage app data” via the button  in the header or via the navigation bar “Home” → tile “Manage app data”
2. ➔ Click on the  button in the “Active” section to open the extended actions.
3. ➔ Click on the  button to open the Loading report.
➔ The Loading report shows the results of the last loading.

Related topics

➔ [Configurations – Introduction](#)

➔ [Window – “Manage app data”](#)

6.5.3 Working with configurations

ctrlX devices are provided with a "REST" interface allowing an external access to the configuration data of the ctrlX device.

It can be accessed as follows:

- Via the ctrlX OS - Solutions API, see also the help menu  in the header of the ctrlX OS web interface:
 “API reference → ctrlX OS → Solutions API”
- Via WebDAV, see: ➔ <https://boschrexroth.github.io/ctrlx-automation-sdk>

6.6 Back up and restore

6.6.1 Backup – Introduction

The backup functionality can be used to back up and restore the settings of a device. Settings can also be selected, modified and applied to devices. The scope of a backup file can be modified during creation via "Options". Backup files can also be created automatically or manually to make specific device modifications.

Important note on changes as of version XCR-V-0206

With the XCR-V-0206, the original function forBackup & Restore was replaced by the "Setup" functionality, but the name of the function is still "Backup & Restore".

Further topics

- ➔ Backup – Creating from ctrlX device
- ➔ Backup – Applying to ctrlX device
- ➔ Backup – Showing report
- ➔ Backup – Showing history
- ➔ Backup – Active configuration and configuration archive
- ➔ Backup – Structure of the backup file "ctrlx-setup.json"
- ➔ Backup – Manual modification
- ➔ Backup – Command line
- ➔ Backup – Automatic application upon system startup
- ➔ Backup – Diagnostics

6.6.2 Backup – Creating from ctrlX device

The current device settings can be saved to a backup file.



Notes and prerequisites

- The “Backup & Restore” entry is only provided in the “Settings” window if the required admin rights are available.
- For security reasons, passwords for the device settings are only included in the backup file in encrypted form ("encryptedPassword" in backup file). Enter a password to create the backup.
The password settings can only be restored if the same password is used when applying the backup.
- After applying a backup or a ctrlX device restart, the latest created backup is automatically deleted from the control.
- When creating a system report, a new backup is created.
- The last created backup can be downloaded to the PC again via the interface or deleted on the ctrlX device.
Click on [...] in the header and open the dialog “Last created backup”.

Creating backup

1. ➔ In the side navigation, open the “ Settings” window and click on the tile “Backup & Restore”.
 - ➔ The “Backup & Restore” window opens and shows the “ Backup” tab.
2. ➔ Select the device settings and resources to be added to the backup file, see "Options during creation" in the following.

3. ➤ Confirm the settings with “Next”
 - ➔ The dialog to select the format and an optional password is shown.
 - ➔ Note: If external data carriers are connected, an additional input field is provided to specify the storage location of the backup file (local computer or external data carrier).
4. ➤ click on “Backup”.
 - ➔ After a successful creation, the backup file is saved in the download directory of the engineering PC.

Related topics

➔ [Backup – Applying to ctrlX device](#)

Backup – Options when creating

Parts of the system settings can be selected or deselected when creating the backup file. If options with resource files are selected, the backup file is created in ZIP format. The backup file can also be used to output "certificate keys". These can optionally be encrypted by specifying a password. Also refer to: ➔ [Backup – Certificates and keys](#)

Option	Description	Note
No option selected	• ➔ licenseManagement • ➔ packageManagement • ➔ systemInfo • ➔ typeplate	If no option is selected, a backup is created that only contains data for documentation purposes. If the backup is used on a ctrlX device, it does not change its properties.
Device-settings	• ➔ connectivity • ➔ dateTime • ➔ hosts • ➔ identityManagement • ➔ licenseManagement • ➔ packageManagement • ➔ proxy • ➔ ssh • ➔ system • ➔ systemInfo • ➔ typeplate	

Option	Description	Note
Apps	Additional app binaries	In the ZIP archive folder: packageManagement/installedApps Allows to restore app versions of the same architecture (amd64, arm64). Note: Operating system-related apps are not backed up. These include: core, bare, snapd, Linux kernel, hardware support. These apps can be selected explicitly, but this increases the size of the backup file considerably.
Certificates and keys	Additional output of certificate files and keys	Output of the certificate key and PKCS12 container files. In the ZIP archive folder: certificateManagement/applications
Active app data	Additional files of the active configuration	In the ZIP archive folder: configurations/active
Archived app data	Additional output of the archived configuration files	In the ZIP archive folder: configurations/archive

Selective element selection

As of Setup app version 02.02, elements to be added to the backup file can be selected in detail.

Enable the option “Extended view” in the “Backup” on the “Backup & Restore” site. The individually selectable elements are displayed in a table. To sort the rows, click on a column in the table header (alphabetically descending/ascending). Enter a search term or select a filter to limit the displayed rows. The table contains a button in the top left area to select or deselect elements. If no row is selected in the table, the buttons apply to all displayed rows in the table. If at least one row is selected, they only apply to the selected, displayed rows.

6.6.3 Backup – Applying to ctrlX device

The previously made device settings can be applied to the same or to another device.



Important information on applying backups

- The “Backup & Restore” entry is only provided in the “Settings” window if the required admin rights are available.
- The process updates the apps first. The changes to the settings are then applied.
- The “Remove apps and settings which are not part of the backup file” option should only be used for complete backups.
If this option is enabled, all elements that are not in the backup file are removed from the control:
 - Apps that are not included, are uninstalled on the control.
 - Configuration archives that are not included, are deleted from the control.
 - User accounts that are not included, are deleted on the control...
- One or multiple device restarts of the ctrlX device might be required for the update. During the restart, it cannot be communicated with the device.
- If properties change when applying the backup (e.g. hostname) or if apps are installed, these might only be visible after refreshing the GUI (F5).
- If the backup fails, try to apply the backup again. The successfully executed steps during the first attempt are skipped and the remaining steps are executed incrementally.
- As of version XCR-V-0204, passwords for the device settings can be restored if the backup was created with a password. Also refer to "encryptedPassword".

Apply saved backup to a ctrlX device:

1. In the side navigation, open the “Settings” window and click on the tile “Backup & Restore”.
 - ➔ The “Backup & Restore” window opens.
2. Open the “Restore” tab.
 - ➔ If an external storage medium is mounted on the ctrlX device, backup files stored on that medium can be selected, see: ➔ [Mounting external storage medium](#)
3. Specify the backup file to be used on the ctrlX device in the dialog.
4. If a password was assigned when the backup was created, enter it and click on “Next”.
5. If the ctrlX device on which the backup is applied, should correspond exactly to the settings of the backup file, enable the option “Remove apps and settings which are not part of the backup file”.
6. To start the backup, click on the “Restore” button.
 - ➔ The backup is applied to the target device. Multiple restarts of the ctrlX device might be required during the execution.

Related topics

➔ [Backup – Creating from ctrlX device](#)

6.6.4 Backup – Showing report

To display a report on the result of the last executed backup, proceed as follows:

1. In the side navigation, open the “Settings” window and click on the tile “Backup & Restore”.
 - ➔ The “Backup & Restore” window opens.

2. Click on the [...] button in the top right of the “Backup & Restore” window to open the advanced actions.
3. Click on the “Report” entry.
 - ➔ A dialog with the result of the last executed backup process is displayed.

Related topics

➔ [Backup – Applying to ctrlX device](#)

6.6.5 Backup – Showing history

Using the backup history, the backups performed up to now on the ctrlX device are shown as follows:

1. In the side navigation, open the “Settings” window and click on the tile “Backup & Restore”.
 - ➔ The “Backup & Restore” window opens.
2. Click on the [...] button in the window to open the advanced actions.
3. Click on the “History” entry.
 - ➔ The backups applied up to now are shown in a table in reverse chronological order.
Click on the column headers to change the sorting, to show details on the individual entries or to delete entries.

Related topics

➔ [Backup – Applying to ctrlX device](#)

6.6.6 Backup – Restrictions

The backup functionality transfers settings and resources between the devices the best possible. However, there are technical exceptions to be taken into account during use.

Plain text password

Passwords in plain text format are required to configure some settings, such as LDAP, radius, proxy and network security. For security reasons, these are only included in the backup file if encrypted. Thus, a password has to be entered when creating the backup file.

Improved encryption with SHA-256

If a password-encrypted backup file is created, the passwords and certificate keys in the backup file are encrypted using the SHA-1 encryption algorithm up to Setup app version 02.04. However, this has not been classified as sufficiently secure. Thus, the SHA-256 encryption algorithm is used from Setup app version 02.06. Furthermore, an additionally generated character string is used to additionally secure the password stored in the backup file, see “\$passwordHash”, in: ➔ [Backup – User management\(IdentityManagement\)](#)



Password-encrypted backup files created with the version XCR-V-0206 can be used from Setup app version XCR-V-0206. SHA-1-encrypted backup files are still supported.

Process architecture

The apps are only available in the device in the processor architecture available there (arm, amd). The backup can therefore only be used on devices with the same processor architecture.

Note: The app files can be replaced manually in the backup file. Ensure to adjust the \$path settings of the app files in the configuration file ctrlx-setup.json.

App Hardware Support

The "Hardware Support" app contains device-specific software that can generally only be used on devices of the same type.

Licenses

Licenses are bound to a device and cannot be copied. A transfer between devices using a backup file is not possible. However, it is possible to install licenses on a device via a so-called "capability response" file using backup file.

Note: Licenses are managed in the device in a special memory area. This is even not affected if the device is updated via Image - Update. = The licenses are retained.

System user

For development and diagnostic purposes, a system user can be set up via a so-called assertion. Although this user is transferred with the backup, the status of the system user is not transferred. This also applies when restoring on the identical device.

Configuration files

Configuration files are transferred without content check. It is the responsibility of the apps to ensure compatibility of these files. For example, the PLC app uses "runtime" files that can only be used on one architecture.

6.6.7 Backup – content and structure

Backup – Structure of the backup file "ctrlx-setup.json"

The backup file is available in two variants:

- JSON file: Without resource files
- ZIP file: With resource files

The JSON format is suitable to apply simple device settings that do not require separate resource files.

However, the ZIP format is normally used. The content of the backup is declared in the "ctrlx-setup.json" file. Further resource files in the ZIP file are referenced from "ctrlx-setup.json".

ctrlx-setup.json

The "ctrlx-setup.json" file is in the root directory of the archive. It contains the device settings as text. The JSON file has a respective schema definition.



JSON schema

There is a specific JSON schema providing a really good editing help together with text editors such as Visual Studio Code. It is sufficient to unzip the contents of the ZIP archive and to open the file "ctrlx-setup.json".

Resource files and \$path

Apart from the ctrlx-setup.json file, more files can be stored in the ZIP archive. Resource files are used in the backup if device settings can be exchanged via defined file formats.

Resource files include for example:

- Apps to be installed
- Certificates and certificate keys
- Contents of the active and archived configurations

Resource files are referenced from the ctrlx-setup.json via the "\$path" property.

The path references to the file in the ZIP archive of the backup.



Files in the ZIP archive are only used if they are referenced with a "\$path" entry in ctrlx-setup.json.

The storage structure of the resource files in the ZIP archive is analog to the structure of the ctrlx-setup.json file, refer to: [Backup – file content](#)



- To avoid problems with different file systems, files and directories are not case-sensitive in the archive entries.
- From version 02.04, the system first checks whether the required resource files are available when the backup file is applied.

Example: App entry to rexroth-deviceadmin (ctrlx-setup.json)

```
"packageManagement": {  
  "installedApps": {  
    "rexroth-deviceadmin": {  
      "$path": "packageManagement/installedApps/rexroth-  
deviceadmin-1.16.0.app",  
      "enabled": true,  
      "version": "1.16.0"  
    }  
  }  
}
```

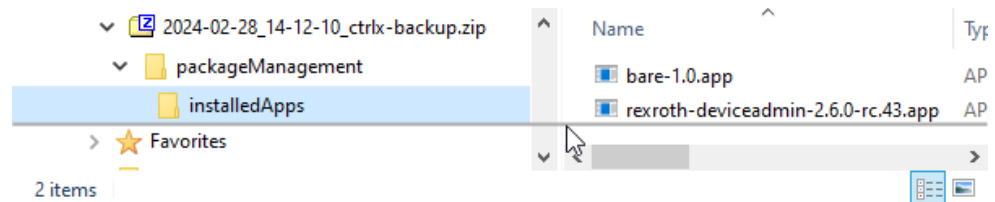


Fig. 6: App file in the backup archive (ctrlx-setup.zip)

Other entries

Additional information is stored in the root element in ctrlx-setup.json.

Path (ctrlx-setup.json)	Description
"\$schema"	This element refers to the schema definition. The schema file is provided as file when the ZIP archive is created. The reference is used in editors such as Visual Studio code to display context-sensitive help.
"\$name"	Optional specification of a backup name. The name is displayed in the overview of applied backups.
"\$description"	Optional description of the backup. The description is displayed in the overview of applied backups.
"\$version"	Optional specification of a backup version stored in the backup file. The version is displayed in the overview of applied backups.

Path (ctrlx-setup.json)	Description
"\$passwordHash"	As of XCR-V-0202, a hash value is stored used to check the password for decryption before applying the backup. From version XCR-V-0206, the encryption variants are defined here. The current variant stores an additionally generated byte sequence to secure the password. => The encrypted content can only be decrypted with the generated password hash.

Backup – file content

The following sections can be adjusted or retrieved via the backup and restore functionality in the device.



Not every backup contains all the areas listed.

Path (ctrlx-setup.json)	Access	Description	Available as of the app.setup version
↪ certificateManagement	rw	Certificates and keys, certificate revocation lists and PKCS12 containers	01.14 (02.06)
↪ configurations	rw	Active configuration and configuration archive	01.16
↪ connectivity	rw	Network adapters, bridges, (network services)	01.14 (0.18)
↪ dateTime	rw	Time setting and time synchronization	01.14
↪ hosts	rw	Static "Hosts" table of the device. (etc/hosts)	01.14
↪ identityManagement	rw	User accounts, user groups, right settings	01.14
↪ licenseManagement	r (w)	License information	01.16 (01.18)
↪ packageManagement	rw	Installed apps	01.14
↪ proxy	rw	Proxy settings	01.14
↪ ssh	rw	SSH settings	01.14
↪ system	rw	System settings	01.14
↪ systemInfo	r	System information	01.16
↪ typeplate	r	Type plate information	01.16

Table 1: Legend

Access	Description
r	The content cannot be changed by applying the backup. Possible reasons: • The element is a structuring element • The element is read-only • The element results from the scope of the installed apps
rw	The element contains parts that can be changed via the backup.

Backup – Certificates and keys

Certificates, keys, certificate revocation lists and PKCS12 containers can be transferred with the backup file. Certificates and keys in "pem" format, certificate revocation lists in "crl" format and PKCS12 containers in "pfx" format are each saved as separate file in the backup file.

Secured key transfer

Files of the keys are stored password-protected if a password is specified when creating the backup. In this case, the extension .aes is attached to the files.

Entries of the "own" category

Certificates and keys of the "Own" category are device-specific. They can be transferred to exactly one device. "Own" certificates and "Own" keys are stored under the serial number of the device.

Example

Certificate storage under der serial number "7261403575795", see: [Backup – Type plate information](#)

```
"certificateManagement": {
  "applications": {
    "rexroth-opcua-client": {
      "certificates": {
        "7261403575795": {
          "5EF1AA4208DDB8D5D055BA42260EA8BAD83C5499.der": {
```

Table 2: Elements of the backup file for certificates and keys

Path	Access	Description	Available from version
certificateManagement/applications	r	Applications in the certificate management	01.14
certificateManagement/applications/<app>	r	Certificates, keys and PKCS12 containers of an app. The app entries result from the installed apps.	01.14
certificateManagement/applications/<app>/certificates	r	Certificates of an app	01.14
certificateManagement/applications/<app>/certificates/<category>	rw	Category of an app Note: Certificates of the "Own" category can be stored under the serial number of the device.	01.14
certificateManagement/applications/<app>/certificates/<category>/<name>	rw	Certificate file of an app	01.14
certificateManagement/applications/<app>/certificates/<category>/<name>/<path>	rw	Reference to a file in ctrlx-setup.zip	01.14
certificateManagement/applications/<app>/keys	r	Keys of an app	01.14
certificateManagement/applications/<app>/keys/<category>	rw	All keys of a category of an app Note: Keys of the "Own" category can be stored under the serial number of the device.	01.14

Path	Access	Description	Available from version
certificateManagement/applications/<app>/ keys/<category>/<name>	rw	Key file of an app	01.14
certificateManagement/applications/<app>/ keys/<category>/<name>/ \$path	rw	Reference to a file in ctrlx-setup.zip	01.14
certificateManagement/applications/<app>/certificateRevocationLists	r	Certificate revocation lists of an app	02.06
certificateManagement/applications/<app>/certificateRevocationLists/<name>	rw	Certificate revocation list file of an app	02.06
certificateManagement/applications/<app>/certificateRevocationLists/<name>/ \$path	rw	Reference to a file in the backup archive	02.06
certificateManagement/applications/<app>/pkcs12Containers	r	PKCS12 container of an app	02.06
certificateManagement/applications/<app>/pkcs12Containers/<name>	rw	PKCS12 container file of an app	02.06
certificateManagement/applications/<app>/pkcs12Containers/<name>/ \$path	rw	Reference to a file in the backup archive	02.06
certificateManagement/pkis	r	Public key infrastructure	02.06
certificateManagement/pkis/<url>	rw	Server entry	01.18

Table 3: Legend

Access	Description
r	The content cannot be changed by applying the backup. Possible reasons: • The element is a structuring element • The element is read-only • The element results from the scope of the installed apps
rw	The element contains parts that can be changed via the backup.

Restrictions
Hardware-related keys (hsm) are included in the backup as of the version XCR-V-0120. These entries are normally applied during manufacturing.
Certificates and keys with "_undeletable" in the file name are neither deleted nor updated.

Backup – Active configuration and configuration archive

The active configuration is saved in configurations/active of the backup file.
Configuration archives are saved in configurations/archive of the backup file.
In the ctrlx-setup.json file, there are corresponding entries in the "configurations" section.

Example: Section "configurations" in ctrlx-setup.json

```
"configurations": {  
  "active": {  
    "$path": "configurations/active"  
  },  
  "archive": {  
    "config-1": {  
      "$path": "configurations/archive/config-1"  
    },  
    "config-2": {  
      "$path": "configurations/archive/config-2"  
    },  
  },  
}
```

Restoration is controlled via these entries.

The configuration referenced in the "active" entry is loaded to the apps of the target device. The configuration previously active on the device is replaced.

The configurations listed under "archive" are uploaded to the configuration archive of the target device. Configurations with the same name are replaced and new configurations are created. The configuration directories in the backup file are identified via "\$path".

If the path is modified, a configuration stored in the "archive" directory can be loaded when applying the backup for example:

Example: - Loading configuration from the directory "configurations/archive/config-1"

```
"active": { "$path": "configurations/archive/config-1" }
```

The following entries of the ctrlx-setup.json file have a special meaning:

"active": null

Resets the active configuration on the target device (corresponds to the loading of an "empty" configuration).

Example: "configurations" section in ctrlx-setup.json

```
"configurations": {  
  "active": null  
}
```

"archive": null

Deletes all configurations from the configuration archive of the target device

Example: "configurations" section in ctrlx-setup.json

```
"configurations": {  
  "archive": null  
}
```

"archive": { "config-1": null }

Deletes individual configurations (in the example "config-1") from the configuration archive of the target device

Example: "configurations" section in ctrlx-setup.json

```
"configurations": {  
  "archive": {  
    "config-1": null  
  }  
}
```

Behavior in case of missing configuration sections

If the "configurations" section is removed from the ctrlx-setup.json file, the configurations on the target device remain unchanged when applying the backup. Analogously applies that when omitting the entries "active" or "archive", the active configuration or the configuration archive remains unchanged on the target device.

Backup – Connectivity

This section contains settings for wired network adapters, network bridges and a WiFi network adapter connected via USB. Network services can also be enabled.

NOTICE

Incorrect settings of the network interfaces can cause that the control is not reachable anymore.

Network services

The following entries in the ctrlx-setup.json file can be used to define the state of the network services in the ctrlX device:

```
ctrlx-setup.json  
"connectivity": {  
  "discoveryServices": {  
    "gratuitous-arp": {  
      "configuration": {  
        "count": 3,  
        "interval": 900,  
        "onStartup": true  
      },  
      "enabled": false  
    },  
    "llmnr": {  
      "enabled": true  
    },  
    "mdns": {  
      "enabled": true  
    },  
    "mdns-resolver": {  
      "enabled": true  
    },  
    "upnp": {  
      "enabled": true  
    }  
  }  
}
```



New as of the Setup app 02.02: gratuitous-arp with the respective settings.

Network interfaces

Backup supports:

- ethernet: Wired interfaces
- wifi: WiFi adapter (max. 1)
- bridge: Bridges

Security

Network interfaces support security protocols that can be set up for the different interfaces or networks (for WiFi adapters).

Supported security protocols:

- PSK
- EAP-TLS
- EAP-PEAP

The available protocols depend on the type of network interface.



If no password is entered when creating the backup, passwords cannot be read from the device.

To set up the Security of networks, add "password" or "privateKeyPassword" explicitly.

Adding the Security mode "IEEE 802.1X / EAP-TLS" for Ethernet adapters

For Ethernet adapters, access to the network can be protected using EAP-TLS. Make the following specifications to add the Security mode "IEEE 802.1X / EAP-TLS" to an Ethernet adapter.

```
ctrlx-setup.json
{
  "connectivity": {
    "ethernet": {
      "XF10": {
        "security": {
          "caCertificate": "network-ca.cert",
          "clientCertificate": "network.cert",
          "enabled": true,
          "identity": "identity",
          "privateKey": "network.key",
          "privateKeyPassword": "password",
          "type": "ieee8021x__eap-tls"
        }
      }
    }
  }
}
```

Handling passwords in the Security mode "IEEE 802.1X / EAP-TLS"

When using the Security mode "IEEE 802.1X / EAP-TLS", the password "privateKeyPassword" for the private key has to be specified if a password is required to read out the private key. If a backup is created without a password (Sichern), the password to read the key is not included in the backup file. To use such a backup file again, enter the password ("privateKeyPassword") explicitly into the backup file before to read the private key.

Alternatively as of XCR-V-0204: If a backup is created and a password is specified, an "encryptedPassword" is backed up used to restore the password.

Removing the "IEEE 802.1X / EAP-TLS" Security mode for Ethernet adapters

The Security mode can be removed with the following specification:

```
ctrlx-setup.json
{
  "connectivity": {
    "ethernet": {
      "XF10": {
        "security": null
      }
    }
  }
}
```

WiFi adapter



The backup supports a WiFi network adapter connected via USB.

Error with multiple WiFi adapters in the backup or on the target device

Under the following conditions, an error message is displayed when the backup is executed:

- If multiple WiFi adapters are connected to the target device.
- If more than one WiFi adapter is included in the backup to be used.

Cause:

The system normally names the WiFi adapters and is not always unique. Thus, the settings of an adapter in the backup can only be applied to a single adapter on the target device.

Related topics

➔ [Connecting to WiFi network](#)

WiFi networks

A WiFi adapter can support multiple WiFi networks. Each wireless network is provided with an own SSID and own security settings. The WiFi adapter is used as network "client".

Example of a WiFi adapter with four WiFi networks: The SSID of the first wireless network is "ssid_none", that of the second "ssid_psk", etc.:

```
ctrlx-setup.json
{
  "connectivity": {
    "wifi": {
      "wlx7cc2c6023ec1": {
        "mode": "client",
        "network": {
          "ssid_none": {
            "autoConnect": true,
            "security": {
              "type": "open-wifi"
            }
          },
          "ssid_psk": {
            "autoConnect": true,
            "security": {
              "type": "wpa2__psk",
              ...
            }
          },
          "ssid_eapTLS": {
            "autoConnect": true,
            "security": {
              "type": "wpa2__eap-tls",

```

```
...
    },
    "ssid_eapPEAP": {
        "autoConnect": true,
        "security": {
            "type": "wpa2__eap-peap",
            ...
        }
    }
}
}
```

Specification of network client or hotspot ("mode", from version 1.20)

A WiFi adapter can be used as network client or as hotspot (access point) if technically supported by it. The "mode" from the Setup app version 1.20, determines whether the WiFi adapter is used as network client (mode: client) or as hotspot (mode: hotspot). Since backup files created with the Setup app version 1.18 or older do not contain the "mode", as of Setup app version 1.20; when applying (restore) a backup without specifying "mode", it is assumed that the WiFi adapter is used as network client as it was used before.

Automatic connection to the wireless network ("autoConnect" as of the Setup app version 1.18)

For the WiFi adapter to connect to the specified wireless network, the setting "autoConnect" has to be either set to true or it must not be specified. In the latter case (downward compatibility), the "autoConnect" is implicitly set to true. If "autoConnect" is explicitly set to false, the WiFi adapter does not connect to the specified wireless network.

Adding a WiFi network with the Security mode "None"

With the following specifications made in the ctrlx-setup.json file, a WiFi adapter can be added to a network with the SSID (here in the example) "ssid_none" and the Security mode "None":

```
ctrlx-setup.json
{
  "connectivity": {
    "wifi": {
      "wlx7cc2c6023ec1": {
        "mode": "client",
        "network": {
          "ssid_none": {
            "security": {
              "type": "open-wifi"
            }
          }
        }
      }
    }
  }
}
```

Adding a WiFi network with the Security mode "WPA/WPA2 Personal"

With the following specifications made in the ctrlx-setup.json file, a WiFi adapter can be added to a network with the SSID (here in the example) "ssid_psk" and the Security mode "WPA/WPA2 Personal". The password has to consist of at least 8 characters and has to be specified.

```
ctrlx-setup.json
{
  "connectivity": {
    "wifi": {
      "wlx7cc2c6023ec1": {
        "mode": "client",
        "network": {
          "ssid_psk": {
            "security": {
              "password": "password",
              "type": "wpa2__psk"
            }
          }
        }
      }
    }
  }
}
```

Handling passwords in the Security mode "WPA/WPA2 Personal"

When adding a WiFi network with the Security mode "WPA/WPA2 Personal", the password has to be specified and consist of at least 8 characters. If a backup is created without password specification, the password is not included in the backup file. To use such a backup file again, manually enter the password into the backup file before.

Alternative from version V-0204: If a backup is created (back up) and a password is specified, the password is saved and also restored via "encryptedPrivateKeyPassword".

Adding a WiFi network with the Security mode "WPA/WPA2 Enterprise (TLS)"

With the following specifications, a WiFi adapter can be added to a network with the SSID "ssid_eapTLS" and the Security mode "WPA/WPA2 Enterprise (TLS)". The password for the private key has to be specified if a password is required to read out the private key.

```
ctrlx-setup.json
{
  "connectivity": {
    "wifi": {
      "wlx7cc2c6023ec1": {
        "mode": "client",
        "network": {
          "ssid_eapTLS": {
            "security": {
              "caCertificate": "network-ca.cert",
              "clientCertificate": "network.cert",
              "identity": "identity",
              "privateKey": "network.key",
              "privateKeyPassword": "password",
              "type": "wpa2__eap-tls"
            }
          }
        }
      }
    }
  }
}
```

```
    }
  }
}
```

Handling passwords in the Security mode "WPA/WPA2 Enterprise (TLS)"

When adding a WiFi network with the Security mode "WPA/WPA2 Enterprise (TLS)", the password for the private key has to be specified if a password is required to read out the private key. If a backup is created without a password (Sichern), the password to read the key is not included in the backup file. To use the password, add it to the backup file.

Alternatively as of V-0204: If a backup is created (back up) and a password is specified, an "encryptedPassword" is backed up used to restore the password.

Adding a WiFi network with the Security mode "WPA/WPA2 Enterprise (PEAP)"

With the following specifications, a WiFi adapter can be added to a network with the SSID "ssid_eapPEAP" and the Security mode "WPA/WPA2 Enterprise (PEAP)".

```
ctrlx-setup.json
"connectivity": {
  "wifi": {
    "wlx7cc2c6023ec1": {
      "mode": "client",
      "network": {
        "ssid_eapPEAP": {
          "security": {
            "anonymousIdentity": "anonymousIdentity",
            "caCertificate": "network-ca.cert",
            "identity": "identity",
            "password": "password",
            "phase2Auth": "MSCHAPV2",
            "type": "wpa2__eap-peap"
          }
        }
      }
    }
  }
}
```

Handling passwords in the Security mode "WPA/WPA2 Enterprise (PEAP)"

Password handling in Security mode "WPA/WPA2 Enterprise (PEAP)" is identical to that in the Security mode "WPA/WPA2 Personal" (see there).

Deleting WiFi networks

With the following specifications made in the ctrlx-setup.json file, the WiFi networks "ssid_none", "ssid_eapTLS", "ssid_eapPEAP" and "ssid_psk" can be deleted:

```
ctrlx-setup.json
"connectivity": {
  "wifi": {
    "wlx7cc2c6023ec1": {
      "mode": "client",
      "network": {
        "ssid_none": null,
        "ssid_eapTLS": null,
        "ssid_eapPEAP": null,
        "ssid_psk": null
      }
    }
  }
}
```

Hotspot (from version XCR-V0120)

A WiFi adapter can only be used either as network client or as hotspot (access point) if technically supported by it, also refer to: [Specification of network client or hotspot](#) ("mode", from version 1.20)

To operate a WiFi adapter as hotspot, the following information is required in the ctrlx-setup.json file. The wireless network in the example has the SSID "ssid", the Security mode has to be of type "WPA/WPA2 Personal". The mode (mode: hotspot) has to be specified. If it is missing, the WiFi adapter is handled like a network client (mode: client) when the backup is applied.

```
ctrlx-setup.json
"connectivity": {
  "wifi": {
    "wlx7cc2c6023ec1": {
      "hotspot": {
        "security": {
          "password": "password",
          "type": "wpa2__psk"
        },
        "ssid": "ssid"
      },
      "mode": "hotspot"
    }
  }
}
```

Adding network bridges

Unlike Ethernet and WiFi adapters, network bridges can be added or deleted via the backup. With the following specifications in the ctrlx-setup.json file, a network bridge can be added to the ctrlX control (in the example "bridge0"):

```
ctrlx-setup.json
"connectivity": {
  "bridge": {
    "bridge0": {
      ...
    }
  }
}
```

Deleting network bridges

With the following specifications in the ctrlx-setup.json file, a network bridge can be deleted on the ctrlX device (in the example "bridge0"):

```
ctrlx-setup.json
{
  "connectivity": {
    "bridge": {
      "bridge0": null
    }
  }
}
```

Adding the "IEEE 802.1X / EAP-TLS" Security mode for network bridges

For bridge adapters, access to the network can be protected using the Security mode EAP-TLS. The details in the backup file correspond to those when using an Ethernet adapter.

```
ctrlx-setup.json
{
  "connectivity": {
    "bridge": {
      "bridge0": {
        "security": {
          "caCertificate": "network-ca.cert",
          "clientCertificate": "network.cert",
          ...
        }
      }
    }
  }
}
```

Removing the "IEEE 802.1X / EAP-TLS" Security mode for bridge adapters

As for Ethernet adapters, the Security mode can also be removed for bridge adapters by specifying "security": null, see: [➔ Removing the "IEEE 802.1X / EAP-TLS" Security mode for Ethernet adapters](#)

VLAN adapter

The user can neither modify, add nor delete VLAN adapters. They are thus not included in the backup.

Names of wired network adapters

There are different names for wired network adapters in the individual control versions. This is considered when applying a backup by adapting the adapter names in the backup file to the names of the current control. For this purpose, two groups are used. The names in one group are used as synonyms.

Name	Description
eth0, XF10	Group 1
eth1, XF11, XF51	Group 2

Restrictions

Backup behavior for Ethernet and WiFi adapters Ethernet and WiFi network adapters can neither be added nor deleted. Only the properties of the network adapters provided by or connected to the hardware can be changed.
Backup behavior with multiple WiFi adapters If a backup is applied that contains more than one WiFi adapter, or if more than one WiFi adapter is connected to the ctrlX control when applying a backup, an error is reported.
Exclusive Ethernet adapter for EtherCAT master If an Ethernet adapter is used exclusively by an EtherCAT master, the adapter is not included in the backup file when a backup is created. If a backup file contains an Ethernet adapter that is currently used exclusively by an EtherCAT master in the control, the adapter is ignored when the backup is applied.
Device restart after changing the MTU (Maximum Transmission Unit) If the MTU (Maximum Transmission Unit) of at least one network adapter is changed, the ctrlX device has to be restarted. If at least one of the affected network adapters is connected to a network when a backup is applied, it is restarted automatically. If the affected network adapters are not all connected to one network, restart the ctrlX device manually after loading the backup to apply the changed MTU values.
With DHCP, only the "Client Identifier" property is supported up to and including the Setup app version 1.18. The properties "Use DNS", "Use NTP", "Use MTU" and "Route Metric" are only supported as of the Setup app version 1.20.
WiFi adapter –Automatic connection The automatic connection of a WiFi adapter to a wireless network can only be prevented as of the Setup app version 1.18 using the entry "autoConnect": false, see: Automatic connection to the wireless network ("autoConnect" as of the Setup app version 1.18)
Support for hotspots Hotspots are supported from the Setup app version 1.20.
Read/write state of network services The state of the network services can only be read or written as of the Setup app version 1.20.

Table of the read and write permissions for the connectivity

Path	Access	Description
connectivity/bridge/<bridge-adapter>	rw	Network bridges
connectivity/ethernet/<eth-adapter>	rw	Ethernet adapter
connectivity/wifi/<wifi-adapter>	rw	WiFi network adapter
connectivity/discovery-Services/<service>	rw	Network services (llmnr, mdns, mdns-resolver, upnp)

Backup – Date and time

Use the ctrlx-setup.json file to see and modify the time zone set on the device and the time server used.

The time synchronization can be switched on and off separately via the time server.



The current date and the current time can neither be retrieved nor read via backup and restore.

Backup – Hosts

To read and write the static host table of the device (etc/hosts), go to "Hosts" in the file ctrlx-setup.json. Each entry of the host table contains the IPv4 or IPv6 address, its host names and a list of other host names used as alias.

When applying the backup file, all entries of the host table of the device are replaced.

NOTICE

The device might not be reachable anymore via the network due to an incorrect project planning and configuration.

Backup – User management(IdentityManagement)

Prerequisite

The user requires admin rights on the ctrlX device for the backup.

The settings of the user management include the settings of the users, user groups, different policy settings and the connection to the LDSP or Radius server.

Active user account settings can be modified while applying the backup. Changes are only applied after executing the backup. Thus, removing the default user "boschrexroth" is even possible if the backup application was started under this user.



References from user definitions to non-existent groups are ignored.

New as of Setup app version 1.20

The user settings contain the password hash.

The extension can be used to transfer the validity of a password to other devices and also other user accounts.

- The password hash allows to check a specific password.
- The password cannot be determined from the password hash.
- The password hash is created via the Linux operating system.

A random number (salt) is used when creating the password hash.

Generating a hash multiple times for the same password generates different password hashes, see:

➔ <https://manpages.ubuntu.com/manpages/focal/man1/passwd.1ssl.html>

- The password policies (PasswordPolicies) are not checked when the password hash is applied.

The password hash is ignored in Setup app versions < 1.20.

If the function to apply the password hash is not available (for deviceadmin version lower than V-0120), a warning is issued.

The settings can be modified partially. Refer to the following handling instructions.

Deleting a user account

Upon delivery of the ctrlX device, "boschrexroth" is the default user with admin rights.

To ensure IT security, the default user should not be available anymore at runtime and should be replaced by customized user accounts.

ctrlx-setup.json: Deleting the user "boschrexroth"

```
{
  "identityManagement": {
    "users": {
      "boschrexroth": null
    }
  }
}
```



In order not to "lock out" oneself, ensure that another user was set up. This user requires admin rights and you have to know this password.

Deleting a user group

ctrlx-setup.json: Deleting the user group "myUserGroup"

```
{
  "identityManagement": {
    "groups": {
      "myUserGroup": null
    }
  }
}
```

Creating a user with initial password

Normally, an administrator creates user accounts for others. To ensure that an account is not hijacked, specify an initial password when creating an account. The respective user can now enter this password to allow a login. Passwords can be set for new user accounts.

Example: The user "foo" should change the password after the initial login. The admin can force it via the entry "forcePasswordChange": true.

ctrlx-setup.json: Creating the user "foo"

```
{
  "identityManagement": {
    "users": {
      "foo": {
        "enabled": true,
        "forcePasswordChange": true,
        "initialPassword": "mysecretpassword"
      }
    }
  }
}
```

Changing password rules

Password rules

```
{
  "identityManagement": {
    "passwordPolicies": {
      "enforcePasswordHistory": 5,
      "minimumLowercaseChars": 0,
      "minimumNumericChars": 0,
      "minimumPasswordLength": 1,
      "minimumSpecialChars": 0,
      "minimumUppercaseChars": 0,
      "mustNotContainUsername": false,
      "passwordExpiration": 0
    }
  }
}
```

Backup – License information

The license details are available in the backup file for information purposes. Information is not considered when applying the backup.

In 1.18 and higher, licenses can be imported via a backup to a ctrlX device. However, this requires a manual addition in the ctrlx-backup.json file of the backup. Also refer to:

➔ [Backup – Structure of the backup file "ctrlx-setup.json"](#)

Addition to the ctrlx-setup.json file

Please note:

- To transfer licenses to a target device, a so-called Capability Response file is required. Refer to the note below
- 1. ➔ Create the section „capabilityFiles“ in the ctrlx-setup.json file in the section „licenseManagement“.
- 2. ➔ Create a section below „capabilityFiles“ for each device with its serial number, in the example "7261403575795".
The device serial number is displayed in the web interface of the ctrlX device, see:
➔ [Widget – “Typeplate”](#)
- 3. ➔ Insert the "Capability Response" file into the ZIP archive of the backup. e.g. into the root directory "/7261403575795.bin".
Download the Capability Response file from the Rexroth license server.
- 4. ➔ Reference this file in the respective section of the ctrlx-setup.json file via "\$path" as shown in the following example.

Example: Applying capability files

```
{
  "licenseManagement": {
    "capabilityFiles": {
      "7261403575795": {
        "$path": "7261403575795.bin"
      }
    }
  }
}
```



Capability Response files are excerpts from the current licensing state of a device on the license server. The files can only be applied successfully in their creation order. It is not possible to apply the file again. A corresponding information is output.

The licensing state cannot be undone, not even by an image update of the device.

Backup – Package management (apps)

When creating a backup in ZIP format, the apps installed on the device can be downloaded and backed up in the subfolder */packageManagement/installedApps* of the backup file.

In the *ctrlx-setup.json* file, entries are created in the *"packageManagement"* section when creating a backup. They control the app loading to the target device when applying the backup.

Example: Section "packageManagement" in ctrlx-setup.json

```
"packageManagement": {
  "allowUnknownApps" : false,
  "installedApps": {
    "core22": {
      "$path": "packageManagement/installedApps/
core22-20230531.app",
      "appType": "base",
      "description": "The base snap based on the Ubuntu
22.04 release.\n",
      "size": 77434880,
      "summary": "Runtime environment based on Ubuntu
22.04",
      "vendor": "Canonical",
      "version": "20230531"
    },
    "rexroth-arch02-hw": {
      "$path": "packageManagement/installedApps/rexroth-
arch02-hw-2.2.0-rc.2.app",
      "appType": "gadget",
      "description": "This gadget enables ctrlX Arch02
devices to work with Ubuntu Core\n",
      "size": 36872192,
      "summary": "Hardware specific components",
      "title": "Hardware Support",
      "vendor": "Bosch Rexroth AG",
      "version": "2.2.0-rc.2"
    },
    "rexroth-arch02-kernel": {
      "$path": "packageManagement/installedApps/rexroth-
arch02-kernel-5.10.156-rt76-arch02-18.app",
      "appType": "kernel",
      "description": "Provides the Linux Kernel running
on ctrlX CORE.",
      "size": 359538688,
      "summary": "Realtime Linux Kernel",
      "title": "Realtime Kernel",
      "vendor": "Bosch Rexroth AG",
      "version": "5.10.156-rt76-arch02-18"
    },
    "rexroth-automationcore": {
      "$path": "packageManagement/installedApps/rexroth-
```

```
automationcore-2.2.0-rc.2338.app",
    "appType": "app",
    "description": "Provides basic automation
functionalities, e.g. the ctrlX Data Layer interface.",
    "size": 20754432,
    "summary": "Basic automation functionalities",
    "title": "Automation Core",
    "vendor": "Rexroth Admin",
    "version": "2.2.0-rc.2338"
  },
  ...
}
```

Handling different app types

All active (i.e. not disabled) apps with appType app can be included in the backup file. From Setup app version 02.02, apps with appType base (e.g. core22), core (e.g. core), gadget (e.g. rexroth-archxx-hw), kernel (e.g. rexroth-archxx-kernel) and snapd as well. The latter can, also as of Setup app version 02.02, be selected in the table of addable elements and added to the backup, see: [Backup – Options when creating](#)



New as of Setup app version 02.02

Apps backed up in the backup file contain the entry "\$path" in the ctrlx-setup.json file.

Apps can also be added manually to the backup file to apply a backup.

Specify the app type if it is not a "normal" app (AppType="app") and the app is newly installed on the target device. This can be the case for core18, core20, for example.

Notes

- After installing an app with appType gadget, a device restart might be required depending on the control type.
- Apps with appType base, core, gadget, kernel and snapd are not included in the backup by default.
However, they can be added manually to the backup file using the table of addable elements, see: [Backup – Options when creating](#)
- The "rexroth-deviceadmin" app can only be updated with a version >= v14.0.0.
- As of setup app version 02.06, the packageManagement setting "allowUnknownApps" can be transferred.
This allows the installation of unsigned apps.

Warning: If this setting is selected, the warranty claims of the device are lost permanently.

Handling unsigned apps

As of Setup app version 02.06, the "allowUnknownApps" option can be used to install unsigned apps on the ctrlX device.

NOTICE

Loss of warranty

Enabling the "allowUnknownApps" option on the ctrlX device results in loss of warranty claims!

Restrictions

- Apps can only be installed on the same target system on which they were backed up (ctrlX CORE X3, ctrlX CORE X7 or VirtualControl).
- The Setup app (internal name "rexroth-setup") can only be updated with a later version.
- Enabling and disabling apps is not supported.
- Disabled apps are not taken into account.
- Apps with appType base, core, gadget, kernel and snapd can only be replaced by newer versions.

Backup – Deleting apps

With the following specifications in the ctrlx-setup.json file, apps can be deleted on the ctrlX device, in this example the "rexroth-firewall" app.

```
ctrlx-setup.json
{
  "packageManagement": {
    "installedApps": {
      "rexroth-firewall": null
    }
  }
}
```



Some system-related apps such as core22, gadget or kernel cannot be deleted.

Backup – Proxy

Store or specify the settings of the device proxy configuration under "Proxy".

The configuration allows the general enabling and disabling of the proxy configuration, the specification of the proxy server used and a list with exception addresses for which no proxy server should be used.

An additional authentication can be enabled for the proxy servers. In this case, username and password have to be specified in the file ctrlx-setup.json if this information has not been provided before due to the application of a previously created backup.



New as of Setup app version 02.04:

The password of the proxy settings is saved as "encryptedPassword" in the backup if the backup is created with a password. The password of the proxy settings is restored from the "encryptedPassword".

Backup – SSH

Store or specify the SSH settings of the device under "SSH".

The SSH access can always be enabled/disabled and different security parameters and a banner are specified.

If the access is enabled, an SSH login is possible for users being "system user" s on the device.

Backup – System

To see and modify the host name of a device, go to "System". Furthermore, theming settings are included to modify the visual design of the device.

As of the Setup version 02.04, the allocation of CPU cores for realtime and non-realtime tasks can also be viewed and modified. Note that at least one core has to be available for non-realtime tasks. Changed settings are applied after device restart. If necessary, the restart is triggered automatically when the backup is applied.

Backup – System information

System information is available in the backup for information purposes. Information is not considered when applying the backup to a target device.

Backup – Type plate information

Type plate information is available in the backup for information purposes. Information is not considered when applying the backup to a target device.

6.6.8 Backup – Manual modification

Partial changes

The backup functionality allows to make specific changes in specific settings. A backup can for example be applied in JSON format only containing the "PasswordPolicies".

In this case, the current device settings are determined when applying the backup. The backup content is included into the determined settings. The result is then applied.

Removing optional elements

The backup content is applied to the currently existing settings of the device. Changes and extensions can be applied easily. Especially changes can be specified explicitly.

To remove optional parts, e.g. a user account or a user group, use a specific notation. The element has to be specified with the value zero.

For examples, refer to the chapter: ➔ [Backup – User management\(IdentityManagement\)](#)

6.6.9 Backup – Diagnostics

Progress and result of the backup actions are shown in the web interface.

Information on the process and possible problems are logged in the logbook.

To show additional details, enable the trace messages und filter for the "web.setup" unit.

6.6.10 Backup – Command line

With the command line, the "Backup and restore" function of the device can be used automatically.

The command line tool is installed on an Engineering PC via the ctrlX WORKS installation.

The storage location depends on the installation path. Default is: "C:\Program Files\Rexroth\ctrlX WORKS\ctrlx-backup.exe"

Command overview

Name	Description
apply	Applying backup to a device
create	Creating device backup
verify	Checking a backup file
help	Output of the help
schema	Creating the JSON schema file for ctrlx-setup.json (can be used as editing help)

Applying backup to a device

Command: apply

Call: ctrlx-Backup.exe apply host [flags]

host: Device address (hostname or IP including port)

Example

```
"C:\Program Files\Rexroth\ctrlX WORKS\ctrlx-backup.exe" apply localhost:8443
```

Switch

Name	Abbreviation	Description	Default
--path		Name of the backup file	ctrlx-setup.json
--help	-h	Output of the help on "apply"	
--level	-l	Logging level (ERROR WARNING INFO DEBUG)	INFO
--options	-o	Optional file to specify further "Apply" options. Password to decrypt the certificate key. Also refer to: --encryptpwd	apply-options.json
--password	-p	Password to access the device	"boschrexroth"
--username	-u	User to access the device (admin rights required)	"boschrexroth"
--encryptpwd	-e	Password to decrypt secrets (alternative to specifying the password via "Apply options")	""
--rejectinsecure-connections		Connection to the device is only set up if a secure connection can be established. Example: If there is no secure certificate (issued by an authentication authority) on the device, no connection is established	false

"Apply" options

Advanced options can be specified as JSON file, refer to the switch "--options".

Name	Description	Example
\$password	Password to decrypt secrets	{ "\$password": "geheim" }

Creating a backup from a device

Command: create

Call: ctrlx-backup.exe create host [flags]

host: Device address (hostname or IP including port)

Example

```
"C:\Program Files\Rexroth\ctrlX WORKS\ctrlx-Backup.exe" create localhost:8443
```

Switch

Name	Abbrevia- tion	Description	Default
--filename	-f	This switch is still supported, but it is obsolete: Please use --path. Name of the backup file. If the file extension .zip is specified, not only the device settings, but also resources such as apps, certificates and configurations are stored in the backup file. Note: Use "Create" options to additionally affect the content of the backup file.	ctrlx-setup.json
--help	-h	Output of the help on "create"	
--level	-l	Logging level (ERROR WARNING INFO DEBUG)	INFO
--options	-o	Optional file to specify further "Create" options	create-options.json
--password	-p	Password to access the device	"boschrexroth"
--path		Name of the backup file If the file extension .zip is specified, not only the device settings, but also resources such as apps, certificates and configurations are stored in the backup file Note: Use "Create" options to additionally affect the content of the backup file.	ctrlx-setup.json
--username	-u	User to access the device (admin rights required)	"boschrexroth"
--rejectinsecure-connections		Connection to the device is only set up if a secure connection can be established. If there is no secure certificate (issued by an authentication authority) on the device for example, no connection is established.	false

"Create" Options

Advanced options can be specified as JSON file, refer to the switch "--options".

Name	Description	Default	Example
\$password	Password to encrypt and to add "encryptedPassword" entries to the device settings. (Proxy, Network Security, ldap, radius)		{ "\$password": "geheim" }
\$format	Output format of the backup file: "zip" or "json" Note: Resource files have to be excluded for the "json" output format	zip	Creating the backup file in the JSON file format. { "\$format": "json" }

Name	Description	Default	Example
\$content	Specifying the backup content on top level: • "none": Not creating any backup elements • "items": Creating only element in the JSON structure • "files": Creating the respective resource files in addition to the elements of the JSON structure (requires "\$format": "zip")	"items", if "\$format" : "json" Otherwise, "files"	Excluding all resource files <pre>{ "\$content": "items" }</pre>
... \$content	Specifying the backup content for child elements, e.g.: • identityManagement • hosts • connectivity • ... It includes \$content with: • "none": Not creating any elements of this node • "items": Creating only the element of this node in the JSON structure • "files": Creating the respective resource files in addition to the elements of the JSON structure (requires "\$format": "zip") The settings apply to the element and all child elements if these are not overwritten explicitly.		Refer to the examples below

Examples on the "create" options: \$content

Excluding app binaries

```
{
  "packageManagement": {
    "$content": "items"
  }
}
```

Only dateTime settings

```
{
  "$format": "json",
  "$content": "none",
  "dateTime": {
    "$content": "items"
  }
}
```

Only user settings

```
{
  "$format": "json",
  "$content": "none",
  "identityManagement": {
    "$content": "items"
  }
}
```

Only active configuration

```
{
  "$format": "zip",
  "$content": "none",
  "configurations": {
    "active": {
      "$content": "files"
    }
  }
}
```

Checking a backup file

The "verify" command checks the following points:

- Validity of the password switch e (--encryptpwd)
Based on the optional \$passwordHash entry in the file.
- Availability of resource files referenced via \$path entries.
- JSON schema rules. Transferred JSON file or ctrlx-setup.json in the backup zip file.



The check is on the local computer based on the version of the executable file. A ctrlX device is not required.

Command: verify

Call: ctrlx-backup.exe verify [flags]

host: Device address (hostname or IP including port)

Example

```
"C:\Program Files\Rexroth\ctrlX WORKS\ctrlx-backup.exe"
verify --path ctrlx-backup.zip -l INFO
```

Switch:

Name	Abbrevia- tion	Description	Default
--path		Name of the backup file	ctrlx-setup.json
--help	-h	Output of the help on "create"	
--level	-l	Logging level (ERROR WARNING INFO DEBUG)	INFO
--encryptpwd	-e	Password to decrypt secrets (alternative to specifying the password via "Apply options")	""

6.6.11 Backup – Automatic application upon system startup

It is possible that the device cannot be operated interactively anymore, e.g. due to an incorrect network configuration or if the login information is no longer available.

In such cases, the automatic application of a backup can solve the problem.



The automatic backup is applied via an external data carrier and triggered once when the device is restarted.

If there are files of an image update on the data carrier, the image update is executed first. After an automated system restart, the backup is executed.

Prerequisites

Data carrier

- The data carrier is plugged into the ctrlX device and logged in (SD card or USB stick).
- The data carrier is formatted in FAT32 file format.
- The ctrlX device has write access to the data carrier.



The USB stick is preferably used to execute the backup.

The SD card is considered if no USB stick is inserted.

Also refer to: [SD cards and USB support](#)

Administrator rights

Administrator rights are required on the device to automatically apply a backup upon system start.

Alternative 1

- The data carrier contains a certificate key of the ctrlX device, which activates the autorestore authorization.

There is at least one file in the root directory on the data carrier that matches a certificate key of the device, see: [Tab - "Keys"](#)

Also refer to: [Autorestore authorizations](#)

Alternative 2

- Device state of the ctrlX device after image update. The user "boschrexroth" exists and can be logged in with the default password. The user has "full access". The device and its data are not secured, as "boschrexroth" has full access as a publicly known account with a known password.

Backup file exists

At least one suitable backup is stored on the data carrier.

A backup can be provided as ZIP archive or as single JSON file.



Directory names and file names are case sensitive. Thus, use lower case letters!

	Description	Example
Backup file	Autorestore searches for backup files with the name "setup" and the extension .zip or .json	setup.zip setup.json
Optional name part (prefix)	Name part to identify the backup content.	Wickler-setup.zip Wickler-setup.json
Optional serial number	Serial number Identification of the target device. The file is only used on this device.	Wickler-setup.7261403575795.zip Wickler-setup.7261403575795.json

Search order:

- First, it is searched for a device-specific file with serial number (see line 1).
- If this is not available, the system searches for names without serial number (see line 2).

Search sequence	Description	Name	Example
1	Device-specific (with serial number: 7261403575795)	<Prefix>setup<.SerialNumber>	/setup.7261403575795.zip /setup.7261403575795.json /wickler1-setup.7261403575795.zip /wickler1-setup.7261403575795.json
2	Device-unspecific (without serial number)	<Prefix>setup	/setup.zip /setup.json /wickler1-setup.zip /wickler1-setup.json

Encrypted backups

For encrypted backups, an extra file with a password is required in addition to the backups. As for the backups, these files can be available on the data carrier for different devices.

Search sequence	Storage	Description	Example
1	/<optional-Prefix>setup.<serial-number>.options.json	Device-specific option file	/setup.7261403575795.options.json /wickler1-setup.7261403575795.options.json
2	/<optional-Prefix>setup.options.json	General option file	/setup.options.json /wickler1-setup.options.json

Example: setup.options.json

```
{
  "$password": "secret"
}
```



Certain special characters in JSON strings have to be 'escaped'.

Example:

Password: secretWith"DoubleQuotesAnd\Backslash

Escaped password: secretWith\"DoubleQuotesAnd\\Backslash

One-time backup application

A backup should be applied automatically once when the device is restarted.

After the backup has been uploaded successfully to the ctrlX device, a device-specific file "/setup.<serialnumber>.done" is automatically created on the removable medium preventing repeated execution upon the next device restart.

To execute the backup again when the device is started, delete the file manually from the removable data carrier.

Generated files on the data carrier

Name	Description
/setup.log	Diagnostic information on the last automatic application of a backup upon system startup.
/setup.<Serial-Number>.done	Marker file to prevent repeated application of a backup upon system startup. Considered only on the device with the matching serial number. Note: This file has to be deleted to enable the repeated application of a backup on the device during startup.

Autorestore authorizations

The automatic application of a backup upon system startup requires extensive rights on the device. Authorization keys are used to enable this authorization even without user interaction. A backup can be applied automatically if at least one authorization key on the data carrier matches the content of an authorization key on the ctrlX device (all .pem files are taken into account).

The storage location on the backup data carrier is the root directory.

The storage location on the ctrlX device is *"Settings → Certificates & Keys → Autorun → TAB Keys"*

Creating permission keys on the device

Creating key file manually

1. In the web interface, open the tab *"Settings → Certificates & Keys → Autorun → TAB Keys"*
2. To create a new key, click on [+].
Note: Create the key with the extension .pem.
3. Confirm the dialog.

Uploading key file from another device

1. In the web interface, open the tab *"Settings → Certificates & Keys → Autorun → TAB Keys"*
2. Click on [⬆].
➡ The file explorer opens.
3. Select the key file to be uploaded (.pem file).
4. Confirm the upload.

Apply authorization key to the backup data carrier

1. In the web interface, open the tab *"Settings → Certificates & Keys → Autorun → TAB Keys"*
2. In the list, select the authorization to apply to the backup data carrier.
3. To reset the extended actions, click on [⋮].
4. To download the key file to the engineering PC, click on [⬇] in the extended actions.

5. → Apply the authorization key manually via the file system of the engineering PC to the root directory on the backup data carrier.
The key has to be created as .pem file.
The name of the key file is irrelevant.
Keys of multiple devices can be stored on a backup data carrier.

Diagnostics

Diagnostic information is provided as log file on the external data carrier.
The log file is written to the data carrier upon the regular device start (file name: /setup.log).



Prerequisites to create the log file

For security reasons, the external data carrier is only mounted to the file system if the user "boschrexroth" is available with default password and admin rights on the device or if there is an authorization key on the device.

LED display

Automatic backup application is visualized via the status LED of the ctrlX CORE device. The LED flashes blue during the update.



After booting the control (LED flashes blue), it is determined whether a backup is to be applied automatically.

First, the LED indicates the control state: Run (continuous green light), Warning (flashes yellow)

Depending on the ctrlX device type and the configuration, this can take up to one minute.

The control then switches to the "Automatic backup" status (LED flashes blue).

Device Portal backup

Backup and restore can be triggered via the Device Portal.

6.6.12 Backup – Extensions to the backup file format

This section summarizes the extensions of the backup content in an overview.

Version release XCR-V-0114

Path (ctrlx-setup.json)	Description
system	System settings Refer to: → Backup – System
ssh	SSH settings Refer to: → Backup – SSH
proxy	Proxy settings Refer to: → Backup – Proxy
packageManagement	Installed apps. App binaries can be added manually for installation Refer to: → Backup – Package management (apps)
identityManagement	User accounts, user groups, right settings Refer to: → Backup – User management (IdentityManagement)
hosts	Static "Hosts" table of the device (etc/hosts) Refer to: → Backup – Hosts

Path (ctrlx-setup.json)	Description
dateTime	Time setting and time synchronization Refer to: ↗ Backup – Date and time
connectivity	Network adapter and bridges Refer to: ↗ Backup – Connectivity
certificateManagement	Certificates and keys Refer to ↗ Backup – Certificates and keys

Version release XCR-V-0116

Path (ctrlx-setup.json)	Description
configurations	Active configuration and configuration archive Refer to: ↗ Backup – Active configuration and configuration archive
licenseManagement	Retrieving the licenses Refer to: ↗ Backup – License information
packageManagement	Apps are backed up Refer to: ↗ Backup – Package management (apps)
systemInfo	System information Refer to: ↗ Backup – System information
typeplate	Type plate information Refer to: ↗ Backup – Type plate information

Version release XCR-V-0118

Path (ctrlx-setup.json)	Description
metainfo	\$name / \$version / \$description
certificateManagement	PKIs (public key infrastructure servers) Refer to: ↗ Backup – Certificates and keys
licenseManagement	"Capability Response" file "Applying licensing", see: ↗ Backup – License information
connectivity	Discovery services Refer to: ↗ Backup – Connectivity
System	Theming Refer to: ↗ Backup – System

Version release XCR-V-0120

Path (ctrlx-setup.json)	Description
connectivity	- Advanced DHCP settings - WLAN support (client or hotspot) Refer to: ↗ Backup – Connectivity
identityManagement	Validity of user passwords (password hashes) Refer to: ↗ Backup – User management (IdentityManagement)

Version release XCR-V-0202

Path (ctrlx-setup.json)	Description
connectivity	- Mapping of the interface names (X3, X7) - gratuitous-arp - WiFi advanced security settings Refer to: ↗ Backup – Connectivity
identityManagement	- System-related apps. Kernel, Hardware Support snapd, core - Detailed information on apps Refer to: ↗ Backup – User management(IdentityManagement)

Version release XCR-V-0204

Path (ctrlx-setup.json)	Description
metainfo	\$passwordHash: Checking the backup password (SHA-1)
apps	Settings of third-party apps
connectivity	Encrypted passwords (SHA-1) Refer to: ↗ Backup – Connectivity
proxy	Encrypted passwords (SHA-1) Refer to: ↗ Backup – Proxy
identityManagement	- Named password policies - Encrypted passwords for LDAP server and radius server (SHA-1) Refer to: ↗ Backup – User management(IdentityManagement)
system	Assignment of CPU cores (rt, nrt) Refer to: ↗ Backup – System

Version release XCR-V-0206

Path (ctrlx-setup.json)	Description
metainfo	\$passwordHash: Use of SHA-256 when creating a backup. SHA-1 is still supported when using it.
certificateManagement	- Support of PKCS12 container - Support for extended PKI settings (public key infrastructure server) Refer to: ↗ Backup – Certificates and keys

6.6.13 Backup – Dealing with changes

This section describes how backup and restore is designed to be as compatible as possible with extensions of the system and the backup file format.

There are three essential aspects:

- Version of the Setup app
- Version of the apps on the ctrlX device
- The backup file content

Extensions to the backup file format

New information is introduced additively. Older setup app versions ignore these "unknown" entries. New settings are not taken into account.



If other apps were updated, it is advisable to update the Setup app to apply the complete scope of the backup file.

App data (configuration files) is not interpreted by the Setup app.

New Setup app functionality in "old" system environment

A later Setup app version usually also uses new functions in the device. These accesses fail in an older environment. The Setup app issues corresponding warnings in these cases. The newer settings do not become active.

Updating the Setup app

The Setup app is a system app. It is usually updated with the other system apps, in particular with the Device Admin app. As a rule, a consistent version is installed on the ctrlX device.

An explicit update of the Setup app is possible without risk. The app only includes the functionality to create and apply the device settings to a backup file.

Updating the Setup app does not affect other system functions except for the factory reset function. Backup and restore is implicitly used there to restore the device state.



Updating the Setup app using a backup file

If the Setup app exists as part of the backup file, it is updated first. The updated version of the Setup app then performs the changes in the device. This ensures that the information in the backup file can completely be applied. When applying the backup, no older version of the Setup app can be enabled. There will be no downgrade!

6.6.14 Backup – HowTo's

Backing up device settings before XCR-V-0116

With restrictions, a backup via the command line can also be created by devices as of XCR-V-0112.9.

Prerequisites

- 1 ctrlx-backup.exe has to be installed in a current release (XCR-V-0118 or higher) via ctrlx-works, see: ➔ [Backup – Command line](#)
2. Use a file to hide content not yet available in XCR-V-0112.9. To do this, create a file named "create-options.json" with the following content.

```
CreateOptions
{
  "$content": "files",
  "$format": "zip",
  "certificateManagement": {
    "pkis": {
      "$content": "none"
    }
  },
  "configurations": {
    "active": {
      "$content": "none"
    }
  },
  "hosts": {
    "$content": "none"
  },
  "packageManagement": {
    "$content": "items"
  }
}
```

Create a backup with the following command line.

Replace "localhost:8443" with the IP address of the ctrlX device.

ctrlx-backup.exe is in the installation directory of ctrlx-works.

Command line

```
"C:\Program Files\Rexroth\ctrlX WORKS\ctrlx-backup.exe"
create localhost:8443 --options create-options.json --path
backup.zip --local
```



The option switch "--local" causes the backup functionality of the Windows application to be used instead of the version installed on the device.

Restrictions

In XCR-V-0112.9, the active configuration cannot be backed up. → The active configuration has to be handled explicitly.

- **Before** creating the backup, transfer the active configuration to the archive via "Save configuration".
- **After** installing the backup, activate the configuration from the archive.

In version XCR-V-0112.9, the snap installation files cannot be retrieved from the device. The apps must be downloaded and installed separately from the product portal.

6.6.15 Backup – FAQ

Q: The app is part of the ZIP archive. However, it is not installed. Why?

A: The backup is based on the ctrlx-setup.json file. The app with the respective version has to be stored there.

Q: Why are all settings read first when applying the backup?

A: The settings are read to determine the differences to the target state described in the ctrlx-setup.json. If there are differences, the settings are modified respectively on the device. This procedure allows to apply backups only containing the settings to be changed.

6.7 Hardware reset in case of ctrlX CORE and ctrlX DRIVEplus with integrated ctrlX CORE

Overview

The hardware reset (image update) allows to set the device to a defined basic state using an external data carrier.

The following external data carriers can be used:

- microSD card → ctrlX CORE X3, ctrlX COREplus X3 and ctrlX DRIVEplus with integrated ctrlX CORE
- USB stick → ctrlX CORE X3, ctrlX COREplus X3 and ctrlX CORE X7

This is for example required if:

- The user forgot its username and/or password
- The user does not reach the web interface of the device anymore (e.g. due to an incorrect NAT or firewall configuration)

Effects of the hardware reset

The device is set to its initial state. Only the operating system and the system apps Device Admin, Automation Core, Solutions, Setup and VersionGuard are available.

Removed are:

- All other apps
- App-specific keys and certificates not saved under "Permanent Storage"
- App-specific project data und device configuration
- Device-specific data. This includes for example:
 - Network configuration
 - Configurations of user and group management, e.g. users, groups, password guidelines and session guidelines
 - Further app-specific and device-specific configurations (e.g. Firewall)

The licenses on the device remain

It is mandatory to back up data relevant for the restoration before.



6.7.1 Hardware reset in case of ctrlX CORE X3, ctrlX COREplus X3 and ctrlX DRIVEplus

Required elements

- microSD card or USB stick (only ctrlX CORE X3 and ctrlX COREplus X3), minimum size 4 GB, maximum 32 GB
- Update tool
- Image file

Update tool and image file are provided as complete file structure (X3_System-image_xx.zip) in the Bosch Rexroth ctrlX AUTOMATION Collaboration Room.

Prepare the external data carrier for the hardware reset

1. ➤ Download the complete file structure for the external data carrier (X3_Systemimage_xx.zip) via the Collaboration Room.
2. ➤ Connect the external data carrier to the engineering PC.
3. ➤ Delete existing files from the external data carrier and partition the external data carrier if required. The external data carrier may only have one partition formatted as FAT32.
4. ➤ Unzip the archive "X3_Systemimage_xx.zip" to the external data carrier.

NOTICE

The prepared external data carrier can be used for resets on multiple devices, as the process is based on the serial number and the reset information of devices is stored in different folders on the external data carrier. If the reset should be repeated on the same device, delete the folders created on the external data carrier during the reset. The folder name is the serial number of the ctrlX CORE previously deleted from the external data carrier.

Executing the reset on the ctrlX CORE X3, ctrlX COREplus X3

1. ➤ Switch off the ctrlX CORE (de-energize).
2. ➤ Insert the prepared external data carrier (microSD card into the CF01 slot or the USB stick into the XF01 slot).
3. ➤ Switch on the ctrlX CORE again.
 - ➔ • The diagnostic LED on the device starts flashing red and blue.
 - The update is completed when the LED is not flashing anymore in red and blue.
 - LED flashes blue: Update completed successfully.
 - LED flashes red: An error occurred.

NOTICE

To localize the error, use the flash.log file. The log file is stored on the external data carrier in the folder created during the reset. The folder name is the serial number of the ctrlX CORE

4. ➤ Switch off the ctrlX CORE again (de-energize).
5. ➤ Remove the external data carrier from slot CF01 (microSD card) or XF01 (USB stick).
6. ➤ Switch on the ctrlX CORE again.

NOTICE

The initial start after switching on the system again takes approximately 4-5 minutes.

7. ➤ LED is constantly green: Start completed successfully.

NOTICE

After the successful initialization, the device can be reached under the standard IP address 192.168.1.1.

➔ The reset is completed. Continue commissioning the device.

Executing the reset on the ctrlX DRIVEplus with integrated ctrlX CORE

1. ➤ Switch off the ctrlX DRIVEplus the device (de-energize).
2. ➤ Plug the prepared microSD card into the CF01 slot.
3. ➤ Switch on the ctrlX DRIVEplus device again.
 - ➔ • The diagnostic LED on the device starts flashing red and blue.
 - The update is completed when the LED is not flashing anymore in red and blue.
 - LED flashes blue: Update completed successfully.
 - LED flashes red or in another color: An error occurred during the update or the ctrlX DRIVE device reports an error.

NOTICE

To localize the error, use the flash.log file. The log file is stored on the microSD card in the folder created during the reset. The folder name is the serial number of the ctrlX CORE

4. ➤ Switch off the ctrlX DRIVEplus device again (de-energize).
5. ➤ Remove the microSD card from the CF01 slot.
6. ➤ Switch on the ctrlX DRIVEplus device again.

NOTICE

The initial start after switching on the system again takes approximately 4-5 minutes.

7. ➤ LED is constantly blue: Update completed successfully
 The ctrlX DRIVEplus device overwrites the green state of the ctrlX CORE after startup, since the EtherCAT communication to the ctrlX DRIVEplus is not yet available after the update.

NOTICE

Note: After the successful initialization, the device can be reached under the standard IP address 192.168.1.1

➔ The reset is completed. Continue commissioning the device

6.7.2 Hardware reset in case of ctrlX COREplus X7**Required elements**

- USB stick minimum 4 GB and maximum 32 GB
- Rescue system binary file → bootx64.efi
- Image file → rexroth-arch02.ova

The files are provided as complete USB stick file structure (X7_Systemimage_xx.zip) in the Bosch Rexroth ctrlX AUTOMATION Collaboration Room.

Prepare the external data carrier for the hardware reset

1. ➤ Download the complete USB stick structure (X7_Systemimage_xx.zip) via the Collaboration Room.
2. ➤ Connect the USB stick to the engineering PC.

NOTICE

The USB stick may only contain one **GPT partition** (GUID partition table) formatted as FAT32.

↪ <https://learn.microsoft.com/de-de/windows-server/storage/disk-management/change-an-mbr-disk-into-a-gpt-disk>

3. ➤ Delete existing files from the USB stick and partition the USB stick again if required.
4. ➤ Unzip the archive "X7_Systemimage_xx.zip" to the USB stick.

NOTICE

The prepared USB stick can only be used for one image update at a time.

To repeat the image update, delete the following folders and files created on the USB flash drive during the process:

- /rescue/.done
File signaling that the update is completed and should not be restarted
- /rescue/backup
Backup folder of the persistent partition
- /rescue/flash.log
Contains information on the image update

Executing the reset on the ctrlX COREplus X7

1. ➤ Switch off the ctrlX COREplus X7.
2. ➤ Plug the prepared USB stick into the XF01C slot.

NOTICE

Please note that only the XF01C slot can be used for the image update.

Only use the XF01C slot for the image update and not as an external memory extension at runtime. The slots XF01A and XF01B can be used as external memory extension at runtime.

3. ➤ Switch on the ctrlX COREplus again.
 - ➔ The image update starts automatically.

NOTICE

Do not switch off the ctrlX COREplus X7 until the update is complete.

While updating the image, the ctrlX COREplus X7 is restarted multiple times and the operating system is initialized for the first time. The entire update, including an initial initialization, can take around 4-5 minutes

4. ➤ The image update is complete
 - when the LED is constantly green or
 - when the LED flashes yellow. A diagnostics (warning) is then pending in the ctrlX OS.

NOTICE

Errors that occur during the image (static RED LED) can be localized via the flash.log file.

The log file is created in the "rescue" folder on the USB stick during the update

5. ➤ Switch off the ctrlX COREplus again (de-energize).
6. ➤ Remove the USB stick from the XF01C slot.
7. ➤ Switch on the ctrlX COREplus again.

NOTICE

After the successful initialization, the device can be reached under the standard IP address 192.168.1.1.

- ➔ The reset is completed.
Continue commissioning the device.

6.8 Setting up ctrlX OS

6.8.1 Device identification

The device identification allows to identify every individual device in the ctrlX ecosystem explicitly. ctrlX services (i.e. licensing, remote management, ...) can rely on the identification to be secure and trustworthy.

Due to the different device manufacturers and vendors and heterogeneity of implementations of identifiers (e.g. SMBIOS, electronic typeplate) we need an overarching solution.

Therefore we introduce the ctrlX device ID which uniquely identifies a device in the ctrlX ecosystem.

ctrlX Device ID

The ctrlX device ID is the main identifier for your device and is defined by the ctrlX device certificate.

It is mandatory for your ctrlX OS device and is used as a possible license identifier string, see [Chapter 6.8.3 Licensing a ctrlX OS device on page 128](#) for details.

Schema

The ctrlX device ID follows the structure of [Unified Resource Name](#).

Schema:

urn:ctrlx:device:<partner>:<partnerserial>

While <partner> is provided by Bosch Rexroth, it is up to you, how you define <partnerserial>.

Example:

urn:ctrlx:device:rexroth:08154711

Schema validation

Here is a regex for Go programs to validate a ctrlX device ID:

```
^urn:ctrlx:device:(?P<partner>[-+a-zA-Z0-9]+)(?P<partnerNIDs>(?:[-+a-zA-Z0-9]*)*):(P<partnerserial>[-+a-zA-Z0-9]+)$
```



The ctrlX device ID must be a maximum of 64 characters long.

This is because a Common Name for X.509 certificates may only have up to 64 characters.

See [RFC 5280](#)

You are free to use further Namespace Identifiers (NID) beyond your NID <partner>.

For example, you could introduce a NID to identify different architectures:

- urn:ctrlx:device:<partner>:arch01:fbd47df1865e4932994532fc51162b
- urn:ctrlx:device:<partner>:arch02:15c9e0d98b5a4cc5b4dfb13e97a957

The named capturing groups *partner*, *partnerNID* and *partnerserial* in the regex above are only a suggestion.

ctrlX Device certificate

Each device needs to have a ctrlX device certificate according to [X.509 standard](#).

It contains the ctrlX device ID as a Common Name, and provides standard techniques to provide trust.

Example:

Certificate:

Data:

```
Version: 3 (0x2)
Serial Number: 4660 (0x1234)
Signature Algorithm: ecdsa-with-SHA256
Issuer: C=DE, ST=Bavaria, L=Lohr am Main, O=Bosch Rexroth AG, CN=com.ctrlx-automation.os.rexroth
Validity
  Not Before: Jul  3 07:40:13 2023 GMT
  Not After : Dec 31 23:59:59 9999 GMT
Subject: C=DE, ST=Bavaria, L=Lohr am Main, O=Bosch Rexroth AG, CN=urn:ctrlx:device:rexroth:08154711
Subject Public Key Info:
  Public Key Algorithm: id-ecPublicKey
  [public key information]
X509v3 extensions:
  [extensions ...]
X509v3 CRL Distribution Points:
  Full Name:
    URI:http://your-company.com/path-to-crl
Signature Algorithm: ecdsa-with-SHA256
[signature]
```

The corresponding private key must be bound to the TPM (Trusted Platform Module) and must be of type ECC with 256 bits ("prime256v1").

Public Key Infrastructure (PKI)

For a better understanding, it is recommended to have a basic knowledge of [PKI](#) (Public Key Infrastructure).

With the help of the PKI and certificates, it is possible to maintain trust. If you trust the "Certificate Authority" (CA), also the "Subordinate Certificate Authority" (Sub-CA) and the devices by means of their certificates can be trusted. All this together is called "chain of trust".

The basic structure of this "chain of trust" is as follows:

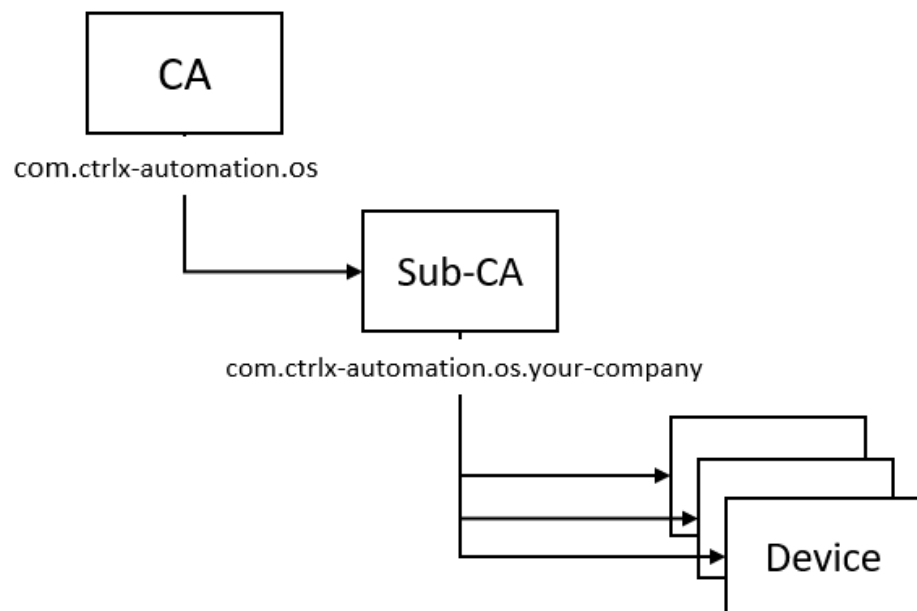


Fig. 7: Process for device registration

Device manufacturers have the responsibility of operating a "Subordinate Certificate Authority" (Sub-CA) according to state-of-the-art of security.

Please be aware of the following:

- As ctrlX device certificates are valid for the lifetime of the device, it is highly recommended providing a path to a certificate revocation list (CRL) in your ctrlX device certificates.
- Certificate authorities are very sensitive in the context of security. If there are any issues with a Sub-CA (e.g. signing key leakage), it is a common technique to revoke its certificate.

6.8.2 Requesting a device certificate for your device

For this article, it is mandatory to understand the concept of .



The certification process outlined here is provisional in nature. We retain the right to revoke any device certificates issued through this method.



Please be advised that this process is intended solely for testing purposes and should not be employed in a production environment

Generating a device key

1. ➤ Open the window “Settings” on your device.
2. ➤ Select the submenu “Certificates & Keys”.

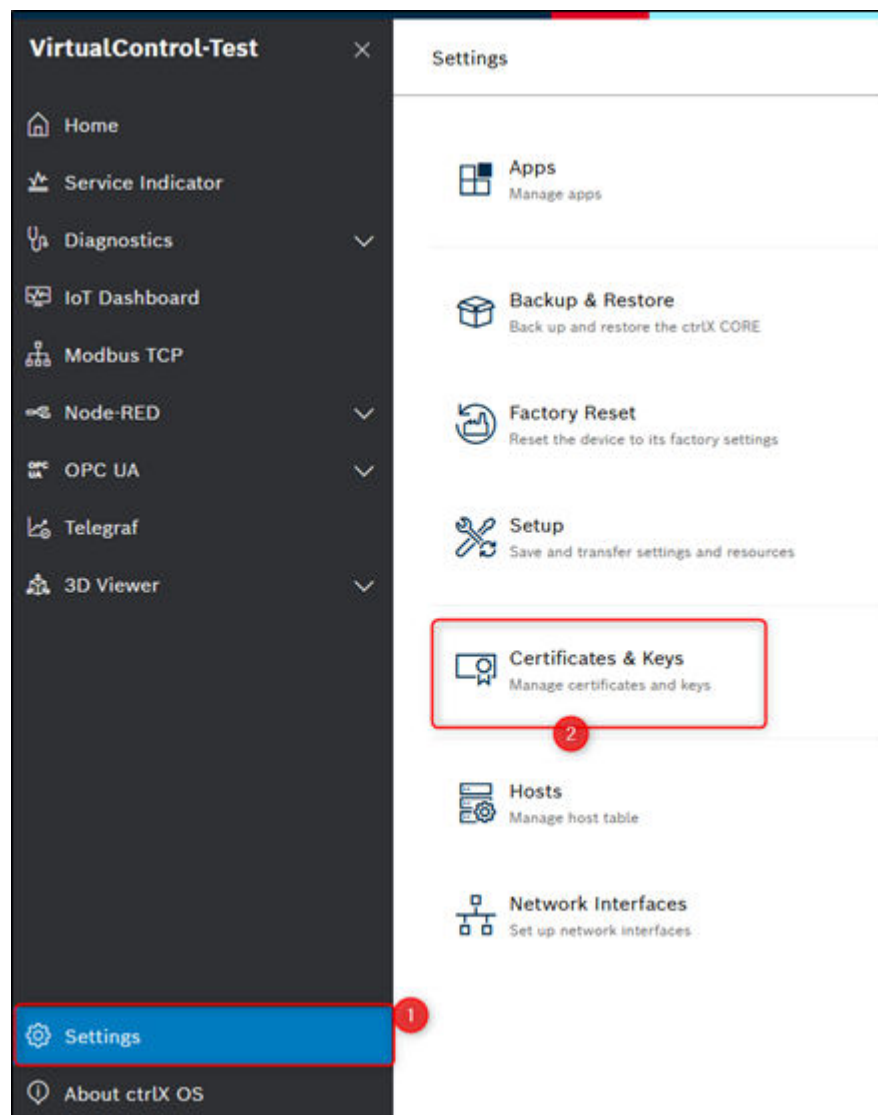


Fig. 8: “Settings → Certificates & Keys”

3. Select the service “Permanent Storage” in the table.

Service	Description
Data Layer	Stores keys that are used for secure Data Layer communication (curveMQ)
Web Server	Manage webserver and token validation credentials
Network security	Keys and certificates required for client and server-side authentication
SSH	Manage the SSH host keys
Setup	Authentication keys required for autorun
Permanent Storage	Certificates and keys in this storage will not be deleted upon a factory reset

Fig. 9: Service “Permanent Storage”

4. Click on the “Keys” tab and on “Generate key”.

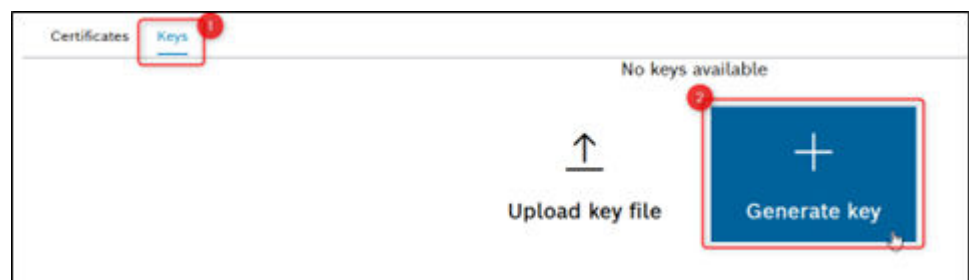


Fig. 10: Generate device key

5. Generate the device key in the “Generate key” dialog with the following settings:

- “Name”: device
- Algorithm: P-256
- Click on the checkbox “Secure key by HSM”
- Extension: .hsm

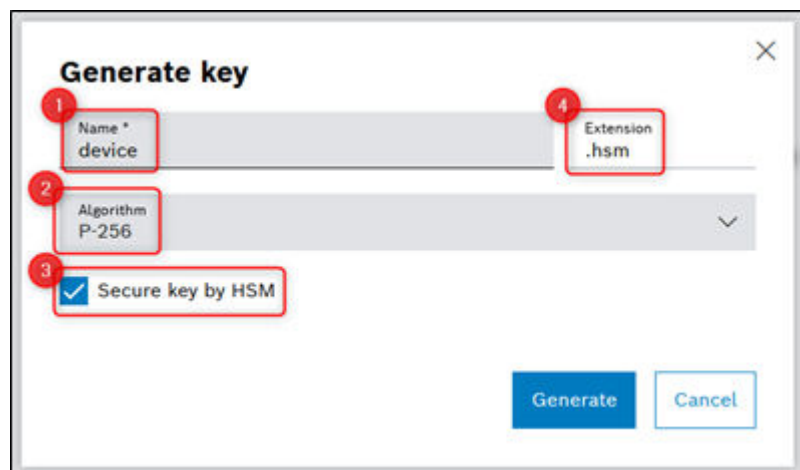


Fig. 11: Settings device key

- ➔ After successfully creating the key, it will be displayed as shown below.

Certificates Keys							
1 item							
Name	Category	Type	Algorithm	Length	HSM secured	Status	Actions
device.hsm	Own	Private	P-256	256	Yes	Available	...

Fig. 12: Device keys overview



In case the key is removed from the device (i.e. due to an unintended destructive flash) you can not restore the device certificate without first restoring the device key.

If you lose your device key you will have to create a new device key and request a new device certificate.

As a consequence you will also have to migrate all services that depend on the device certificate as a secure trust anchor (i.e. app licenses).

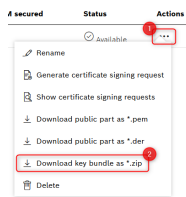


Fig. 13: Manual backup of the device key

Generating certificate signing request

After generating the key, you must generate the signing request for the certificate.

Please follow the steps below, to do so.

1. Click on the symbol ... in the column “Actions”.
2. Select “Generate certificate signing request” as demonstrated below.

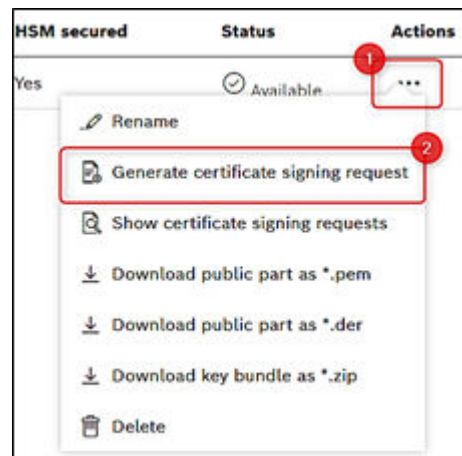
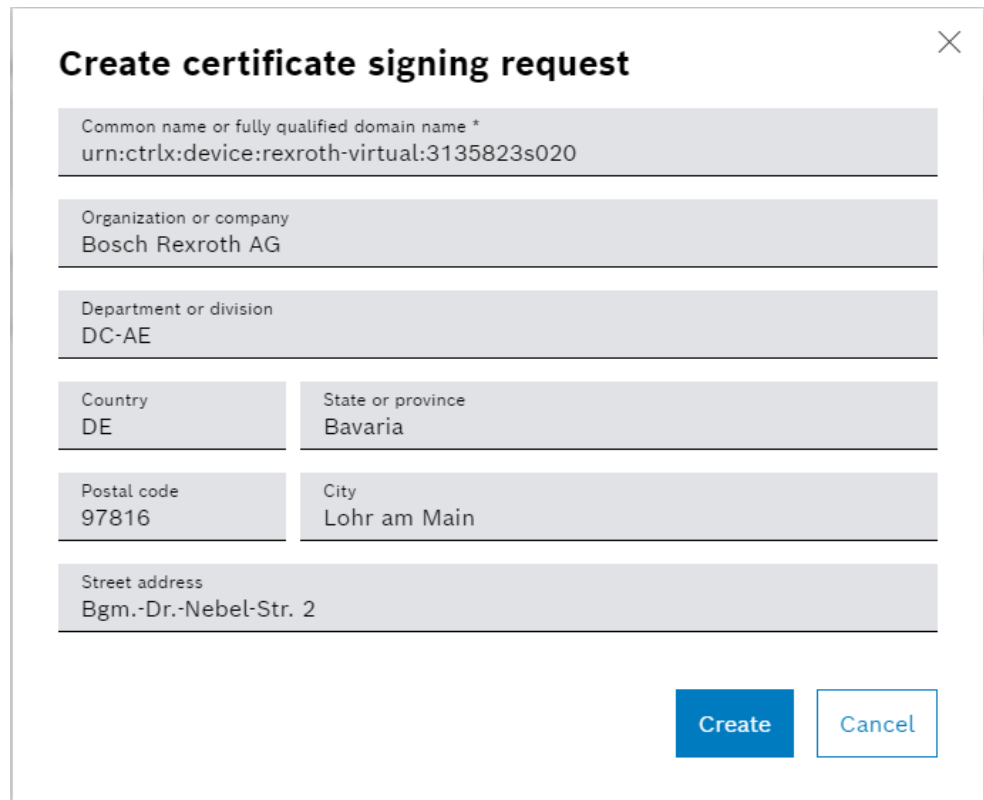


Fig. 14: Generate certificate signing request (CSR) device certificate

➔ A dialog opens.

3. In the dialog enter the required information.



The dialog titled "Create certificate signing request" contains the following fields:

- Common name or fully qualified domain name *: urn:ctrlx:device:rexroth-virtual:3135823s020
- Organization or company: Bosch Rexroth AG
- Department or division: DC-AE
- Country: DE
- State or province: Bavaria
- Postal code: 97816
- City: Lohr am Main
- Street address: Bgm.-Dr.-Nebel-Str. 2

Buttons: Create, Cancel

Fig. 15: Example for a CSR for the device certificate

- ➔ The common name must be a ➔ ctrlX device ID with the following schema:
urn:ctrlx:device:rexroth-virtual:<your-device-serial>
4. After filling out the form, kindly proceed by clicking on button "Create".
- ➔ Once the certificate signing request is successfully generated, a new dialog will open.
5. Please proceed by downloading the .pem file.



Fig. 16: Download the CSR for the device certificate

- ➔ The downloaded file should typically be named something like *device_csr.pem*.

Requesting the device certificate

Please send the certificate signing request to the following mail address:

➔ ctrlX.os.license@boschrexroth.de

Please provide the following information:

- Reason and application area (e.g. testing new hardware, integration test track)
- The ".pem" file as an attachment

You will receive an email containing the certificate within a few days.

Uploading the device certificate

1. ➤ After receiving the certificate, please check the following:
 - File name must be "device.crt"
 - Category must be "Own"
 2. ➤ Navigate to the "Certificates" tab.
 3. ➤ Upload the certificate file with .
 4. ➤ Select your "device.crt" file.
 - ➔  To enable the device certificate, a reboot of the device is required.
- Your device should now be successfully certified.

Automation



This process can be achieved easily with the [Certificates API](#).

Do not forget to register your device for licensing ([Chapter Registering a new device for licensing on page 128](#)).

6.8.3 Licensing a ctrlX OS device

Secure hardware anchor for licensing

Licenses are tied to a secure hardware anchor.

This can be one of the following:

- A µSD licensing dongle plugged into your device
 - Get in contact with your sales partner to order a µSD licensing dongle
- A device certificate with a [ctrlX Device ID](#) that is tied to the TPM (Trusted Platform Module) of the device



Licensing dongles do not come with a pre-assigned ctrlX OS System License - Standard Runtime. You will need to purchase the licenses you require and assign them to your licensing dongle.

Registering a new device for licensing

You will need to make your ctrlX OS device known to our licensing infrastructure during manufacturing so that you can assign licenses to it.

See [FAQ Licensing with ctrlX AUTOMATION](#) on how to register your ctrlX OS device.



Make sure to register your device with the same ctrlX Device ID as in your device certificate.

ctrlX Device ID schema:

urn:ctrlx:device:<partner>:<partnerserial>



ctrlX CORE devices (and the licensing dongles) are already registered by Bosch Rexroth.

It might be that you need to [CLAIM](#) the ctrlX CORE devices if they are not visible for you.

Manual licensing

If you are new to licensing a ctrlX OS device start with the [Quick Start Guide Licensing](#).

For more detailed information you can have a look at [↗ FAQ Licensing with ctrlX AUTOMATION](#).

Trial licenses for development

It might be that you want to use additional apps of the ctrlX AUTOMATION ecosystem in your system design.

For development purposes you can use [↗ trial licenses](#) for these apps.



Once a trial license runs out you can renew it. This can be done for up to 150 days.

6.8.4 Enabling Secure Boot

Prerequisites

If you are new to the topic of Secure Boot on UEFI systems or want to refresh your memory, Rod Smith provides a great resource for learning the concepts of Secure Boot and some practical examples on his [↗ website](#).

Note that the target audience of the article series seems to be the owner of a personal computer, rather than a hardware manufacturer or system integrator.

Nevertheless, the rest of this article assumes that you are already familiar with the topic or read the articles linked above.

The [↗ Arch Linux wiki article on Secure Boot](#) offers a more concise overview.



Secure Boot is mandatory for using ctrlX OS in production.

If you deactivate Secure Boot for your devices in production it will break the ctrlX OS security concept.

Default ctrlX OS vs. customized ctrlX OS

How to enable your device for Secure Boot depends on what kind of ctrlX OS you want to deploy on your device:

- Default ctrlX OS as provided by Bosch Rexroth
- Your customized ctrlX OS image

Secure Boot for default ctrlX OS

For Secure Boot to accept the EFI binaries contained in the default ctrlX OS image you will have to add the public Bosch Rexroth certificate to your device's DB.



The public certificates for default ctrlX OS are available in the ctrlX OS Collaboration Room.

You will need to be in control of the KEK on your device to enroll the certificate into the DB yourself.

See the chapter [↗ Chapter Deploying your Secure Boot certificates to device NVRAM on page 131](#).



While you can build the default ctrlX OS image from source you can only use it for development purposes with Secure Boot disabled on your test device. As you do not have access to the Bosch Rexroth private keys you can not sign the default kernel and gadget EFI binaries in your own build.

If you want to use default ctrlX OS for production you will have to use the image provided in the Collaboration Room.

Steps to own Secure Boot for customized ctrlX OS

In order to take control of Secure Boot with a customized ctrlX OS image you have to note the following:

- You have to maintain your own Secure Boot keys.
- You have to sign your EFI binaries (e.g. your customized kernel) with your Secure Boot keys.
- Deploy your public Secure Boot certificates on each device you manufacture.

Managing your Secure Boot keys

An example on how to create Secure Boot keys with OpenSSL can be found here: [↗ Creating Secure Boot keys](#)

NOTICE

Your private Secure Boot keys are very sensitive data. If they are leaked attackers may take control of your devices.
Make sure you take appropriate measures to protect them.

We recommend using an [↗ HSM \(hardware security module\)](#) to secure and restrict access to your private keys.

This can range from a low-cost solution (e.g. Nitrokey) to sophisticated solutions for your company.

How to handle your HSM securely depends on your company and processes. Take into account how to restrict access to the HSM to authorized personnel and how to back up your private keys.

If you use a low-cost solution like a portable HSM we recommend using a separate HSM for your PK and KEK private keys and for your DB private keys.

In case the portable HSM with the DB private key gets leaked you still have control over the PK and KEK to revoke the leaked DB key.

The DB private key will need to be made accessible for your build environment so that you can sign your EFI binaries for Secure Boot.

Signing EFI binaries for Secure Boot with a customized ctrlX OS

Use the "sbsigntools" and the "sbsign" command to sign your EFI binaries:

- Shim
- GRUB
- Kernel
- MOKManager



While it is possible to use MOK keys to take control of your Secure Boot we do not recommend it. Enrolling MOK keys on your device requires lots of manual effort (see [↗ here](#)). Thus adding a MOKManager mmx64.efi binary to your image is optional.

An example can be found here: [↗ Signing EFI Binaries](#)

You do not need to build the Shim and GRUB binaries yourself. Instead, we recommend to use prebuilt binaries already signed by Canonical.

You can then add your own signature to these binaries via dual signing:

```
sbsign --engine pkcs11 --key $(SECUREBOOT_DB_KEY) --cert $(SECUREBOOT_DB_CRT) \  
--output shim.efi.signed $(SNAPCRAFT_STAGE)/usr/lib/shim/shimx64.efi.signed; \  
sbsign --engine pkcs11 --key $(SECUREBOOT_DB_KEY) --cert $(SECUREBOOT_DB_CRT) \  
--output mmx64.efi $(SNAPCRAFT_STAGE)/usr/lib/shim/mmx64.efi; \  
sbsign --engine pkcs11 --key $(SECUREBOOT_DB_KEY) --cert $(SECUREBOOT_DB_CRT) \  
--output grubx64.efi $(SNAPCRAFT_STAGE)/usr/lib/grub/x86_64-efi-signed/  
grubx64.efi.signed; \  

```

Make sure to include the signed Shim and GRUB binaries (as well as the optional MOKManager) in your customized gadget snap via the snapcraft.yaml.

Your customized kernel snap must contain a signed kernel binary.

Deploying your Secure Boot certificates to device NVRAM

Here are some options to enroll Secure Boot certificates on your devices:

- Deploy the Secure Boot certificates manually.
E.g. by copying your certificates to a USB stick and using your UEFI to add them to your Secure Boot databases.
- Build LockDown.efi from the efitools package and use it to enroll your Secure Boot certificates on your devices during manufacturing (see [example](#)).



LockDown.efi can handle certificates only in the .crt format. Many UEFI implementations accept only .der format and thus are incompatible with Lock-Down.efi.

- For the ctrlX CORE devices we have an arrangement with our board manufacturer, so that devices already contain our public Secure Boot certificates in their NVRAM.

Target state for your device

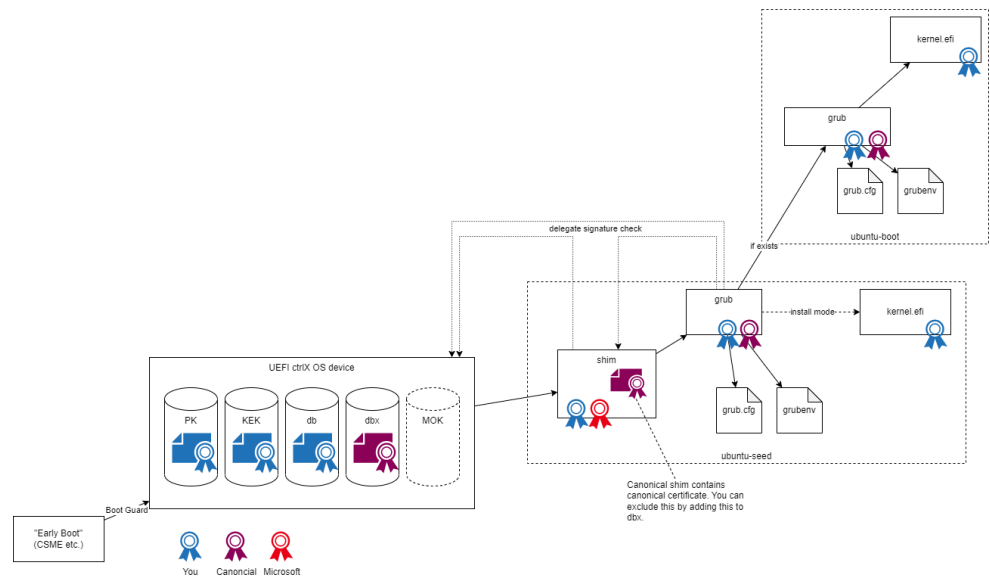


Fig. 17: Target Secure Boot setup on your device

You can use the efi-reader command provided by efitools to have a look at the Secure Boot keys on your device (see [here](#)).

Revoking keys and hashes on your device

See [this example](#) for key revocation and considerations of the [Boot Hole](#) vulnerabilities.

As only your Secure Boot certificates are deployed to the device it is up to you to use a secure version of GRUB. Secure Boot will prevent the installation of a malicious GRUB binary that is signed with a 3rd party key.

UEFI password protection



The UEFI password protection is mandatory!

If you do not use UEFI password protection it will break the ctrlX OS security concept!

Secure your device UEFI settings via a password. We recommend that the password is unique for each device.

How to set a password depends on your UEFI (Unified Extensible Firmware Interface).

Make sure to safely store these passwords, e.g. for service usecases.

6.8.5 Image conversion

Precondition

- Providing an image in the ".ova" format
- Installing "qemu-img"
 - "qemu-img" is provided with your QEMU installation.
 - Note that "qemu-img" is not automatically added to the default path during installation and needs to be manually added first.
 - In a default QEMU installation on a Windows host "qemu-img" can be found at *C:\Program Files\qemu*.
 - For tips on how to use "qemu-img" you can see also ➔ <https://docs.openstack.org/image-guide/convert-images.html>.

Conversion

Target format	Conversion guidance	Example	Use case
.ova	-	-	Vsphere, VirtualBox, Proxmox, KVM, ...
.vmdk	Untar the ".ova" file (e.g. using 7-Zip)	-	Vsphere, VirtualBox, Proxmox, KVM, ...
.img	Use "qemu-img"	<code>qemu-img convert -f vmdk -O raw image.vmdk image.img</code>	Creating a bootable USB stick or harddrive
.qcow2	Use "qemu-img" to convert the ".vmdk" file	<code>qemu-img convert -f vmdk -O qcow2 image.vmdk image.qcow2</code>	QEMU (Note: Rather use ".vmdk" due to compatibility reasons)

7 Diagnostics

7.1 Logbook – Overview

Log messages of the ctrlX applications are shown in the logbook.

Use filters to limit the number of shown log messages for analytic purposes.

There are three different types of log messages:

- **Diagnostic messages:**
The user right "View diagnostic logs" is required
- **Trace messages:**
The user right "View trace logs" is required
- **System messages:**
The user right "View system logs" is required

To configure the user rights, refer to ➔ [Chapter 13.2.19 Window – “Users” on page 305](#)

Mechanism to save logbook messages

The Linux mechanism "journal" is used as storage medium for logbook messages. journal uses a ring memory for storage.



Depending on the runtime of the control and the number of issued messages, later messages might be shown in the ring memory instead of older messages. Thus, not all messages can be shown in the logbook. This response also affects the export function of the logbook.

Note that Linux system messages might be shown instead of diagnostic messages which are not shown in the logbook when the default setting is used. In this case, no diagnostic messages might be displayed in the logbook.

Logbook messages are not saved persistently in the factory setting of the ctrlX device. Thus, all accumulated log messages are no longer available after a ctrlX device restart. Persistent storage can be enabled in the logbook settings, see:

➔ [Dialog – “Settings”](#) (Logbook)

Settings

Different settings of the logbook can be made in the “Settings” dialog (only available with the user right “View system logs”) such as:

- Persistent storage of log messages.
- Configuration of the displayed log data (standard/developer view).

Refer to: ➔ [Dialog – “Settings”](#) (Logbook)

Exporting log messages

Use an export function to export log messages to a table, see:

➔ [Logbook – Exporting](#)



Automated/remote retrieval of the log

The log can be read out using the REST API. Bosch Rexroth provides the required API specification if necessary, see ➔ [Chapter 15 Service and support](#) on page 383.

Call

ctrlX OS side navigation “*Diagnostics → Logbook*”

Further information

- ➔ [Chapter 13.2.2 Window – Logbook](#) on page 271

7.2 Logbook – Exporting

Entries in the logbook of the ctrlX device can be exported, e.g. for archiving purposes or for the error analysis by the system manufacturer or a service provider.

The amount of data of the export file depends on the following factors:

- Export of the currently pending messages via the “Alarms” tab or export of the message history via the “History” tab, see the following descriptions.
- Set view of the logbook (standard/developer view), see:

➔ [Dialog – “Settings”](#) (Logbook)

- User-specific configuration of the displayed data columns, see:

➔ [Dialog – “Select columns”](#) (Logbook)

Perform export in the “Alarms” tab

- ➔ 1. Open the logbook in the ctrlX OS web interface and open the “Alarms” tab, see:

➔ [Tab – “Alarms”](#)

2. In the command bar, click on [↓].
 - ➔ The export file is created and stored in the "Download" directory on the local PC.



The export file of the “Alarms” tab only contains the currently pending errors and warnings.

Perform export in the “History” tab

1. Open the logbook in the ctrlX OS web interface and open the “History” tab, see:
➔ Tab – “History”
2. In the command bar, click on [↓].
 - ➔ A submenu with the following selection options opens:
 - Download filtered data (disabled if no filter is set).
 - Download all data.
3. Select one of the options and confirm the dialog.
 - ➔ The export file is created and stored in the "Download" directory on the local PC.



Please note:

The number of log messages varies depending on the configured selection and can include a large number of entries.

The number of log messages per export file is limited to a maximum of 10,000 entries.

If there are more than 10,000 entries, these are divided into further export files.

Structure/description of the export file – in the standard view

Column header	Description
Level	Message category (Information / Warning / Error)
Date	Time stamp of the diagnostic entry: • For ctrlX OS diagnostics, the point in time when the entry was received in the ctrlX OS diagnostic system, see: journal Metadata R_TIMESTAMP • For Linux journal diagnostics, the point in time when the entry was received in the Linux journal diagnostic system, see: journal Metadata REALTIME_TIMESTAMP Note: In an exported CSV file, the fields "Date" and "Journal date" are identical in this case
Main diagnostic code	Main diagnostic number (unique 32 bit number)
Detailed diagnostic code	Detailed diagnostic number (unique 32 bit number with direct reference to the main diagnostic number)
Description 1	Short text describing the meaning of the main diagnostic number. The exact meaning of the main diagnostic number is described in detail in the documentation of the respective diagnostic number.
Description 2	Short text describing the meaning of the detailed diagnostic number. The exact meaning of the detailed diagnostic number is described in detail in the documentation of the respective diagnostic number.
Description 3	Additional log description that can output dynamic elements

Column header	Description
Source 1	Entity/resource Example: The entity corresponds to the system URI of the Data Layer address if the resource is provided with a respective address.
Source2	Origin (Name of the application/snap).

Structure/description of the export file – in the developer view

The number of exported datasets and the information per dataset (columns in the table) depend on the set configuration.

Also refer to: [↗ Log structure \(message format\) *journald* and its metadata](#)

Column header	Description
Level	Message category (Information / Warning / Error)
Date	Time stamp of the diagnostic entry: - For ctrlX OS diagnostics, the point in time when the entry was received in the ctrlX OS diagnostic system, see: journal Metadata R_TIMESTAMP - For Linux journal diagnostics, the point in time when the entry was received in the Linux journal diagnostic system, see: journal Metadata REALTIME_TIMESTAMP Note: In an exported CSV file, the fields "Date" and "Journal date" are identical in this case
Main diagnostic code	Main diagnostic number (unique 32 bit number)
Detailed diagnostic code	Detailed diagnostic number (unique 32 bit number with direct reference to the main diagnostic number)
Main diagnostic text	Short text describing the meaning of the main diagnostic number. The exact meaning of the main diagnostic number is described in detail in the documentation of the respective diagnostic number.
Detailed diagnostic text	Short text describing the meaning of the detailed diagnostic number. The exact meaning of the detailed diagnostic number is described in detail in the documentation of the respective diagnostic number.
Dynamic description	Additional log description that can output dynamic elements
Entity	Entity/resource Example: The entity corresponds to the system URI of the Data Layer address if the resource is provided with a respective address.
Origin	Origin (Name of the application/snap).
Unit name	If the journal entry contains the Rexroth-specific journal metadata "R_UNIT", it is used. Otherwise, the journal metadata "_SYSTEMD_UNIT" is used.
File name	Name of the file in which the log entry is issued
Function name	Name of the function in which the log entry is issued. The name is output with namespaces if required.
Line number	Line number at which the log entry is issued
User ID	Unique user ID. This element is only filled in case of user-relevant and security-relevant logs.
Message type	Logbook message type: - - - n - rexroth_diag: Diagnostic messages - rexroth_trace: Trace messages - system: System messages

Column header	Description
Journal date	See "Date" above

7.3 Remote Logging – Configuring external server

The logbook of the ctrlX device allows to load diagnostic messages to one or multiple external logging servers. The "Syslog" protocol is responsible for the transmission between the ctrlX device and the server, also refer to: ➔ <https://wikipedia.org/wiki/Syslog>

Use cases

- Option to send diagnostic messages to a parent master system
- Archiving of diagnostic messages, as they are not saved persistently on the ctrlX device
- Reduction of memory requirement on the ctrlX device

Related topics

➔ [Chapter 13.2.2 Window – Logbook on page 271](#)

7.3.1 Operating principle

The ctrlX device sends copies of all incoming logbook entries to the configured and enabled logging servers. The copy is immediately transmitted after the entry into the logbook of the ctrlX device and with regard to the message filter settings, see: ➔ [Editor – “Remote Logging”](#)

7.3.2 System requirements

The server application on the host PC has to support at least the Syslog protocol RFC3164. A permanent network connection to the ctrlX device is required.

7.3.3 Server reference sources

Rexroth does not provide any Syslog server applications.

However, there are some commercial providers for Syslog server applications and open source sources.

7.3.4 Configuring the external server in the ctrlX device

To allow a connection between the ctrlX device and the external server, connection data of the server has to be configured in the ctrlX device.

Configuring the connection

- ➔ Open the web interface of the ctrlX device and navigate to the window: *“Diagnostics – Logbook”*
- ➔ In the header, click on “Remote Logging”.
 - ➔ An window opens that displays the configured server connections (empty upon delivery), see ➔ [Window – Remote Logging](#)
- ➔ To configure a new Syslog server, click on + “Add remote logging server”
 - ➔ The “Add remote logging server” dialog opens, see ➔ [Dialog – “Add remote logging server”](#)
- ➔ Enter the required information and confirm the dialog with “Add”
 - ➔ The new connection to the Syslog server is set up and shown in the window.

7.3.5 Evaluation of server log messages

Log messages saved on the Syslog server cannot be retrieved via the ctrlX OS web interface. To retrieve log messages, use the Syslog server application.

7.3.6 Backing up and restoring the configuration

The configured connections and message filters on the remote logging server are saved in the app data in the folder „remote-logging“ and have to be included when backing up and restoring app data, see [Backup – Introduction](#)

7.3.7 Encrypted transmission

Encrypted transmission can be set using the protocol selection in the “Add remote logging server” dialog or subsequently in the “Server settings” tab in the server configuration window. The certificate and key for TLS are uploaded/selected in the “TLS encryption” tab in the configuration window.

7.4 LED states

LED states for ctrlX CORE

The following functions are assigned to the ctrlX CORE status LEDs when the system firmware is running:

State	Color	P1 0..7	P2 0..3	LED Flashing pattern ¹⁾
System error (F8 and F9) occurred in the ctrlX OS Runtime	Red	0	0	
Bootng active or shut down	Flashing blue	0	1	
ctrlX OS Rescue System is active	Flashing red and blue	0	2	
Identification	Flashing white	0	3	
Setup active	Flashing blue	1	0	
App execution faulty	Flashing red	1	1	
-	Blue	1	2	
-	Flashing red	2	0	
-	Flashing red	2	1	
Error (F0 – F7) occurred in ctrlX OS Runtime	Flashing red	2	2	
Warning occurred in the ctrlX OS Runtime	Flashing yellow	3	1	
-	Flashing red	4	0	
System state "Service/Setup"	Blue	4	3	
-	Flashing green and yellow	5	1	
System state "Partial Operating"	Green flashing	5	3	
-	Green	6	0	
-	Green flashing	6	1	
-	Green	6	2	
System state "Operating"	Green	6	3	
		7	3	

Legend:

1): Flashing pattern: One square corresponds to a period of 200 ms; the arrow indicates the end of a cycle; designations of the squares: WH = LED flashes white, YE = LED flashes yellow, BU = LED flashes blue, GN = LED flashes green, RD = LED flashes red, -- = LED is not on

The diagnostic states are prioritized. The state with the highest priority is displayed on the diagnostic LED:

(highest priority) system error → Error message → Warning → Operating state (lowest priority)



A new status is only displayed when the previous flashing cycle expired. A status change can therefore be displayed with a delay of up to two seconds.

LED states for ctrlX DRIVE

For the LED states for ctrlX DRIVE, refer to the following documentation:

- "ctrlX DRIVE Diagnostic Messages of Runtime AXS-V-xxRS", DOK-XDRV**-GEN*-DIAG**-RERS-EN-P (chapter → ["Display at the device"](#))

7.5 Creating system report in ctrlX OS

The ctrlX OS system report allows to easily collect information on the current device, e.g. to support service requests or for troubleshooting and error analysis. The ctrlX OS system report is provided for download as ZIP archive and can be customized via options, e.g. collection period, app data or password protection.

Included elements:

- Device settings
- Logbook entries
- App version information
- Active app data (optional)

Notes on use

- The system report can only determine information accessible to the current user with his/her user rights.
For a complete system report, the creator should therefore have administrator rights.
- To create the system report, the backup functionality is for example used in the ctrlX.
If another backup file is still present, it is overwritten when creating the system report.

Procedure to create a system report

1. Click on [?] in the header of the ctrlX OS web interface
➔ The "Help" menu is shown
2. Click on [📦 ctrlX OS system report]
➔ The "Create system report" dialog is opened
3. Select the options to create the system report:
 - Collection period
 - Collecting active app data (yes/no)
4. Confirm your inputs with [Next]
➔ The dialog changes to the step to enter additional information.
The information is optional and added to the system report as "readme.txt" text file. The already existing entries (name, email and description) are only a template.
5. Confirm the dialog with [Next]
➔ The dialog changes to the step to enter an optional password to encrypt the entire system report using the AES 256-bit encryption method

6.  Optionally enter a password and confirm the dialog with [Create]
- ➔ The system report is now created and stored in the browser download directory
 - ➔ If any warnings or errors occurred while creating the system report, the "Log" dialog is displayed. The list shows the creation process.

Creating a system report with limited user rights

When creating the system report, only information available to the logged-in user can be detected. If certain information cannot be accessed due to missing user rights, a note is displayed indicating that the system report is incomplete. In addition, information that cannot be included in the system report is displayed. Due to missing user rights, it is possible that app data is not included in the system report.

Example: If the user does not have the user right to read app data, its data cannot be retrieved for the system report.

Content of the system report

The system report contains a backup file (ctrlx-setup.zip) and log files provided together in a ZIP archive for download. Password-protected archives can be viewed with 7-zip for example.

Elements of the system report:

- Directory logbook
- Directory logs
- Directory snapd
- File ctrlx-setup.zip
Backup file with the device settings such as apps and licenses used, network settings or data of the active configuration ("App data" if selected).
- File readme.txt
Contains additional information entered into the dialog "Create system report" by the user.
- File systemreport.log
Contains the name of the creator ("user") and a list of events ("log") during creation.

Related topics

[➔ ctrlX OS web interface – Overview](#)

7.6 Control monitoring

The following control states are monitored:

- System temperature
- Battery status
- Fan
- eMMC

If limit values defined by the hardware are exceeded, a diagnostics is issued:

Main diagnostic number	Meaning	Detailed diagnostic number	Meaning
➔ 080F0540	Error during control monitoring	➔ 0C66003E	Control battery empty
		➔ 0C66003D	Service life of the integrated memory medium exceeded

Main diagnostic number	Meaning	Detailed diagnostic number	Meaning
		→ 0C660038	Fan failure
		→ 0C660039	Hardware fault
→ 080E0540	Warning during control monitoring	→ 0C66003A	High control temperature
		→ 0C66003B	The service life of the integrated memory medium is nearly elapsing
→ 080F9540	Critical error during control monitoring	→ 0C66003C	Control temperature exceeds maximum value

The sensors to be available are defined for each hardware:

Hardware	Required hardware
M3	battery core-temp cpu-temp emmc system-temp watchdog fan
M4	battery core-temp cpu-temp emmc system-temp watchdog

The control state is shown in the Data Layer:

Node	Description	Data type
<i>health/*</i>	Status of the hardware component • TRUE: Hardware OK • FALSE: Hardware not OK	BOOL
<i>health/*/value</i>	Main value of the hardware component	Variant
<i>health/*/last-checked</i>	Time when the hardware was checked last	STRING
<i>health/*/thresholds</i>	Limit values (threshold) of the hardware component • value: Value of the limit value • mainDiagnosisCode: Main diagnostics issued when exceeding the limit value • detailedDiagnosisCode: Detailed diagnostics issued when exceeding the limit value	<i>types/health/thresholds</i>

8 Device interfaces

8.1 Interface description

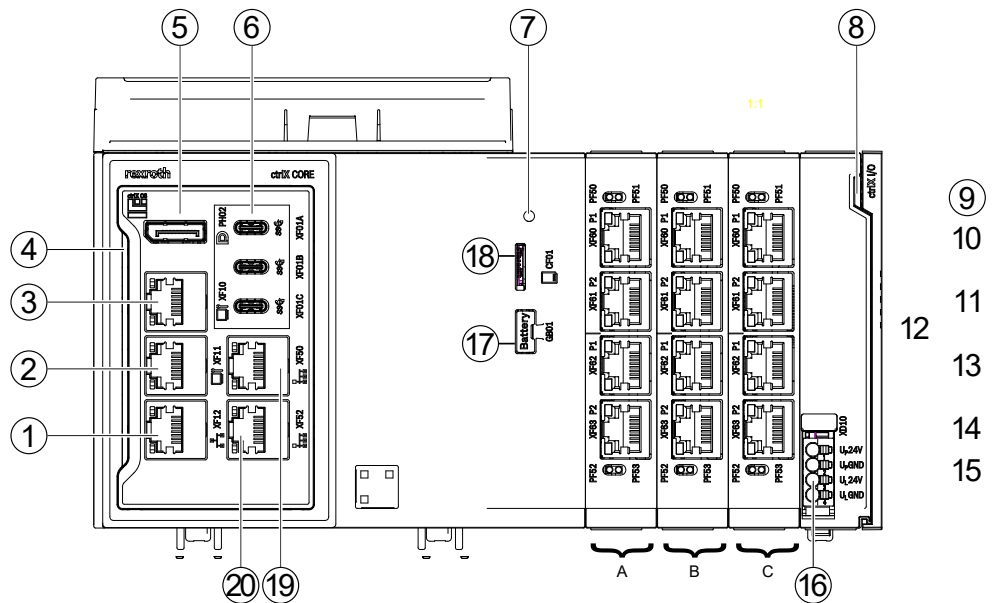


Fig. 18: Interfaces

- A Extension option 1 (optional)
B Extension option 2 (optional)
C Extension option 3 (optional)

No.	Name	Connection type	Connector type (Integrated)	Mating connector and cable (From outside)
①	XF12	Ethernet	RJ45 socket	RJ45 plug
②	XF11	10/100/1000 MBit	8-pin	(twisted pair, 8-wire)
③	XF10	HMI and engineering port		
④	–	ctrlX CORE status LED	–	–
⑤	PH02	Displayport	Displayport socket	Displayport plug
⑥	XF01A XF01B XF01C	USB host, USB 3.1	USB socket, type C	USB plug, type C
⑦	–	Pushbutton	–	–
⑧	–	Device status LED	–	–
⑨	PF50 PF51	Status display PF50 Status display PF51	–	–
⑩	XF60	Multi-Ethernet 10/100 MBit field bus master, field bus slave (configurable)	RJ45 socket 8-pin	RJ45 socket (twisted in pairs, 8-wire)
⑪	XF61	Multi-Ethernet 10/100 MBit field bus master, field bus slave (configurable)	RJ45 socket 8-pin	RJ45 socket (twisted pair, 8-wire)
⑫	–	End cover	–	–
⑬	XF62	Multi-Ethernet 10/100 MBit field bus master, slave (configurable)	RJ45 socket 8-pin	RJ45 socket (twisted pair, 8-wire)
⑭	XF63	Multi-Ethernet 10/100 MBit field bus master, field bus slave (configurable)	RJ45 socket 8-pin	RJ45 socket (twisted pair, 8-wire)

No.	Name	Connection type	Connector type (Integrated)	Mating connector and cable (From outside)
15	PF52	Status display PF52	–	–
	PF53	Status display PF53		
16	XD10	24 V power connected, U _L , U _P , with status display	4-pin	4-pin
17	–	Battery case	–	–
18	–	SD card holder	–	–
19	XF50	Ethernet	RJ45 socket	RJ45 plug
20	XF52	10/100/1000 MBit	8-pin	(twisted pair, 8-wire)

NOTICE

Damage of the device by plug mounting under voltage!

- Before mounting or dismounting components, disconnect the control - including its components - from voltage.
- Connect the voltage only after the control and its components have been set up.

8.2 Storage media

8.2.1 SD cards and USB support

Depending on the ctrlX CORE control hardware, the following conditions apply to the use of SD cards and USB devices:

Conditions for:	ctrlX CORE device variants	
	X3 / X3plus / M3	X7
SD cards	Supported card types: • SD • SDHC • SDXC • SDUC (supported only up to SDXC Standard / 2 TB max.)	Supported card types: • SD • SDHC (32 GB max.)
USB devices	• All USB interfaces are USB2.0-compatible • High, Full and Low Speed support	• All USB interfaces are USB3.1-compatible and downward compatible to USB3.0 and USB2.0 • Super, High, Full and Low Speed support • The maximum cable length for a USB cable is 3 m



Media formatted as Fat32 can also be used under Windows.

Removable media

Available storage media (internal and external) are shown in the “Storage” window, see [Window – “Storage”](#).

For the internal medium (System data), the free memory and its total size are shown.



Data on external storage media is not part of the "Backup and Restore" functionality!

Additional media can be mounted to the system and removed safely.

To open the detailed view of the storage medium, click on the tile of a memory medium. Memory size, memory load and the path of the mounted memory medium (USB stick and micro SD card) are displayed.

Depending on the operating state of the ctrlX device, the following actions can be made in the detailed view:

- ➔ [Mounting external storage medium](#)
- ➔ [Removing external storage medium safely](#)
- ➔ [Formatting external storage medium](#)

USB LAN adapter

USB LAN adapters can be operated at the USB interface of the control.

The following chipsets support USB LAN adapters:

- Realtek 8152
- Realtek 8153

An available USB LAN adapter is shown as additional network interface in the “Network interfaces” window, see ➔ [Window – “Network interfaces”](#).

USB WiFi adapter

USB WiFi adapters can be operated at the USB interface of the control.

The following chipsets support USB WLAN adapters:

- Realtek rtl8192eu
- Realtek rtl8192cu
- Atheros ar9271

An available USB WiFi adapter is shown as additional network interface in the “Network interfaces” window, see ➔ [Window – “Network interfaces”](#).

8.2.2 Mounting external storage medium

To use an external storage medium, mount it to or include it into the ctrlX device. It has to be defined how to use the storage medium:

- Use for data exchange with other devices
- Use as extension of the device memory for system data to store app data for example

Please note the instructions for use below

Use for data exchange

To use FAT32 or EXT4 formatted storage media for data exchange, see: ➔ [Formatting external storage medium](#)

The storage path in the data system is `"/media/<DeviceName>"` and displayed in the details of the storage medium.

The formatting of the storage medium is retained with all its data.

The storage medium can be safely removed and the content can be used on another device.

Several storage media can be mounted parallelly for the data exchange.

Use as system data memory

To be used as system data memory, the storage medium is automatically formatted in EXT4 format and encrypted with `"dmccrypt"`.

Note that all previously existing data on the storage medium is deleted!

The ctrlX device then restarts automatically.

The free memory and the total size are shown for mounted media.

Important notes:

- The storage medium can only be used on this device
- "Unplugging" the storage device can cause unexpected errors in the apps
- Only one storage medium can be mounted to expand the system data memory
- A separate memory area is provided for each application on which the application can store data
- The path in the system is not displayed in the details of the storage medium

Mounting storage medium

1. ➤ Open the ctrlX OS web interface and then the "Storage" window, see [↪ Window - "Storage"](#).
 - ➔ All available storage media are shown in tile representation in the "Storage" window.
2. ➤ Click on the storage medium to be mounted
 - ➔ A detailed view on the storage medium is shown
3. ➤ In the detailed view, click on the "Mount" button
 - ➔ The dialog to select the use opens
4. ➤ Select the use and confirm the dialog
 - ➔ When used for data exchange, the storage medium is mounted to the file system of the control and displayed in the "Storage" window.
 - ➔ When used as system data storage, a confirmation prompt has to be acknowledged before formatting and encrypting. At the end of the process, the ctrlX device is restarted.

Related topics

[↪ Removing external storage medium safely](#)

8.2.3 Formatting external storage medium



Mounted storage media cannot be formatted.

Execute the command "Remove safely" first, see [↪ Removing external storage medium safely](#).

Supported formatting:

- **FAT32**
To be used as removable media
- **EXT4**
To be used as removable media or as system data extension

Proceed as follows:

NOTICE

Formatting deletes all data on the external storage medium.

1. ➤ Open the ctrlX OS web interface and then the "Storage" window, see [↪ Window - "Storage"](#).
 - ➔ All available storage media are shown in tile representation in the "Storage" window.
2. ➤ Click on the storage medium to be formatted.
 - ➔ A detailed view on the storage medium is shown.

3. In the detailed view, click on the “Format” button
 - ➔ The formatting is started and the time required depends on the memory size

8.2.4 Removing external storage medium safely

A "safe removal" of the storage medium ensures that writing to the medium is completed and that it can be removed from the control without losing data (for removable media).

NOTICE

For storage media to expand the system data

After the "safe removal", the storage medium can only be used when formatted again!

All previously saved data is deleted by the formatting!

The "safe removal" automatically restarts the ctrlX device.

Proceed as follows:

1. Open the ctrlX OS web interface and then the “Storage” window, see [Window – “Storage”](#).
 - ➔ All available storage media are shown in tile representation in the “Storage” window.
2. Click on the storage medium to be safely removed.
 - ➔ A detailed view on the storage medium is shown.
3. In the detailed view, click on the “Remove safely” button
 - ➔ The storage medium is safely removed from the file system of the control

9 Real-time processing

9.1 System Handler

9.1.1 Introduction and overview

The System Handler of the ctrlX OS Runtime software monitors the operating states of the ctrlX OS Runtime software and signals these operating states using the integrated LED display.

The System Handler implements a state machine that responds on system error messages and starts a respective system response.

9.1.2 State machine of the System Handler

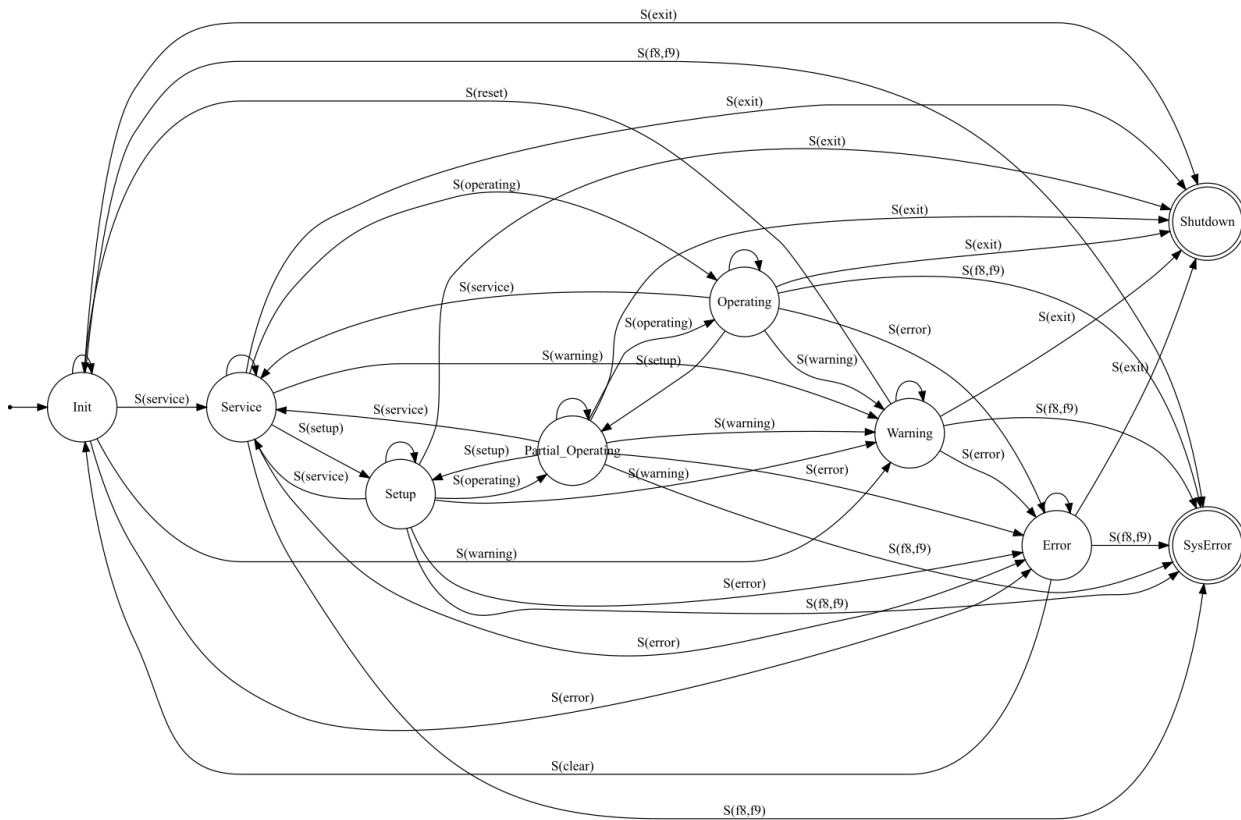


Fig. 19: State machine of the System Handler

After switching on the ctrlX hardware, the initialization phase of the system starts after the boot loader has been loaded.

In the initialization phase, the operating system (Linux) is started, the respective hardware drivers are loaded and then the ctrlX OS Runtime is started.

The system is currently in the operating state "Init".

Different components of the ctrlX OS Runtime report their states to the System Handler. The System Handler aggregates these individual states to a system state.

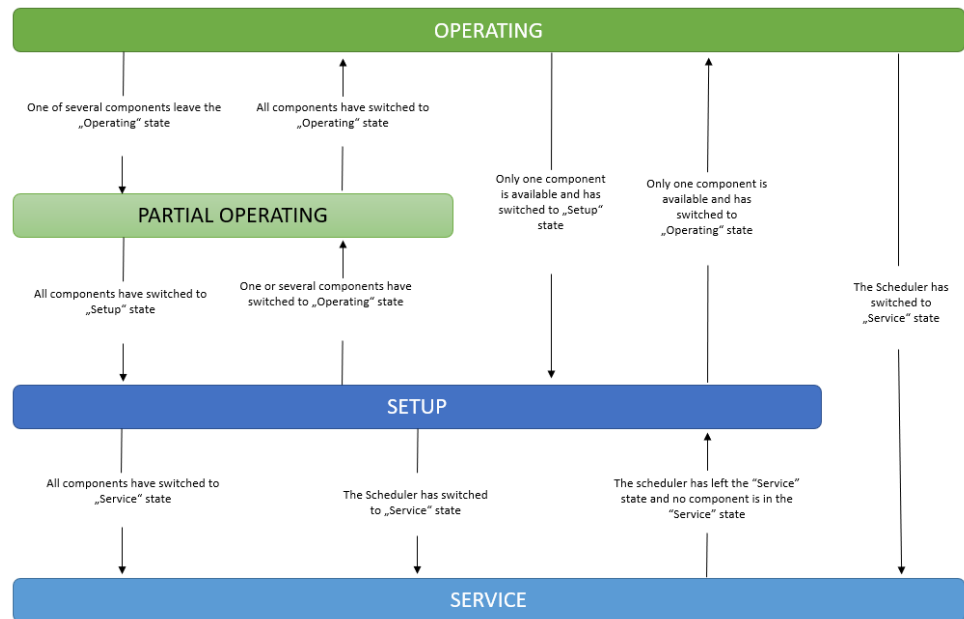


Fig. 20: State transitions of the system status

Operating states

State"Init"

The System Handler monitors the states of the ctrlX OS Runtime components.

State"Service"

In the "Service" state, all components are in the "Service" state or at least the ctrlX CORE Scheduler is in this state.

State"Setup"

All components are in the "Setup" state.

State"Partial Operating"

At least one component is in the "Operating" state.

State"Operating"

All components are in the "Operating" state.

State"Warning"

A warning was signaled in the ctrlX OS Runtime.
To reset the ctrlX OS Runtime, reset the warning.

State"Error"

An error of the error classes F0 to F7 was signaled in the ctrlX OS Runtime (refer to ➔ [Table 34 Detailed structure of a main diagnostic number on page 252](#)).

To reset the ctrlX OS Runtime, clear the error.

State"SysError"

A critical error of the error classes F8 to F9 was signaled in the system (also refer to ➔ [Table 34 Detailed structure of a main diagnostic number on page 252](#)).

- F8 error: The critical error is visualized briefly and then the ctrlX OS Runtime is restarted.
- F9 error: The critical error is visualized and the ctrlX CORE is switched off.

State"Shutdown"

The ctrlX OS Runtime is shut down and closed as expected.

Transitions

Transition/signal	Transition/signal description
S(service)	A component of the ctrlX OS Runtime was switched into the "Service" state
S(setup)	A component of the ctrlX OS Runtime was switched into the "Setup" state
S(operating)	A component of the ctrlX OS Runtime was switched into the "Operating" state
S(warning)	A warning was signaled
S(error)	An error of the state classes F0 to F7 was signaled
S(clear)	The errors of the state classes F0 to F7 and the warnings were deleted
S(reset)	A warning was reset
S(f8,f9)	A critical error of the state classes F8 to F9 was signaled
S(exit)	The ctrlX OS Runtime was shut down and closed

(see [framework/state](#))

Special case Safe Mode

If the Automation Core app does not start properly , the ctrlX CORE is in Safe Mode.

For more information, see [Chapter 9.1.8 Safe mode on page 151](#).

9.1.3 State of the ctrlX OS Runtime components

The System Handler represents the state of all ctrlX OS Runtime components. An aggregated system state as well as the state of all components is displayed. The error states are not considered.

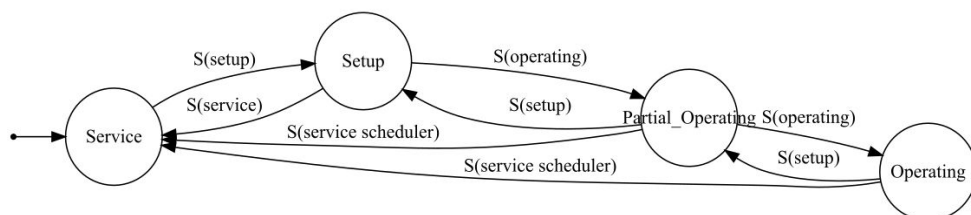


Fig. 21: State of the ctrlX OS Runtime components

Transitions

Transition/signal	Transition/signal description
S(service)	A component of the ctrlX OS Runtime was switched into the "Service" state
S(service scheduler)	The ctrlX CORE Scheduler was switched into the "Service" state

Transition/signal	Transition/signal description
S(setup)	A component of the ctrlX OS Runtime was switched into the "Setup" state
S(operating)	A component of the ctrlX OS Runtime was switched into the "Operating" state

(see [↗ system/state](#))

9.1.4 Multicore support of the PLC app

When the ctrlX device starts up, the processor cores and their property (real-time or non-real-time) available on the current target system are determined automatically by the system handler.

The PLC runtime system queries the properties determined during startup and assigns the system tasks to the first available real-time processor core.

The execution of the PLC user tasks is always assigned to this real-time processor core.

Via the task configuration in the ctrlX PLC Engineering, the user can also the assign the execution of these tasks to other available processor cores.

However, a PLC license with multicore option has to be installed as well on the control for the execution, see: [↗ Licenses – Overview](#)



In ctrlX PLC Engineering, the task configuration always provides 4 processor cores for allocation, independent of the actually existing properties of the target system.

The attempt to download an application via the ctrlX PLC Engineering with a configuration that cannot be executed on the target system is aborted and a respective error message is issued.

9.1.5 LED diagnostic code

The integrated diagnostic LED signals the operating states of the state machine of the System Handler.

The operating states are assigned to the LED states as follows:

Operating state of the System Handler	State of the diagnostic LED
Init	Booting active or shut down
Service/Setup	System state "Service/Setup"
Partial Operating	System state "Partial Operating"
Operating	System state "Operating"
Warning	Warning occurred in the ctrlX OS Runtime
Error	Error (F0 – F7) occurred in the ctrlX OS Runtime
SysError	System error (F8 and F9) occurred in the ctrlX OS Runtime
Shutdown	Booting active or shut down

LED states for ctrlX CORE

For the LED states for the ctrlX CORE, refer to the following chapter:

- [↗ LED states](#)

LED states for ctrlX DRIVE

For the LED states for ctrlX DRIVE, refer to the following documentation:

- "ctrlX DRIVE Diagnostic Messages of Runtime AXS-V-xxRS", DOK-XDRV**-GEN*-DIAG**-RERS-EN-P (chapter → ["Display at the device"](#))

9.1.6 Ensuring a safe machine state after a critical error

If an exception or a critical error of error class F8 or F9 results, a safe machine state and the diagnosability of the cause is ensured for the user.

Description of the current behavior when an exception or an F8/F9 error occurs:

- If a critical error occurs, it is responded to as follows:
 - If there is an exception error or an F8 error, the Automation Core app is shut down and restarts automatically.
 - If there is an F9 error, the ctrlX CORE is shut down and has to be restarted by a voltage cycle.
- In the subsequent start of the Automation Core app, the critical error is detected.
 - The cause is documented in the logbook of the ctrlX CORE.
→ Diagnosability is ensured.
 - An F6 error → (080F60B0) is issued with a variable detailed diagnostics. The cause of the critical error is pointed out.
→ Safe machine state is ensured (e.g. axes cannot be traversed).
- To further run the ctrlX CORE properly, acknowledge the problem and delete the pending F6 error.
- In special cases (e.g. hardware watchdog occurred), the Automation Core app also starts in safe mode (→ [Chapter 9.1.8 Safe mode on page 151](#)).

9.1.7 Detecting a reboot loop of the Automation Core app

If a reboot loop is detected with the Automation Core app, it is intended to always enter safe mode (→ [Chapter 9.1.8 Safe mode on page 151](#)).

Description of the current behavior:

- Detecting reboot loops is divided into three sections, with the following implemented counters, one for each phase of the detection of reboot loops:
 - Safe mode reached
 - Automation Core app started successfully
 - All apps in the process space of the Automation Core app started successfully
- The counters are incremented and reboot upon each restart if the corresponding conditions are met:
 - Initial start counter:
From the start of the Automation Core app until the evaluation of the safe mode is reached
Event "Initial start completed" (evaluation of safe mode) reached
→ Resetting the counter during the safe mode implementation
Evaluation of the counter at the start of the Automation Core app
→ Terminating the process with error in case of counter overflow
 - Core start counter:
From the start of the boot process to the successful start of the Automation Core app (only bundles of the Automation Core app)
Event FRAMEWORK_EVENT_CORE_STARTED occurs
→ Resetting the counter in the callback of the event
Evaluation of the counter in the safe mode implementation

- Safe mode in case of counter overflow
- Framework start counter:
 - From the boot start until the successful start of the Automation Core app and all bundles in monitored directories (bundles of the Automation Core app and all dependent bundles such as Motion app, PLC app and EtherCAT Master app)
 - Event FRAMEWORK_EVENT_STARTED occurs
 - Resetting the counter in the callback of the event
 - Evaluation of the counter in the safe mode implementation
 - Safe mode in case of counter overflow
- If there are more than three reboots of the app without the corresponding conditions being met, a reboot loop is detected and further reboots are prevented
- The F0 error ➔ [080F00B2](#) with a variable detailed diagnostics is issued indicating the cause of the reboot loop of the Automation Core app
- After an update, a reinstallation or a manual restart of the app, the reboot loop detection starts again from the beginning (with reset counters)

9.1.8 Safe mode

The safe mode definition in the Automation Core app ensures that the Automation Core app does not start completely and therefore the apps that depend on it (e.g. Motion app, PLC app, EtherCAT Master app) do not start.

The safe mode can be detected using the diagnostics ➔ [080F00B0](#)/➔ [0C7B4305](#). It is not visualized via the system state.

In this state, only some basic functionalities of the Automation Core app are available.

To further operate the ctrlX CORE properly, the safe mode can be exited by restarting the Automation Core app or the ctrlX CORE. An app that does not run properly might be replaced before.

Causes for safe mode can include:

- Hardware watchdog occurred
- Reboot loop detected in an application

9.2 ctrlX CORE Scheduler

9.2.1 Introduction and overview

The ctrlX CORE Scheduler is part of the system application "Automation Core". The main tasks are the synchronous execution and the monitoring of tasks and their assigned execution units, the so-called "callable".

The ctrlX CORE Scheduler is mainly configured via an individual web interface. However, the Data Layer can also be used to customize the configuration. Some properties can currently only be changed via Data Layer access.

The web interface is started via the entry "Scheduler" in the "Settings" of the ctrlX CORE.

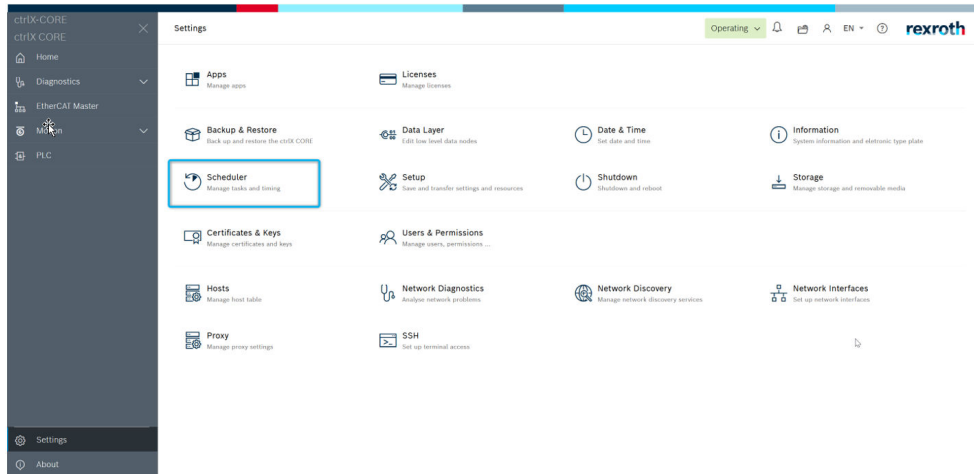


Fig. 22: Starting the web interface of the ctrlX CORE Scheduler

In the “Settings” menu of the Scheduler, basic properties concerning the startup behavior of the Scheduler or the system can be defined. They are explained in detail in the following chapters.

The “Debug” mode can also be enabled in this menu (see [Chapter Mode “Debug” on page 175](#)).

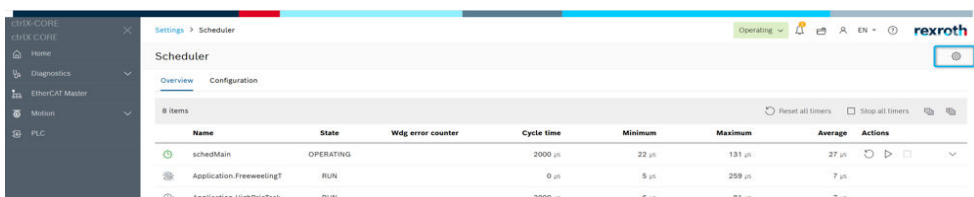


Fig. 23: ctrlX CORE Scheduler, “Overview” tab

The “Overview” tab provides an overview on the status and runtime behavior of the realtime-capable and non-realtime-capable tasks of the installed applications, such as "Motion" or "PLC".

If required, extended runtime measurements can be triggered in this tab (see [Chapter Time measurements on page 168](#)).

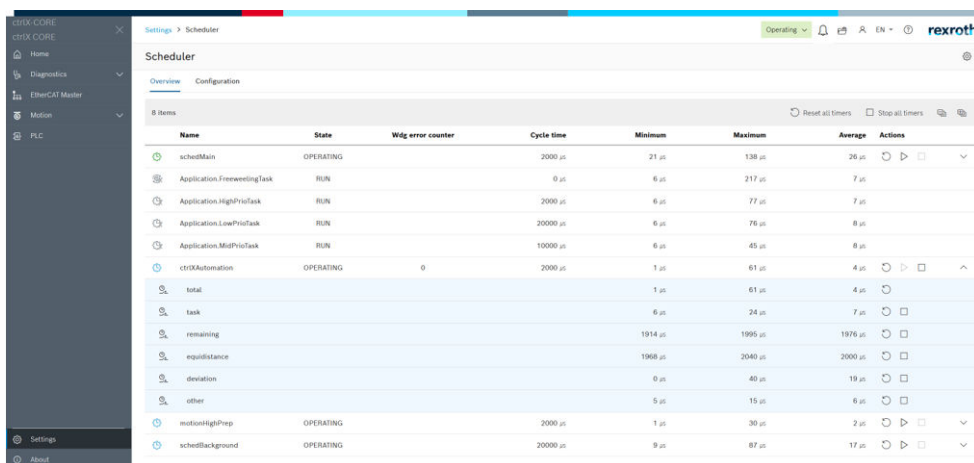
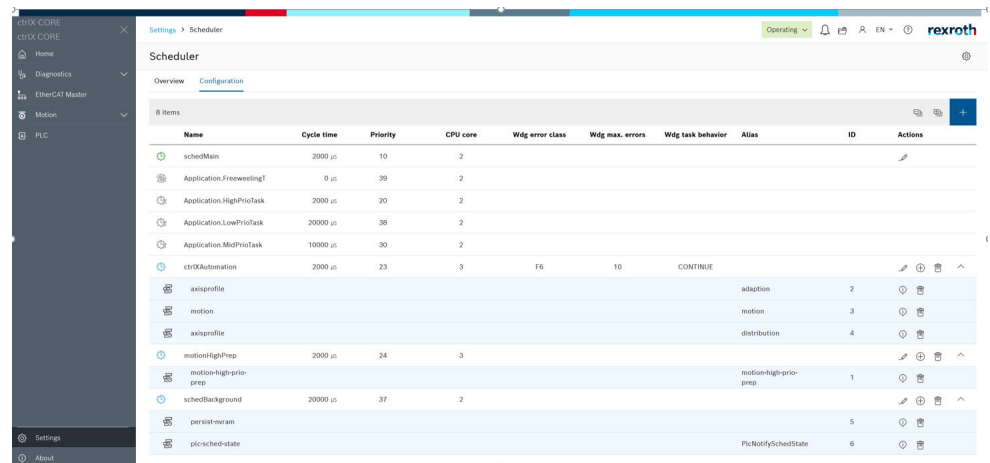


Fig. 24: ctrlX CORE Scheduler, “Overview” tab

In the “Configuration” tab, realtime-capable and non-realtime capable tasks as well as callables of the installed applications, e.g. "Motion" or "PLC", are created, managed and removed.

For details refer to the following chapters.



Name	Cycle time	Priority	CPU core	Wdg error class	Wdg max. errors	Wdg task behavior	Alias	ID	Actions
schedMain	2000 µs	10	2						
Application.Freezing1	0 µs	39	2						
Application.HighPrioTask	2000 µs	20	2						
Application.LowPrioTask	20000 µs	38	2						
Application.MidPrioTask	10000 µs	30	2						
ctrlXAutomation	2000 µs	23	3	F6	10	CONTINUE			
axisprofile							adaption	2	
motion							motion	3	
axisprofile							distribution	4	
motionHighPrep	2000 µs	24	3						
motion-high-prio-prep							motion-high-prio-prep	1	
schedBackground	20000 µs	37	2						
persist-nram								5	
plc-sched-state							PlcNotInSchedState	6	

Fig. 25: ctrlX CORE Scheduler, "Configuration" tab



Tasks of the PLC application (for example "Application.MidPrioTask") are shown for a better overview, but they cannot be modified.

9.2.2 Tasks und callables

The tasks managed by the ctrlX CORE Scheduler, i.e. all user tasks except PLC application tasks are called in their individual cycle time by the Scheduler and processed as per their priority.

Task response

Example

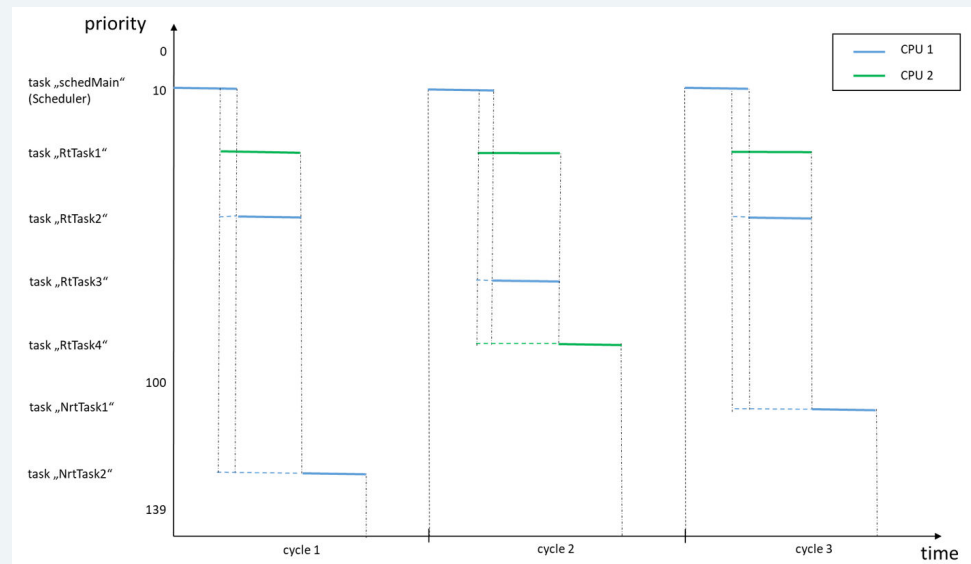


Fig. 26: Task response - Example

Cycle 1

- The scheduler starts the tasks *RtTask1*, *RtTask2* and *NrtTask2* simultaneously.
- The task *RtTask1* runs on another CPU as Scheduler and starts running directly after the activation.

- The task *RtTask2* runs on the same CPU as the Scheduler and starts running only after the Scheduler has been ended due to the lower priority.
- The task *NrtTask2* has a lower, non-real-time-capable priority and it starts running after all real-time-capable tasks were completed on its assigned CPU. As this task has a priority in the real-time-capable area, it can be suppressed by further, non-real-time-capable tasks in this area (time slice, Round Robin).

Cycle 2

- The Scheduler starts the tasks *RtTask1*, *RtTask3* and *RtTask4* simultaneously.
- The tasks *RtTask1* and *RtTask3* run on their respective CPUs according to their priority.
- The task *RtTask4* has a lower priority than the task *RtTask1* running on the same CPU. The task *RtTask4* starts running after the task *RtTask1* was processed completely.

Cycle 3

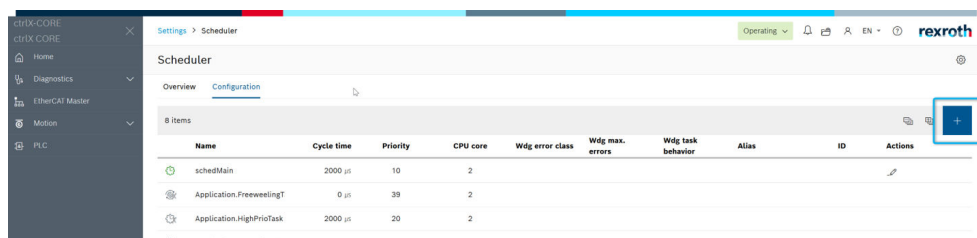
- The Scheduler starts the tasks *RtTask1*, *RtTask3* and *NrtTask1* simultaneously.
- All tasks are processed on their respective CPUs as per their priority.



- Real-time capable tasks are processed according to their priority. If they have the same priority, they are processed according to the FIFO principle ("first come, first served").
- Non-real-time capable tasks are processed in a remaining period of a time slice ("round-robin time-sharing") as long as no real-time-capable tasks have to be processed. The operating system can vary the length of a period for a task.

Task properties

In the "Configuration" tab, new tasks can be created and callables can be created and assigned in a subsequent step.



Name	Cycle time	Priority	CPU core	Wdg error class	Wdg max. errors	Wdg task behavior	Alias	ID	Actions
schedMain	2000 µs	10	2						
Application.FreewheelingT	0 µs	39	2						
Application.HighPriorityTask	2000 µs	20	2						
Application.LowPriorityTask	50000 µs	18	2						

Fig. 27: ctrlX CORE Scheduler, "Configuration" tab, creating a new task

A unique name is mandatory when creating a task. More properties can be modified via the extended menu.

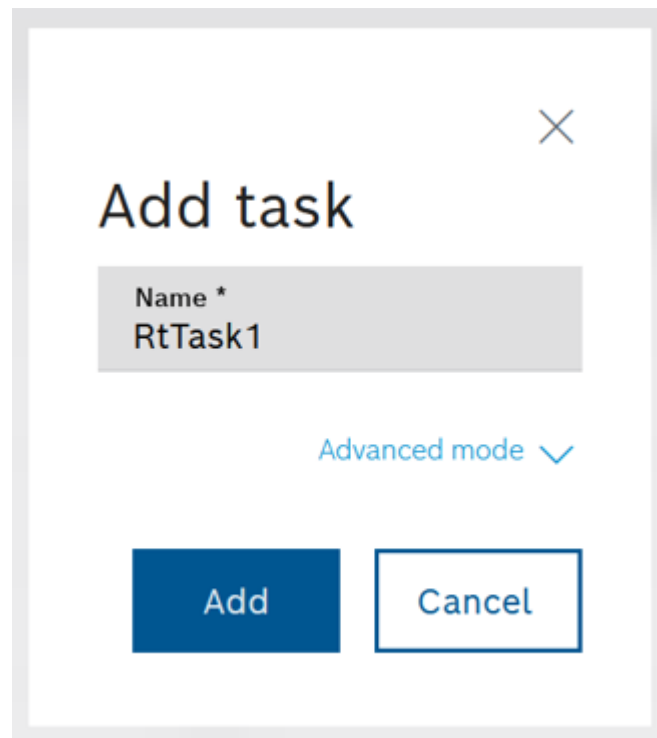


Fig. 28: Dialog to create a new task with the name "RtTask1"

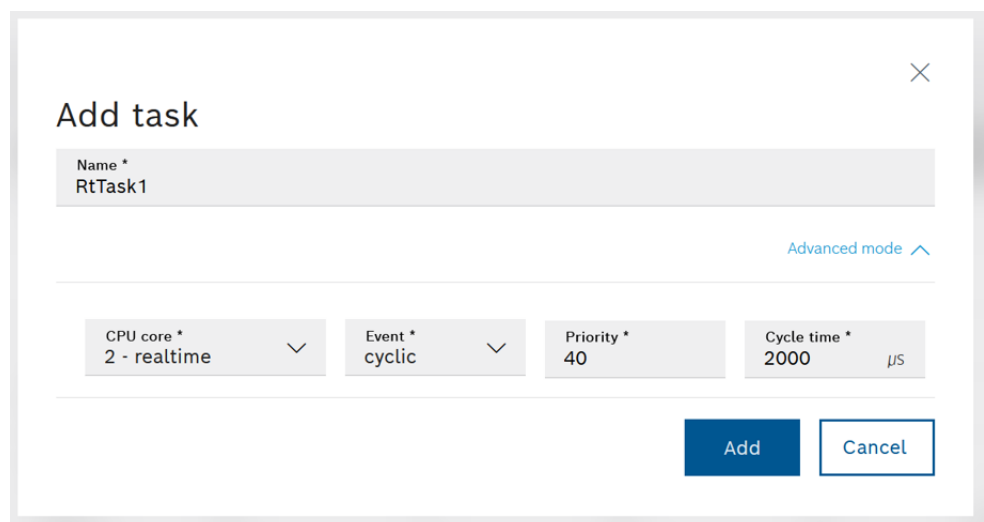


Fig. 29: Dialog to create a new task with the name "RtTask1" with extended properties

A task is defined with the properties shown in the following. Some of these properties can only be modified via Data Layer accesses.

Task properties:

- "Name"
 - The task name can be freely specified.
 - The name can consist of up to 15 characters and only of letters and digits.
 - The task name may only be specified once in the system.

- “CPU Core” (affinity)
 - The affinity specifies on which CPU (Core) the task is to be executed.
 - If the processor has for example 4 CPUs, an affinity between 0 and 3 can be selected.
 - If the processor is a multi-core processor, it has normally real-time and non-real-time capable CPUs.
- “Event”
 - The property "Event" specifies which event triggers the task call.
 - To call the task cyclically in a specified time grid, specify the event as “cyclic”.
- “Priority”
 - The priority of the task can only be selected in the available priority range.
For supplementing notes on the topic, refer to [Chapter Task “Priority” on page 156](#).
- “Cycletime”
 - The cycle time specifies during which cycle (in μs) the task should be called.
 - The possible cycle times can differ on different hardwares.
The following cycle time are possible:
125 μs , 250 μs , 500 μs , 1 ms, 2 ms, further in millisecond steps.
 - The maximum cycle time is one second.
- Stack size
 - The stack size indicates how much stack memory is provided to the task.
 - Typically, the initial value of the stack size is sufficient and does not have to be modified.

(see [scheduler/tasks/<task-name>/properties](#))

(see [Chapter 13.3.8 Tab – “Configuration” on page 328](#))

Task “Priority”

Priorities from 0 to 139 can be specified for tasks.

These priorities are divided into the following sections:

Table 4: Task priorities

Priority	Remark
0	Highest real-time priority
10	ctrlX CORE Scheduler
11...19	High-priority system tasks
20...39	ctrlXAutomation task, PLC tasks...
...	
99	Lowest real-time priority
100	Highest non-real-time priority
...	
139	Lowest non-real-time priority

Real-time priorities

- Priorities 0 – 9
 - This priority range is reserved for system tasks. No tasks can be created in this range.
- Priorities 10

- This priority is only intended for the ctrlX CORE Scheduler. No further task can be created on this priority.
- Priorities 11...19
 - These priorities are intended for high-priority system tasks.
 - Tasks should only be created in exceptional cases. Tasks within this priority range can affect the system stability negatively.
- Priorities 20...39
 - These priorities are intended for tasks with high requirements on the real-time capability with regard to temporal equidistance or interruptions for example. Examples are field bus drivers, Motion computations or the use of inputs and outputs.
 - Tasks with priorities in this range can only be interrupted or omitted by high-priority system tasks. However, the runtime of these system tasks is low and does not interrupt these tasks for a longer period.
- Priorities 40...99
 - In this field, system tasks with the most different tasks are executed.
 - Tasks with priorities in this range can be interrupted or omitted by system tasks.

Non-real-time priorities

- Priorities 100 – 139
 - These priorities are intended for tasks without real-time priorities.
 - These tasks are processed if no task is running with a real-time priority.
 - These tasks can be omitted or interrupted at any time.
 - An equidistant processing cannot be ensured.



Task priority information

- High requirements on the realtime
 - If low cycle times and a high-accurate equidistant call are required, select the highest priority possible. These requirements can be met the best possible if this task is set to the highest priority on the respective CPU.
 - The high-priority system tasks typically run on a CPU. Thus, the task should run on another CPU.
 - The lower a CPU load, the better the response with regard to the runtime jitter and equidistance.
 - The best case possible is if a CPU can be reserved for an individual task.

(see [↗ scheduler/tasks/<task-name>/properties/priority](#))

Callables

Each task executes a program consisting of one or multiple execution units named callables.

Callables are provided by the installed applications such as "Motion" or "PLC" and contain fixedly defined functionalities, such as backing up NvRAM memory or entire modules, such as the command value calculation of axes.

Thus, the customized callable sequence results in the intended program sequence of the provided functionalities.

Example

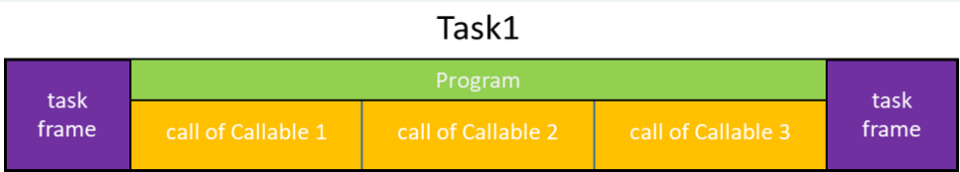


Fig. 30: The “Task1” executes a program consisting of three callables

In the “Configuration” tab, a new callable can be added to an existing task or an existing callable can be removed from a task.

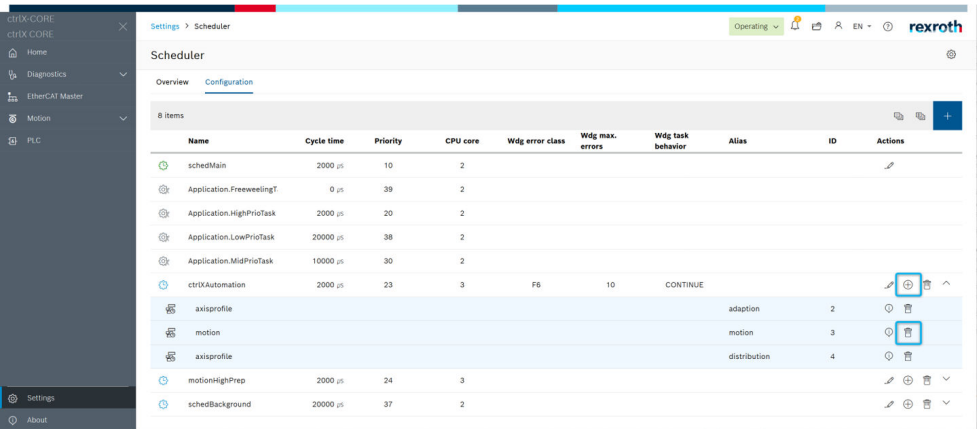


Fig. 31: ctrlX CORE Scheduler, “Configuration” tab, creating or deleting a callable

The available callables depend on the installed applications, such as "Motion" or "PLC" and are shown in the dialog used to create a callable or in the "callables-pool" if available.

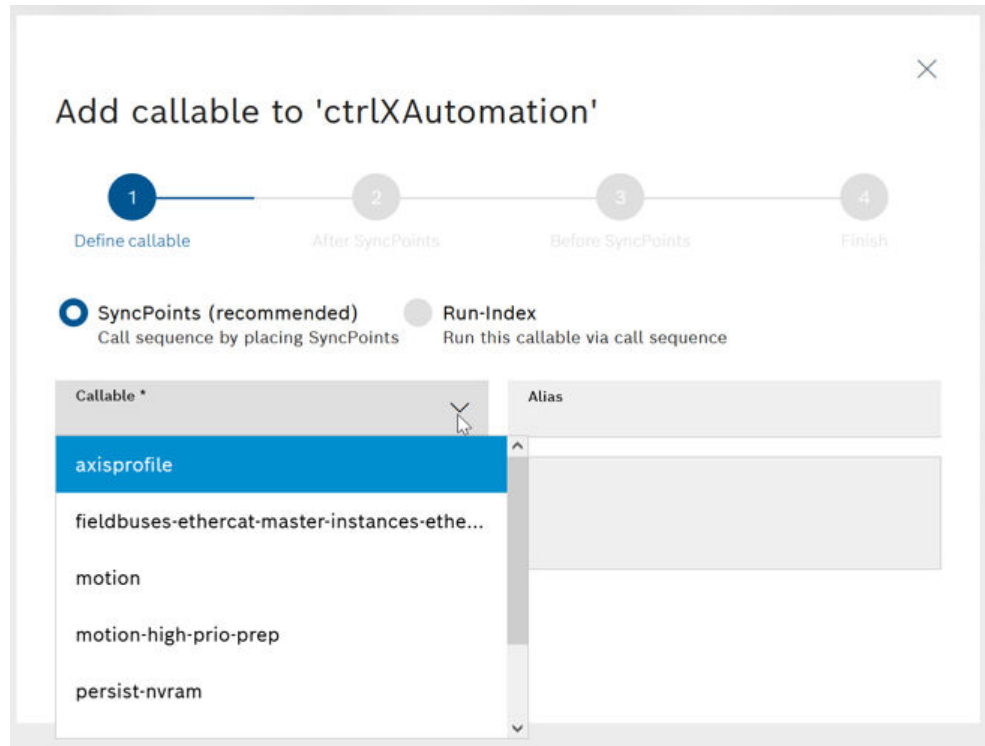


Fig. 32: ctrlX CORE Scheduler, "Configuration" tab, available callables (example)

(see [↗ scheduler/callables-pool](#))

To create a callable, define the method to determine the processing sequence within the program.

There are two options that can also be used together in one program:

- Synchronization points "SyncPoints" (recommended for new projects)
- Call indices "Run index"

Fig. 33: ctrlX CORE Scheduler, “Configuration” tab, creating a callable
(see [↗ scheduler/tasks/<task-name>/program/<callable_id>/run-index](#))
(see [↗ scheduler/tasks/<task-name>/program/<callable_id>/synchronization-points](#))

Method to determine the processing sequence

The callable with the lowest run index above zero is called first in the processing sequence. Further callables are called in ascending run index order. The processing sequence of callables with the same run index to each other is not determined.

Following the callables with a run index unequal to zero, callables with synchronization points and a run index equal to zero are enqueued.

Callables without synchronization points and with a run index equal to zero are executed at the end of the program in any sequence.

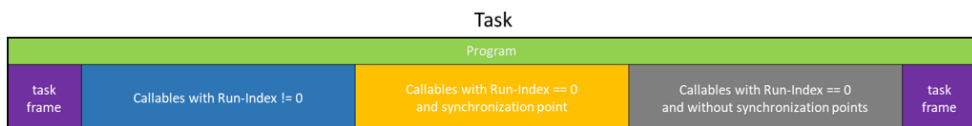


Fig. 34: Processing sequence of the callables within a program

In addition to the strict definition of the processing sequence using a run index, the use of synchronization points allows that a callable always runs before or after a specific callable, even if further applications with new callables are installed.

The configuration of the synchronization points of a callable consists of the lists "after" and "before". The "after" list defines after which synchronization points the callable is called, while the "before" list defines before which synchronization points the callable is called.

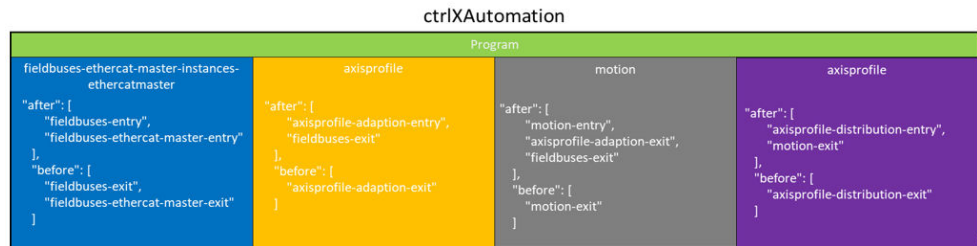


Fig. 35: Exemplary configuration of the ctrlXAutomation task

Example

There is a callable A with a synchronization point "syncPointA" in the list "before".

Another callable B is to be created. It is always executed after callable A.

The requirement is met if callable B is created with the synchronization point "syncPointA" in the list "after".

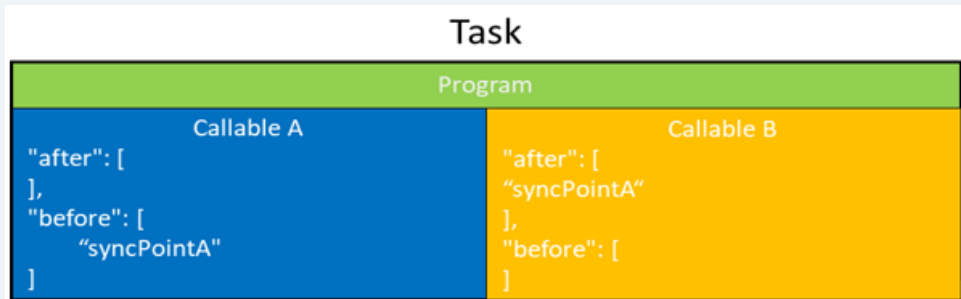


Fig. 36: Defining the sequence of the callables by the synchronization point "syncPointA"



The processing sequence using synchronization points does not enforce the direct call sequence, but ensures the condition [A before B] and [A before C].

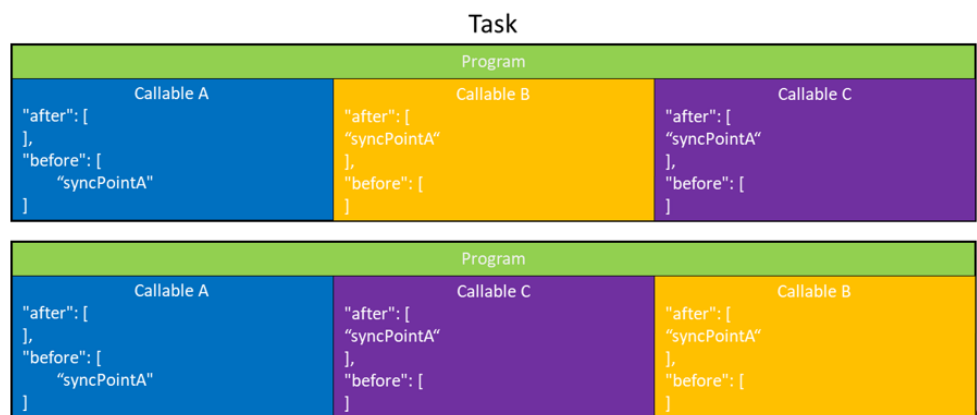


Fig. 37: Defining the sequence of the callables by the synchronization point "syncPointA" (alternative example)



The callables provided by the applications are generally already useful synchronization points that ensure a basic functionality.

A callable can be added any number of times to the respective program of one or multiple tasks using the same or different initialization arguments. Whether this is useful and possible is a basic callable property.

For more information, refer to the description of the respective application that provides the callable.

The possible initialization arguments and their number are also determined by the callables. The arguments used for the instantiation can be determined either via the editing window or via the Data Layer.

For an individual differentiation, the user can assign an alias to a callable instance.

Add callable to 'ctrlXAutomation'

1 Define callable 2 After SyncPoints 3 Before SyncPoints 4 Finish

☒ **SyncPoints (recommended)**
Call sequence by placing SyncPoints

☐ **Run-Index**
Run this callable via call sequence

Callable *
plc-external-event

Alias
CallPlcTouchProbe

Arguments
TASK_EXTERNAL_EVENT_01

Enter arguments (not required)

Next

Fig. 38: Creating a callable of the type “plc-external-event“, alias “CallPcTouchProbe”

(see [➔ scheduler/callables-pool/<callable_factory>/arguments](#))

(see [➔ scheduler/tasks/<task-name>/program/<id>/alias](#))

×

Add callable to 'ctrlXAutomation'

1

2

3

4

Define callable

After SyncPoints

Before SyncPoints

Finish

After which SyncPoints is the callable 'plc-external-event' called:

Available SyncPoints

☐ fieldbuses-ethercat-master-entry
 ☐ fieldbuses-ethercat-master-exit
 ☐ motion-entry
 ☐ motion-exit

Select all

Deselect all

Sync-point

+

After SyncPoints

☐ fieldbuses-exit

Select all

Deselect all

Back

Next

Fig. 39: Specifying the synchronization points "SyncPoints" of the list "after"

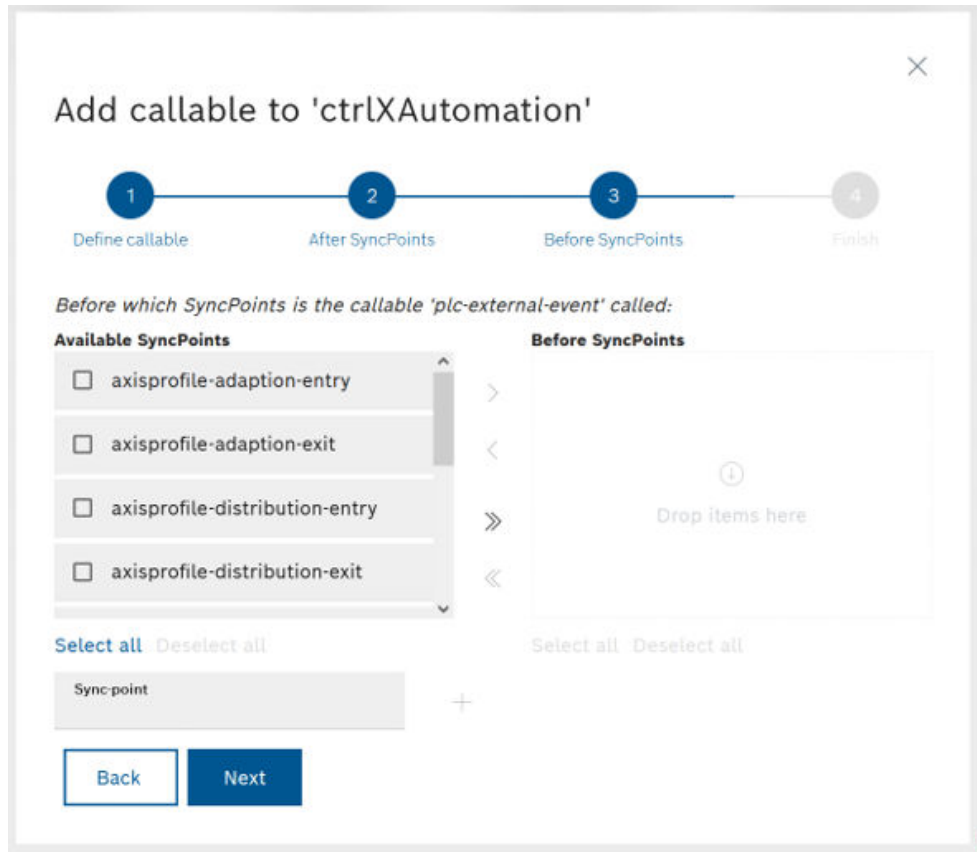


Fig. 40: Specifying the synchronization points "SyncPoints" of the list "before"

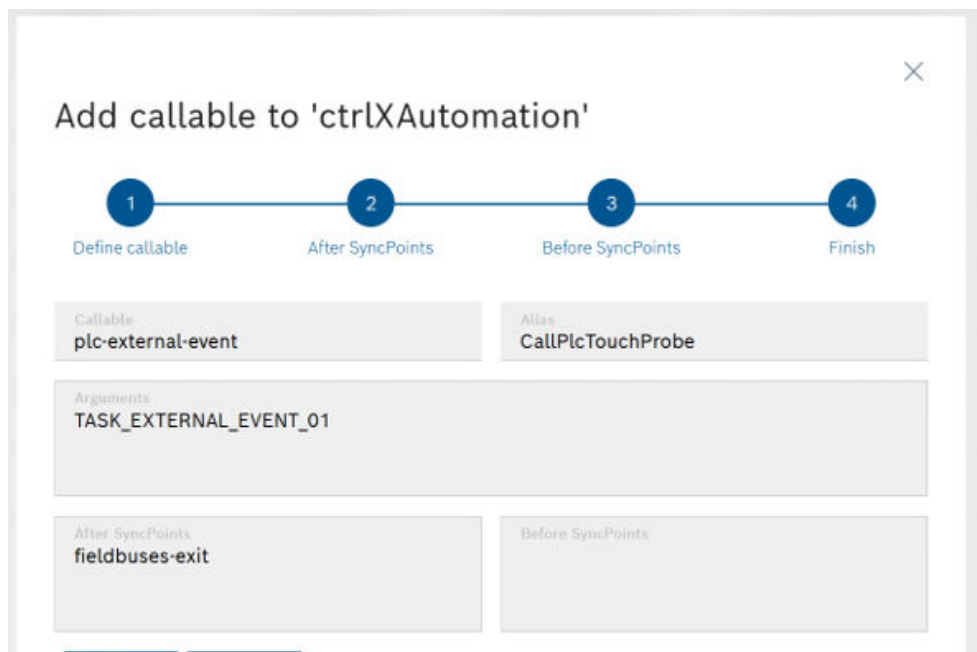


Fig. 41: Summarizing the specification of the synchronization points "SyncPoints"



Any number of additional synchronization points can be created and used by the user.

Name	Cycle time	Priority	CPU core	Wdg error class	Wdg max. errors	Wdg task behavior	Alias	ID	Actions
schedMain	2000 µs	10	2						
Application.FreewheelingT	0 µs	39	2						
Application.HighPrioTask	2000 µs	20	2						
Application.LowPrioTask	20000 µs	38	2						
Application.MidPrioTask	10000 µs	30	2						
ctrlAutomation	2000 µs	23	3	F6	10	CONTINUE			
recovery-recovery-master-instances- performance-tester								8	
axisprofile							adaption	2	
motion							motion	3	
axisprofile							distribution	4	
plc-external-event							CallPicTouchProbe	9	
motionhighPrep	2000 µs	24	3						
schedBackground	20000 µs	37	2						

Fig. 42: Result after successfully creating a callable of the type "plc-external-event"

(see [scheduler/tasks/<task-name>/program/<callable-id>/<callable_factory>](#))

(see [scheduler/tasks/<task-name>/program/<callable_id>/synchronization-points](#))

(see [scheduler/tasks/<task-name>/program](#))

If a callable is added to a task and thus to a program (i.e. to a program sequence) and thus instantiated, an automatically generated ID is assigned to this callable. The ID is shown in the table or in the Data Layer. The ID allows to subsequently assign callables across tasks.

(see [scheduler/tasks/<task-name>/program](#))

(see [scheduler/callables-pool/<callable_factory>/ids](#))

The calling sequence of the callables in a task can be retrieved via the sequence. The sequence is a list of callable IDs. The callables are called in the order given in the sequence.

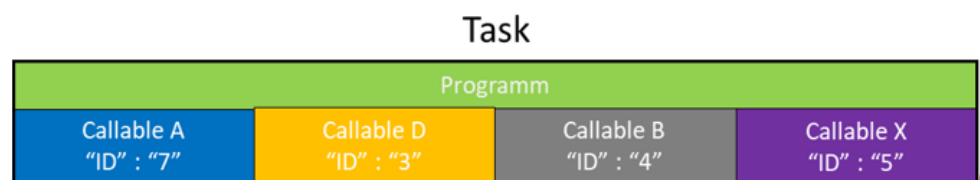


Fig. 43: Task with the sequence "7", "3", "4", "5"

(see [scheduler/tasks/<task-name>/sequence](#))

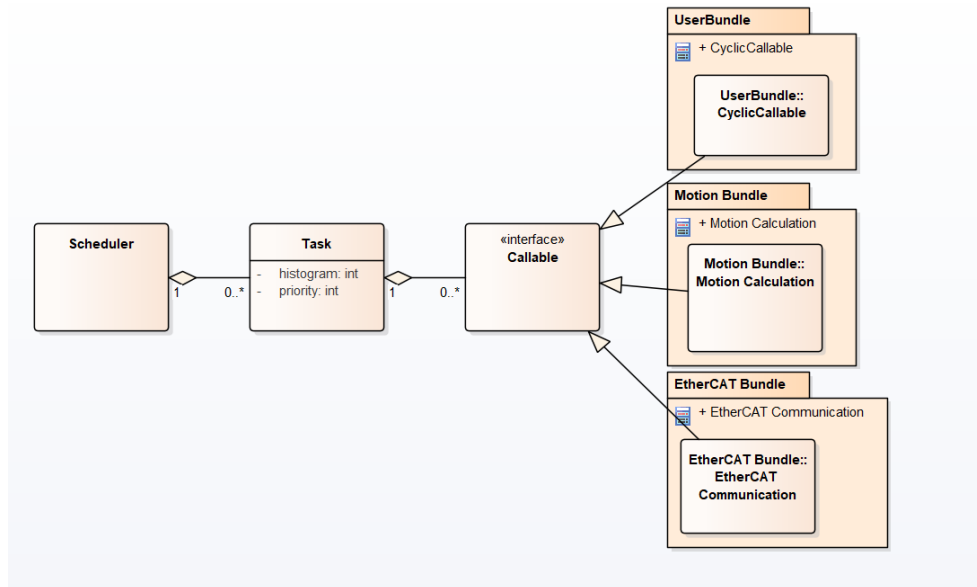


Fig. 44: Tasks and callables

Placeholder

Name	Cycle time	Priority	CPU core	Wdg error class	Wdg max. errors	Wdg task behavior	Alias	ID	Actions
schedMain	2000 µs	10	2						
Application.FreewheelingT	0 µs	39	2						
Application.HighPrioTask	2000 µs	20	2						
Application.LowPrioTask	20000 µs	38	2						
Application.MidPrioTask	10000 µs	30	2						
ctrlXAutomation	2000 µs	23	3	F6	10	CONTINUE			
master-instances-ethercatmaster								8	
_placeholder_axisprofil							adaption	2	
_placeholder_motion							motion	3	
_placeholder_axisprofil							distribution	4	
plc-external-event							CallPtcTouchProbe	9	
motionHighPrep	2000 µs	24	3						
schedBackground	20000 µs	37	2						

Fig. 45: Placeholder in case of missing application (e.g. Motion)

If a configuration with callables is loaded with some or all required applications missing, the respective callables are labeled with the prefix "_placeholder_". These callables do not have any further functionality and are used as note for the user to install the respective applications (in the example "Motion"). Placeholders can be removed optionally.



The placeholder callables can also remain in the configuration, as they do not affect the system behavior.

Program configuration mode

There are two different modes to configure programs. A program is defined by a task and by the call sequence of the callables in this task.

The mode affects the behavior of the following actions:

- Changing task properties
- Deleting and creating callables
- Deleting and creating tasks
- Saving and loading a configuration
- Logging off callable factories via the Scheduler interface

Manual program configuration

- It is possible to create and delete callables.
- Tasks and callables are saved in the configuration. These are restored when the configuration is loaded.
- Placeholders are used for callables.

Automatic program configuration

- Tasks and callables cannot be deleted or created manually.
- Callables are not saved in the configuration.
- The name of a task cannot be changed.
- If the properties of a task were changed, these changes are saved in the configuration.
- No placeholders are created for callables.

Switching the program configuration mode

No further actions are performed when switching to the manual program configuration mode. The current configuration is retained and it is thus now possible to make changes to tasks and callables.

- **Switching to the manual program configuration mode**

No further actions are performed when switching to the manual program configuration mode. The current configuration is retained and it is thus now possible to make changes to tasks and callables.

- **Switching to the automatic program configuration mode**

Switching to the automatic program configuration mode is only possible if the current state is not OPERATING and strict mode is not active. No more changes can be made to tasks or callables when switched. The complete configuration is reset after switching. All tasks and callables are deleted. The automatically generated tasks and callables are then restored. Switching the program configuration mode again is only possible once this process is complete.

Differences between the modes

Table 5: Changing task or callables

	Creating task	Deleting task	Creating callable	Deleting callable	Changing task name	Changing task properties	Changing task watchdog
Automatic program configuration mode	✗	✗	✗	✗	✗	✓	✓
Manual program configuration mode	✓	✓	✓	✓	✓	✓	✓

Table 6: Saving a configuration

	Callables	Non-modified tasks	Modified tasks
Automatic program configuration mode	✗	✗	✓
Manual program configuration mode	✓	✓	✓

Table 7: Loading a configuration

	Callables	Tasks
Automatic program configuration mode	✗	✓
Manual program configuration mode	✓	✓

(see → [scheduler/admin/cfg/program-configuration](#))

Time measurements

The ctrlX CORE Scheduler provides different options to measure runtimes and time intervals. The runtime measurements can be started and stopped in the tab “Configuration” of the Scheduler using the buttons in the “Actions” column.



Data for time measurement as well as the start time of the tasks managed in the Scheduler are available in the Data Layer RT and can for example be applied to the "Oscilloscope" application in the oscilloscope and then analyzed. The start time of the tasks managed by the scheduler are of the type "CLOCK_MONOTONIC" (Linux) or "std::chrono::high_resolution_clock" (Windows).

Time measurements of a task

- Total runtime ("total")
 - This is the time from the start of the task until it was ended.
 - This time can also include interruptions due to tasks with a higher priority or interrupts.
 - This measurement is always performed.
- Task runtime ("task")
 - This is the time required by the task.
 - This time does not include the time of possible task interruptions.
 - This measurement has to be enabled.
- Time during which the task was interrupted ("other")
 - Time during which the task was not running or interrupted due to tasks with a higher priority.
 - This measurement has to be enabled.
- Equidistance ("equidistance")
 - This measurement provides information on how equidistant the task was called.
 - A cyclic task is always started in its cycle time. The calling point in time as well as the start time have a certain jitter. The jitter causes can be different, such as the load of the CPU(s). It is also possible that the start time is changed due to a task with a higher priority.
 - It is always measured from one start to the next start.
 - This measurement has to be enabled. Enabling this measurement also enables the measurement of the deviation.

- Deviation ("deviation")
 - This time indicates the deviation of the start time from the earliest start time.
 - This measurement has to be enabled. Enabling this measurement also enables the measurement of the equidistance.
- Remaining time ("remaining")
 - For a cyclic task, this is the remaining time from the end of the task up to the beginning of a new task.
 - If the task was not completed in its cycle, the remaining time is zero.
 - This measurement has to be enabled.

The minimum, maximum and average time are determined during all messages.

Example

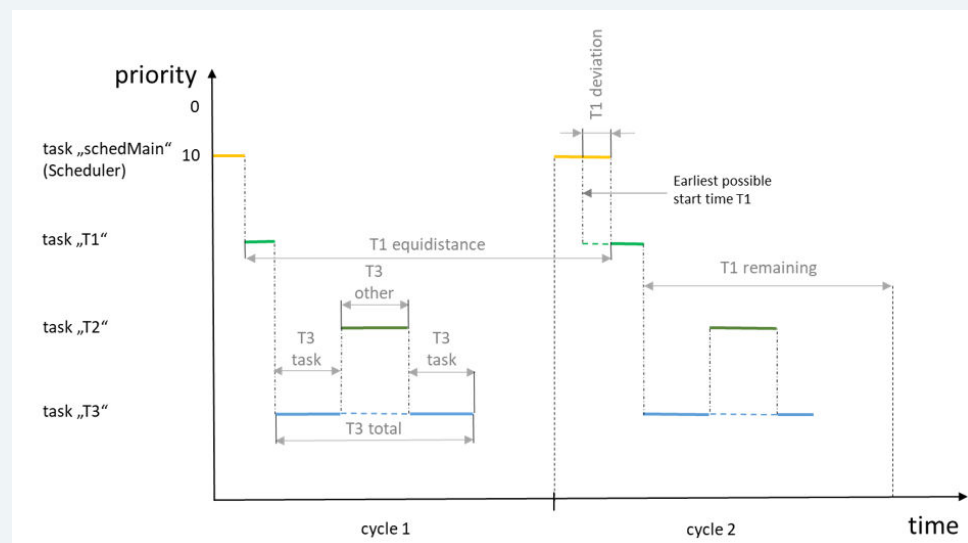


Fig. 46: Time measurements - Example

The time from the task start up to the next start is determined when measuring the equidistance. In case of a deviation, it is measured how much a task deviates from the earliest measured start time. The remaining time can be used to determine the amount of time available from the end of the task up to the beginning of a new task.

Task T2 has a higher priority than T3. Thus, T3 is suppressed at runtime. This interruption as well the runtime are optionally also computed for task T3.

(see [scheduler/tasks/<task-name>/info/duration](#))

Histogram

A histogram can be created via the time measurements. The histogram scaling is adapted automatically to the measured period.

Example

Scaling: 10;

Data: 0, 5, 67, 0, 1, ...

Time span in microseconds	Number of measured times within this time span
0 - 9	0
10 - 19	5
20 - 29	67
30 - 39	0
40 - 49	1

(see [scheduler/tasks/<task-name>/info/histogram](#))



Enabling the optional runtime measurements affects the runtime response and the runtime jitter. The best result is obtained for the respective measurement if only the relevant measurement is active.

Time measurements of callables

For callables, the total time required to execute the cyclic callable functionality is measured.

The following values are determined for each callable of a task during the measurements:

- Minimum time
- Maximum time
- Average time

The time measurements of the callables are identified by their ID.

The time measurement of all callables of a task can be started, stopped and reset.

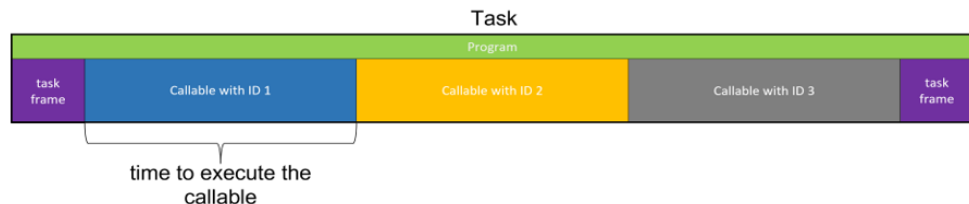


Fig. 47: Time measurement of callables

(see [scheduler/tasks/<task_name>/info/duration-callables](#))

Task status

A task can assume different states:

- SERVICE
 - The task is not executed in this state.
 - The callables of the task are not called.
 - The task counter stopped.
 - The runtime measurements are not executed.
 - Task watchdog is inactive.
- SETUP
 - The task is executed in this state.
 - The callables are called and also in the SETUP state.
 - All measurements and monitorings are active.

- OPERATING
 - The task is executed in this state.
 - The callables are called and also in the OPERATING state.
 - All measurements and monitorings are active.

(see [scheduler/tasks/<task-name>/state](#))

9.2.3 Administration

To administer the ctrlX CORE Scheduler, different settings and information are available in the ctrlX CORE Scheduler menu “Settings” and via the Data Layer.

(see [scheduler/admin/properties](#))

(see [scheduler/admin/startup-config](#))

(see [scheduler/admin/trigger-source](#))

(see [Chapter 13.4.19 Dialog – “Scheduler settings” on page 357](#))

Status

The ctrlX CORE Scheduler can assume the following states:

- SERVICE
- SETUP
- OPERATING

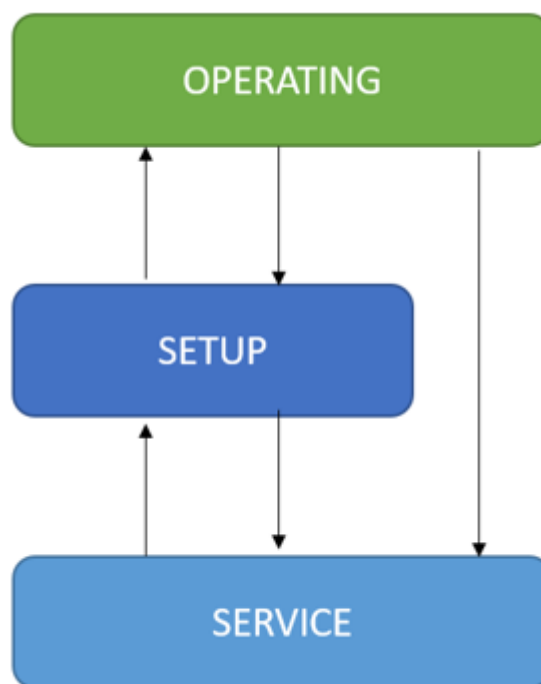


Fig. 48: ctrlX CORE Scheduler states

State meaning

- SERVICE

Actions affecting the realtime can be executed in this state. These actions can for example be the installation and uninstallation of apps.
- SETUP

This state is used to configure the subordinate components of the callables.
- OPERATING

This state is intended for the machine operation.

State properties

- SERVICE
 - The tasks are in the SERVICE state and they are not executed (see [Chapter Task status on page 170](#)).
 - The hardware watchdog is not active.
- SETUP
 - The tasks are in the SETUP state and they are executed (see [Chapter Task status on page 170](#)).
 - The hardware watchdog is active.
- OPERATING
 - The callables are switched to the OPERATING state.
 - Callables can then normally not be configured anymore.

When switching the state, the states of the tasks and thus the callables are switched implicitly. When switching to a higher state, the callables are switched in the sequences in which they are called in the task. If it is switched to a lower state, the callables are switched in reverse sequence.

Switching the callables also switches the subordinate component of the callable to a corresponding state.

Example

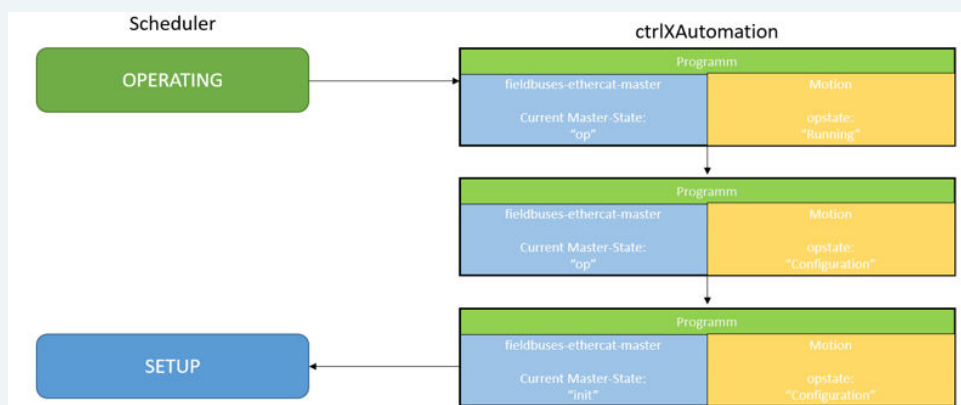


Fig. 49: Switching from OPERATING to SETUP

The subordinate components of the callable and the Scheduler report their state to the system state (see [Chapter 9.1.2 State machine of the System Handler on page 146](#)).

As it is possible that individual tasks cannot change their state, an error message is issued.

Switching SETUP -> OPERATING

- If a task cannot switch, the Scheduler still switches to OPERATING.
- If the error was troubleshooted in the task, the task can be switched individually to OPERATING or it has to be switched again in the Scheduler.

Switching OPERATING -> SETUP

- If a task cannot switch, the Scheduler does not switch to SETUP.
- The Scheduler remains in OPERATING as long as the task remains in this state.
- A possible reason that a task cannot switch into SETUP is that a callable in this task is currently commanding a motion that has to be completed first.

The state switching can be checked via Data Layer node. The *scheduler/admin/state/switching* node is TRUE as long as a switching is active. The *scheduler/admin/state/info* node contains the initial state, the current state and the target state.

Since callables require different times to switch, but the switching is also monitored by the Scheduler in terms of time, it can be required to adjust the limits of the monitoring times via the *scheduler/admin/state/callable-timeout* node.

(see [↗ scheduler/tasks/<task-name>/state](#))

(see [↗ scheduler/admin/state](#))

(see [↗ scheduler/admin/state/info](#))

(see [↗ scheduler/admin/state/switching](#))

(see [↗ scheduler/admin/state/callable-timeout](#))

Trigger source

The trigger source can be configured for the Scheduler task "schedMain". The trigger source affects the accuracy of the Scheduler call. This depends on the hardware and the operating system used.

Trigger source:

- Timer
 - Timers of the operating system are used to trigger the Scheduler.
- Interrupt
 - An internal interrupt is used to trigger the Scheduler.

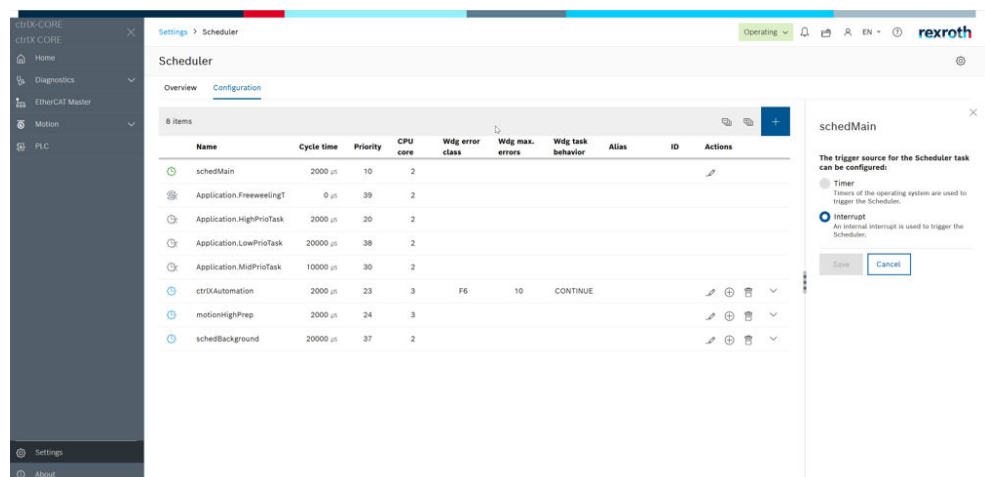


Fig. 50: Selecting the trigger source of the Scheduler
(see [↗ scheduler/admin/trigger-source](#))

Configuring the startup

As the ctrlX CORE is modular, the Scheduler can execute program parts from different components such as "Motion", "PLC" or even third-party manufacturers if required. Thus, it can be required that the Scheduler remains in the startup until all components were loaded completely.

Configuration options

- State
 - Specifies which target state the Scheduler should switch to.
- Timeout
 - Specifies how long the Scheduler waits for additional components during startup.
- Error reaction
 - Specifies how the Scheduler responds to a timeout during startup.
 - The Scheduler can cancel the startup or continue without the missing components.

Scheduler settings

Startup Debug

State - Specifies to which target state the Scheduler should switch after switching on

☒ **Operating**
Tasks are called and callables are ready to operate

☐ **Setup**
Tasks are called and callables can be configured

☐ **Service**
Tasks are not called, callables are inactive and apps can be installed

Timeout - Specifies the time how long the Scheduler waits for additional components during startup

Timeout *
0 ms

Error reaction - Specifies how the Scheduler reacts to a timeout during startup

☐ **Stop**
The Scheduler aborts further startup

☒ **Continue**
The Scheduler continues the startup without the missing components

Save Cancel

Fig. 51: Startup configuration options
(see [↗ scheduler/admin/startup-config](#))

Backing up and restoring the configuration

The complete configuration is backed up or restored. If a configuration is backed up, a previously saved configuration is overwritten. If a configuration is restored, the current configuration is deleted and the saved configuration is restored.

If the active configuration is reset, the complete configuration of the Scheduler is reset to default. The default configuration of the callables is subsequently restored if the callables have a default configuration. Callables without configuration are deleted and not created again.

(see [↗ scheduler/admin/cfg](#))

Resetting the task configuration

Resetting the task configuration allows to restore the configuration of tasks and callables intended by the installed applications.

(see [↗ scheduler/admin/cfg/reset-programs](#))

Selecting an initial CPU core

For an automated commissioning or for operating interfaces, it might be required to suggest an initial CPU core for real-time processes or non-realtime processes to the user.

For this purpose, the Scheduler provides the respective Data Layer nodes.

(see [↗ scheduler/admin/cfg/info/cpu-cores](#))

Mode “Debug”

The Scheduler can be switched to a “Debug” mode.

Different monitoring mechanisms, such as the hardware watchdog, are disabled.

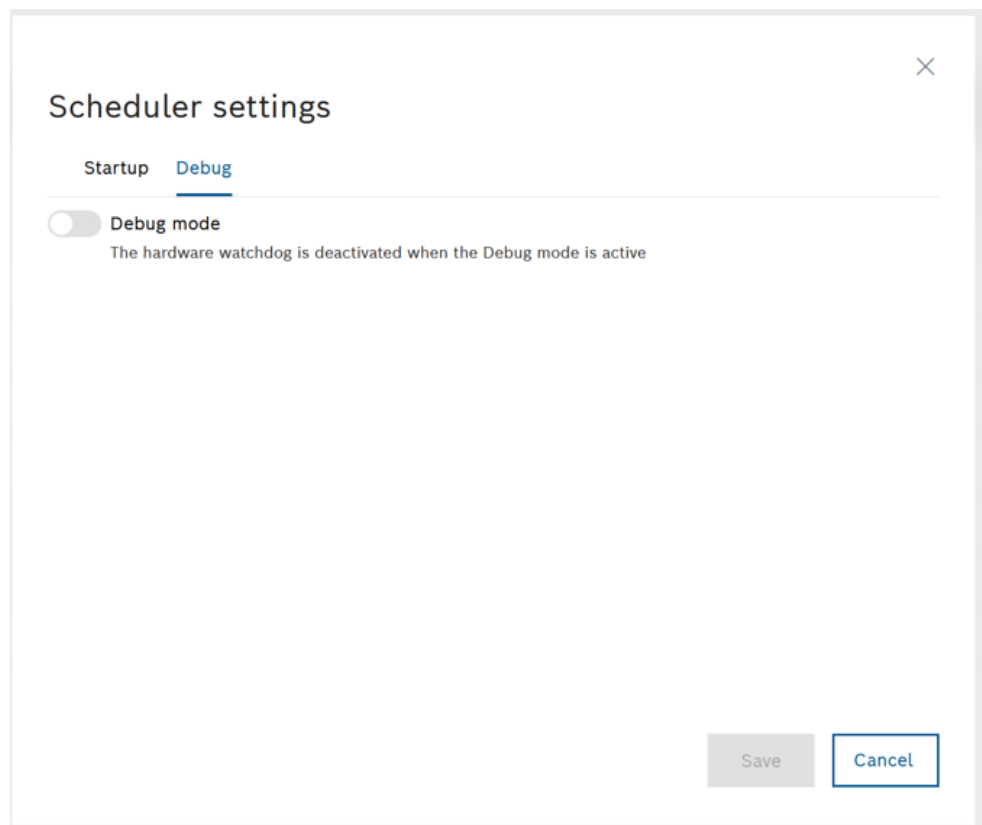


Fig. 52: “Debug” mode

⚠ WARNING

If the monitoring mechanisms are disabled, the safety measures might not apply or unintended motions might be executed. Do not use the “Debug” mode when the machine is running. This mode is only for development purposes.

(see [↗ scheduler/admin/controls](#))

Strict mode

If the Scheduler is in the strict mode, the hardware watchdog is active in the states SETUP and OPERATING states. Some actions, e.g. the installing an application, affect the real-time processing of the tasks.

Thus, it is required that the hardware watchdog and the task watchdog are disabled to execute the action and that no real-time-critical tasks are running.

Thus, always switch to the SERVICE state if the strict mode is active.

The mode is active in the following actions:

- A task watchdog is enabled.
- A task with an active watchdog is created.
- A callable is created with the requirement that the hardware watchdog has to be active.

(see [↗ scheduler/admin/info/strict-mode](#))

Scheduler load

As the same user programs can be used on different hardware platforms (with different performance), monitor the load of the Scheduler in addition to the task watchdogs for the user tasks.

The node *scheduler/admin/utilization/max* provided in the Data Layer, shows in percent how much the Scheduler requires of the available runtime of a Scheduler cycle. Due to unfavorably selected cycle times for the user tasks (e.g. 3 ms and 7 ms), it might be required that the Scheduler runs at a much higher cycle rate (e.g. 1 ms) than apparently necessary. This cause to an unnecessarily high load on less performant systems.

The following parameterizable limits are available for load monitoring:

<i>scheduler/admin/utilization/threshold-warning</i>	50 %	Can be parameterized, a warning is generated if exceeded
<i>scheduler/admin/utilization/threshold-error</i>	75 %	Can be parameterized, an error is generated if exceeded
---	100 %	Cannot be parameterized, an error is generated if exceeded and switched to SERVICE

(see [↗ scheduler/admin/utilization](#))

(see [↗ scheduler/admin/utilization/max](#))

(see [↗ scheduler/admin/utilization/threshold-warning](#))

(see [↗ scheduler/admin/utilization/threshold-error](#))

9.3 Exception handling

9.3.1 Introduction and overview

ctrlX applications are provided with mechanisms to log exceptions.

Logging is used for an advanced error analysis of applications in case of a service call.

If an exception occurs in a ctrlX application, the runtime system intercepts the exception and is logged in an exception log.

The issued exception is signaled by an implemented LED status display and the application is ended. Depending on the restart-condition of the respective application, it is restarted automatically.

It is continuously written to the log file if there are new exceptions during the following program executions.

If the data size of the log file is more than 100 kBytes, a backup file is generated and the log file is reset.

The following information is stored in the log file:

- Name of the application program
- Date and time in UTC format (Universal Time, Coordinated)
- Signal number and code
- Application register
- Call chain of a user program

9.3.2 Storage location of the log file

The log file is stored under the following name with regard to the snap names:

"var/snap/<snap-name>/current/package-run/<snap-name>/logs/<snap-name>.ctrlxex.log"

The extension of the respective backup file is *.bak*:

"var/snap/<snap-name>/current/package-run/<snap-name>/logs/<snap-name>.ctrlxex.log.bak"

Example of the snap *rexroth-automationcore*

"var/snap/rexroth-automationcore/current/package-run/rexroth-automationcore/logs/rexroth-automationcore.ctrlxex.log"

"var/snap/rexroth-automationcore/current/package-run/rexroth-automationcore/logs/rexroth-automationcore.ctrlxex.log.bak"

9.3.3 Structure of the log file

The log file is divided into three sections:

- "Application Exception"
- "Application Register"
- "Application Callchain"

"Application Exception" section

This section includes the following information:

- Name of the application
- Date and time
- Signal number
- Identifier
- Signal code

For more information on the signal numbers and IDs, refer to the **"Portable Operating System Interface (POSIX), Standard IEEE 1003"**.

Example

```
----- APPLICATION EXCEPTION -----  
NAME: <APPLICATION NAME>  
DATE: 2020-03-10 05:47:23 GMT  
SIGNAL NUMBER: 11 (SIGSEGV)  
SIGNAL CODE: 128
```

"Application Register" section

This section includes the current pointers resulting during the exception:

- Instruction Pointer
- Stack Pointer
- Base Pointer

Example

```
----- APPLICATION REGISTER -----
Instruction Pointer (IP): 0x7ffefcf61d2
Stack Pointer (SP): 0x7ffbcfab910
Base Pointer (BP): 0x7ffbcfab950
```

"Application Callchain" section

The section includes the calling hierarchy in the following display format:

**{FrameNumber} {ModuleName} ({optional FunctionName} + {ModuleOffset})
[{FunctionAddress}]**

Example of a callchain with five frames

```
----- APPLICATION CALLCHAIN -----
0 /snap/foo/x0/foo/.cache/bundle1/version0.0/libfoo.so(bundleActivator_stop+0x36) [0x7ffefcf61d2]
1 /lib/x86_64-linux-gnu/libpthread.so.0(+0x76db) [0x7ffff653f6db]
2 /lib/x86_64-linux-gnu/libc.so.6(clone+0x3f) [0x7ffff712a88f]
3 /lib/x86_64-linux-gnu/libc.so.6(start+0x3f) [0x7ffff712345f]
4 /lib/x86_64-linux-gnu/libc.so.6(init+0x113f) [0x7ffff79aaddf]
```

9.3.4 Diagnostics

If an exception resulted and after a log file has been created, an integrated status LED display of the ctrlX signals SysError via the operating state "System error".

For the LED states of the ctrlX CORE, refer to [Chapter 9.1.5 LED diagnostic code on page 149](#).

9.4 Hardware watchdog

The hardware watchdog of the ctrlX OS Runtime software monitors the proper basic functionality of the ctrlX CORE Scheduler. The Scheduler regularly notifies the watchdog after specified periods.

If there is no notification within double the cycle time, as the Scheduler is interrupted or blocked, an F8 error is output. The ctrlX OS Runtime is ended and the EtherCAT master interface is disabled. After triggering the hardware watchdog, the control has to be restarted to reinitialize the hardware watchdog.

If the system cannot be shut down properly in case of a hardware watchdog, the system is restarted.

The hardware watchdog monitoring of the Scheduler can be disabled for debugging purposes. Otherwise, an error response (shutdown, reboot) is triggered immediately after reaching the breakpoint. For disabling purposes, create respective "Debug" controls in the Scheduler. Apply "machine" as parameter and the serial number of the control. Disabling the monitoring can be made persistent. It is signaled with a permanent warning issued while enabling and starting the ctrlX OS Runtime software. Alternatively, monitoring can be disabled in the Scheduler settings.

Disabling of the hardware watchdog for debugging purposes via Data Layer

POST <https://192.168.1.1:443/automation/api/v1.0/scheduler/admin/controls>

```
{
  "control_type": "Debug",
  "control": {
    "machine": "123456789"
  }
}
```

Disabling of the hardware watchdog for debugging purposes via the Scheduler settings

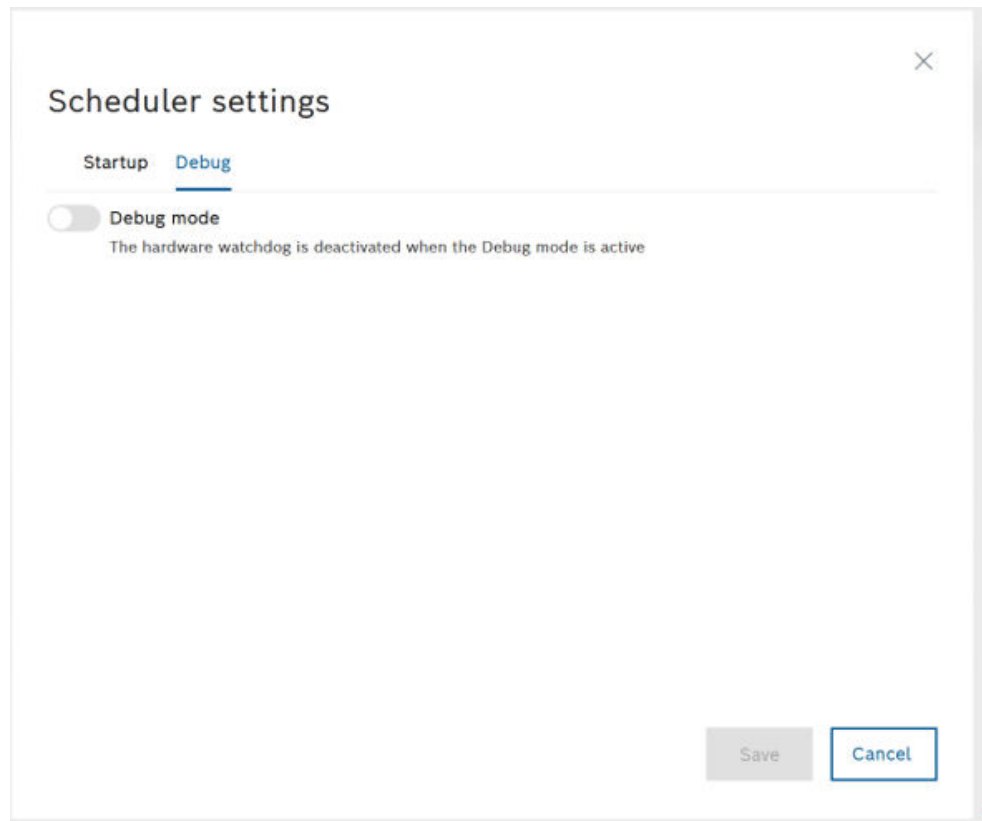


Fig. 53: Debug mode

NOTICE

If the monitoring mechanisms are disabled, the safety measures might not apply or unintended motions might be executed. Do not use the “Debug” mode when the machine is running. This mode is only for development purposes.

Hardware watchdog in the ctrlX CORE Scheduler state SERVICE

The hardware watchdog is disabled in the ctrlX CORE Scheduler state SERVICE.

Further topics

- [Chapter Task status on page 170](#)

9.5 Script parser/interpreter

9.5.1 Introduction and overview

Scripts can be executed with the ctrlX CORE. Scripts are simple text files with commands in the respective script language. The commands of the script file are subsequently processed during execution. Very simple and intuitive command sequences can be created.

To execute a script, an app has to be installed with the respective script language. The ctrlX AUTOMATION currently provides the script language ➔ [Python](#). More script languages are planned. Each script language is extended by specific functions for ctrlX CORE. Their specific syntax is described in the functions/commands.

All scripts are part of the configuration and have to be stored in the configuration to be processed in the ctrlX CORE.

9.5.2 User interface via Data Layer

To process a script, create the script interpreter instance first. Specify the script language in which the instance is to be processed. Any number of script interpreter instances can be created (limited by the ctrlX CORE memory). The instances run completely independent from each other.

After a script interpreter instance was created, an execution script can be specified. The script starts automatically.

To query information on the current state, use the interfaces in the ➔ [Data Layer](#). A sequence control is also possible. It can be used to cancel the script processing for example.

Depending on the script language, the process of the script interpreter instance is created again ("lifecycle once") or only reinitialized ("lifecycle permanent") after the end of the program or after an error.

For example, "lifecycle once" is preset for the Python script language and "lifecycle permanent" for G-code.

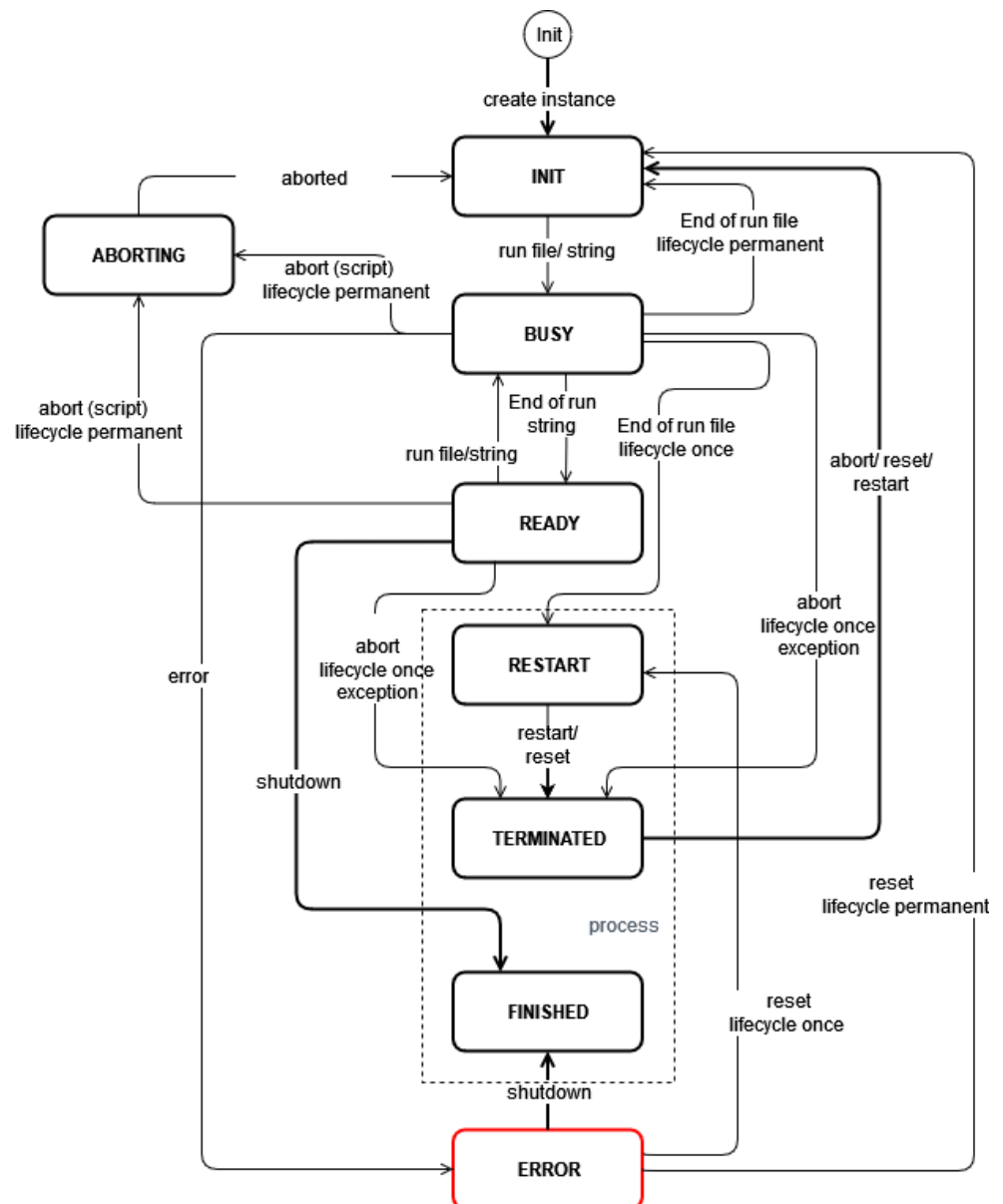


Fig. 54: Sequential control

9.5.3 Initialization script

An initialization script can be configured in the ctrlX CORE. When starting the ctrlX CORE, a script interpreter instance is created automatically and the configured script is started in this instance.

This script can for example:

- Creating additional script interpreter instances
- Starting more scripts in the generated script interpreter instances
- Writing a configuration into the Data Layer
- Executing first commands
- etc.

9.5.4 Search paths for Python modules

The configuration path for the script manager is provided in \$SNAP_COMMON (/var/snap/rexroth-automationcore/common) by the "rexroth-automationcore" app.

Python scripts without relative or absolute path specification are searched in \$SNAP_COMMON/solutions.

Script execution

For the "robot" instance, process the \$SNAP_COMMON/solutions/activeSolution/script/loadWorkpiece.py script.

Under the Data Layer node *script/instances/robot/cmd/file*, the payload {"name":"activeSolution/script/loadWorkpiece.py","param":"pallet1"} is posted.

Optionally, the absolute path can be specified using `"/var/snap/rexroth-automationcore/common/solutions/activeSolution/script/loadWorkpiece.py"`.

Imported Python modules are searched in the following order:

Current script directory

- ./

Application modules:

- \$SNAP_COMMON/solutions/activeConfiguration/scripts/user
- \$SNAP_COMMON/solutions/activeConfiguration/scripts/oem
- \$SNAP_COMMON/solutions/activeConfiguration/scripts/bosch

Libraries if available at the time of instance generation:

- \$SNAP_COMMON/solutions/activeConfiguration/scripts/libraries/user
- \$SNAP_COMMON/solutions/activeConfiguration/scripts/libraries/oem
- \$SNAP_COMMON/solutions/activeConfiguration/scripts/libraries/bosch

9.5.5 Python functions for Data Layer access

Function (nodes) = datalayer.browse (<path>)

(nodes) = datalayer.browse (<path>)

The function browses through a specified Data Layer path and returns all nodes of a level below this path.

- <path>
String, Data Layer path to be browsed through
- (nodes)
Tuple of node names

Function <value> = datalayer.read (<path>)

<value> = datalayer.read (<path>)

The function reads an element of the Data Layer and returns its value. Only simple types are supported.

- <path>
String, Data Layer path to be read
- <value>
Value node (only simple values are supported)

Function datalayer.write(<path>, <value>)

datalayer.write(<path>, <value>)

The function writes an element of the Data Layer. Only simple types are supported.

- <path>

- String, Data Layer path to be written
- **<value>**
Value to be written (only simple values are supported)

Function `datalayer.create(<path>, <value>)`

`datalayer.create(<path>, <value>)`

The function creates an element with the specified value in the Data Layer. Only simple types are supported.

- **<path>**
String, Data Layer path to be created
- **<value>**
Value node (only simple values are supported)

Function `datalayer.remove(<path>)`

`datalayer.remove(<path>)`

The function deletes an element of the Data Layer.

- **<path>**
String, Data Layer path to be deleted

Function `datalayer.read_json(<path>)`

`datalayer.read_json(<path>)`

The function reads an element of the Data Layer and returns its value as JSON string.

- **<path>**
String, Data Layer path to be read
- **<value>**
Value of the node as JSON string

Function `datalayer.write_json(<path>, <value>)`

`datalayer.write_json(<path>, <value>)`

The function writes an element of the Data Layer. It is specified as JSON string.

- **<path>**
String, Data Layer path to be written
- **<value>**
Value to write the JSON string

Function `datalayer.create_json(<path>, <value>)`

`datalayer.create_json(<path>, <value>)`

The function creates an element with the specified value in the Data Layer. The value is a JSON string

- **<path>**
String, Data Layer path to be created
- **<value>**
Value of the node JSON string
- **<result>**
Result during the creation as JSON string (only if successful)

Further information

Further information

Documentation "Python Runtime App" chapter [Python functions](#)

10 ctrlX Data Layer

10.1 ctrlX OS Data Layer - Central data interface of ctrlX OS

10.1.1 Data storage and data access

ctrlX OS Data Layer is the central data storage of ctrlX OS.

All information generated during the configuration and operation of a system is ctrlX OS is stored in the ctrlX OS Data Layer and is generally accessible via the open, documented interface of the ctrlX OS Data Layer.

Using the rights system of the ctrlX OS Data Layer access to individual data nodes and data elements can be restricted by the user.

For example, a machine manufacturer can restrict access to certain areas by the operator and also by Bosch Rexroth.

Bosch Rexroth does not specify any access restrictions upon delivery. Bosch Rexroth also has no way of circumventing any restrictions imposed by a user.

Bosch Rexroth thus guarantees transparent access to all system data ctrlX OS.

10.1.2 Scope of data

The scope of the data in the ctrlX OS Data Layer depends on the installed apps that a customer can install on a ctrlX OS system. (Note: Bosch Rexroth does not install any additional apps upon delivery, unless explicitly agreed otherwise with the customer).

All data generated during the configuration and operation of ctrlX OS system is typically:

- Configuration data that define the behavior of a ctrlX OS app
- Process data that an app creates during production and exchanges with other apps

In addition to this app-specific data, the ctrlX OS Data Layer includes general configuration and telemetry data describing for example the physical interfaces and usage information (e.g. operating hours and memory load) of the ctrlX OS system.

In general, the ctrlX OS Data Layer does not contain any personal data.

No personal data is collected while installing other ctrlX apps either, unless this is explicitly described in the product documentation of a corresponding app. At the time of publication of this documentation, Bosch Rexroth is not aware of such an app.

10.1.3 Further information

For further information on handling the ctrlX OS Data Layer programming interface (API) and on the content of individual Data Layer nodes, go to [ctrlX SDK](#) and refer to the corresponding chapters of the ctrlX OS documentation

The exact design of the interface as well as the content depends on the respective ctrlX OS version and on the specific apps installed.

10.2 Data exchange via Data Layer

The ctrlX Data Layer defines the data interface of the control.

This includes:

- **Access:**
 - How is data accessed? How is data provided?
 - How does the access or provision take place on the control or from other devices?
- **Addressing:**
What is the address of a datum; how is data structured hierarchically?
- **Representation:**
How does the data type of a datum look like (simple or complex data types)?
- **Runtime behavior:**
How is real-time capable or non-real-time-capable data accessed?
- **Security:**
How does the authentication work? How is access to data authorized?
- **Transport:**
Which mechanisms are used to exchange data? How is data coded during its transport?

The following figure provides an overview on all subsections of the ctrlX Data Layer.

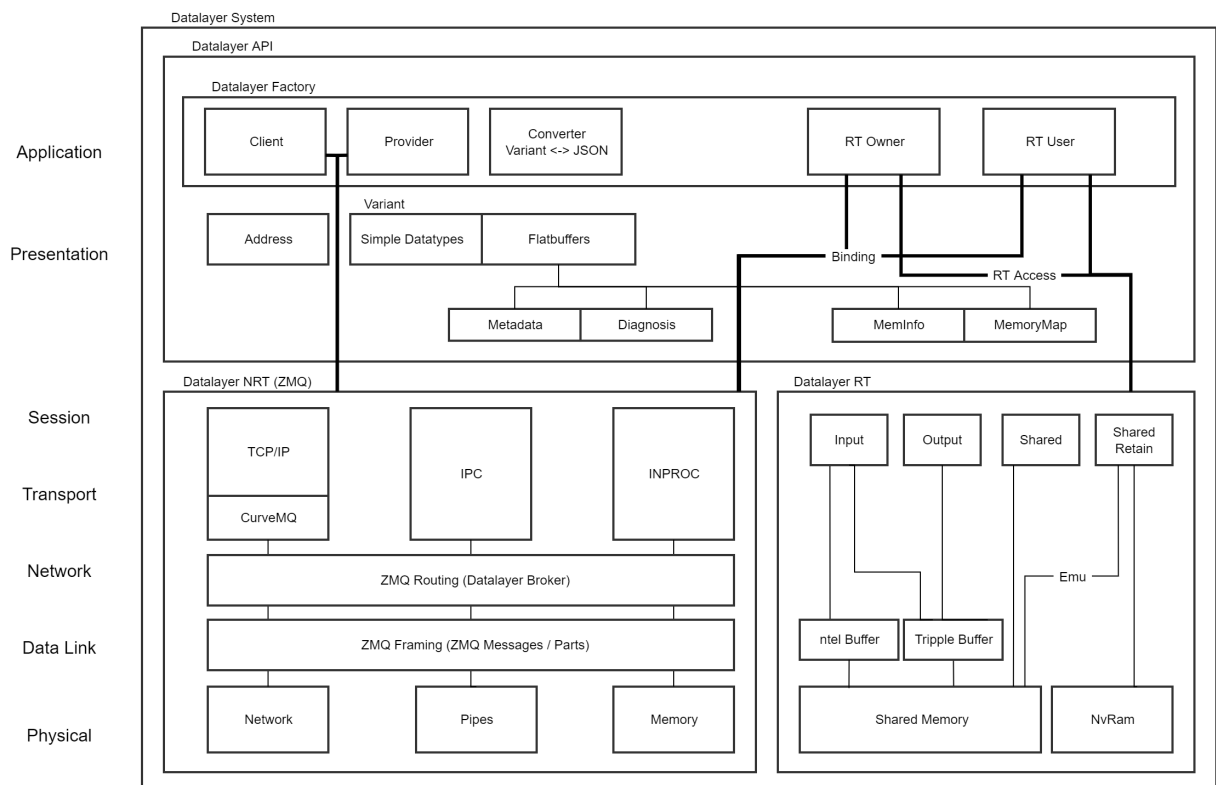


Fig. 55: ctrlX Data Layer - Overview

Further information

- [Chapter 13.2.7 Window – “Data Layer” on page 288](#)

10.3 Addresses

Each data (node) has an address in the ctrlX Data Layer. An address is a hierarchical path to the node. The individual components of the path are separated by a “/”. This structures all data hierarchically. Subaddresses only used to set up the hierarchical structure do not have to include any data.

Examples of addresses:

- datalayer/nodes
- plc/admin/cmd/start
- motion/axes/1/cur-values/cur-pos

An address does not contain any “/” at its beginning and at its end. In case of queries, “/” are removed automatically at the beginning and at the end.

10.4 Data types

10.4.1 Introduction and overview

All Data Layer data is stored in an own type called Variant.

One Variant can contains different data types.

The following data types are defined:

Table 8: Definition of the data types in the Data Layer

Data Layer data types:				
#	VariantType (enum)	Description	Data Layer reference type alias Conversion: • MemoryMapHelper::equalTypes(const std::string type1, VariantType type2) • VariantType MemoryMapHelper::toVariantType(std::string type)	Short name string (json type) alias Conversion: • std::string Variant::typeAsString() • VariantType Variant::getVariantTypeByString(std::string type)
0	UNKNOWN	Unknown data type		unknown
1	BOOL8	Bool 8 Bit	types/datalayer bool8 types/plc/bit (One Bit)	<u>bool8</u>
2	INT8	Signed INT 8 Bit	types/datalayer/int8 types/plc/sint	<u>int8</u>
3	UINT8	Unsigned INT 8 Bit	types/datalayer/uint8 types/plc/byte types/plc USINT	<u>uint8</u>
4	INT16	Signed INT 16 bit	types/datalayer/int16 types/plc/int	<u>int16</u>
5	UINT16	Unsigned INT 16 Bit	types/datalayer/uint16 types/plc/word types/plc/uint	<u>uint16</u>
6	INT32	Signed INT 32 Bit	types/datalayer/int32 types/plc/dint	<u>int32</u>
7	UINT32	Unsigned INT 32 Bit	types/datalayer/uint32 types/plc/dword types/plc/uint	<u>uint32</u>
8	INT64	Signed INT 64 Bit	types/datalayer/int64 types/plc/lint	<u>int64</u>
9	UINT64	Unsigned INT 64 Bit	types/datalayer/uint64 types/plc/lword types/plc/ulint	<u>uint64</u>

Data Layer data types:				
#	VariantType (enum)	Description	Data Layer reference type alias Conversion: - MemoryMapHelper::equalTypes(const std::string type1, VariantType type2) - VariantType MemoryMapHelper::toVariantType(std::string type)	Short name string (json type) alias Conversion: - std::string Variant::typeAsString() - VariantType Variant::getTypeByName(std::string type)
10	FLOAT32	Float 32 Bit	types/datalayer/float32 types/plc/real	float float32
11	FLOAT64	Float 64 Bit	types/datalayer/float64 types/plc/lreal	double float64
12	STRING	String (UTF-8)	types/datalayer/string types/plc/string (not for DL-RT)	string
13	ARRAY_OF_BOOL8	Array of Bool 8 bit	types/datalayer/array-of-int8 types/plc/array-of-sint (not for DL-RT)	arbool8
14	ARRAY_OF_INT8	Array of Signed INT 8 Bit	types/datalayer/array-of-int8 types/plc/array-of-sint	arint8
15	ARRAY_OF_UINT8	Array of Unsigned INT 8 Bit	types/datalayer/array-of-uint8 types/plc/array-of-byte types/plc/array-of-usint	aruint8
16	ARRAY_OF_INT16	Array of Signed INT 16 Bit	types/datalayer/array-of-int16 types/plc/array-of-int	arint16
17	ARRAY_OF_UINT16	Array of Unsigned INT 16 Bit	types/datalayer/array-of-uint16 types/plc/array-of-word types/plc/array-of-uint	aruint16
18	ARRAY_OF_INT32	Array of Signed INT 32 Bit	types/datalayer/array-of-int32 types/datalayer/int32 types/plc/array-of-dint	arint32
19	ARRAY_OF_UINT32	Array of Unsigned INT 32 Bit	types/datalayer/array-of-uint32 types/plc/array-of-dword types/plc/array-of-uint	aruint32
20	ARRAY_OF_INT64	Array of Signed INT 64 Bit	types/datalayer/array-of-int64 types/plc/array-of-lint	arint64
21	ARRAY_OF_UINT64	Array of Unsigned INT 64 Bit	types/datalayer/array-of-uint64 types/plc/array-of-lword types/plc/array-of-ulint	aruint64
22	ARRAY_OF_FLOAT32	Array of Float 32 Bit	types/datalayer/array-of-float32 types/plc/array-of-real	arfloat arfloat32
23	ARRAY_OF_FLOAT64	Array of Float 64 Bit	types/datalayer/array-of-float64 types/plc/array-of-lreal	ardouble arfloat64
24	ARRAY_OF_STRING	Array of String (UTF-8)	types/datalayer/array-of-string types/plc/array-of-string (not for DL-RT)	arstring
25	RAW	Raw Bytes		raw

Data Layer data types:				
#	VariantType (enum)	Description	Data Layer reference type alias Conversion: - MemoryMapHelper::equalTypes(const std::string type1, VariantType type2) - VariantType MemoryMapHelper::toVariantType(std::string type)	Short name string (json type) alias Conversion: - std::string Variant::typeAsString() - VariantType Variant::getTypeByName(std::string type)
26	FLATBUFFERS	Bytes as complex data type codes as FlatBuffer		flatbuffers object (for type in JSON)
27	TIMESTAMP	timestamp (FILETIME) 64 bits 100 ns since 1.1.1601 (UTC)	types/datalayer/timestamp	timestamp
28	ARRAY_OF_TIMESTAMP	Array of timestamp (FILETIME) 64 bits 100 ns since 1.1.1601 (UTC)	types/datalayer/array-of-timestamp	artimestamp

Table 9: List of data types in the Data Layer and the equivalents in the different programming languages

Data Layer data types		Respective data types in the programming languages							
#	VariantType (enum)	C++	.NET (C#)	Java	Golang	Python	JavaScript	IEC	FlatBuffers
0	UNKNOWN	nullptr	null	null	nil	none	null		null
1	BOOL8	bool8	bool (System.Boolean)	boolean (primitive)	boolean	bool	boolean	BIT	bool
2	INT8	int8	sbyte (System.Byte)	byte (primitive)	int8	int	number	SINT	ubyte uint8
3	UINT8	uint8	byte (System.Byte)	UnsignedByte UByte (no built-in type)	uint8	int	number	BYTE USINT	ubyte uint8
4	INT16	int16	short (System.Int16)	short (primitive)	int16	int	number	INT	short int16
5	UINT16	uint16	ushort (System.UInt16)	UnsignedShort UShort (no built-in type)	uint16	int	number	WORD UINT	ushort uint16
6	INT32	int32	int (System.Int32)	int (primitive)	int32	int	number	DINT	int int32
7	UINT32	uint32	uint (System.UInt32)	UnsignedInteger UInteger (no built-in type)	uint32	int	number	DWORD UDINT	uint uint32
8	INT64	int64	long (System.Int64)	long (primitive)	int64	int	number	LINT	long int64
9	UINT64	uint64	ulong (System.UInt64)	UnsignedLong ULong (no built-in type)	uint64	int	number	LWORD ULINT	ulong uint64

Data Layer data types		Respective data types in the programming languages							
#	VariantType (enum)	C++	.NET (C#)	Java	Golang	Python	JavaScript	IEC	FlatBuffers
10	FLOAT32	float	float (System.Single)	float (primitive)	float32	float	number	REAL	float
11	FLOAT64	double	double (System.Double)	double (primitive)	float64	float	number	LREAL	double
12	STRING	char* (null terminated)	string (System.String)	string	string	str	string	STRING	string
13	ARRAY_OF_BOOL8	bool8[]	bool[]	boolean[]	[]Boolean	typing.List[bool]	object	ARRAY OF SINT	[bool]
14	ARRAY_OF_INT8	int8[]	sbyte[]	byte[]	[]int8	typing.List[int]	object	ARRAY OF SINT	[byte] [int8]
15	ARRAY_OF_UINT8	uint8[]	byte[]	UnsignedByte[] UByte[] (no built-in type)	[]uint8	typing.List[int]	object	ARRAY OF BYTE ARRAY OF USINT	[ubyte] [uint8]
16	ARRAY_OF_INT16	int16[]	short[]	short[]	[]int16	typing.List[int]	object	ARRAY OF INT	[short] [int16]
17	ARRAY_OF_UINT16	uint16[]	ushort[]	Unsigned-Short[] UShort[] (no built-in type)	[]uint16	typing.List[int]	object	ARRAY OF WORD ARRAY OF UINT	[ushort] [uint16]
18	ARRAY_OF_INT32	int32[]	int[]	int[]	[]int32	typing.List[int]	object	ARRAY OF INT32 ARRAY OF DINT	[int] [int32]
19	ARRAY_OF_UINT32	uint32[]	uint[]	UnsignedInteger[] UInteger[] (no built-in type)	[]uint32	typing.List[int]	object	ARRAY OF DWORD ARRAY OF UDINT	[uint] [uint32]
20	ARRAY_OF_INT64	int64[]	long[]	long[]	[]int64	typing.List[int]	object	ARRAY OF LINT	[long] [int64]
21	ARRAY_OF_UINT64	uint64[]	ulong[]	UnsignedLong[] ULong[] (no built-in type)	[]uint64	typing.List[int]	object	ARRAY OF LWORD ARRAY OF ULINT	[ulong] [uint64]
22	ARRAY_OF_FLOAT32	float[]	float[]	float[]	[]float32	typing.List[float]	object	ARRAY OF REAL	[float]
23	ARRAY_OF_FLOAT64	double[]	double[]	double[]	[]float64	typing.List[float]	object	ARRAY OF LREAL	[double]
24	ARRAY_OF_STRING	char**	string[]	string[]	[]string	typing.List[str]	object	ARRAY OF STRING	[string]

Data Layer data types		Respective data types in the programming languages							
#	VariantType (enum)	C++	.NET (C#)	Java	Golang	Python	JavaScript	IEC	FlatBuffers
25	RAW	uint8[]	byte[]	UnsignedByte[] UByte[] (no built-in type)	[]uint8	typing.List[int]	object		[ubyte]
26	FLATBUFFERS		ByteBuffer				object	Flatbuffer library	[ubyte]
27	TIMESTAMP	uint64	DateTime	LocalDateTime	uint64 time.time		Date		uint64
28	ARRAY_OF_TIME-STAMP	uint64[]	DateTime[]	LocalDate-Time[]	[]uint64 []time.time		Array<Date>		[uint64]

10.4.2 Simple data types

The following simple data types are supported:

Table 10: ctrlX Data Layer - Simple data types

Data type	Description
BOOL8	Boolean 8 bits
INT8	Signed 8 bit number
UINT8	Unsigned 8 bit number
INT16	Signed 16 bit number
UINT16	Unsigned 16 bit number
INT32	Signed 32 bit number
UINT32	Unsigned 32 bit number
INT64	Signed 64 bit number
UINT64	Unsigned 64 bit number
FLOAT32	Floating point number (IEEE-754 simple accuracy)
FLOAT64	Floating point number (IEEE-754 double accuracy)
STRING	Character string
ARRAY_OF_BOOL8	Array of BOOL8
ARRAY_OF_INT8	Array of INT8
ARRAY_UINT8	Array of UINT8
ARRAY_INT16	Array of INT16
ARRAY_UINT16	Array of UINT16
ARRAY_INT32	Array of INT32
ARRAY_UINT32	Array of UINT32
ARRAY_INT64	Array of INT64
ARRAY_UINT64	Array of UINT64
ARRAY_FLOAT32	Array of FLOAT32
ARRAY_FLOAT64	Array of FLOAT64
ARRAY_OF_STRING	Array of character strings
TIMESTAMP	timestamp (FILETIME) 64 bits 100 ns since 1.1.1601 (UTC)
ARRAY_OF_TIMESTAMP	Array of timestamp (FILETIME) 64 bits 100 ns since 1.1.1601 (UTC)

10.4.3 Complex data types

Complex data types are represented using FlatBuffer. FlatBuffer is a Google-engineered efficient, platform-comprehensive and programming language-comprehensive serialization platform for data. More complex data can be described, transferred and accessed using different programming languages.

The structure of a datum is described using a schema file (.fbs). Access methods are then generated from the individual programming languages. Data can still be converted from and into JSON.

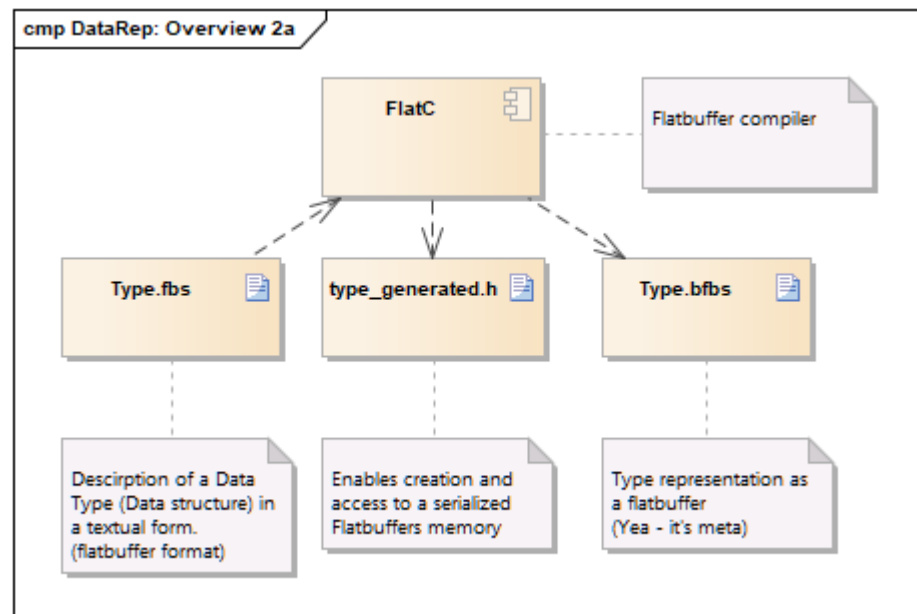


Fig. 56: ctrlX Data Layer - Complex data types

- **Type.fbs:**
Type description
- **FlatC:**
Tool to generate access methods and descriptions of a FlatBuffer type description
- **Type_generated.h:**
Generated methods to generate a FlatBuffer of this type or to access a FlatBuffer of this type
- **Type.bfbs:**
Generic type in binary format. This file describes the type. It is used to store the type in the type tree

Complex data types are stored in the type tree. For the type tree, go to the path "types/..". All saved types are described by reflection using the FlatBuffer. The description type is schema.fbs. This description can be generated automatically from the schema file (.fbs). A self-defined type can be registered at the ctrlX Data Layer (see provider).

Data Layer - FlatC

FlatC is the compiler for FlatBuffer schemas. FlatC generates source code from FlatBuffer schemas for different programming languages. Use the source codes to access the FlatBuffer of the type and to generate new FlatBuffer of the type. The access code for all static types is provided via the [SDK](#).

There are various dynamically generated types in the control for which no source code is provided for access:

- ctrlX OPC UA Client connection to an external, remote OPC UA Server
- Loading an OPC UA companion model to the control
- PLC-defined structures from the PLC program
- MATLAB-defined structures from the MATLAB program

For these types, the FlatC compiler in the Data Layer can be used to generate the access code for every supported programming language.

Using FlatC in the Data Layer

FlatC can be used via the Data Layer node *datalayer/flatbuffers/flatc*. The compiler can be started on the node with create.

Transfer values:

- "name":
Name of the compiler run. The name is used as name for the result file (*<name>.zip*)
- "addresses":
Array of Data Layer type addresses to be compiled
- "options":
Options for the FlatC compiler

Possible programming languages and the corresponding FlatC options:

- --cpp, -c: Generates C++ Header
- --java, -j: Generates Java code
- --kotlin, -k: Generates Kotlin code
- --csharp, -n: Generates C# code
- --go, -g: Generates Go code
- --python, -p: Generates Python code
- --js, -s: Generates JavaScript code
- --ts: Generates TypeScript code
- --php: Generates PHP code
- --grpc: Generates RPC stub code for GRPC
- --dart: Generates Dart code
- --lua: Generates Lua code
- --lobster: Generates Lobster code
- --rust, -r : Generates Rust code
- --swift: Generates Swift code

For further compiler switches, refer to ➡ https://flatbuffers.dev/flatbuffers_guide_using_schema_compiler.html.

The status of the compiler run can be monitored via the corresponding child node. After a successful compiler run, the result can be downloaded from the control using the child node *<name>.zip*.

10.4.4 Data type “Variant”

The data type “Variant” combines simple and complex data types. Values of these data types can be set and read. The data type is used for multiple interface methods.

A “Variant” can set data using the following methods:

- **setValue(<type> value)**
 - value is a c-data type:
Variant type is identified automatically (incl. std::string)
 - value is a “vector” or “set”:
Array type of the Variant is identified automatically
Values are copied

- **setValue(<type> values, size_t count)**
value is a c-array
 - Variant type is automatically identified using the array type
 - Values are copied
- **shareValue<type> values, size_t count)**
 - In some cases, it is not required to copy the values to the “Variant”. It is sufficient to reference data. Referenced data has to remain valid up to the end of the service life of the “Variant”.
 - The “Variant” type is defined automatically
 - Only the pointer on data is saved
 - The method is faster than the “setValue”, as no data is copied
- **copyFlatbuffers()**
 - Copies data of a FlatBuffer to the “Variant”
- **shareFlatbuffers()**
 - References data of a FlatBuffer into the “Variant”
 - No data is copied. Only the pointer on the data is saved
 - FlatBuffer has to remain at runtime of the "Variant"
- **setSharedPointer(type, uint8_t* data, size_t size)**
 - Sets the “Variant” to a type with data and size
 - Data is not copied. Only the pointer on data is saved
 - Data has to remain at runtime of the “Variant”
- **setType(type, size_t) + copyRaw(data, size_t size)**
 - Sets a “Variant” with a type and data with a size
 - Data is copied

The data type is provided with the following interface methods to read data:

- **checkConvert(type)**
 - Check whether a Variant can be converted into a type
- **Simple data types**
 - A Variant can be assigned to a simple C-data type (type conversion). If a Variant value can be converted into the target data type, the value is assigned.

```
Variant variant;
uint32_t value1 = 4711;
variant.setValue(value1);
uint32_t value2 = variant; // value2 is 4711
uint8_t value3 = variant;  // value3 is 0 (variant is too big)
```

- **Arrays**
 - A Variant can be assigned to a C-array (type conversion). Use the method getCount() to determine the array size

```
Variant data; // data containing an array of string
const char** strArray = data;
for (size_t i = 0; i < data.getCount(); i++)
    printf("%s", strArray[i]);
```

- **Complex data types (FlatBuffer)**
 - To check whether a “Variant” contains a specific FlatBuffer type, the generated verifier of the FlatBuffer type can be used. Then, access the FlatBuffer using the generated access methods

```
Variant data; // data with a flatbuffer
if (STATUS_SUCCEEDED(data.verifyFlatbuffers(VerifyAxisBuffer))); // VerifyAxisBuffer is
generated out of .fbs file
{
    auto axis = GetAxis(data->getData()); // GetAxis is generated out of .fbs file
}
```

10.5 Node structure

10.5.1 Introduction and overview

A node consists of the following parts

- Address:
Path used to address this node
- Date:
Node value
- Metadata:
Additional node information

10.5.2 Metadata

Metadata describes a node additionally.

Metadata is described using a FlatBuffer. Metadata is defined as follows:

```

namespace comm.datalayer;

enum NodeClass : byte {
    Node,
    Method,
    Type,
    Variable,
    Collection,
    Resource,
    Program,
    Folder,
} // methods are commands
enum DisplayFormat : byte {Auto, Bin, Oct, Dec, Hex}

table Extension {
    key: string (key);
    value: string (required);
}

table LocaleText {
    id: string (key); // iso 639.1 - to be defined
    text: string (required);
}

table AllowedOperations
{
    read: bool = false; // get
    write: bool = false; // put
    create: bool = false; // post
    delete: bool = false; // delete
    browse: bool = true; // browse
}

table Reference
{
    type: string (key); // nodeid of type "readType", "writeType", "createType"
    targetAddress: string (required); // full qualified address of target
}

table Metadata {
    nodeClass: NodeClass = Node;
    operations : AllowedOperations (required);
    description: string (required); // markdown text only
    descriptionUrl: string (required); // URL to description
    displayName: string;
    displayFormat: DisplayFormat = Auto;
    unit: string;
    extensions : [Extension]; // extended key value pair
    references : [Reference];
    descriptions: [LocaleText];
    displayNames: [LocaleText];
}

root_type Metadata;

```

Table 11: ctrlX Data Layer - Metadata

Field	Description
nodeClass	Describes the node type • Node: Basic class of all nodes • Variable: The node represents a status value and a configurable value • Resource: The node represents an object encapsulating more variables as child nodes • Collection: The node is a parent object of a group of resources of the same type or at least of the basic type • Folder: The node improves the readability of the tree by grouping nodes • Program: The node stands for a long running action • Method: The node stands for a short running action that returns a result immediately • Type: A type definition or a schema of another node
operations	Describes the permitted accesses on the nodes
description	Node documentation
descriptionUrl	Link to the node description
displayName	Display name of the name
displayFormat	Display name of the node
unit	Node unit
extensions	Extended properties of the node as key and value pair
references	References to other nodes (also refer to References on page 197) • readType: Reference to the type in the type tree to be used for reading (read-only access) • writeType: Reference to the type in the type tree to be used for writing (write access) • createType: Reference to the type in the type tree to be used for creation (read-only access)

Node

The basic class is an unspecified node. Even though it is possible, it is not recommended to provide a node without specifying a node class via metadata. The ctrlX Data Layer consists of interconnected nodes.

A node consists of:

- A unique address is like a key and identifies your instance
- Up to five different operations that can be applied to the node
- Additional metadata that can be read by a client to return the node of the semantics better. Metadata is to be provided. Otherwise, default values are used.

Variable

A variable contains a value that can be read and also be optionally written.

It can only be distinguished between Read-Only variables and Write-Read variables.

Example

- Position of an axis (Read-Only variable)
- PLC variable mapped to the ctrlX Data Layer

The represented value can either be a primitive value or a complex structure. A variable can contain other variables as children to represent complex units.

Resource

The resource is an object that can abstract a real or a logical unit (e.g. axis, motor, kinematics). Resources can be created or deleted in the system. This can be performed explicitly by a direct user interaction or implicitly due to changes in state.

The resource name should be a unique identifier, e.g. a unique name or key. A resource normally contain a fixed set of variables, methods or programs that can be considered as interface or resource type.

A resource can consist of child resources or be linked to other resources. A resource can return a summary of its state as value.

The writing to a resource can be used to change significant settings of the resource. Settings can for example be changed automatically or the resource can be renamed (example: changing the name of an axis).

Collection

A Collection summarizes resources of the same type or the basic type as children. It can allow the setting of additional resources via the onCreate() operation. The collection can return an aggregated view (e.g. a summarized object or array of child data) as own value. The collection value is read-only.

Folder

Folder can be used to group the nodes for an improved readability. A Folder does not have a value and does not allow that any operations are applied to the it.

Method

A method is a short call (e.g. less than one second) returning a result with the response to the request. The method type of input and output data can be different.

Program

A program is a long running task (e.g. longer than one second) running asynchronously and does not return a result immediately or has a side effect that does not mandatorily be completed if the result of request returns to the caller. The request returns a unique taskId instead that can be used to follow the progress and the state of the started program.

Type

A type describes the syntax of a value. A type is described as FlatBuffer schema.

The type can be queried as one of the following schemata:

- fbs schema (FlatBuffer schema)
- fbs schema (binary FlatBuffer schema)
- JSON schema

Operations

Operation	Variable	Resource	Collection	Folder	Method	Program	Type
browse	+	+	+	+	-	+	+
create	-	+	+	-	-	+	-
delete	-	+	-	-	-	-	-
read	+	+	+	-	+	+	+
write	+	+	-	-	+	-	-

References

Metadata of a node can reference to other nodes.

The reference consists of a type and a targetAddress.

There are some predefined reference types.

Type	Description
readType	Type when reading a value (absolute node address), input type and output type are identical
readInType	Input type when reading a value
readOutType	Output type when reading a value
writeType	Type when writing a value (absolute node address), input type and output type are identical
writeInType	Input type when writing a value
writeOutType	Output type when writing a value
createType	Type when creating a value (absolute node address)
hasSave	Reference to the address to save nodes that have to be called after a node change to maintain the new remanent value

10.5.3 Metadata database (Mddb)

Introduction and overview

Metadata for nodes can be provided as follows:

- Via the onMetadata() of a ProviderNodes
Via a metadata database (Mddb)

This chapter describes the option to provide metadata via a metadata database.

The following table compares the cases in which it is better to provide metadata via the source code or via a metadata database:

	Provision in source (onMetadata)	Provision via Mddb
Advantages	• Entries can be dynamically generated using other settings.	• Clearly-structured central location of all metadata • Smaller footprint • Less maintenance effort
Disadvantages	• Spread in the source code • Higher footprint • High maintenance effort	• Only static metadata

A metadata database is provided via CSV file. This file is compiled via the Mddb compiler. To register the generated file at a provider using the registerType method. If the respective ProviderNode returns an error to the provider after a metadata request, it is tried to find the requested address in the Mddb.

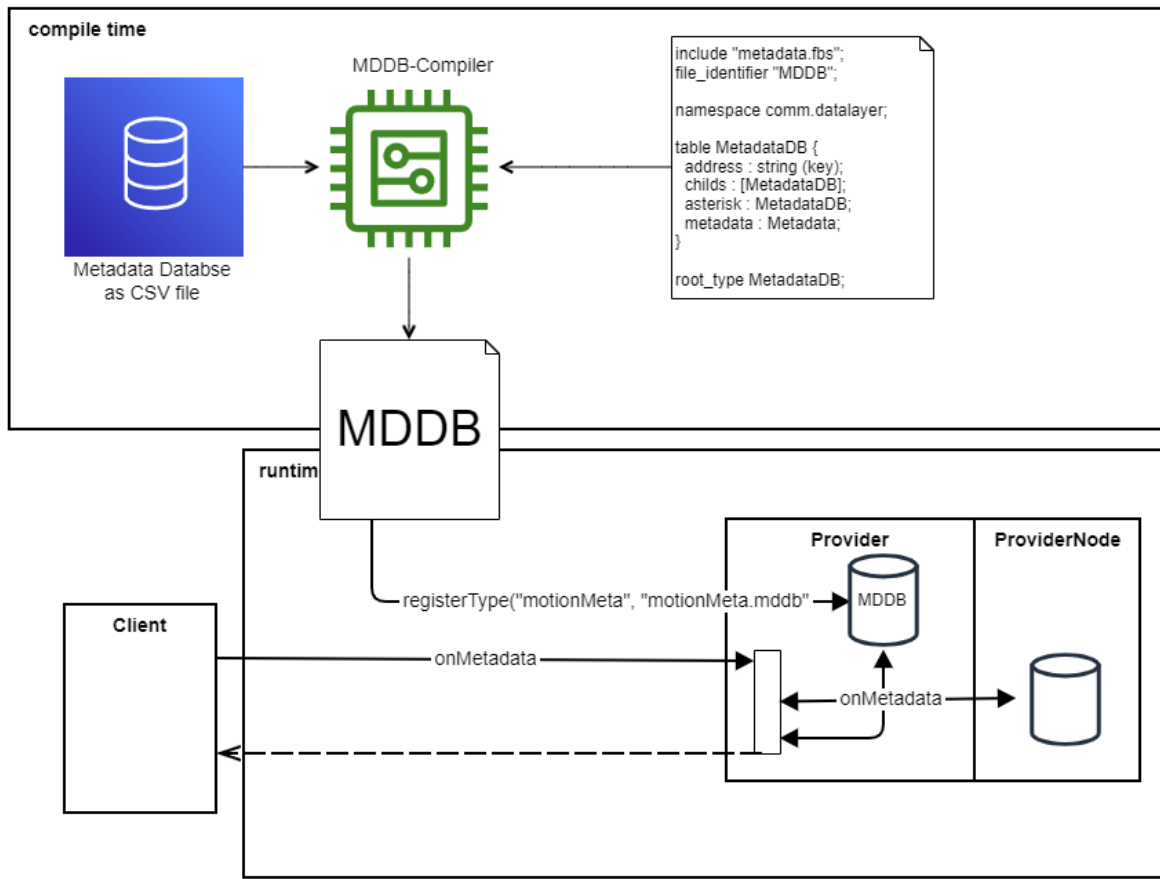


Fig. 57: Overview on the metadata database

CSV structure

The metadata database (Mddb) is a CSV file.

Columns are split by ";". Column headers have to be in the first line.

- Columns can be sorted again (address has to be at first position)
- String arrays can remain empty
- Additional columns ref.* and ext.* can be extended and are included into the metadata as references and extensions
- Additional columns (e.g. comment) can be added and then ignored
- Do not insert any type nodes (e.g. *types/motion/axis*) into this table. Metadata is returned by the type node itself

Table 12: Mandatory columns

Column	Type	Description
address	string	Node address • Placeholders "*" are permitted • Double placeholders "***" are not permitted

Column	Type	Description
nodeClass	NodeClass	• Method • Type • Variable • Collection • Resource • Program • Folder
operations.read	bool	Address can be read
operations.write	bool	Address can be written
operations.create	bool	Node can be created on this address
operations.delete	bool	Address can be deleted
operations.browse	bool	Address can be browsed
description	bool	Node description
descriptionURL	string	URL on the node description
displayName	string	Display name
displayFormat	Display-Format	• Auto (default) • Bin • Oct • Dec • Hex
unit	string	Node unit

Table 13: Optional columns

Column	Type	Description
references.readType	string	Type for the onRead method
references.readIn-Type	string	Input type for the onRead method
references.readOut-Type	string	Output type for the onRead method
references.write-Type	string	Type for the onWrite method
references.writeIn-Type	string	Input type for the onWrite method
references.write-OutType	string	Output type for the onWrite method
references.create-Type	string	Type for the onCreate method
references.<key>	string	Key/value pair added to metadata.references
extensions.<key>	string	Key/value pair added to metadata.extensions
descriptions.<language>	string	Language/description pair added to metadata.descriptions
displayNames.<language>	string	Language/display name added to metadata.displayName

MDDB compiler

Upon delivery, the MDDB compiler is provided with the ctrlX Data Layer and is available for Windows and Linux.

Table 14: Compiler options

Option	Description
-in <infile>	Sets the input file
-out <outfile>	Sets the output file
-verbose	Enables additional log outputs

The compiler can be integrated via CMake.

Table 15: build_mddb parameters

Parameter	Description
target	Build target that is generated
mddb files	MDDB files to be compiled
dest folder	Target folder for compiled files

10.6 Data transport NRT

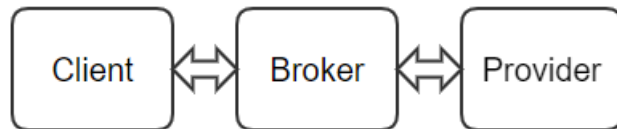


Fig. 58: ctrlX Data Layer - Data transport NRT

Clients can log in at a broker and send queries to nodes. Providers provide data and also log in at the broker. The broker forwards the client requests to the responsible provider and returns the response to the client.

Clients and providers report at the broker via a connection string:

tcp://max:muster@www.example.com:8080?p1=A&p2=B

/ _/_ _/_/_ _/_/_/_/_ _/_/_ _/_/_/_/_
 |
 Schema Kennwort Host Port Query
 Benutzer

Table 16: Connection types of clients and providers

Type	Optional	Address	Description
Schema		tcp	Different device
		ipc	Same device
		inproc	Same process space
Users	x		Username
Password	x		Password
Host	x		Device address
Port	x		Port at TCP (default 2069)
Query	x	sslport	SSL port to query user tokens (default 443)
	x	token	JWT to be used for the connection

The address is specified when creating a client or a provider. Irrespective of the connection type, all methods of clients and providers can be used. These connection types allow to use the same interface to query data from the same and from other controls and to provide them.

10.7 Accessing data (NRT)

10.7.1 Introduction and overview

A client is used to access data. There are different methods to access data. Only the C/C++ interface is described. The specific interface descriptions apply to other programming languages.

10.7.2 Synchronous methods

Synchronous methods wait for the result of the request. As they block the program sequence, use them only where possible.

Table 17: ctrlX Data Layer - Synchronous methods

Method	Value	Type	I/O	Description
readSync	address	string	Input	Reading a node
	data	Variant	Input/output	
	return	DLResult	Return value	
writeSync	address	string	Input	Writing a node
	data	Variant	Input/output	
	return	DLResult	Return value	
browseSync	address	string	Input	Reading the child nodes on the next node level Child nodes are returned as ARRAY_OF_STRINGS
	data	Variant	Output	
	return	DLResult	Return value	
createSync	address	string	Input	Generating a resource (if supported)
	data	Variant	Input/output	
	return	DLResult	Return value	
removeSync	address	string	Input	Deleting a resource (if supported)
	data	Variant	Input/output	
	return	DLResult	Return value	
metadata-Sync	address	string	Input	Reading metadata of a node
	data	Variant	Output	
	return	DLResult	Return value	
readBulk-Sync	request	VecBulkRequest	Input	Reading multiple nodes
	response	VecBulkResponse	Output	
	return	DLResult	Return value	
writeBulk-Sync	request	VecBulkRequest	Input	Writing multiple nodes
	response	VecBulkResponse	Output	
	return	DLResult	Return value	
createBulk-Sync	request	VecBulkRequest	Input	Reading the child nodes on the next node level of multiple nodes
	response	VecBulkResponse	Output	
	return	DLResult	Return value	
deleteBulk-Sync	request	VecBulkRequest	Input	Creating multiple resources (if supported)

Method	Value	Type	I/O	Description
	response	VecBulk-Response	Output	
	return	DLResult	Return value	
browseBulk-Sync	request	VecBulkRequest	Input	Deleting multiple resources (if supported)
	response	VecBulk-Response	Output	
	return	DLResult	Return value	
metadata-BulkSync	request	VecBulkRequest	Input	Reading metadata of multiple nodes
	response	VecBulk-Response	Output	
	return	DLResult	Return value	

There are two structures for asynchronous requests:

- VecBulkRequest
- VecBulkResponse

VecBulkRequest is a vector containing the address of the requested node and its data of each entry:

```

//! Structure for bulk request
typedef struct BulkRequest
{
    std::string address;        //!< Address for the request
    Variant data;              //!< Data of the request
}BulkRequest;

```

```
typedef std::vector<BulkRequest> VecBulkRequest;
```

VecBulkResponse is a vector containing the address of the requested node and its data of each entry. It also includes the status code of the request and the time stamp:

```

//! Structure for bulk request
typedef struct BulkResponse
{
    std::string address;        //!< Address for the response
    Variant data;              //!< Data of the response
    DLResult result;           //!< Result of the response
    uint64_t timestamp;        //!< Timestamp of the response
}BulkResponse;

```

```
typedef std::vector<BulkResponse> VecBulkResponse;
```

If a query is sent, the responses in VecBulkResponse and VecBulkRequest have the same order.

10.7.3 Asynchronous methods

Asynchronous methods issue requests. This method call returns immediately. If a response to the query is available, a callback is called that was sent together with the request. Callback structure:

```
void callback(DLResult status, const Variant* data)
```

The callback contains the status and the data of the request. The callback function is called with the client.

The following methods are available:

Table 18: ctrlX Data Layer - Asynchronous methods

Method	Value	Type	I/O	Description
readAsync	address	string	Input	Reading a node
	data	Variant	Input	
	callback	ResponseCall-back	Input	
	return	DLResult	Return value	
writeAsync	address	string	Input	Writing a node
	data	Variant	Input	
	callback	ResponseCall-back	Input	
	return	DLResult	Return value	
browseAsync	address	string	Input	Reading the child nodes on the next node level
	callback	ResponseCall-back	Input	
	return	DLResult	Return value	Child nodes are returned as ARRAY_OF_STRINGS
createAsync	address	string	Input	Generating a resource (if supported)
	data	Variant	Input	
	callback	ResponseCall-back	Input	
	return	DLResult	Return value	
removeAsync	address	string	Input	Deleting a resource (if supported)
	callback	ResponseCall-back	Input	
	return	DLResult	Return value	
metadataA-sync	address	string	Input	Reading metadata of a node
	data	ResponseCall-back	Input	
	return	DLResult	Return value	
readBulkA-sync	request	VecBulkRequest	Input	Reading multiple nodes
	callback	BulkCallback	Input	
	return	DLResult	Return value	
writeBulkA-sync	request	VecBulkRequest	Input	Writing multiple nodes
	callback	BulkCallback	Input	
	return	DLResult	Return value	
createBulkA-sync	request	VecBulkRequest	Input	Reading the child nodes on the next node level of multiple nodes
	callback	BulkCallback	Input	
	return	DLResult	Return value	

Method	Value	Type	I/O	Description
deleteBulkAsync	request	VecBulkRequest	Input	Creating multiple resources (if supported)
	callback	BulkCallback	Input	
	return	DLResult	Return value	
browseBulkAsync	request	VecBulkRequest	Input	Deleting multiple resources (if supported)
	callback	BulkCallback	Input	
	return	DLResult	Return value	
metadataBulkAsync	request	VecBulkRequest	Input	Reading metadata of multiple nodes
	callback	BulkCallback	Input	
	return	DLResult	Return value	

The callback for an asynchronous request is structured as follows:

```

/// Response function when a asynchronous call is ready.
/// @param[out] status    Response status of the call
/// @param[out] data      Response data
typedef std::function<void(DLResult status, const Variant* data)> ResponseCallback;

/// This callback delivers a vector of results for a bulk request.
/// It is called when the whole bulk request is finished.
/// The order of the items is the same as in the request vector.
/// @param[in] response   Vector of responses
typedef std::function<void(const VecBulkResponse& response)> BulkCallback;

```

10.8 Providing data (NRT)

10.8.1 Introduction and overview

The providers provide data to the ctrlX Data Layer. A provider can manage multiple nodes. A node responds to client requests for a specific address. To respond to requests, nodes can log in at the provider.

To log them in, use the following method:

```

void callback(DLResult status, cprovider::registerNode(const std::string &address,
IProviderNode* node)onst Variant* data)

```

If a login address contains wildcards, the logged-in node is responsible for multiple addresses.

10.8.2 Simple wildcard "*"

A simple wildcard is labeled with the "*" character in the login address. The simple wildcard can replace one or multiple path locations. Only the complete part (between two "/") may be replaced by a simple wildcard. The node is then valid for any replacements. A simple wildcard replaces one hierarchy level.

Example:

A node is logged in under the address "motion/axes/*/pos". This node should provide the current position of an axis. All requests provided with any replacements for "*" on this hierarchical level are forwarded to this node. Examples: "motion/axes/1/pos", "motion/axes/axis1/pos" or "motion/axes/foobar/pos".

10.8.3 Double wildcard "***"

A double wildcard is labeled with the "***" character in the login address. The double wildcard can replace any number of hierarchical levels and can only be used at the end of a login address. It can be used to label that a node is responsible for a complete subtree.

10.8.4 Responsibility of nodes

The address is used upon each request to decide which node is responsible for the request. The node with the "best matching login address" is used. Login addresses with a specific value on one level have a higher priority than login addresses with a wildcard on this level. A simple wildcard has a higher priority than a double wildcard.

Only nodes with a wildcard at the end of their login address receive browse or create requests.

10.8.5 Example of node responsibility

The following nodes are logged in:

Table 19: Logged-in nodes

Login address	Node
test/*	Node1
test/foo/*/bar	Node2
test/*/foo	Node3
test/*/*	Node4
test/bar/bar	Node5
test/**	Node6
test/bar/**	Node7

The following table shows which node is responsible for which specific request.

Table 20: Responsibility of nodes

Request address	Read/Write/Delete	Browse/Create
Test		Node1
test/1	Node1	Node4
test/1/foo	Node3	Node6
test/1/bar	Node4	Node6
test/foo	Node1	Node4
test/bar	Node1	Node7
test/bar/bar	Node5	Node7
test/bar/foo	Node7	Node7
test/bar/bar/foo	Node7	Node7
test/bar/foo/foo	Node7	Node7
test/foo/foo	Node3	Node6
test/foo/foo/bar	Node2	Node6
test/foo/foo/foo	Node6	Node6
test/1/bar	Node4	Node6
test/1/bar/1	Node6	Node6

10.8.6 Special features for browse requests

Browse requests return nodes below the requested node. Nodes resulting due to login addresses are automatically returned in case of such requests. The specific nodes resulting for wildcards are requested at the responsible node and merged with the nodes of the login addresses.

Example of logged-in nodes listed above:

Table 21: Special features for browse requests

Request address	Read/Write/Delete
	"test"
Test	"foo", "bar" + return from node1
test/1	"foo" + return from node4
test/1/foo	return from node6
test/1/bar	return from node6
test/foo	"foo" + return from node4
test/bar	"foo" + "bar"+ return from node7
test/bar/bar	return from node7
test/bar/foo	return from node7
test/bar/bar/foo	return from node7
test/bar/foo/foo	return from node7
test/foo/foo	"bar" + return from node6
test/foo/foo/bar	return from node6
test/foo/foo/foo	return from node6
test/1/bar	return from provider6
test/1/bar/1	return from provider6

10.8.7 Node methods

A node has to implement the following methods:

Table 22: Node methods

Method	Description	Side effects	Idempotent	Input data	Output data
onCreate	Creates an object	Yes	No	Yes	Yes
onRemove	Deletes an object	Yes	Yes	No	No
onBrowse	Returns the child node	No	Yes	No	Yes
onRead	Reads the value of a node	No	Yes	Yes	Yes
onWrite	Writes the value of a node	Yes	Yes	Yes	Yes
onMetadata	Returns metadata of a node	No	Yes	No	No

Side effect:

Indicates whether the method call may change the node state. When values are read (onRead, onMeatdata, onBrowse), the node state may not change.

Idempotent:

Specifies whether two equal subsequent calls cause the same state in the node.

Input data:

Indicates whether the method has input data.

Output data:

Indicates whether the method has output data.

The methods are provided with the following transfer parameters:

Table 23: Methods and their transfer parameters

Parameter	Type	Description
address	std::string	Requested address This address can (especially for nodes with wildcard login addresses) specify the exact address requested
data	Variant	Specifies input data for the method (only for a method with input data)
Callback	ResponseCallback	Callback Has to be called when the result of the request is present

10.8.8 Response of a provider

To respond to a request, a provider starts a callback. The response can be synchronous or asynchronous.

A return status (type: DLResult) is used for the callback and the data (type: Variant).

comm::datalayer::DLResult::DL_OK has to be used for positive responses. The "Variant" includes the result of the request. If there is no result, nullptr can be returned.

10.8.9 Returning errors

The callback is used as follows for incorrect requests.

An error code is returned as first transfer parameter.

The following values are available for error codes:

Define	Usage
DL_INVALID_ADDRESS	The specified address does not exist
DL_UNSUPPORTED	The method is not supported for the specified address
DL_OUT_OF_MEMORY	Insufficient memory or resource not available (RAM, handling, hard disk memory...)
DL_LIMIT_MIN	The value of the request is too small
DL_LIMIT_MAX	The value of the request is too big
DL_TYPE_MISMATCH	The value type is incorrect
DL_SIZE_MISMATCH	The value size is incorrect
DL_INVALID_FLOATING-POINT	Invalid floating point number
DL_INVALID_HANDLE	Invalid handle or null pointer
DL_INVALID_OPERATION_MODE	Cannot be accessed due to an incorrect state (writing not possible)
DL_INVALID_CONFIGURATION	Value in conflict with other configured values
DL_INVALID_VALUE	Invalid value
DL_SUB-MODULE_FAILURE	Error in child module
DL_TIMEOUT	Timeout upon access
DL_ALREADY_EXISTS	Create: Resource already exists
DL_CREATION_FAILED	Error upon creation
DL_VERSION_MISMATCH	Version conflict
DL_DEPRECATED	The method is not supported anymore

Define	Usage
DL_PERMIS- SION_DENIED	Access not allowed
DL_NOT_INITIALIZED	Object not yet initialized

To return detailed diagnostics, return a variant with a FlatBuffer as second transfer parameter. The FlatBuffer has to be of the type “diagnosis”.

FlatBuffer structure:

```
namespace comm.datalayer;

table Diagnosis
{
    mainDiagnosisCode : uint32 = 0;           // Main diagnosis code for diagnosis system.
    E.g.: 0x3F0A0901
    detailedDiagnosisCode : uint32 = 0;       // Detailed diagnosis code for diagnosis
    system. E.g.: 0xC00E2050

    dynamicDescription: string;               // A human-readable explanation with even more
    detailed information specific to this occurrence of the problem.
}

root_type Diagnosis;
```

A main diagnostics, a detailed diagnostics and a description can be specified.

10.8.10 Synchronous response

If there is a synchronous response, the prompt result is determined synchronously to the method call. The callback call thus takes place while calling the on-method. Use synchronous responses when there is a direct result.

10.8.11 Asynchronous response

It is often not possible to respond directly to a request. Data has to be obtained first or the operation requires more time. To allow new requests, the on-method may not be blocked. Thus, an asynchronous response of the callback is not called during the on-method. The callback is buffered, data provision or operation requested and the on-method exited. The callback is called either when data is available or when the operation is completed.

10.9 Exchanging data RT

10.9.1 Introduction and overview

In case of NRT access to data, a request is sent for each access. This is not suitable for a cyclic access.

There is a specific access method in the ctrlX Data Layer.

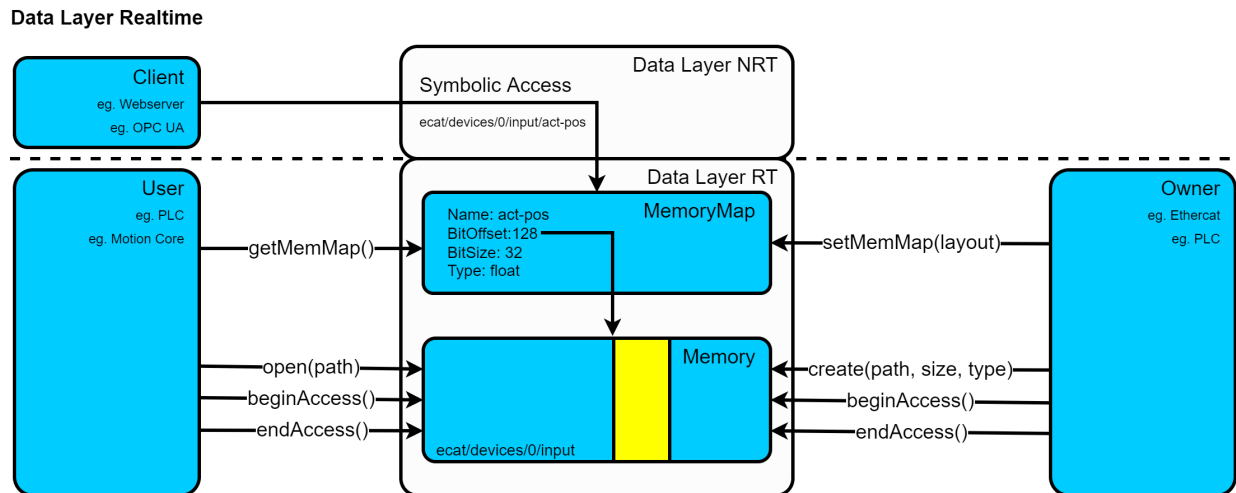


Fig. 59: ctrlX_CORE_MANUAL_Data_Layer_data_exchange_RT_overview

Data is exchanged via memory ranges. Owners define these memory ranges. Owners define the access address (path), the size and the type of memory range depending on the application area. Owners also define the internal structure of the memory range. Users can access these memory ranges. Writing to and reading this commonly used memory can exchange real-time-capable data.

Properties of the ctrlX Data Layer RT:

- Memory ranges can be created and deleted at runtime. User and owner can be initialized individually. Owners and users are connected automatically at runtime if they are available.
- The internal structure of a memory range can change at runtime. Users are informed on changes. If a memory range changes, it is ensured that users always access the correct revision of the memory (see revision number).
- NRT mechanisms can be used to access variables in the memory ranges using the ctrlX Data Layer.
- Depending on the type of memory range, consistent access to multiple variables is ensured.
- Accessing the memory ranges can be restricted for users and clients due to security reasons.

10.9.2 Types of memory ranges

To exchange data, the following memory range types are available. These types are provided with different properties to cover different areas of application.

Type	Properties	Use case
Input	• Data flow from owner to user • An owner produces data • Many users consume data • Access from different processes • Consistency during access • Unblocking access for owners • Access between users: - triple buffer: blocking - nth buffer: non-blocking for n-2 user	Input mapping of a field bus

Type	Properties	Use case
Output	• Data flow from user to owner • An owner consumes data • Many users produce data • Access from different processes • Consistency during access • Unblocking access for owners • Blocking access between users	Output mapping of a field bus
Shared	• No direct data flow • Simultaneous access to the memory • Access from different processes • Direct access to memory	Common status between components
SharedRe- tain	• Memory remains after control reboot • No direct data flow • Simultaneous access to the memory • Access from different processes • Direct access to memory	Retain memory of the PLC
TriggerInput	• Data is organized in a queue - Specifying the size of the queue - blocking/non-blocking • Owner can place data in the queue • If endAccess from owner, user is informed • Communication across process boundaries	Producing camera data
Trigger- Output	• Data is organized in a queue - Specifying the size of the queue - blocking/non-blocking • User can place data in the queue • If endAccess from the user, the owner is informed • Communication across process boundaries	External triggering

10.9.3 Structure of a memory range

The owner specifies the organization of data in a memory range.
MemoryMap is used for the definition.

```
include "metadata.fbs";
namespace comm.datalayer;

table Variable {
  name : string (key);      // Name of the variable
  bitoffset : uint;         // Offset (in bits) of variable in memory
  bitsize : uint;          // Size (in bits) of variable in memory
  type : string;            // type information
  metadata : Metadata;      // Metadata for the variable
}

table MemoryMap {
  variables : [Variable];
  revision : uint = 0;      // changes every time on variables changes
}

root_type MemoryMap;
```

The owner defines the variables in the memory range.

The variable definition includes:

- **name:**
Variable name in address format. This name and the address of the memory range create together an address used to access this variable using NRT mechanisms.
- **Bitoffset:**
Offset of the variable in the memory (counted from the start of the memory range) in bits
- **Bitsize:**
Variable size in bits
- **Type:**
Variable type as string
- **Metadata:**
Optional: Metadata of the variable. If no metadatum is specified, it is generated automatically.

The following variable types are available:

Table 24: Variable types

Define	Type string
TYPE_DL_BOOL8	types/datalayer/bool8
TYPE_DL_INT8	types/datalayer / int8
TYPE_DL_UINT8	types/datalayer/uint8
TYPE_DL_INT16	types/datalayer/int16
TYPE_DL_UINT16	types/datalayer/uint16
TYPE_DL_INT32	types/datalayer/int32
TYPE_DL_UINT32	types/datalayer/uint32
TYPE_DL_INT64	types/datalayer/int64
TYPE_DL_UINT64	types/datalayer/uint64
TYPE_DL_FLOAT32	types/datalayer/float32
TYPE_DL_FLOAT64	types/datalayer/float64
TYPE_DL_STRING	types/datalayer/string
TYPE_DL_ARRAY_OF_BOOL8	types/datalayer/array-of-bool8
TYPE_DL_ARRAY_OF_INT8	types/datalayer/array-of-int8
TYPE_DL_ARRAY_OF_UINT8	types/datalayer/array-of-uint8
TYPE_DL_ARRAY_OF_INT16	types/datalayer/array-of-int16
TYPE_DL_ARRAY_OF_UINT16	types/datalayer/array-of-uint16
TYPE_DL_ARRAY_OF_INT32	types/datalayer/array-of-int32
TYPE_DL_ARRAY_OF_UINT32	types/datalayer/array-of-uint32
TYPE_DL_ARRAY_OF_INT64	types/datalayer/array-of-int64
TYPE_DL_ARRAY_OF_UINT64	types/datalayer/array-of-uint64
TYPE_DL_ARRAY_OF_FLOAT32	types/datalayer/array-of-float32
TYPE_DL_ARRAY_OF_FLOAT64	types/datalayer/array-of-float64
TYPE_DL_ARRAY_OF_STRING	types/datalayer/array-of-string
TYPE_PLC_BIT	types/plc/bit
TYPE_PLC_BOOL	types/plc/bool
TYPE_PLC_BYTE	types/plc/byte
TYPE_PLC_WORD	types/plc/word
TYPE_PLC_DWORD	types/plc/dword

Define	Type string
TYPE_PLC_LWORD	types/plc/lword
TYPE_PLC_SINT	types/plc/sint
TYPE_PLC_USINT	types/plc/usint
TYPE_PLC_INT	types/plc/int
TYPE_PLC_UINT	types/plc/uint
TYPE_PLC_DINT	types/plc/dint
TYPE_PLC_UDINT	types/plc/udint
TYPE_PLC_LINT	types/plc/lint
TYPE_PLC_ULINT	types/plc/ulint
TYPE_PLC_REAL	types/plc/real
TYPE_PLC_LREAL	types/plc/lreal
TYPE_PLC_STRING	types/plc/string
TYPE_PLC_WSTRING	types/plc/wstring
TYPE_PLC_TIME	types/plc/time
TYPE_PLC_TIME_OF_DAY	types/plc/time-of-day
TYPE_PLC_DATE	types/plc/date
TYPE_PLC_DATE_AND_TIME	types/plc/date-and-time
TYPE_PLC_LTIME	types/plc/ltime

Metadata of an RT variable

Metadata of an RT variable can come from different sources.

The order used for the metadata sources is listed here:

- Metadata from the MemoryMap
- Metadata from the MDDB (search with the complete address path)
- Metadata from the MDDB (search with the variable name)
- Generic metadata

10.9.4 Revision of a memory range

The MemoryMap includes the revision of the memory range in addition to the variables. The revision indicates the definition release of the memory range.

If a memory range is accessed cyclically, the owner writes the specified revision number to the memory.

When a user accesses a memory range, the specified revision number is checked against the revision number in the memory range. If these revision numbers do not match, access is denied. Thus, data is not misinterpreted.

10.9.5 Owner interface

```
DIResult createMemorySync(std::shared_ptr<IMemoryOwner>& owner, const std::string &addr,
    IProvider* provider, size_t sizeBytes, MemoryType type)
```

```
DIResult createMemoryAsync(const std::string &addr, IProvider* provider, size_t sizeBytes,
    MemoryType type, ResponseCallbackOwner &callback)
```

Creates a memory range with an address, a size (in bytes) and a type. A provider has to be specified to respond to this memory range.

Methods can then be applied via an analogously created owner:

Table 25: Interfaces of the owner - Methods

Method	Description
DIResult getSize(size_t& sizeBytes)	Returns the size of the memory range
DIResult getType(MemoryType& type)	Returns the type of the memory range

Method	Description
DIResult setMemoryMap(const Variant &memoryMap)	Sets a MemoryMap for the memory range
DIResult getMemoryMap(Variant &memMap)	Returns the current MemoryMap for this memory range
DIResult beginAccess(uint8_t*& data, uint32_t revision)	Starts accessing a memory range
DIResult endAccess()	Ends accessing a memory range

10.9.6 User interface

```
DIResult openMemory(std::shared_ptr<IMemoryUser>& user, const std::string &addr)
```

The method openMemory creates a user. The address of the memory range of the owner has to be specified. The memory range does not have to exist right now.

The user has the following access methods:

Table 26: User interface - Access methods

Method	Description
DIResult getSize(size_t& sizeBytes)	Returns the size of the memory range
DIResult getType(MemoryType& type)	Returns the type of the memory range
DIResult getMemoryMap(Variant &memMap)	Returns the current MemoryMap for this memory range
DIResult beginAccess(uint8_t*& data, uint32_t revision)	Starts accessing a memory range
DIResult endAccess()	Ends accessing a memory range

10.9.7 State machine of the user

The state machine is responsible that a user connects to the owner and provides all information on the memory range.

The new MemoryMap is maintained when changing the revision number. If the complete memory cannot be accessed anymore, it is tried to connect again to the server.

The state machine is executed in the background.

All accesses to NRT nodes in the state machine are executed asynchronously. If required, the state machine changes its state by responding or calling user methods (beginAccess, endAccess, getMemoryMap, getSize, getType).

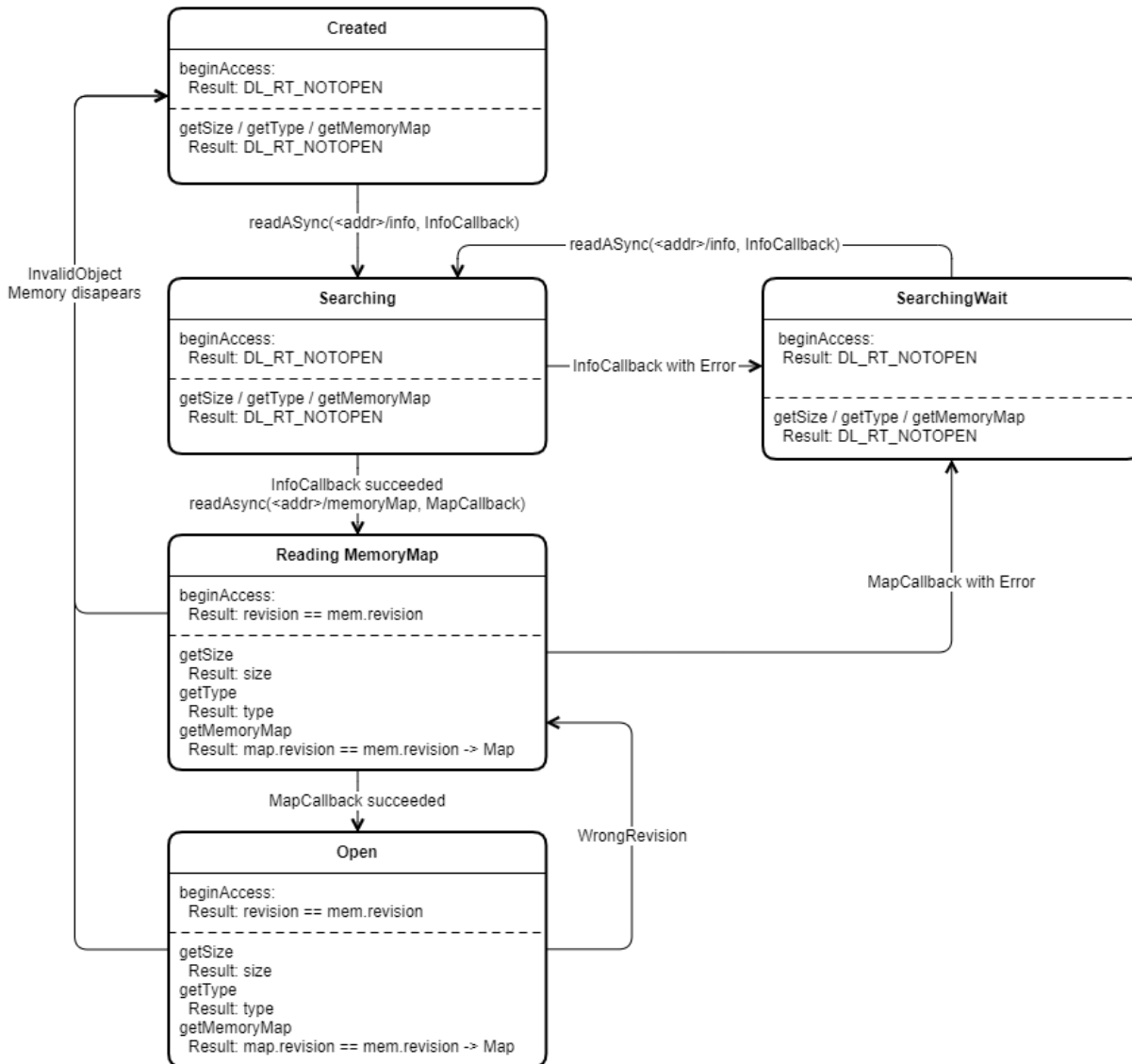


Fig. 60: State machine of the user

10.9.8 NRT access to RT data

If an owner is created, an NRT access to this data is automatically possible. The following data is published in the ctrlX Data Layer:

Address	Data	Type
<Address RT>/access/read	Consistent NRT read access to RT data	<i>types/datalayer/rt-data</i>
<Address RT>/access/write	Consistent NRT write access to RT data	<i>types/datalayer/rt-data</i>
<Address RT>/info	Information on the memory (type, size, ID)	<i>types/datalayer/memory</i>
<Address RT>/map	MemoryMap of the memory	<i>types/datalayer/memory_map</i>
<Address RT>/data	Memory content	ARRAY_OF_UINT8
<Address RT>/data/<Address variable>	Access to individual variables of the memory	Variable type

Consistent NRT access to RT data

To access RT data consistently via NRT access, the *access/read* and *access/write* nodes are available under each RT address. The *access/write* node is only possible if the RT range allows write access.

The following values are required for access:

- revisionNumber:
Current revision number of the memory range from the MemoryMap. If the memory structure changes in the meantime, the revision number also changes.
- Array of data:
 - bitoffset:
Start bit of the data range to be accessed
 - bitsize:
Length of the data range to be accessed
 - data:
Data to be written

This data can be used to perform a read or write operation. As a result, a structure with the same design is returned.

10.9.9 Special features of the memory type SharedRetain

Introduction

SharedRetain memory is remanent: It remains after a control restart.

The remanence on the ctrlX CORE is achieved due to the hardware support. The available memory size depends on the hardware:

- ctrlX CORE X3: 128 kB retain memory
8 kB of this memory are used for internal purposes. If the control is used with a PLC, 64 kB of this memory is used for PLC retain data by default.
- ctrlX CORE X7: 2048 kB retain memory
8 kB of this memory are used for internal purposes. If the control is used with a PLC, 64 kB of this memory is used for PLC retain data by default.
- ctrlX CORE Virtual:
The retain memory is emulated via a Memory Mapped File. The size can be set via the Data Layer node *datalayer/server/settings/serverEmulatedNvramSize*.

Remanence of retain memory

Save retain memory cyclically to be available again after a control restart.

There are different options:

- Via the Scheduler:
The "persist-nvram" callable can be used to trigger the saving of the retain memory. By default, this callable is in the low-priority task "schedBackground".
- Via a Data Layer node:
Via a Data Layer node *datalayer/retain/sync* can be used to trigger the saving of the retain memory.

Forcing the emulation of retain memory

The size of physical NVRAM memory is limited. If more retain memory is required, the emulation of retain memory can be forced and the size can be specified.

Thus, make the following settings:

- *datalayer/server/settings/forceEmulatedNvram*: TRUE
- *datalayer/server/settings/serverEmulatedNvramSize*: Size of desired retain memory

Emulated retain memory is mapped as Memory Mapped File. Synchronization between memory and file takes place on command. However, this command is only accepted and executed every 10 seconds to increase the service life of the internal flash memory.

Thus, proceed as follows when using emulated memory:

- Use a UPS
- Remove the callable call persist-nvram from the background task
- Call the *datalayer/retain/sync* node via Rest, OPC UA or the PLC when the UPS detects a power failure

Reserving memory in the retain area

Memory can be reserved by the system via a memory owner. An address is transferred when creating an owner. This address is used as key when searching in the retain memory.

There are different search cases:

Case	Behavior
Address not found	Memory is created and emptied with the specified size
Address found (specified size is smaller than or equal to the memory range found)	The memory range is used
Address found (specified size is greater than or equal to the memory range found)	The old range is deleted and a new range is created and emptied

The revision of the RetainShared memory is also remanent. An owner can thus use the revision number to detect an older version between the restarts and to convert it into a new format if required. An owner can open a *getMemoryMap*. This method returns a *MemoryMap* with the current revision number set.

For *SharedMemory* ranges, go to the node *datalayer/retain/chunks* in the ctrlX Data Layer. The login name of the memory is used as subpath. The memory of this node can be accessed on this node.

If the *SharedMemory* is emulated, go to *datalayer/retain/emu* for the complete emulated memory. As it is an RT memory, the data is of an RT memory.

10.9.10 RT MemoryType trigger

Creating a trigger

The *MemoryType* trigger can be used to create a queue between process spaces and to inform another process about events. The length of the queue can be specified.

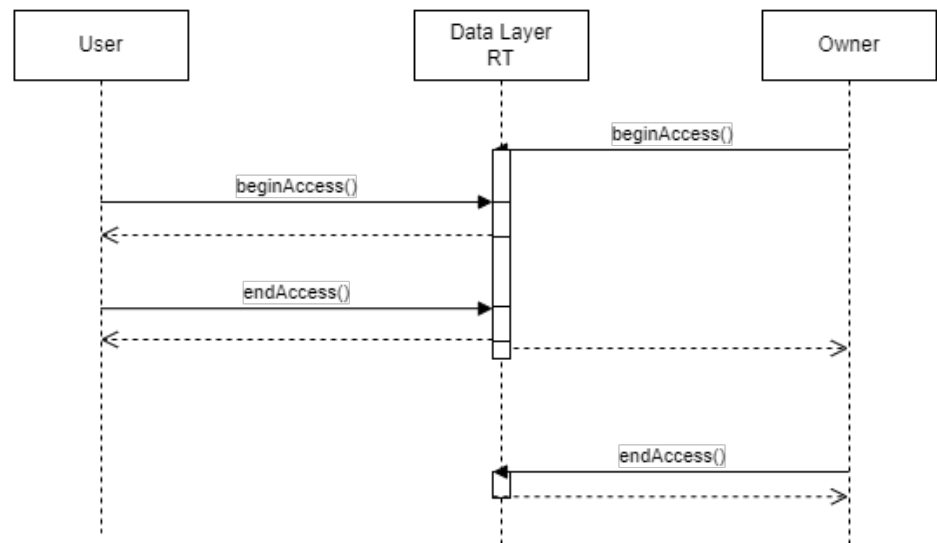
```
flatbuffers::FlatBufferBuilder builder;
uint32_t queueSize = 5;
builder.Finish(comm::datalayer::CreateTriggerOptions(builder, queueSize));
comm::datalayer::Variant options;
options.copyFlatbuffers(builder);
m_datalayer->factory()->createMemorySync(owner, address, provider, 4, type, options);
```

Temporal behavior

If the queue is empty and it is tried to retrieve an element from the queue, *beginAccess()* blocks until an element is available in the queue.

If the queue is full and it is tried to place an element in the queue, beginAccess() blocks until there is space in the queue again.

Trigger Output



Trigger Input

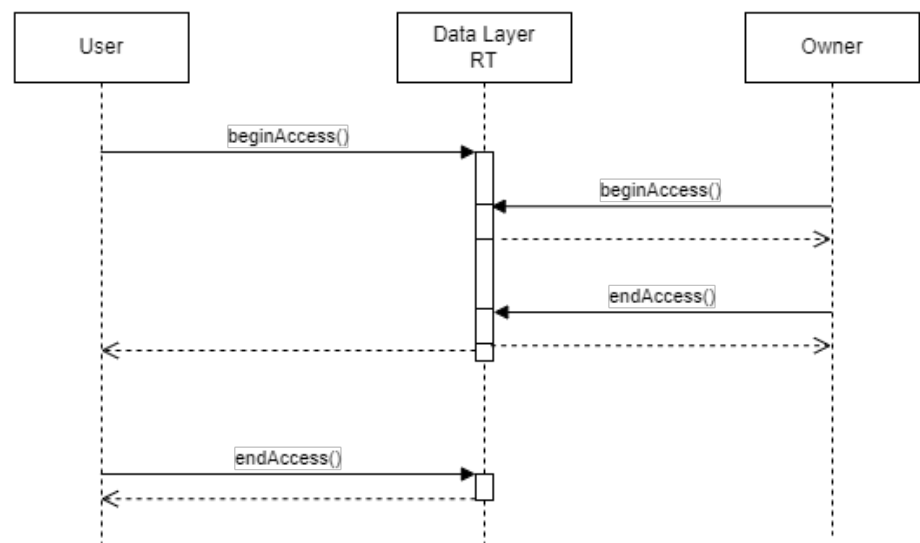


Fig. 61: Temporal behavior

Non-blocking access

To prevent blocking access to the queue, proceed as follows:

```

uint8_t* data;
data = DL_RT_NON_BLOCKING;
result = user->beginAccess(data, revision);
if (result == comm::datalayer::DLResult::DL_RT_WOULD_BLOCK)
{
    // access would block
}

```

```
if (result == comm::datalayer::DIResult::DL_OK)
{
    // access ok
}
user->endAccess();
```

Revision check

To check only the revision of the memory range, proceed as follows:

```
uint8_t* data;
data = DL_RT_CHECK_REVISION;
result = user->beginAccess(data, revision);
if (result == comm::datalayer::DIResult::DL_OK)
{
    // revision is right
} else {
    // revision is wrong
}
user->endAccess();
```

10.10 NRT data exchange between controls

10.10.1 Introduction and overview

The Data Layer of a control can be shown in the Data Layer of another control. This can be used to manage all controls of a system with multiple controls in one control without setting up a direct communication channel to these controls.

There are two options to integrate one Data Layer into another:

	Remote Connection	Reverse Connection
It is used:	Client	Provider
Establishing a connection:	Connection to the Data Layer to be integrated	Connection to the Data Layer that should integrate the own Data Layer
Node address of the configuration:	<i>devices/remotes</i>	<i>datalayer/reverse-connections</i>
Works through NAT:	no	yes
Data Layer integration:	The Data Layer of the other control is integrated into the ownData Layer	The own Data Layer is integrated into the Data Layer of the other control

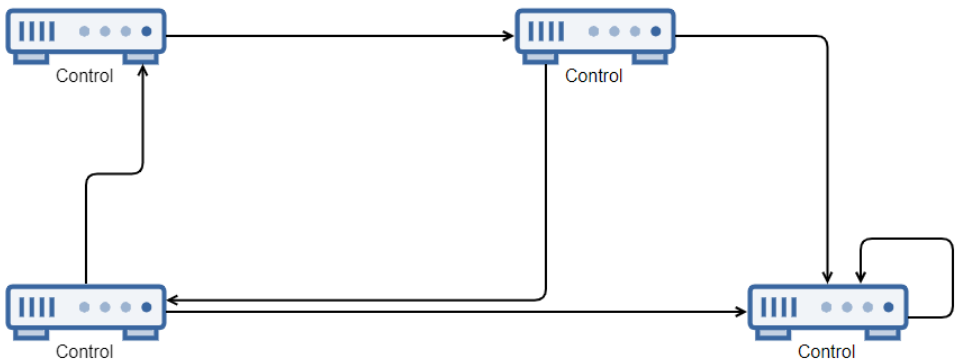


Fig. 62: Overview on the NRT data exchange between controls

10.10.2 Configuration

Creating a Remote Connection

For the configuration, connections to controls to be shown are configured in a control.

Send a "create" request to the *devices/remote* node with the following content:

```
{
  "name" : "otherControl",
  "address" : "tcp://<user>:<pw>@192.168.1.1:2069"
}
```

Feld	Behavior
name	Connection name Access the node of the control via <i>devices/remote/<name>/...</i>
address	Address of the other control (see Chapter 10.6 Data transport NRT on page 200)

Creating a Reverse Connection

Reverse Connection is currently an experimental function. Incompatible changes may therefore be made to this function. To enable this experimental function, set "reverse-connections" in the *datalayer/server/settings* node.

```
{
  ...
  "experimentalFeatures": [
    "reverse-connections"
  ]
}
```

For the configuration, connections to the controls - in which your own control is to be displayed - are configured in a control.

Send a "create" request to the *datalayer/reverse-connections* node with the following content:

```
{
  "connectionId": "foo",
  "connectionAddress": "tcp://<user>:<pw>@192.168.1.1",
  "name": "bar"
}
```

Feld	Behavior
connectionID	Connection ID This ID is shown below <i>datalayer/remote-connections</i>
connectionAddress	Address of the other control (see Chapter 10.6 Data transport NRT on page 200)
name	Connection name The nodes of the control can be accessed on the other control via <i>devices/remote/<name>/...</i>

10.10.3 Access to data of another control

Use the child nodes of *devices/remote/<name>* to access NRT values of the other control. All requests are forwarded to the configured control. The configuration for Remote Connection can be read out, changed or deleted via the node *devices/remote/<name>*.

10.11 Data Layer - Global settings

To make global settings for the Data Layer, go to the *datalayer/server/settings* node.

Setting	Description	Default setting
serverIdlePingTimeout	Time in milliseconds in which the system checks whether a provider is still reachable if there was no communication with this provider in the meantime	30000 ms
serverWaitResponseTimeout	Time in milliseconds that is waited for a response from a provider	30000 ms
serverMaxMessageSize	Maximum size of a Data Layer message in bytes	52428800 bytes
serverMaxRtSize	Maximum size of an RT range in bytes	1048576 bytes
serverEmulatedNvramSize	Size of the NVRAM in bytes if this is emulated	122880 bytes
experimentalFeatures	Enables experimental features from Data Layer	
tcpPortFrontend	Data Layer TCP port for Data Layer clients 0 switches off the port completely. The change is only applied after a control restart.	2069
tcpPortBackend	Data Layer TCP port for Data Layer provider 0 switches off the port completely. The change is only applied after a control restart.	2070
forceEmulatedNvram	Forces NVRAM emulation. For more on this topic, see Chapter 10.9.9 Special features of the memory type SharedRetain on page 215	FALSE
serverZmqHighWaterMark	High Water Mark for ZMQ	1000

To make settings for the Data Layer web server, go to the *datalayer/server/websettings* node.

Setting	Description	Default setting
timeout	Timeout for a request to the Data Layer	3000 ms
maxJsonInput	Maximum size for a JSON request in bytes	1048576 bytes
maxFlatbufferToJson	Maximum size of Data Layer responses converted to JSON	1048576 bytes

To make settings for the Data Layer Real Time, go to the *datalayer/real-time/broker/settings* node

Setting	Description	Default setting
defaultInputAccessType	Buffer type used for Real Time Input: - NTelBuf: n-fold buffer system that allows simultaneous access by n-2 users - TripleBuffer: Triple buffer system that allows simultaneous access by one user	NTelBuf
nTelBufConfig	N for the n-fold buffer system	"defaultN": 6

10.12 ctrlX Data Layer Rest interface

10.12.1 Introduction and overview

To open the complete functionality of the ctrlX Data Layer, use HTTPS (Hypertext Transfer Protocol Secure).

Compared to REST API V1, the following points change in REST API V2:

- The web server supports the Content Negotiation
- The Data Layer with parameters is read via HTTP-GET requests using a query parameter
- The REST-API V2 supports generic HTTP requests to Data Layernodes and events sent by the server via Data Layer subscriptions

10.12.2 Protocol mapping

Addressing via Node address

Standard URI to access to the REST API version V2:

<https://<hostname>/automation/api/v2/nodes/<address of node>>

Redirecting HTTP requests to the ctrlX Data Layer

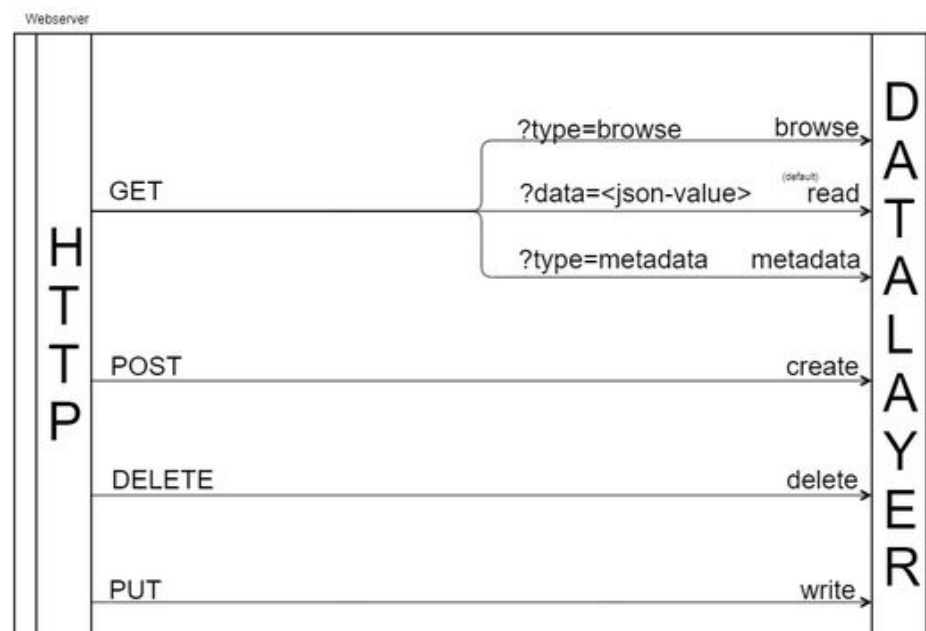


Fig. 63: Overview - Redirecting HTTP requests to the ctrlX Data Layer

The following table includes more detailed information on request and response data.

Data Layer method	HTTP request				HTTP response	
Method	Method	Path	User data	Note	HTTP status code if successful	User data in case of success
read	GET	<i><node address></i>	none		200	Returns the node value
	GET	<i><node-URL>?data=<json-value></i>	none	The request parameter data <i><json-value></i> is interpreted as JSON string considered as argument of the node. For a method, <i><json-value></i> can for example represent the input parameter of the mentioned method.	200	Returns the node value
write	PUT	<i><node address></i>			200	Returns written data
create	POST	<i><node address></i>			200 or 201	Returns a response including information on the created resource
delete	DELETE	<i><node address></i>	none		200 or 204	none
browse	GET	<i><node address></i>	none		200	Returns the subnodes of the node
metadata	GET	<i><node address></i>	none		200	Returns meta data of the node

10.12.3 Access rights

Range	Permitted HTTP method	Access rights
rexroth-device.all.rwx	GET PUT DELETE POST	read,write,execute
rexroth-automation.datalayer.r	GET	read
rexroth-automation.datalayer.rw	GET PUT	read,write
rexroth-automation.datalayer.x	POST DELETE	execute

10.12.4 Content mapping

The web server can interpret transferred data and convert them into different media types

The “Content-Type-Header“ and the “Accept-Header“ are interpreted in the HTTP request and transferred data is converted accordingly.



The error response of a request is always shown as media type *application/json*.

For detailed information on the error response, refer to [Chapter 10.12.5 Troubleshooting on page 223](#).

The table shows the respective supported media types and their standard conversion schemes.

Table 27: Supported media types and their standard conversion schemes

Request Content type	Request Accept type	Data Response Content type	Error Response Content type	Note
<i>application/json</i>	<i>application/json</i>	<i>application/json</i>	<i>application/json</i>	Default
<i>application/json</i>	<i>application/octet-stream</i>	<i>application/octet-stream</i>	<i>application/json</i>	
<i>application/json</i>	<i>text/plain</i>	<i>text/plain</i>	<i>application/json</i>	
<i>application/json</i>	<i>*/*</i>	<i>application/json</i>	<i>application/json</i>	
<i>application/json</i>	every other type	<i>application/json</i>	<i>application/json</i>	
<i>application/octet-stream</i>	<i>application/json</i>	<i>application/json</i>	<i>application/json</i>	
<i>application/octet-stream</i>	<i>application/octet-stream</i>	<i>application/octet-stream</i>	<i>application/json</i>	Default
<i>application/octet-stream</i>	<i>text/plain</i>	<i>text/plain</i>	<i>application/json</i>	
<i>application/octet-stream</i>	<i>*/*</i>	<i>application/octet-stream</i>	<i>application/json</i>	
<i>application/octet-stream</i>	every other type	<i>application/octet-stream</i>	<i>application/json</i>	
<i>text/plain</i>	<i>application/json</i>	<i>application/json</i>	<i>application/json</i>	
<i>text/plain</i>	<i>application/octet-stream</i>	<i>application/octet-stream</i>	<i>application/json</i>	
<i>text/plain</i>	<i>text/plain</i>	<i>text/plain</i>	<i>application/json</i>	Default
<i>text/plain</i>	<i>*/*</i>	<i>text/plain</i>	<i>application/json</i>	
<i>text/plain</i>	every other type	<i>text/plain</i>	<i>application/json</i>	

10.12.5 Troubleshooting

In every error case, the web server provides an HTTP error status code (e.g. 404) and additional information on the background. The error is always coded as media type *application/json* and shown as JSON object as defined in RFC3986.

The FlatBuffer schema (problem.fbs) defines the error response time with the following format:

```
namespace comm.datalayer;

enum Severity : byte {
  Emergency,
  Alert,
  Critical,
  Error,
  Warning,
  Notice,
  Informational,
  Debug
}

table Problem
{
  type : string (required);           // A URI reference [RFC3986] that identifies the problem type.
                                     // This specification encourages that, when dereferenced,
```

```

// it provide human-readable documentation for the problem type
// (e.g., using HTML [W3C.REC-html5-20141028]). When this member
// is not present, its value is assumed to be "about:blank".
title : string (required); // A short, human-readable summary of the problem type.
// It SHOULD NOT change from occurrence to occurrence of the problem,
// except for purposes of localization (e.g., using proactive content
// negotiation; see [RFC7231], Section 3.4).
// The HTTP status code ([RFC7231], Section 6) generated by the origin server
status : int32;
for this occurrence of the problem.
// This will be added by the webserver - so you don't have to fill this field.
detail : string; // A human-readable explanation specific to this occurrence of the problem.
instance : string; // A URI reference that identifies the specific occurrence of the problem. It
may or may not yield further information if dereferenced.
mainDiagnosisCode : string; // The main diagnosis code was/is issued by the system that caused the problem.
// This code can be used to track down the root cause and source of the error.
// It can be used to search in the documentation for a solution.
// It SHOULD NOT change from occurrence to occurrence of the same problem.
// example: 080F0100
detailedDiagnosisCode : string; // The detailed diagnosis code issued by the system that caused the problem.
// This code can be used to track down the detailed cause and source of the
error.
// It can be used to search in the documentation for a solution.
// The detailed diagnosis code gives additional information about the cause of
the error.
// It SHOULD NOT change from occurrence to occurrence of the same problem.
// example: 00666001
dynamicDescription : string; // A dynamic description gives detailed information about the occurrence of a
problem.
// It can change between different occurrences of the same error.
// example: value = -4.5
severity : Severity; // Severity of a problem as defined RFC5424 of the Syslog standard, see https://
tools.ietf.org/html/rfc5424
links : [string]; // Collection of links to fix the problem. E.g. links to online user manual, to
an online community (using tags) or a developer chat (e.g. Slack).
}

root_type Problem;

```

Example:

```

{
  title: 'Error on Read',
  type: 'about:blank',
  status: 404,
  detail: 'Your current balance is 30, but that costs 50.',
  instance: '/account/12345/msgs/abc',
  mainDiagnosisCode: 'F0360001',
  detailedDiagnosisCode: '00666001',
  dynamicDescription : 'This could be a dynamic description',
  severity: 'ERROR'
}

```

10.13 Subscriptions

10.13.1 Introduction and overview

In contrast to a permanent information retrieval (polling), the Data Layer provides a far better option - a so-called subscription. A Data Layer client can subscribe a selection of node and it can let them being monitored. The client is only notified in case of changes, e.g. change in values. This mechanism largely reduces the amount of transferred data. Apart from reducing the required band width, this mechanism has more advantages. It is also recommended when “reading” information if information is required cyclically.

A subscription consists of a subscription ID and a rule set. A rule set describes the subscription settings. Data Layer nodes can be added to and removed from a subscription.

Subscription Methods

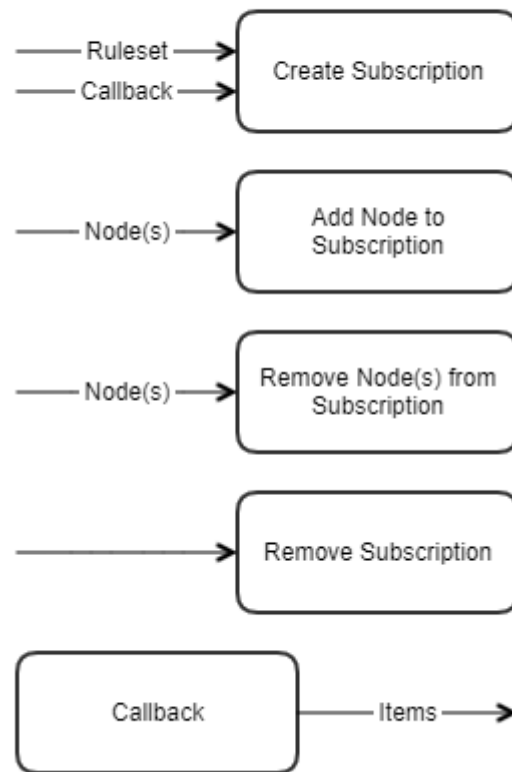


Fig. 64: Methods of the subscriptions

Use a rule set and a callback to create a subscription. The rule set describes the subscription settings. A callback opens when new data is available for logged-in nodes.

Nodes can be added to or removed from a subscription. If a subscription is not required anymore, it can be deleted.

10.13.2 Subscription - Rule set

A subscription is provided with settings specifying the behavior. These settings are called rule set.

Overview

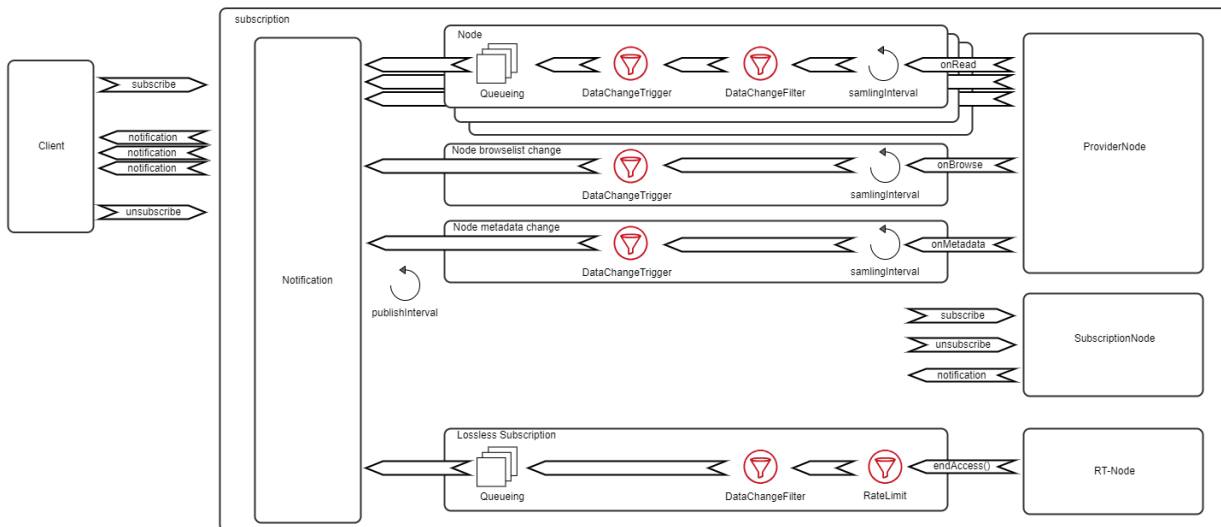


Fig. 65: Subscription overview

ID

- **Description:** Identification name of the subscription

SamplingInterval

- **Description:** The SamplingInterval defines the rate the system uses to check for changes in the data source. The value is only detected if it changes. The SamplingInterval can be set far faster than the client notification. In this case, the server can queue the samples and transfer the complete queue at once. To protect the system against high load from the sampling, configure the minimum SamplingInterval in the global settings.
- **Default setting:** 1000 ms

PublishInterval

- **Description:** Interval during which a notification is sent to the user. A notification includes all newly captured values. If no new data is captured, no notification is sent.
- **Default setting:** 1000 ms

DataChangeFilter (optional)

- **Description:** The DataChangeFilter describes when a new value is to be captured. Only if the difference of the latest captured value is higher than the current value of the DataChangeFilter, the value is captured.
- **DeadBandValue:** If $(\text{abs}(\text{lastCapturedValue} - \text{newValue}) > \text{deadBandValue})$ capture(newValue);

Queueing (optional)

- **Description:** If more values are captured than sent, these values can be buffered in a queue. Size and behavior can be defined if the queue is full.
- **queueSize:**
 - **Description:** Queue size
 - **Default setting:** 10

LosslessRateLimit (optional)

- **Description:** Only every nth new file is recorded. This is useful to limit the amount of data transferred. This function is only implemented in Lossless Subscription.

ChangeEvents (optional)

- **DataChangeTrigger (optional):** Filters whether to save a new value if status, value or time stamp change.
- **valueChange:**
 - **Status:** Value is captured when the status changes during reading.
 - **StatusValue (default):** Value is captured when the status changes or the value changes during reading.
 - **StatusValueTimestamp:** value is captured when the status changes, the value changes or the time stamp changes during reading (continuous recording).
- **brwoseListChange:** Informs when the browse list of the node changes.
- **metdataChange:** Informs when meta data of the node changes.

The rule set type for a subscription is `sub_properties.fbs`

```
namespace comm.datalayer;

table Sampling {
    /// sample time in  $\mu$  seconds for minimum sampling of data - currently only multiples of 1000 are supported
    samplingInterval: uint64 = 1000000;
}

enum QueueBehaviour : int { DiscardOldest, DiscardNewest }

table Queueing {
    /// size of buffer
    queueSize: uint = 10;

    /// The queuing strategy. Decided what data should be sent or discarded when the queue runs full before it could
    be sent.
    behaviour: QueueBehaviour = DiscardOldest;
}

table DataChangeFilter {
    /// If (abs(lastCapturedValue - newValue) > deadBandValue) capture(newValue);
    deadBandValue : float;
}

enum DataChangeTrigger : int {
    /// Inform if STATUS changes
    Status,

    /// Inform if Value or Status changed
    StatusValue,

    /// Inform if Status Or Status Or Timestamp changes (inform always)
    StatusValueTimestamp
}

table ChangeEvents {
    valueChange : DataChangeTrigger = StatusValue;
    browselistChange : bool = false;
    metadataChange : bool = false;
}

table Counting {
    /// Nodes are counted if subscribed multiple times. Unsubscribe only accures if counter for this node is 0.
    countSubscriptions : bool = false;
}

table LosslessRateLimit {
    /// Only every n-th new data will be recorded - this is usefull to limit amount of data transmitted
    /// This feature is only implemented in lossless subscriptions
    rateLimit : uint = 0;
}

union Properties { Sampling, Queueing, DataChangeFilter, ChangeEvents, Counting, LosslessRateLimit }
```

```

/// The table Property can be used to set additional optional properties which influence the subscription
behaviour.
/// If no Properties are given, default values will be used automatically.
table Property
{
    rule: Properties;
}

table SubscriptionProperties {
    /// user identification for subscription Use it update and dispose the old subscription
    id : string (required);

    /// keepalive in milli seconds for subscription - this time is used to detect life status of providers
    keepaliveInterval : uint = 60000;

    /// time in milli seconds for upper limit of publish messages to not get flooded with new messages
    publishInterval: uint = 1000;

    /// Rules for subscription
    rules : [Property];

    /// time in milli seconds to re-read data if an error was received for a nodeX
    errorInterval : uint = 10000;
}

root_type SubscriptionProperties;

```

Timing

NRT Subscription

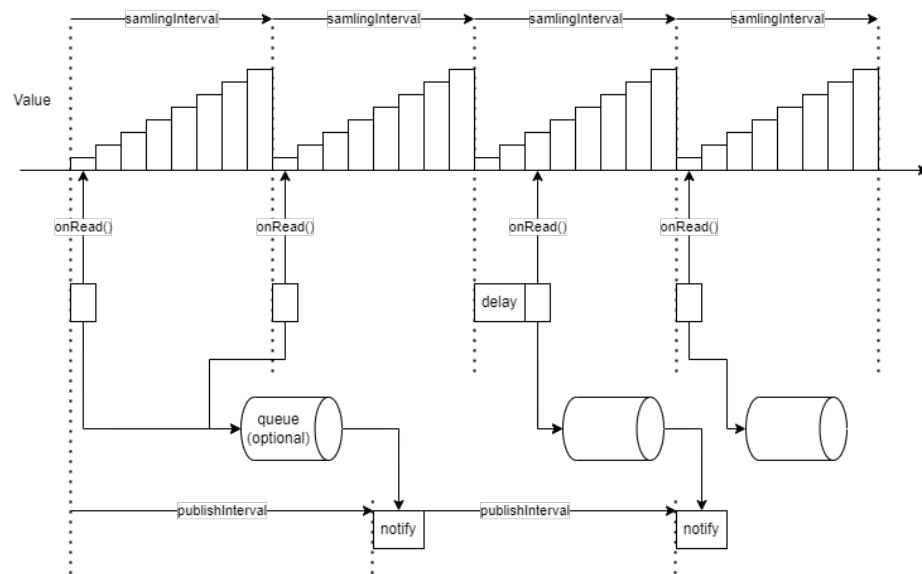


Fig. 66: NRT Subscription

Lossless Subscription

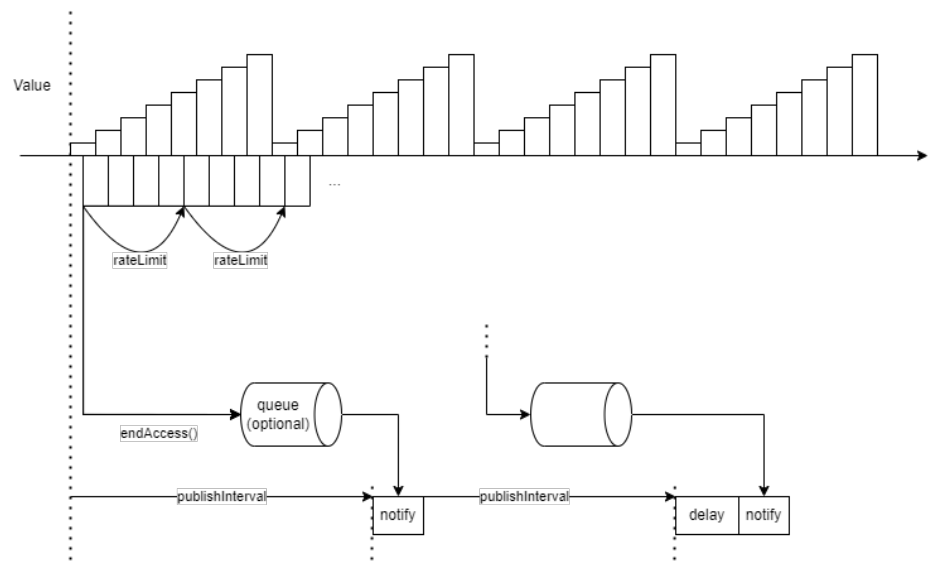


Fig. 67: Lossless Subscription

10.13.3 Subscription - Notifications

Changes are reported as notifications. A notification consists of multiple elements.

Structure of each element

Information	Description
Address	Node address for this element
Value	Element value
Time stamp	Time stamp of the value The time stamp format is "FILETIME". A "FILETIME" is a 64-bit value representing 100 nanoseconds passed since January 1st 1601 00:00 o'clock.

10.13.4 Subscription - Global settings

To make global settings for subscriptions, go to the `datalayer/subscriptions/settings` node.

Setting	Description	Default setting
minimumSampleInterval	If the SamplingInterval of a subscription is smaller than the minimum SamplingInterval, the SamplinInterval is limited to the minimum SamplingInterval	100000 μ s
minimumPublishInterval	If the PublishInterval of a subscription is smaller than the minimum PublishInterval, the PublishInterval is limited to the minimum PublishInterval	100 ms
minimumErrorInterval	If the ErrorInterval of a subscription is smaller than the minimum ErrorInterval, the ErrorInterval is limited to the minimum ErrorInterval	10000 ms
maximumBufferSize	If the queue size of a subscription is greater than the maximum buffer size, the queue size is limited to the maximum buffer size	50

Setting	Description	Default setting
maximumRTSubscribedNodes	Specifies the maximum number of nodes that can be subscribed without loss	50

10.13.5 Active subscriptions

For an overview on the running subscriptions, all subscriptions are tracked and shown under the address *datalayer/subscriptions*.

Address	Description	Type	Type path
<i>datalayer/subscription/clients/<client>/subscription/<subscriptionID></i>	All information on a subscription (rule set and monitored addresses)	FlatBuffers (sub_data.fbs)	<i>types/datalayer/subscription_data</i>
<i>datalayer/subscription/clients/<client>/subscription/<subscriptionID>/nodes</i>	List of all monitored addresses of a subscription	Array of string	
<i>datalayer/subscription/clients/<client>/subscription/properties</i>	Rule set of a subscription	FlatBuffers (sub_properties.fbs)	<i>types/datalayer/subscription_properties</i>
<i>datalayer/subscription/settings</i>	Global settings of subscriptions (global limits)	FlatBuffers (sub_settings.fbs)	<i>types/datalayer/subscription_settings</i>

10.13.6 Subscription - Language binding

Subscriptions can be used with different programming languages and interfaces.

Programming language / interface	Documentation
C/C++, Python, C#	➔ SDK
HTTP (REST)	➔ ctrlX Data Layer Rest Interface
OPC-UA	➔ OPC UA Server App

10.13.7 Subscription - Client examples

Subscriptions can be used with different programming languages and interfaces.

Use case	C++	HTTP REST
User interface (low sampling rate)	<pre> dl::Variant ruleSet; flatbuffers::FlatBufferBuilder builder; builder.Finish(comm::datalayer::CreateSubscriptionPropertiesDirect(builder, "ID1")); ruleSet.shareFlatbuffers(builder); client->createSubscriptionSync(ruleSet, callback(subId)); client->subscribeSync("ID1", std::set<std::string>({ "my/counter", "my/counter2" })); </pre>	GET: /automanagement/api/v2/events?nodes=my/counter,my/counter2

Use case	C++	HTTP REST
User interface ● Informs on value change ● Sampling of 250 ms	<pre> dl::Variant ruleSet; flatbuffers::FlatBufferBuilder builder; auto dataChange = comm::datalayer::CreateSampling(builder, 250000); auto rule = comm::datalayer::CreateProperty(builder, comm::datalayer::Properties::Properties_Sampling, dataChange.Union()); auto rules = builder.CreateVector(&rule, 1); auto id = builder.CreateString("ID1"); builder.Finish(comm::datalayer::CreateSubscriptionProperties(bui lder, id, 60000, 250, rules)); ruleSet.shareFlatbuffers(builder); client->createSubscriptionSync(ruleSet, callback(subId)); client->subscribeSync("ID1", std::set<std::string>{{ "my/ counter", "my/counter2" }}); </pre>	POST: /automation/api/v2/ events <pre> { "properties": { "id": "ID1", "publishInterval": 250, "rules": [{ "rule_type": "Sampling", "rule": { "samplingInterval": 250000 } }], "nodes": ["my/counter", "my/counter2"] } } </pre>
Oscilloscope ● Sampling of 100 ms ● Also recording in case of equal values ● Recording in a queue ● Communication only every second	<pre> flatbuffers::FlatBufferBuilder builder; auto ququeing = comm::datalayer::CreateQueueing(builder,10); auto dataSample = comm::datalayer::CreateSampling(builder,100000); auto changeTrigger = comm::datalayer::CreateChangeEvents(builder, comm::datalayer::DataChangeTrigger_StatusValueTimestamp); std::vector<flatbuffers::Offset<comm::datalayer::Property>> vecRules; vecRules.push_back(comm::datalayer::CreateProperty(builder, comm::datalayer::Properties::Properties_Queueing, ququeing.Union())); vecRules.push_back(comm::datalayer::CreateProperty(builder, comm::datalayer::Properties::Properties_Sampling, dataSample.Union())); vecRules.push_back(comm::datalayer::CreateProperty(builder, comm::datalayer::Properties::Properties_ChangeEvents, changeEvents.Union())); auto rules = builder.CreateVector(vecRules); auto id = builder.CreateString("subscription"); builder.Finish(comm::datalayer::CreateSubscriptionProperties(bui lder, id, 60000, 1000, rules)); ruleSet.shareFlatbuffers(builder); result = client->createSubscriptionSync(ruleSet, callback()); result = client->subscribeSync("subscription", "my/counter"); </pre>	POST: /automation/api/v2/ events <pre> { "properties": { "id": "subscription", "keepaliveInterval": 60000, "publishInterval": 1000, "rules": [{ "rule_type": "Queueing", "rule": { "queueSize": 10 } }, { "rule_type": "Sampling", "rule": { "samplingInterval": 100000 } }, { "rule_type": "ChangeEvents", "rule": { "valueChange": "StatusValueTimestamp" } }], "nodes": ["my/ counter"] } </pre>

Use case	C++	HTTP REST
Lossless Subscription • Sampling Interval 0 • Queueing 50 • Communication every second • RateLimit: only every 20. Transferring value	<pre> flatbuffers::FlatBufferBuilder builder; auto queueing = comm::datalayer::CreateQueueing(builder,50); auto dataSample = comm::datalayer::CreateSampling(builder,0); auto losslessRateLimit = comm::datalayer::CreateLosslessRateLimit(builder,20); std::vector<flatbuffers::Offset<comm::datalayer::Property>> vecRules; vecRules.push_back(comm::datalayer::CreateProperty(builder, comm::datalayer::Properties::Properties_Queueing, queueing.Union())); vecRules.push_back(comm::datalayer::CreateProperty(builder, comm::datalayer::Properties::Properties_Sampling, dataSample.Union())); vecRules.push_back(comm::datalayer::CreateProperty(builder, comm::datalayer::Properties::Properties_LosslessRateLimit, losslessRateLimit.Union())); auto rules = builder.CreateVector(vecRules); auto id = builder.CreateString("subscription"); builder.Finish(comm::datalayer::CreateSubscriptionProperties(bui lder, id, 60000, 1000, rules)); ruleSet.shareFlatbuffers(builder); result = client->createSubscriptionSync(ruleSet, callback()); result = client->subscribeSync("subscription", "scheduler/admin/ info/common-data-rt/data/counter"); </pre>	POST: /automation/api/v2/ events <pre> { "properties": { "id": "subscription", "keepaliveInterval": 60000, "publishInterval": 1000, "rules": [{ "rule_type": "Queueing", "rule": { "queueSize": 50 } }, { "rule_type": "Sampling", "rule": { "samplingInterval": 0 } }, { "rule_type": "LosslessRateLimit", "rule": { "rateLimit": 20 } }], "nodes": ["scheduler/ admin/info/common-data-rt/ data/counter"] } </pre>

10.13.8 Subscription - Web interface

Simple subscription

To use a GET request for a simple subscription interface, go to the URL */automation/api/v2/events*.

Parameters:

- nodes: List of subscribed nodes
 - separated by comma (",")
 - Required
- publishIntervalMs: PublishInterval in milliseconds
 - optional
 - Default value 1000

The response is an "SSE (Server Send Event) Stream".

Complex subscription

Complex subscriptions can be used to control a subscription response and to make additional settings. Generating a complex subscription to the URL */automation/api/v2/events* via a POST request.

The following JSON object has to be included into the request:

JSON-Objekt für Post

```
{
  nodes: ["node1", "node2"],
  properties : {}, // see sub_properties.fbs
}
```

Also refer to [Chapter 10.13.2 Subscription - Rule set on page 225](#)

If the POST request is successful, the URL to the generated subscription is returned. The ID field in the “Properties” is used as address section. Access the subscription via a GET request to this URL. Delete unused subscriptions via a DELETE request. Unused subscriptions are still automatically deleted after one minute.

Response to a subscription

Notification	Response	Fields in the data element
error	id: <id> notifyType: <event> data: <problem object>	Refer to the “Problem” definition { "type": "about:blank", "title": "DL_INVALID_ADDRESS", "status": 404, "severity": "Error", "notifyType": <type>, "instance": <nodeaddress> }
data	id: <id> event: update data: {"node", <nodeaddress>, "type":<type>, "value":<data>, "time- stamp":<timestamp>}	- node: Node address of the element - type: Value type - Basic type or "object" - schema: Reference to the FlatBuffer type in the Data Layer (only if type is "object") - value: Data itself - timestamp: 64 bit Filetime
browse	id: <id> event: browse data: {"node", <nodeaddress>, "type":<type>, "value":<data>, "time- stamp":<timestamp>}	
metadata	id: <id> event: metadata data: {"node", <nodeaddress>, "type":<type>, "value":<data>, "time- stamp":<timestamp>}	
keepalive	id: <id> event: keepalive data: {"node", <nodeaddress>, "type":<type>, "value":<data>, "time- stamp":<timestamp>}	

10.13.9 Subscription - Lossless subscription to real-time data

In some applications, real-time data has to be subscribed so that all data changes of this node are provided.

Thus, subscribe to the variable address below the RT base address.

Subscription address

<RT base address>/data/<Variable address>

Properties of a lossless subscription:

- **SamplingInterval = 0:**
With a SamplingInterval of 0, a lossless subscription is created. Only nodes that have "qos.lossless: true" in the metadata extensions can be subscribed without loss.
- **Queueing:**
Furthermore, it is recommended to configure a queue with sufficient depth in order not to directly transfer the value upon each value change. With a queue, the content of the queue is only transferred in the PublishInterval. If no queue is configured, a queue of 10 is used for lossless subscriptions by default.
- **LosslessRateLimit:**
Optional parameter with which only every nth value change is transferred.

For examples of such a subscription, go to the chapter ➔ [Chapter 10.13.7 Subscription - Client examples on page 230](#).

10.14 Compliance Checker

10.14.1 Introduction and overview

The Data Layer Compliance Checker is a tool to check that your Data Layer nodes comply with the Data Layer rules.

Upon delivery, the Data Layer Compliance Checker is provided with the Data Layer and available for Windows and Linux.

```
> dl_compliance.exe
Compliance checker 1.6.0 - 55 rules installed
Usage: dl_compliance <options>
  -rules: prints rules
  -connection: connection string to target
  -path: node of datalayer to be checked (default: all datalayer nodes will be checked)
  -ignore: ignore rules with format <ruleID1>:[<path1>,<path2>];<ruleID2>:[<path1>,<path3>]
```

- The tool issues all errors and warnings found for each checked node
- Returns the number of errors as return value

Table 28: Rules

#Rule	Name
1	Metadata has to exist
2	Metadata has to be a FlatBuffer
3	Metadata has to be of the type <i>types/datalayer/metadata</i>
4	Metadata description may not empty
5	Metadata NodeClass may not be Node
10	Name may only contain the characters [a-z][A-Z][0-9]-._
11	Name may only start with a letter
20	Browse list has to be an array of strings
30	Reference type has to exist
31	Reference type has to be reachable
32	Referenced readType exists, but Node is not readable
33	Referenced readInType exists, but Node is not readable
34	Referenced readOutType exists, but Node is not readable
35	Referenced writeType exists, but Node is not writable
36	Referenced writeInType exists, but Node is not writable
37	Referenced writeOutType exists, but Node is not writable
38	Referenced createType exists, but Node is not creatable
39	Different in/out types are only allowed for methods (read/write)
100	Variable has to be readable
101	Variable has to provide the status "OK" while reading
102	Variable readType and writeType have to be equal (if the variable is writable)
103	Variable may not have the operation create
104	Variable may not have the operation delete
200	Resource readType and writeType have to be equal (if Resource is writable)
201	Resource is writable, but not readable
202	Resource may not have the operation create
300	Collection may not be writable
301	Collection may not have the operation delete
302	Collection has to be browsable
303	All child nodes of a Collection have to have the same NodeClass
400	Folder has to be browsable
401	Folder may not be readable
402	Folder may not be writable
403	Folder may not have the operation create
404	Folder may not have the operation delete
405	Folder has to return errors while reading
406	Folder has to return "OK" while browsing
500	Method may not have the operation create
501	Method may not have the operation delete
502	Method may not be browsable

#Rule	Name
503	Method has to be readable or writable (both is not permitted at the same time)
600	Program may not be writable
601	Program has to have the operation create
602	Program may not have the operation delete
603	Program has to be browsable
604	The child nodes of a program have to be resources
700	Type has to be readable
701	Type may not be writable
702	Type may not have the operation create
703	Type may not have the operation delete
704	Type may not be browsable
705	Reading Type has to be successful
706	Type has to be in the Type tree
707	Type has to have the reflection readType
708	Type has to be a FlatBuffer
709	Type has to be a FlatBuffer of the type "Schema"
801	All event references have to refer to types with the prefix types/ events
802	A hasEventType reference is only allowed in nodes with NodeClass Resource, Collection or Folder
803	Types in types/events have to be derived from each other if they are located directly under another type: • A derived type is located directly below another type • A derived type has to have all fields of the superimposed type • A derived type can have additional fields added to the end of the type

Type compatibility check

The Compliance Checker can check FlatBuffers schema for compatible changes.

These changes are:

- Fields can only be added in the schema at the end of a table definition. Older data continues to be read correctly and indicates the default value when reading. Older code will simply ignore the new field.
- Fields that are no longer used, cannot be deleted from the schema. However, these fields can be marked as outdated.

The Compliance Checker performs compatibility checks with bfbs files. A reference directory containing bfbs files can be compared to a new directory containing new bfbs files with the same name.

```
> dl_compliance.exe -is>compatible <path to reference bfbs files> <path to new bfbs files>
```

10.15 Security

10.15.1 Introduction and overview

Overview

This function restricts user access to the Data Layer.

Scopes can be defined. Access Data Layer sections are configured in this scope. These scopes can be assigned via the user and group settings.

Nodes

Address	Node-class	Description
<i>data-layer/security/check-operations</i>	Method	Checks the permitted operations for one Token-Address combination • inType: <i>types/datalayer/check-operations</i> • outType: <i>types/datalayer/allowed-operations</i>
<i>datalayer/security/scopes</i>	Collection	Data Layer scopes: • <i>read</i>: List of all scopes Type: <i>types/datalayer/scopes</i> • <i>create</i>: Creates a new scope Type: <i>types/datalayer/scope</i>
<i>datalayer/security/scopes/*</i>	Resource	Data Layer scopes: • <i>read</i>: Reads the scope definition Type: <i>types/datalayer/scope</i> • <i>write</i>: Changes the scope definition Type: <i>types/datalayer/scope</i>

Scope definition

User-defined scopes can be created and access to individual addresses or address ranges can be configured using the node *datalayer/security/scopes*.

A scope consists of:

- identifier:
Scope name
 - Scope name that can be assigned to a name or a group
 - This scope is part of the token if a user or a group has this token
 - The identifier has to start with "datalayer."
- name:
User-defined name shown in the rights management
- description:
Scope description

A scope contains the following permission lists:

- permissionR:
List of addresses that may be accessed for reading, browsing and reading metadata
- permissionRW:
List of addresses that may be accessed for reading, writing, browsing and reading metadata
- permissionX:

List of addresses that may be accessed for creating, deleting, browsing and reading metadata

- **permissionRWX:**
List of addresses that may be accessed for creating, deleting, reading, writing, browsing and reading metadata

An address in one of the lists may contains placeholders:

- ***:**
 - Stands for any node
 - Can be used as part between two "/" or at the end
- ****:**
 - Stands for any node trees
 - Can be used at the end

Examples

- **datalayer/nodes:**
Allows to access the node *datalayer/nodes*
- **motion/**:**
Allows to access all nodes below *motion*
- **devices/drives/**:**
Allows to access all nodes below *devices/drives*
- **motion/axs/*/pos:**
Allows to access all *pos* nodes of all axes

10.15.2 Scope check

General behavior

The access rights are checked in a request to the Data Layer if a token is available. The access is permitted if a "Scope" in the token allows the access to the requested address. The token has to be sent as well in a TCP connection. The token is optional for IPC and INPROC, but it is considered if it is available. All provider responses are always permitted.

Source	Request	Required access rights
client	read	read
	write	write
	create	execute
	delete	execute
	browse	browse
	metadata	browse
	subscription	read
	unsubscribe	read
	keepalive	read
provider	register	execute
	unregister	execute

Browse and metadata access

A defined path in a scope definition also allows to access the parent node for browse and metadata access. Browsing to a parent node returns only the nodes that may be accessed by the token.

Example: *motion/axs/***

- Allows to access all nodes below *motion/axs*
- Browsing to the address "" does only return *motion*. All other child nodes of "" are not returned with the browse request, as the token does not allow to access these subnodes
- Browsing to the address *motion* returns *axs*
- Reading metadata on the addresses *motion* and *motion/axs* is permitted

Troubleshooting

If accessing is not permitted, the error code DL_SEC_UNAUTHORIZED (0x80070004) is returned.

The extended response of a diagnostics is returned containing the following content:

- mainDiagnosisCode: 080f0600
- detailedDiagnosisCode: 0
- dynamicDescription: Lists the scopes allowing to access the address/request combination

When using the REST interface, the error code FORBIDDEN (403) is returned in case of error. A JSON object as per RFC 7807 is generated as content.

10.15.3 Method *check-operations*

The node *datalayer/security/check-operations* allows to read out the permitted operations for a Token-Address combination.

Input values:

- address:
Address to be checked
- token:
A JWT token to be checked

Output values:

- read:
Read operation permitted
- write:
Write operation permitted
- create:
Create operation permitted
- delete:
Delete operation permitted
- browse:
Browse operation permitted

Example**Eingangswerte:**

```
{
  "address": "datalayer",
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzY29wZSI6WyJyZXhyb3RoLWRldm1jZS5hbGwucnd4Il0sImp0aSI6ImRjMDA3NDViLTEyOGItNDAtZC1iMDgwLTl2OTg3ZGJmMmQ5OCIsIm1hdCI6MTU5OTQ2ODg1NSwiZXhwIjoxNTk5NDcyNDU1fQ.gI6i0ZiKEKhC35v_v2d8VmyeUxEQopq_nqTkGLqfgYQ"
}
```

Ausgangswerte:

```
{  
  "read": true,  
  "write": true,  
  "create": true,  
  "delete": true,  
  "browse": true  
}
```

10.16 Description of the nodes

For the description of the individual nodes of the ctrlX Data Layer, go to:

➔ [ctrlX CORE - Data Layer nodes](#), Reference Book

(DOK-XCORE*-BASE*DL****-RERS-EN-P)

11 ctrlX diagnostic system

11.1 Introduction into the diagnostic system

To use the diagnostic system, the difference between the main diagnostics and the detailed diagnostics have to be known and the following information has to be considered.

The main target of the diagnostic system is to support the user while troubleshooting

11.1.1 Messages, warnings and errors

It is distinguished between messages, warnings and errors:

- **Messages:**
These are messages providing information (e.g. temperature monitoring active). They do not trigger a response.
- **Warnings:**
These are messages informing on a possible upcoming error that can still be avoided (e.g. overtemperature warning). They do not trigger a response.
- **Error:**
These are messages informing on a deviation from the command state (e.g. overtemperature reached - System is paused). They trigger a response. This response depends on the error class (see ➔ [Table 34 Detailed structure of a main diagnostic number on page 252](#)).

11.1.2 Main diagnostics and detailed diagnostics

It is distinguished between main diagnostics and detailed diagnostics:

- **Main diagnostics** (and the respective help entry):
It describes a situation or a general problem and provides a broad overview on the occurrences in the system.
- **Detailed diagnostics** (and the respective help entry):
It describes the situation or the problem in detail and provides detailed information to understand the situation or to solve the problem.

11.1.3 Information on the use of diagnostics

- There can be one or multiple detailed diagnostics for one main diagnostics (for example, refer to [Chapter 11.6 Examples on page 248](#)).
 - If the main diagnostics is present without detailed diagnostics, the description of the situation is sufficient.
 - Detailed diagnostics cannot be issued on their own.
- One detailed diagnostics can also be used for multiple main diagnostics (for examples, refer to [Chapter 11.6 Examples on page 248](#)).
 - Recommendation: Use detailed diagnostics only at one position in the program of a component/bundle.
- The meaning of the main diagnostics (main diagnostic number + text) always has to be unique in the system. However, the meaning of a detailed diagnostics (detailed diagnostic number + text) can be ambiguous. As only unique diagnostics are permitted in the system (e.g. to identify an exact diagnostic source/diagnostic text), the combination of main and detailed diagnostics has to be unique (for examples, refer to [Chapter 11.6 Examples on page 248](#))

11.1.4 Logbook

The Linux service "*journald*" is used as logbook.

The main and detailed diagnostics are entered there and can be displayed on the website of the control under "*Diagnostics → Logbook*".

The following representation exemplary shows an invented excerpt from the logbook and the description of the individual elements.

The screenshot shows a web-based logbook interface. At the top, there's a header with 'Logbook' on the left and user information 'boschrexroth EN' on the right. Below the header is a table with columns: Level, Date, Code, Entity, Description, and a status button. Three rows are visible, each with a different background color (red, yellow, and light blue). Callouts point to various parts of the interface: 'Main diagnostic number' points to the 'Code' column; 'Detailed diagnostic number' points to the 'Entity' column; 'Main diagnostic text' points to the 'Description' column; 'Detailed diagnostic text' points to the 'Description' column; 'Dynamic description' points to the 'Description' column. The status buttons are labeled 'F0 Error', 'Warning', and 'Message'.

Level	Date	Code	Entity	Description	Status
F0 Error	12.11.2021 07:50:55.028	0E1F0001 00000004	devices/io/5	Connection to I/O is not possible The cable is not plugged	F0 Error
Warning	12.11.2021 07:50:49.564	0E1E0001 00000001	diagnosis/temperature	Temperature problem of control detected Temperature exceeded maximum temperature Maximum temperature: 99°C	Warning
Message	12.11.2021 07:50:43.512	0E0A0001		Successfully finished boot process	Message

Fig. 68: Excerpt from the logbook of the ctrlX CORE

journald Namespace

To avoid that Linux logs replace the logs of the ctrlX CORE system for example, *journald Feature Namespace* is used.

The logs of the apps *Device Admin* and *Automation Core* are issued in an own *Namespace* called *snap-systemapps-journald*. All other logs like Linux operating system logs or all logs of other apps (if not assigned to their own *Namespace*) are issued in the *Namespace* standard.

The logs in the *Namespace snap-systemapps-journald* are backed up in a ring buffer to ensure that logs in the *Namespace* standard do not replace those in the logbook. This allows to visualize relevant messages of the ctrlX CORE system for a longer period in the logbook.



Due to the different ring buffers of the standard *Namespace* and the *Namespace snap-systemapps-journald*, it is possible that older logs of a *Namespace* have already been overwritten by its ring buffer due to a higher log frequency, whereas the older logs of the other *Namespace* are still available.

How to read out *Namespace* logs via a Linux command, refer to: ➔ [Chapter 11.12 Log structure \(message format\) *journal*d and its metadata on page 259.](#)

11.1.5 Further information

- ➔ [Chapter 13.2.2 Window – Logbook on page 271](#)
- ➔ [Chapter 7.1 Logbook – Overview on page 132](#)

11.2 Using the diagnostic system

The diagnostic system provides its functionalities via two interfaces:

- **C++ bundle interface:**
For this interface description, refer to the SDK documentation or in the interface headers of the "common.log.diagnosis" component.
- **Data Layer:**
For the documentation on this interface, go to ➔ [Chapter 11.10 User interface via Data Layer on page 256.](#)

The diagnostic system provides the following different functionalities and is exemplarily shown in the following.



For details on which interface supports which functionalities, refer to the description in the SDK, the interface documentation or the Data Layer description ➔ [Chapter 11.10 User interface via Data Layer on page 256.](#)

- Register and unregister main and detailed diagnostics.
 - To use the diagnostic system, at least the main diagnostics have to be created and used.
 - Using detailed diagnostics is optional, but recommended.
 - The diagnostics to be used have to be registered at the diagnostic system and unregistered after ending the user program.
- Issue and reset diagnostics (messages, warnings, errors).
 - Errors and warnings can be set to the statuses "active" and "reset". Messages are always in the status "active".
For more information, go to the chapter ➔ [Chapter 11.3 Error categories, log types and \(error\) deletion on page 242.](#)
- Clear single or all pending errors.
 - Errors in the "reset" status can be deleted from the list of pending errors and warnings.
(see ➔ [diagnosis/get/actual/list](#))
- Read out the currently pending message (error or warning) with the highest priority.
- Read out the list of the pending errors and warnings.
- Read out texts of registered main and detailed diagnostics.
- Register and unregister on diagnostic events.
 - Inform when a specific diagnostic event occurs, e.g. F2 error was issued, status of a pending high-priority message changed, command to delete own errors was executed.

11.3 Error categories, log types and (error) deletion

Since the introduction of bit 20 ("Reset") in the main diagnostic number (➔ [Table 34 Detailed structure of a main diagnostic number on page 252](#)), different log types are required.

This dynamic bit defines the following:

- Whether an error or a warning is active (bit 20 = 0) →, the diagnostics is in the "active" status

Example: 0x080F0100 "Error in diagnostic system"

- Whether an error or a warning is not active anymore (bit 20 = 1) →, the diagnostics is in the "reset" status

Example: 0x081F0100 "Error in diagnostic system"

- Messages are only possible in the "active" status (with bit 20 = 0)

Due to that reason and depending on the diagnostic type to be issued, two different error categories are defined:

- "Fire and forget":

Defines an error or a warning. Its cause is resolved immediately after the diagnostics has been issued.

- "Fire and remember":

Defines an error or a warning. Its cause is not resolved after the diagnostics has been issued.



Messages do not have any cause that can be resolved.

Category "Fire and forget"

Errors and warnings of diagnostics of this category do not have to be followed up, as the diagnostics have only an informing purpose. The diagnostics is issued via the diagnostic system. Further dealing is not required.

An entry with the "reset" status is generated for the diagnostics in the log (bit 20 = 1).

Example:

- Command error: E.g. "Backing up the configuration failed"
- Execution error: E.g. "Watchdog of task 'Task1' elapsed"

Diagnostics of this category can be issued via the diagnostic system (e.g. via the Data Layer address: → [diagnosis/set/set-and-reset](#)).

Errors and warnings are set and reset immediately. Messages are only set. Warnings are immediately and automatically deleted. Errors are set to the "reset" status. The errors have to be deleted separately. The user has to analyze errors and then to delete them manually.

Category "Fire and remember"

Errors and warnings of diagnostics of this category have to be followed up. If the error occurs, the diagnostics with the "active" status has to be issued via the diagnostic system. If the error cause does not exist anymore, the diagnostics with the "reset" status has to be issued via the diagnostic system.



When using this category, it is recommended to register on the diagnostic event "reset-all-pending" to allow a response on the command "Resetting all pending errors and warnings". The event informs that all connected instances have to try to reset all errors and warnings as well as entities or resources currently in error state.

For more information, refer to the Data Layer address → [diagnosis/events/reset-all-pending](#).

One entry each is generated for the diagnostics in the logbook:

- Entry for the "active" status (bit 20 = 0) if the error is reported as active
- Entry for the "reset" status (bit 20 = 1) if the error is reset

Examples:

- Cyclically resulting error: E.g. "Position limit value exceeded"
- Command errors blocking further execution steps: E.g. "Incorrect scheduler configuration"

Diagnostics of this category can be set to the "active" status via the diagnostic system (e.g. via the Data Layer address: ➔ [diagnosis/set/set-active](#)) and set to the "reset" status (e.g. via the Data Layer address: ➔ [diagnosis/set/reset-active](#)).

Errors, warnings and messages are issued when "Setting to active". Warnings and errors are set to the "active" status.

Errors and warnings in the "active" status are reset when "Reset". Messages are not supported. Warnings are immediately and automatically deleted. Errors are set to the "reset" status. The errors have to be deleted separately.



When shutting down or disabling the bundle or an app, it is mandatory that dependent active diagnostics (errors and warnings) of the bundle are reset before these diagnostics are logged off. Otherwise, these diagnostics cannot be deleted anymore. Use for example the Data Layer address: ➔ [diagnosis/set/reset-active](#).

(Error) clearing

The diagnostic system manages a list with all active and reset errors and all active warnings. This list can be shown on the website of the control under: *"Diagnostics → Logbook → tab Alarms"*

Refer to: ➔ [Tab – "Alarms"](#)

Warnings are automatically deleted as defined if they are set to the "reset" status. The user has to analyze errors and then to delete them manually.

If all pending errors are deleted (respective Data Layer address: ➔ [diagnosis/confirm/all-errors](#)), all errors in the "reset" status are removed from the list and the errors in the "active" status remain. Active warnings also remain.

To delete individual errors, use the Data Layer address: ➔ [diagnosis/confirm/error](#).

11.4 Use cases

11.4.1 General information

It can be distinguished between two types of diagnostics:

- **Runtime diagnostics:**
These are messages issued at system runtime and not caused by an external event.
- **Event diagnostics:**
These are messages caused by an external event, e.g. the user executes an action causing an error.

Some possible use cases including examples are shown in the chapters ➔ [Chapter 11.4.2 Runtime diagnostics on page 245](#) and ➔ [Chapter 11.4.3 Event diagnostics on page 246](#) to show the possible use of main and detailed diagnostics. They are described using error cases. The same applies to messages and warnings.

11.4.2 Runtime diagnostics

Use case 1:

Direct topic-specific main diagnostics without further detailed diagnostics

- Description:
The main diagnostics are topic-specifically and contain sufficient information. A detailed diagnostics is not required.
- Example of a main diagnostics:
"Critical temperature reached; processor is stopped"

Use case 2:

Component-comprehensive general group main diagnostics with a more precise detailed diagnostics

- Description:
The main diagnostics is used as general collector for component-comprehensive errors and the detailed diagnostics describes the error more in detail.
General runtime errors of the respective detailed diagnostics can thus be collected in one single main diagnostics.
- Example of a main diagnostics:
"Error in diagnostic system"
- Example of a detailed diagnostics:
"Main diagnostic number is not registered"

Use case 3:

Component-comprehensive topic-specific group main diagnostics with a more precise detailed diagnostics

- Description:
The main diagnostics is used as topic-specific collector for component-comprehensive errors and the detailed diagnostics describes the error more in detail.
Topic-specific runtime errors of the respective detailed diagnostics can thus be collected in one single main diagnostics.
- Example of a main diagnostics:
"Error in task watchdog"
- Example of a detailed diagnostics:
"Watchdog of task "Task1" elapsed"

11.4.3 Event diagnostics

Use case 4:

Component-comprehensive topic-specific group main diagnostics with a more precise detailed diagnostics

- Description:
The main diagnostics is used as topic-specific collector for component-comprehensive errors and the detailed diagnostics describes the error more in detail.
Topic-specific event errors of the respective detailed diagnostics can thus be collected in one single main diagnostics.
- Example of a main diagnostics:
"Error while creating an axis"
- Example of a detailed diagnostics:
"Axis 1: Incorrect acceleration configuration"

Use case 5:

Component-comprehensive topic-specific group main diagnostics with a more precise detailed diagnostics

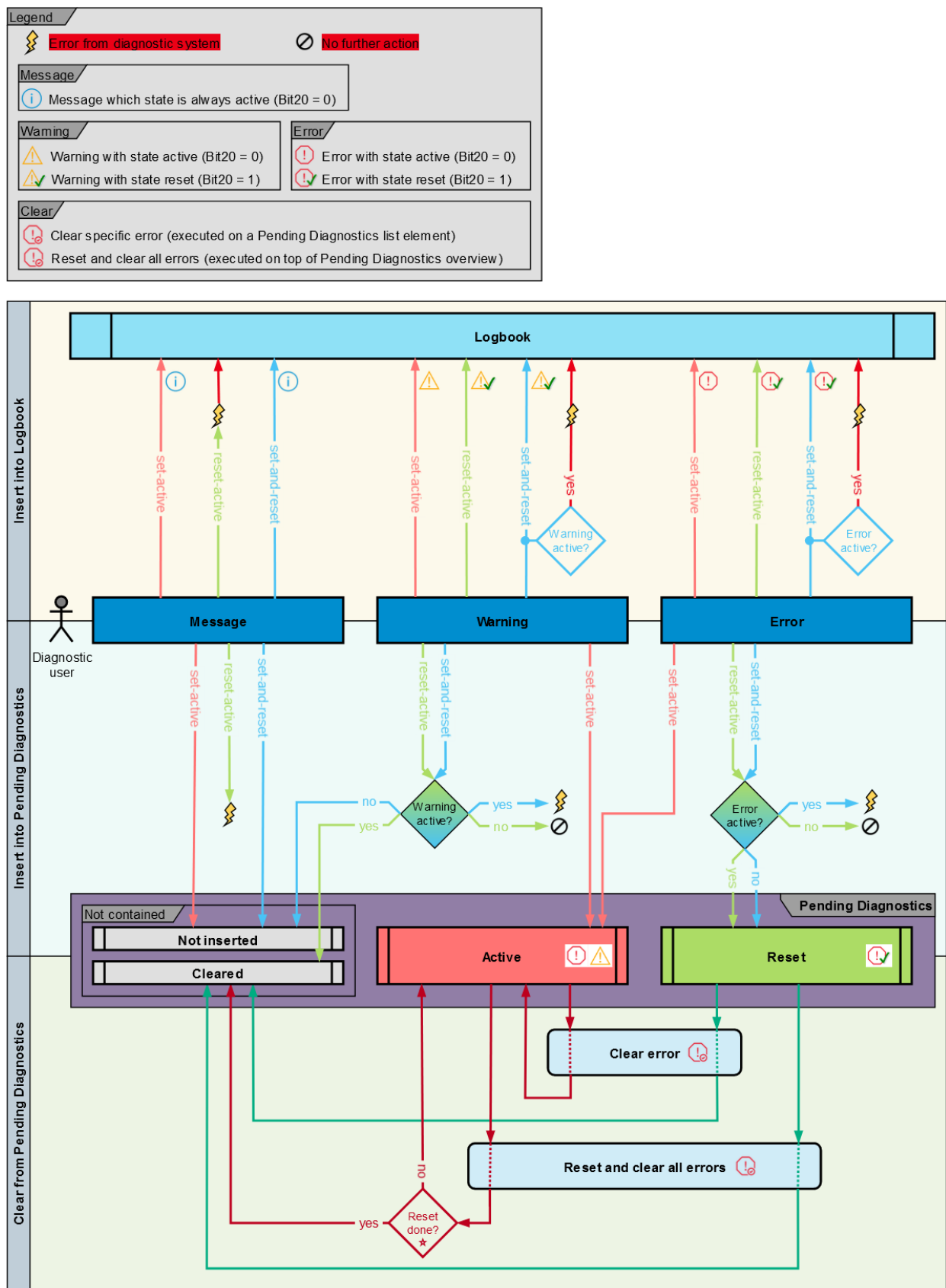
- Description:
The main diagnostics is used as topic-specific collector for component-comprehensive errors and the detailed diagnostics describes error more in detail (which component caused the error and what happened).
Topic-specific event errors of the respective detailed diagnostics can thus be collected in one single main diagnostics.
- Example of a main diagnostics:
"Error while backing up/restoring"
- Example of a detailed diagnostics:
"Restoring the EtherCAT configuration failed"

11.5 Diagnostic handling summary

Wording and key facts

- The status of errors and warnings can either be "active" or "reset". The message status is always "active".
 - Alternative for "reset" is "marked as clearable".
- All errors are shown in the "Pending Diagnostics" irrespective of the log type (set-active, reset-active, set-and-reset) used.
 - The ctrlX CORE LED thus flashes red.
 - The user has to analyze errors and "cleared" them manually.
- All warnings are displayed in "Pending Diagnostics" as long as they are "active".
 - The ctrlX CORE LED thus flashes orange if no errors are pending.
 - Warnings are automatically "cleared" if they are "reset".
- Messages are not displayed in "Pending Diagnostics".
- Each diagnostics (consisting of main diagnostic number, detailed diagnostic number and entity) can only be displayed once in "Pending Diagnostics".
- If a "clearing" is executed once, all errors in the status "reset" are removed from the "Pending Diagnostics" list. Errors in the status "active" also remain. Active warnings also remain "active".

Flow diagram of the diagnostics (“Logbook” and “Pending Diagnostics”)



★ Reset all pending: An event is triggered which indicates that all connected instances need to try to reset all warnings and errors as well as reset specific entities that are in error state

Fig. 69: Flow diagram of the diagnostics (“Logbook” and “Pending Diagnostics”)

11.6 Examples



The diagnostic numbers and diagnostic texts in the following use invented values.

11.6.1 Examples of possible diagnostic combinations

Using a main diagnostics without detailed diagnostics

Main diagnostic number	Main diagnostic text	Detailed diagnostic number	Detailed diagnostic text
0A0F0123	Error while booting	-	-
0A0F2456	Error in the library XYZ	-	-

Using multiple detailed diagnostics with the same main diagnostics

Main diagnostic number	Main diagnostic text	Detailed diagnostic number	Detailed diagnostic text
0A0F0123	Error while booting	00231234	Configuration could not be loaded
0A0F0123	Error while booting	00235678	Library ABC not ready

Using the same detailed diagnostics with different main diagnostics

Main diagnostic number	Main diagnostic text	Detailed diagnostic number	Detailed diagnostic text
0A0F0123	Error while booting	00231234	Configuration could not be loaded
0A0F2456	Error in the library XYZ	00231234	Configuration could not be loaded

Using ambiguous detailed diagnostics

Main diagnostic number	Main diagnostic text	Detailed diagnostic number	Detailed diagnostic text
0A0F0123	Error while booting	00231234	Configuration could not be loaded
0A0F2456	Error in the library XYZ	00231234	Function block returned an error

11.6.2 Examples of unpermitted/impossible diagnostic combinations

Using a detailed diagnostics without main diagnostics

Main diagnostic number	Main diagnostic text	Detailed diagnostic number	Detailed diagnostic text
-	-	00231234	Configuration could not be loaded

Using ambiguous main diagnostics

Main diagnostic number	Main diagnostic text	Detailed diagnostic number	Detailed diagnostic text
0A0F0123	Error while booting	-	-
0A0F0123	Error in the library XYZ	-	-

Using ambiguous detailed diagnostics with the same main diagnostics

Main diagnostic number	Main diagnostic text	Detailed diagnostic number	Detailed diagnostic text
0A0F0123	Error while booting	00231234	Configuration could not be loaded
0A0F0123	Error while booting	00231234	Function block returned an error

11.7 Structure and use of customer, OEM and partner diagnostics

Details using the diagnostic numbers for:

- OEM customers
- ctrlX World & Rexroth-internal partners

The structure of the Bosch Rexroth ctrlX CORE diagnostics are not further described in detail.

When using the ctrlX diagnostic handler, follow the structure of the main diagnostic numbers.

OEM customers

Table 29: Main diagnostics (OEM customer)

Bits	31-30	29-24	23-21	20	19 -16	15-12	11-0
Description	Reserved	Source Type	Reserved	Reset	Diagnostic class	Status code: Priority	Status code: Number
Range	0x0E, 0x0F		0x0, 0x1		0xA, 0xE, 0xF	e.g. 0x0	0x000 -> 0xFFF

Table 30: Detailed diagnostics (OEM customer)

Bits	OEM-specific (31-0)
Description	OEM-specific
Range	OEM-specific

ctrlX World & Rexroth-internal partners

Table 31: Main diagnostics (ctrlX World & Rexroth-internal partners)

Bits	31-30	29-24	23-21	20	19 -16	15-12	11-0
Description	Reserved	Source Type	Reserved	Reset	Diagnostic class	Status code: Priority	Status code: Number
Range	0x30 -> 0x37		0x0, 0x1		0xA, 0xE, 0xF	e.g. 0x0	0x000 -> 0xFFFF
Note	One or multiple main diagnostics per diagnostic class and priority is provided to each partner. They have to be reserved at Bosch Rexroth						

Table 32: Detailed diagnostics (ctrlX World & Rexroth-internal partners)

Bits	Partner-specific (31-0)
Description	Partner-specific
Range	Partner-specific

General notes

The following criteria apply to main diagnostic numbers:

- The 32 bit-wide diagnostic numbers have a specified structure (see [↗ Table 29 Main diagnostics \(OEM customer\) on page 249](#) or [↗ Table 31 Main diagnostics \(ctrlX World & Rexroth-internal partners\) on page 249](#))
- The numbers have to be unique for the complete ctrlX AUTOMATION system (For details, refer to [↗ Chapter 11.1.3 Information on the use of diagnostics on page 241](#))
- The numbers are within a specified number range (see [↗ Additional information for the OEM customer on page 250](#) or [↗ Additional information for ctrlX World & Rexroth-internal partners on page 251](#))

The following criteria apply to the detailed diagnostic numbers:

- The 32 bit-wide diagnostic number have a freely definable structure (OEM- or partner-specific)
- The numbers can be ambiguous in the ctrlX AUTOMATION system, but have to be unique when combined with a main diagnostics (For details, refer to [↗ Chapter 11.1.3 Information on the use of diagnostics on page 241](#))

Additional information for the OEM customer

General information:

- As an OEM customer can use all main diagnostics in the source types 0x0E and 0x0F, no other application of another OEM customer may be executed in a ctrlX AUTOMATION system, as the main diagnostic numbers might overlap and are thus not unique anymore
- Thus, uploading OEM applications to the ctrlX World App Store is not allowed

Main diagnostics:

- To ensure unambiguity in the ctrlX AUTOMATION system, use the source types 0x0E und 0x0F
- All other fields (e.g. the complete use of the status code) can be freely used within the specifications defined in [↗ Chapter 11.8.2 Main diagnostics on page 252](#)
- Bosch Rexroth does not have to be consulted with regard to the number reservation
- Examples of usable number ranges:
 - Messages:
0E0A0000 -> 0E0A0FFF und 0F0A0000 -> 0F0A0FFF
 - Warnings:
0E0E0000 -> 0E0E0FFF und 0F0E0000 -> 0F0E0FFF
 - Error priority 2:
0E0F2000 -> 0E0F2FFF and 0F0F2000 -> 0F0F2FFF

Detailed diagnostics:

- Using "00000000" is not permitted
- Bosch Rexroth does not have to be consulted with regard to the number reservation

Additional information for ctrlX World & Rexroth-internal partners

Main diagnostics:

- To ensure unambiguity in the ctrlX AUTOMATION system, use the source types 0x30 -> 0x37
- One or multiple main diagnostics per diagnostic class and priority is provided to each partner
- All other fields (e.g. the complete use of the status code) can be freely used within the specifications defined in [Chapter 11.8.2 Main diagnostics on page 252](#)
- Bosch Rexroth does not have to be mandatorily consulted with regard to the number reservation.
Please contact your Partner Manager or send an email to ctrlx.world@boschrexroth.com
- Examples of usable number ranges:
 - Partner 1:
0x300A0000, 0x300E0000,...
 - Partner 2:
0x300A0001, 0x300E0001,... + 0x300A0002, 0x300E0002,... + 0x300A0003, 0x300E0003,...
 - ...

Detailed diagnostics:

- Using "00000000" is not permitted
- Bosch Rexroth does not have to be consulted with regard to the number reservation

11.8 Structure of the diagnostic number

11.8.1 General information

The diagnostic entry also consists of a main and a detailed diagnostics.

Each main and detailed diagnostics contain a diagnostic number, a diagnostic text and further information.

The diagnostic numbers are each shown as 8-digit hexadecimal number.

Simplified example of a virtual diagnostic entry:

Main diagnostics: 090F2ABC axis error

Detailed diagnostics: 0C5512EF Position beyond the limit value

To ensure that each diagnostics is unique, the following applies to the complete ctrlX AUTOMATION system:

- Each main diagnostic number may only be defined once
- Each detailed diagnostic number together with a main diagnostics may only be defined once

The detailed structure of main and detailed diagnostic numbers differs essentially.

The structure of the main diagnostic number is explained in the chapter [Chapter 11.8.2 Main diagnostics on page 252](#).

The structure of the detailed diagnostic number is not explained in detail, as this detailed diagnostic number can be defined by the OEM customer and the ctrlX World & Rexroth-internal partners.

Sole exception: Using "00000000" is not permitted.

11.8.2 Main diagnostics

Structure of a main diagnostics

Table 33: General structure of a main diagnostic number

Bits	31-30	29-24	23-21	20	19 -16	15-12	11-0
Description	Reserved	Source Type	Reserved	Reset	Diagnostic class	Status code: Priority	Status code: Number
Example	09		0		F	2	00C

Table 34: Detailed structure of a main diagnostic number

Bits	Value	Description	Remarks
31-30	0b00	Reserved	
29-24		Source Type	Defines the functional scope of the main diagnostics
	0x00-0x0D	Reserved	
	0x0E	OEM machine-specific	To be used only by the OEM customer to create individual diagnostics Example: Machine-specific application of an OEM
	0x0F	OEM machine-HMI application	To be used only by the OEM customer to create individual diagnostics Example: Visualization of the OEM managing the machine
	0x10-0x2F	Reserved	
	0x30-0x37	ctrlX World & Rexroth-internal partners	Only to be used by ctrlX World & Rexroth internal partners to create own diagnostics As the diagnostic number have to remain unique in the system, the number to be used is assigned to each Bosch Rexroth partner
	0x38-0x3F	Reserved	
23-21	0x0	Reserved	

Bits	Value	Description	Remarks
20	0b0/0b1	Reset	Defines whether the diagnostics is still active or not active anymore. Note: If the ctrlX OS diagnostic interface (or the respective Data Layer interface at the address ↪ diagnosis) is used, the diagnostic system manages this bit. Thus, it always has to be set to "0". Only used for the classes "Warning" (0xE) and "Error" (0xF). "Messages" (0xA) are always active (0b0). 0: Cause of the error/warning is (still) active. Behavior after <i>Clear error</i>: - Error: The error remains active and is still shown in the dialog ↪ Pending Diagnostics - Warning: Not affected by <i>Clear error</i> 1: Cause of the error/warning not active (any more) Behavior after <i>Clear error</i>: - Error: The error is reset and deleted and it is not displayed anymore in ↪ Pending Diagnostics dialog - Warning: Not affected by <i>Clear error</i>
19-16		Diagnostic class	Defines the class and thus the type of the main diagnostics
	0xA	Message	Use: Information on operating states or other information Example: e.g. "Booting completed", "Application download completed"
	0xE	Warning	Use: Warning against specific states in the system Example: e.g. "Temperature warning", "Hard disk memory at minimum"
	0xF	Errors	Use: Reporting an error on specific states in the system Example: e.g. "Maximum temperature exceeded - System shutdown", "General error of the axis"
	Other	Reserved	
15-0	0x0000 - 0xFFFF	Status code	Defines the priority and a specific number - The status code contains 4 bits with the priority (bits 15-12) and a specific number (bits 11-0). - The complete range of the specific number defined in the following can be used. Ensure that each main diagnostic number is unique in the ctrlX AUTOMATION system The structures and responses described in the following differ for the diagnostic classes (0xA, 0xE, 0xF): 0xA - Status codes "Message": Range: 0x0000 → 0x0FFF (consequently: 0xA0000 → 0xA0FFF) → max. 4096 diagnostics allowed, as only 0xA0 is valid Priority: No priorities, but priority bits are reserved and have to be set to 0

Bits	Value	Description	Remarks									
			Responses: No response 0xE - Status codes "Warning": Range: 0x0000 → 0x0FFF (consequently: 0xE0000 → 0xE0FFF) → max. 4096 diagnostics allowed, as only 0xE0 is valid Priority: No priorities, but priority bits are reserved and have to be set to 0 Responses: No response 0xF - Status codes "Error": Range (per priority): 0x000 → 0xFFFF (e.g for priority 2: 0xF2000 → 0xF2FFF) → max. 4096 diagnostics allowed per priority Priority: Refer to the following table Responses: Refer to the following table <table><tr><th>Priority</th><th>General significance</th><th>Response and notes</th></tr><tr><td>0</td><td>Non-fatal error</td><td>No error response</td></tr><tr><td>2</td><td>Stop of an individual axis/kinematics</td><td> • A response is executed for: – an individual axis or kinematics, managed by the ctrlX OS Motion component – Errors reported by all ctrlX CORE components and user components via the ctrlX OS diagnostic interface – Errors issued via the Data Layer diagnostic interface • A response is only executed if the error contains the correct entity of an individual axis or kinematics </td></tr></table>	Priority	General significance	Response and notes	0	Non-fatal error	No error response	2	Stop of an individual axis/kinematics	• A response is executed for: – an individual axis or kinematics, managed by the ctrlX OS Motion component – Errors reported by all ctrlX CORE components and user components via the ctrlX OS diagnostic interface – Errors issued via the Data Layer diagnostic interface • A response is only executed if the error contains the correct entity of an individual axis or kinematics
Priority	General significance	Response and notes										
0	Non-fatal error	No error response										
2	Stop of an individual axis/kinematics	• A response is executed for: – an individual axis or kinematics, managed by the ctrlX OS Motion component – Errors reported by all ctrlX CORE components and user components via the ctrlX OS diagnostic interface – Errors issued via the Data Layer diagnostic interface • A response is only executed if the error contains the correct entity of an individual axis or kinematics										

Bits	Value	Description	Remarks	
			6	Stop of all axes/ kinematics • A response is executed for: - all axes or kinematics managed by the ctrlX OS Motion component - Errors reported by all ctrlX CORE components and user components via the ctrlX OS diagnostic interface - Errors issued via the Data Layer diagnostic interface
			8	Fatal application error (exception or forced F8 error) in an app (e.g. rexroth-automationcore) The ctrlX OS Run-time (and thus all included applications) is shut down and restarted automatically.
			9	Fatal system error (exception or forced F9 error) in an app (e.g. rexroth-automationcore) The ctrlX CORE is switched off and has to be restarted by a voltage cycle.

11.9 Connection to the Data Layer RT (real time)

The diagnostic system provides diagnostic messages for the real-time access using the Data Layer RT. Thus, the user receives detailed information about the current (diagnostic) status of the control.

Functional principle and configuration

Only one - the last issued diagnostics (consisting of the main diagnostics and, if available, the detailed diagnostics), i.e. the active diagnostics is provided, not the diagnostics with the highest priority.

- Use the Data Layer address "[↗diagnosis/cfg/realtime/numbers](#)" to enable or disable the Data Layer RT nodes, displaying the current diagnostic numbers (main diagnostic number and detailed diagnostic number).
- Use the Data Layer address "[↗diagnosis/cfg/realtime/numbers/class](#)" to configure the diagnostic class active for the Data Layer RT node.

The following can be selected:

- ERROR: Only errors are shown in the Data Layer RT
- WARNING: Only warnings and errors are shown in the Data Layer RT
- MESSAGE: Only messages, warnings and errors are shown in the Data Layer RT

- The Data Layer RT address "[↗ diagnosis/realtime-data/input/data/main-diagnosis](#)" shows the main diagnostic number of the latest issued diagnostics, irrespective of the selected configuration
- The Data Layer RT address "[↗ diagnosis/realtime-data/input/data/detailed-diagnosis](#)" shows detailed diagnostic numbers belonging to the main diagnostic number if available

The addresses are described in detail in the respective Data Layer documentation.

Use case

This functionality can for example be used for the oscilloscope function of the control.

By using the oscilloscope, diagnostic messages can be recorded in an exact temporal interdependency to other events (such as events occurring during the Motion calculation, events at the EtherCAT bus, PLC variable changes).

11.10 User interface via Data Layer

For more information and description of the *diagnosis* node, go to:

- [↗ "diagnosis" node](#)
- [↗ Chapter 10.2 Data exchange via Data Layer on page 184](#)

11.11 Registering and unregistering main and detailed diagnostics

11.11.1 General information

A JSON file is used to register and unregister main and detailed diagnostics. The texts belonging to the diagnostic numbers are registered in the diagnostic system. Thus, only the diagnostic numbers have to be used in case of further use of the diagnostic interface. The diagnostic texts registered in the diagnostic system are default diagnostic texts, which are only used if the diagnostic texts were stored in a language variant that was not entered. Only the default diagnostic texts are used in the device-internal logbook (journald). The JSON structure is described and the options to register and unregister diagnostics in the following.

11.11.2 Structure of the JSON file

The JSON structure described in the following applies to:

- Files to register the default diagnostic texts in the diagnostic system
 - Rexroth diagnostics are default diagnostic texts in English.
 This is also recommended for all other diagnostics to be registered.
- Files to support multilingualism of diagnostic texts

The JSON file is defined by the following structure:

```
{
  "product": "<PRODUCT_VERSION>",
  "mainDiagnostics": {
    "<MAIN_DIAG_1_NUMBER>": {
      "text": "<MAIN_DIAG_1_TEXT>",
      "version": "<MAIN_DIAG_1_VERSION>",
      "detailedDiagnostics": {
        "<DETAILED_DIAG_1_NUMBER>": {
          "text": "<DETAILED_DIAG_1_TEXT>",
```

```

    "version": <DETAILED_DIAG_1_VERSION>
  },
  "<DETAILED_DIAG_2_NUMBER>": {
    "text": "<DETAILED_DIAG_2_TEXT>",
    "version": <DETAILED_DIAG_2_VERSION>
  }
},
"<MAIN_DIAG_2_NUMBER>": {
  "text": "<MAIN_DIAG_2_TEXT>",
  "version": <MAIN_DIAG_2_VERSION>,
  "detailedDiagnostics": {
    "<DETAILED_DIAG_3_NUMBER>": {
      "text": "<DETAILED_DIAG_3_TEXT>",
      "version": <DETAILED_DIAG_3_VERSION>
    }
  }
},
"<MAIN_DIAG_3_NUMBER>": {
  "text": "<MAIN_DIAG_3_TEXT>",
  "version": <MAIN_DIAG_3_VERSION>
}
}
}

```

Explanations on the format:

- All main diagnostic texts are listed one after another.
- Below each main diagnostic number, there is a diagnostic text and the version and all respective detailed diagnostic numbers. The respective diagnostic text and the version are then below those detailed diagnostic numbers.
- The version is always optional and only required if for example multiple instances register the same diagnostics, as always the latest version of the registered diagnostic texts is used.
- The tag "product" is also optional. The tag is not analyzed and only for information purposes.

Example of a JSON file structure:

```

{
  "product": "XCR-V-0114",
  "mainDiagnostics": {
    "0E0A0001": {
      "text": "Successfully finished boot process",
      "version": 1,
      "detailedDiagnostics": {
        "00000001": {
          "text": "Finished boot process without errors",
          "version": 1
        }
      }
    },
    "0E0E0001": {
      "text": "Temperature problem of control detected",
      "version": 1,
      "detailedDiagnostics": {
        "00000001": {
          "text": "Temperature exceeded maximum temperature"
        },
        "00000002": {
          "text": "Defect of temperature sensor"
        }
      }
    },
    "0E0E0002": {
      "text": "User id was changed",

```

```

    "version": 2
  },
  "0E0E0003": {
    "text": "Watchdog occurred",
    "detailedDiagnostics": {
      "00000001": {
        "text": "Unsafe machine state DEBUG"
      }
    }
  },
  "0E0F0001": {
    "text": "Connection to I/O is not possible",
    "version": 1,
    "detailedDiagnostics": {
      "00000001": {
        "text": "The cable is not plugged",
        "version": 1
      }
    }
  },
  "0E0F2001": {
    "text": "General axis command error",
    "version": "1",
    "detailedDiagnostics": {
      "00000001": {
        "text": "Axis for axis command was not found",
        "version": 1
      }
    }
  },
  "0E0F6001": {
    "text": "Fatal axis command error",
    "version": 1,
    "detailedDiagnostics": {
      "00000001": {
        "text": "Stop commissioning of all axes",
        "version": 1
      }
    }
  }
}

```



Notes on formatting diagnostic texts

If there are special characters like <\> or <">, the masking character <\> always has to be placed in front (e.g. <\\> or <\>).

File name specification

The following naming schema is recommended for file name:

<ComponentName/productName>.diagnostics.<xx-YY>.json

- <xx-YY> describes the language code valid for the file (example: en-US).

For details, refer to ➔ https://en.wikipedia.org/wiki/Language_localisation#Language_tags_and_codes.

11.11.3 Options to register and unregister main and detailed diagnostics

There are the following options to register and unregister main and detailed diagnostics for default diagnostic texts:

- Via the JSON file in the Data Layer stored on the ctrlX CORE.
 - Respective Data Layer addresses:
 - ➔ [diagnosis/registration/register-file](#)
 - ➔ [diagnosis/registration/unregister-file](#)
 - The JSON file has to be stored on the ctrlX CORE.

It can be added to the current configuration via the WebDAV protocol. Create the "diagnostics" folder in the current configuration and save the JSON file to this folder.

When calling the respective Data Layer addresses, it is sufficient to specify the JSON file as file path (e.g. "*test.diagnostics.en-US.json*").

The complete file path can also be specified (e.g. "*/var/snap/rexroth-solutions/common/solutions/DefaultSolution/configurations/appdata/diagnostics/test.diagnostics.en-US.json*").

- Via the content of the JSON file in the Data Layer.
 - Respective Data Layer addresses:
 - ➔ [diagnosis/registration/register-file-content](#)
 - ➔ [diagnosis/registration/unregister-file-content](#)
 - The JSON file does not have to be stored on the control, but on a device communicating with the ctrlX CORE. The file content has to be interpreted binarily and then transferred.
- Via the JSON file in the C++ Bundle Interface of the component "common.log.diagnosis" (only available if an own bundle is provided) provided in the bundle.

For sample implementations on the different options, refer to the SDK.

11.12 Log structure (message format) *journald* and its metadata

Rexroth logs are entered into *journald* in a defined message format (see ➔ [Chapter 11.12.1 Message format on page 259](#)) and with defined metadata (see ➔ [Chapter 11.12.2 Metadata on page 261](#)). This metadata can also be retrieved directly using the Linux Shell command *sudo journalctl* for example. To further define this command, specify metadata and thus, not all system messages are output.

For details on the command *journalctl*, refer to the Linux documentation:

➔ <https://www.freedesktop.org/software/systemd/man/journalctl.html>

As certain logs are entered into an own *journald Namespace*, different options to read out logs via the command *journalctl* are described in the following:

- Reading logs of specific *Namespace*:
`sudo journalctl --namespace=snap-systemapps-journald`
- Reading logs of the standard *Namespace*:
`sudo journalctl`
- Reading logs of all *Namespace*s:
`sudo journalctl --namespace=*`

For general information on *Namespace*s, see: ➔ [journald Namespace on page 241](#).

11.12.1 Message format

A ctrlX CORE log is shown as follows in the *journald* by default:

```
<Time stamp>|<User ID>|<Main diagnostic number>|<Main diagnostic text>|<Detailed diagnostic number>|
<Detailed diagnostic text>|<Entity>|<Origin>|<Unit name>|<File name>|<Function name>|<Line number>|
<Dynamic description>
```

The individual elements are separated by pipes (|).

Table 35: Explanation of the elements and specification about its use in diagnostics and trace

Element	Meaning	Use in diagnostics	Use in trace
<Time stamp>	Time stamp when issuing the log in the ctrlX CORE diagnostic system acc. to ISO 8601. Format: "yyyy-mm-ddThh:mm:ss.microsZ" with micros = microseconds.	Optional	Optional
<User ID>	Unique user ID. This element is only filled in case of user-relevant and security-relevant logs.	Optional	Never
<Main diagnostic number>	Unique 32 bit number	Always	Always
<Main diagnostic text>	Short text describing the meaning of the main diagnostic number. The exact meaning of the main diagnostic number is described in detail in the documentation of the respective diagnostic number.	Always	Always
<Detailed diagnostic number>	Unique 32 bit number. This number directly refers to the main diagnostic number.	Optional	Never
<Detailed diagnostic text>	Short text describing the meaning of the detailed diagnostic number. The exact meaning of the detailed diagnostic number is described in detail in the documentation of the respective diagnostic number.	Optional	Never
<Entity>	Entity/resource. Example: This corresponds to the system URI of the Data Layer address if the resource is provided with a respective address.	Optional	Optional
<Origin>	Name of the application. Specification: This corresponds to the snap name if the application is a snap.	Optional	Optional
<Unit name>	Name of the respective unit (bundle/component).	Optional	Always
<File name>	Name of the file in which the log entry is issued.	Optional	Optional
<Function name>	Name of the function in which the log entry is issued. The name is output with namespaces if required.	Optional	Optional
<Line number>	Line number at which the log entry is issued.	Optional	Optional
<Dynamic description>	Additional log description that can be output in dynamic elements. The recommended language is English. l18n (multilingualism) does not affect this text. It remains in the language issued.	Optional	Optional

Example of a diagnostic log:

```
2020-07-31T19:10:42.123456Z|101|080F0100|Error in diagnosis system|00666006|Main diagnosis code
is not registered|diagnosis/set|rexroth-automationcore|common.log.diagnosis|handler.cpp|logThread|
600|Not registered main diagnosis code 0x080F01FFF
```

Example of a trace log:

```
2020-07-31T19:10:42.123456Z||080A0112|Trace common message|||trace|rexroth-automationcore|
common.log.trace|manager.cpp|setEnablingStateTraceUnit|211|Set enabling state for trace unit
'common.log.diagnosis' (Messages: true, warnings: true, errors: true).
```

11.12.2 Metadata

In case of a *journald* entry, different metadata can also be set to allow filtering via a specific metadata for example.

When issuing a *journald* entry, the *journald* system sets specific system metadata by default.

Additionally, some system metadata is adjusted when issuing a ctrlX CORE log entry if the respective information was specified.

Additional Rexroth metadata was specified. It starts with the prefix "R_".

Metadata can for example be retrieved in the JSON Pretty format via:

```
sudo journalctl --namespace=* -r R_ORIGIN=rexroth-
automationcore -o json-pretty
```

The ctrlX CORE system adjusts the following metadata or it is defined for use:

Metadatum	Meaning	Example of a diagnostics	Example of a trace		
MESSAGE	Contains the complete message format of a log.	"MESSAGE" : "2020-07-31T19:10:42.123456Z 101 080F0100 Error in diagnosis system 00666006 Main diagnosis code is not registered diagnosis/set rexroth-automationcore common.log.diagnosis handler.cpp logThread 600 Not registered main diagnosis code 0x080F01FFF"	"MESSAGE" : "2020-07-31T19:10:42.123456Z 080A0901 Trace common message trace rexroth-automationcore common.log.trace manager.cpp setEnablingStateTraceUnit 211 Set enabling state for trace unit 'common.log.diagnosis' (Messages: true, warnings: true, errors: true)."		
PRIORITY	Defines the priority and the respective diagnostic class of the log.	"PRIORITY" : "3"	"PRIORITY" : "6"		
	Possible values:				
	Definition			Value	Class
	LOG_INFO			6	0xA
	LOG_WARNING			4	0xE
LOG_ERR	3	0xF			
R_ENTRY_TYPE	Defines that this is a log entry of the Rexroth ctrlX CORE system.	"R_ENTRY_TYPE" : "rexroth_diag"	"R_ENTRY_TYPE" : "rexroth_trace"		

Metadatum	Meaning	Example of a diagnostics	Example of a trace
R_TIMESTAMP	Contains the time stamp when issuing the log in the ctrlX CORE diagnostic system. Format: Microseconds since EPOCH UTC. Depends on the set system time of the control.	"R_TIMESTAMP" : "1593076853184818" "	"R_TIMESTAMP" : "1593076853184818" "
R_USER_ID	Corresponds to <i><User ID></i> .	"R_USER_ID" : "101"	"R_USER_ID" : ""
R_DIAG_MAIN	Corresponds to <i><Main diagnostic number></i> .	"R_DIAG_MAIN" : "080F0100"	"R_DIAG_MAIN" : "080A0112"
R_DIAG_DETAILED	Corresponds to <i><Detailed diagnostic number></i> .	"R_DIAG_DETAILED" : "00666006"	"R_DIAG_DETAILED" : ""
R_ENTITY	Corresponds to <i><Entity></i> .	"R_ENTITY" : "diag-nosis/set"	"R_ENTITY" : "trace"
R_ORIGIN	Corresponds to <i><Origin></i> .	"R_ORIGIN" : "rexroth-automation-core"	"R_ORIGIN" : "rexroth-automation-core"
R_UNIT	Corresponds to <i><Component name></i> .	"R_UNIT" : "common.log.diagnostics"	"R_UNIT" : "common.log.trace"
CODE_FILE	Corresponds to <i><File name></i> .	"CODE_FILE" : "handler.cpp"	"CODE_FILE" : "manager.cpp"
CODE_FUNC	Corresponds to <i><Function name></i> .	"CODE_FUNC" : "logThread"	"CODE_FUNC" : "setEnablingStateTraceUnit"
CODE_LINE	Corresponds to <i><Line number></i> .	"CODE_LINE" : "600"	"CODE_LINE" : "211"

12 How to

12.1 Cleaning up ctrlX OS and updating to a new software release (backup)

This "How to" describes how to update a device to a new software release and how to apply the device settings including app configuration.

The following scenarios are taken into account:

- **Scenario 1 (Clean up device and update software)**
The device is updated to a new software version by manually installing the apps.
- **Scenario 2 (Clone device)**
The ctrlX device is cleaned up, updated to the current image version and app versions, settings and configuration are transferred from a device backup (template).
A backup is created on one device as template and applied to other devices.
- **Scenario 3 (Automated device cloning)**

Prerequisite

The ctrlX device has to be equipped with at least ctrlX OS version XCR-V-0120.

The backup functionality in the device is not available for devices with a version lower than XCR-V-0120.

12.1.1 Scenario 1: Clean up device and update software

If the device software is upgraded to a new major version (e.g. from XCR-V-0120 to XCR-V-0206), it is recommended to clean up the ctrlX device as described in the following instructions.

The boot loader and the system apps are updated.

Internal memory areas are also reset.

Handling instruction

Save the settings and configuration of the device (create backup)

1. To restore the device settings and the device configuration, save the settings before using the backup functionality, see: [Backup – Creating from ctrlX device](#).

When creating the backup, use a password for the complete transmission of encrypted content.

The password is required later when applying the backup.

Download system image and prepare the external data carrier

2.  An external storage medium is required for the image update: [SD cards and USB support](#).
The ctrlX CORE X7 device type only supports the USB connection by the XF01C slot.

- **Step 1:**

Download the system image as ZIP archive from the [Rexroth Collaboration Room](#) (note that the images are device-specific).

- **Step 2:**

Format the data carrier with FAT32 file system.

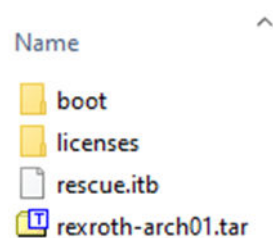
- **Step 3:**

Clean up the data carrier (SD card or USB memory stick). In particular, temporary backup files and the image update may not be available.

- **Step 4:**

Apply the files from the ZIP archive of the image.

- ➔ The content of the external data carrier should look as follows:



Executing image update

- The image update cleans up the device and updates the system apps to the image version.
- An external storage medium is required for the image update.
- Do not switch off the ctrlX device during the update.
- The licensing information is retained during an image update. However, new licenses might be required to use newer app versions.

3. ➤• **Step 1:**

Disconnect the ctrlX device from voltage.

• **Step 2:**

Insert the external data carrier into the disconnected ctrlX device (for the ctrlX CORE X7 device type, the USB connection is only supported by the XF01C slot).

• **Step 3:**

Switch on the ctrlX device.

• **Step 4:**

The device installs the new image. The process takes approximately 5 minutes.

- ➔ The current image update is signaled via the LED (blue-black). The display deviates briefly during required boot processes.
- ➔ The factory settings are active after the image update. Note that the default admin user "boschrexroth" is active.

Installing apps on the ctrlX device**4. ➤**

The new versions of the apps are available via the download portal.

Alternatively, apps are also available in the [➔ Rexroth Collaboration Room](#), under: *"Start → ctrlX AUTOMATION → ctrlX CORE APPS Releases"*.

For general information on installing apps on ctrlX devices, see: [➔ Reinstalling apps](#).

Importing settings and configuring the device**5. ➤**

Prerequisites:

- You are logged in as admin on the ctrlX device (default user: boschrexroth)
- If a password was used when creating the backup, it has to be entered before it is applied.

Apply the backup created in step 1 to the ctrlX device, see: [➔ Backup – Applying to ctrlX device](#).

After applying the settings, access the device settings under the network settings and the user settings saved in the backup file.

12.1.2 Scenario 2 (Clone device)

Create a backup file first and then load the apps, device settings and device configuration.

Note that a backup file can be installed automatically via an external data carrier together with an image update.

Handling instruction

Save the settings and configuration of the ctrlX device to be cloned (create backup)

1. ➤ The apps and device settings to be transferred to the target device are included in the backup file, see: ➔ [Backup – Creating from ctrlX device](#). System-related apps can be explicitly selected via the “Extended view” if required.

When creating the backup, use a password for the complete transmission of encrypted content.

The password is required later when applying the backup.

Download system image and prepare the external data carrier

2. ➤ ⓘ An external storage medium is required for the image update: ➔ [SD cards and USB support](#). The ctrlX CORE X7 device type only supports the USB connection by the XF01C slot.

- **Step 1:**

Download the system image as ZIP archive from the ➔ [Rexroth Collaboration Room](#) (note that the images are device-specific).

- **Step 2:**

Format the data carrier with FAT32 file system.

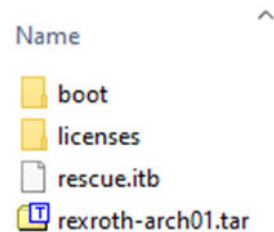
- **Step 3:**

Clean up the data carrier (SD card or USB memory stick). In particular, temporary backup files and the image update may not be available.

- **Step 4:**

Apply the files from the ZIP archive of the image.

- ➔ The content of the external data carrier should look as follows:



Executing image update on the ctrlX target device (apply backup)

- The image update cleans up the ctrlX target device and updates the system apps to the image version.
- An external storage medium is required for the image update.
- Do not switch off the ctrlX target device during the update.
- The licensing information is retained during an image update. However, new licenses might be required to use newer app versions.

3. ➤• **Step 1:**

Disconnect the ctrlX target device from voltage.

• **Step 2:**

Insert the external data carrier into the disconnected ctrlX target device.

• **Step 3:**

Switch on the ctrlX target device.

• **Step 4:**

The device installs the new image. The process takes approximately 5 minutes.

- ➔ The current image update is signaled via the LED (blue-black). The display deviates briefly during required boot processes.
- ➔ The factory settings are active after the image update. Note that the default admin user "boschrexroth" is active.

Importing apps, settings and configuration for the template**4. ➤****Prerequisites**

- You are logged in as admin on the ctrlX target device (default user: boschrexroth).
- If a password was used when creating the backup, it has to be entered before it is applied.

Apply the backup created in step 1 to the ctrlX target device, see:

➔ [Backup – Applying to ctrlX device.](#)

After installing the apps and applying the settings, the device settings can be accessed under the network settings and the user settings set in the backup file.

Make individual device settings**5. ➤**

In the last step, make the individual device settings of the device, e.g. network settings, device name, user settings.

Some files are stored device-specifically on the data carrier via the serial number.

For the serial number of the device, go to Settings/Information/Type plate.

For the history of the image update, go to the file <Serial number>\flash.log on the external data carrier.

Repeated execution of the image update on this device is prevented by a marker file on the external data carrier: <Serial number>\.done.

12.1.3 Scenario 3: (apply external data carrier to device)

A ctrlX device can automatically import an image update and then apply a backup.

The image and backup have to be provided on an external data carrier for this purpose.

Handling instruction**Save the settings and configuration of the device (create backup)**

1. ➤ To restore the device settings and the device configuration, save the settings before using the backup functionality, see: ➔ [Backup – Creating from ctrlX device](#).

When creating the backup, use a password for the complete transmission of encrypted content.

The password is required later when applying the backup.

Download system image and prepare the external data carrier

2. ➤ ⓘ An external storage medium is required for the image update: ➔ [SD cards and USB support](#).

The ctrlX CORE X7 device type only supports the USB connection by the XF01C slot.

- **Step 1:**

Download the system image as ZIP archive from the ➔ [Rexroth Collaboration Room](#) (note that the images are device-specific).

- **Step 2:**

Format the data carrier with FAT32 file system.

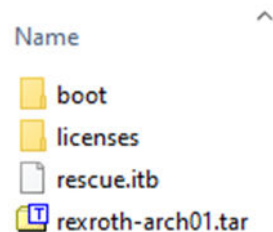
- **Step 3:**

Clean up the data carrier (SD card or USB memory stick). In particular, temporary backup files and the image update may not be available.

- **Step 4:**

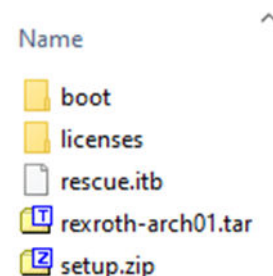
Apply the files from the ZIP archive of the image.

- ➔ The content of the external data carrier should look as follows:

**Copy the backup to the data carrier**

3. ➤ Copy the backup created in step 1 to the data carrier.

- ➔ Note: The backup file has to be on the external data carrier under the name "setup.zip":



Executing image update



- The image update cleans up the device and updates the system apps to the image version.
- An external storage medium is required for the image update.
- Do not switch off the ctrlX device during the update.
- The licensing information is retained during an image update. However, new licenses might be required to use newer app versions.

4. ➔

• **Step 1:**

Disconnect the ctrlX target device from voltage.

• **Step 2:**

Insert the external data carrier into the disconnected ctrlX device.

• **Step 3:**

Switch on the ctrlX device.

• **Step 4:**

The device installs the new image. The process takes approximately 5 minutes.

➔ The current image update is signaled via the LED (blue-black). The display deviates briefly during required boot processes.

➔ The factory settings are active after the image update. Note that the default admin user "boschrexroth" is active.



Some files are stored device-specifically on the data carrier via the serial number.

For the serial number of the device, go to Settings/Information/Type plate.

For the history of the image update, go to the file <Serial number>\flash.log on the external data carrier.

Repeated execution of the image update on this device is prevented by a marker file on the external data carrier: <Serial number>\.done.

13 User interface

13.1 Navigation

13.1.1 ctrlX OS web interface – Overview

The web interface of the ctrlX CORE device consists of three sections:

- ① **Header**
- ② **Workspace**
- ③ **Side navigation**

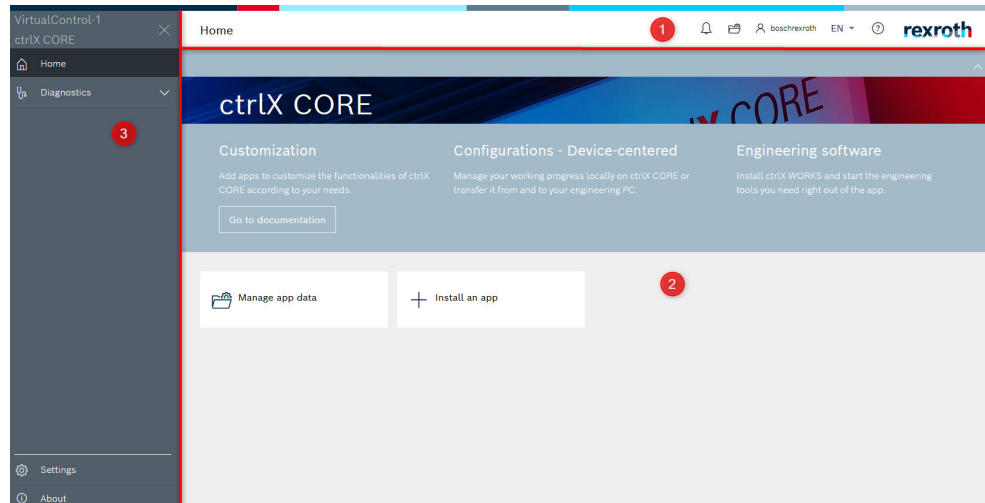


Fig. 70: ctrlX OS web interface (start page "Home")

① Header

The header provides the following functions:

- Shows the status of pending device diagnostics.
The button opens the logbook or the “Alarms” tab with the currently pending messages, see:
➔ **Tab – “Alarms”**
 - Pending errors are shown with a red symbol and the number of errors
 - Pending warnings are shown with a yellow symbol and the number of warnings
- Opens the window “Manage app data” that shows and manages active app data, see ➔ **Window – “Manage app data”**
- Symbolizes your user account and provides the following functions:
 - “Change password”
Opens a dialog to change the password ➔ **Documentation**
 - “Log Out”
Option to log out of the device
- **DE ▾/EN ▾**
GUI language can be selected, currently available in **DE ▾** (German) and **EN ▾** (English)
- Opens a menu with the following information:
 - “Application manual”
Opens the respective documentation in the browser
 - “Current page help”
Opens the help on the currently shown window
 - “How-tos”
Opens the "How-to" with information on the ctrlX AUTOMATION in the browser, such as FAQs, examples, technical notes or how-tos
 - “Forum”
Opens the ctrlX AUTOMATION forum in the browser
 - “API reference”

- Opens the “API reference” window
-  “GitHub”
Opens the GitHub platform in the browser
-  “ctrlX OS system report”
Refer to [↗ Creating system report in ctrlX OS](#)

② Workspace



The sites shown in the workspace are called “window” in the documentation.

The main work affects the ctrlX CORE device, e.g.:

- Creating/enabling/saving the control configuration
- Installing and managing apps
- Configuring field bus communication (e.g. EtherCat master)
- Creating and configuring axes and kinematics
- Opening engineering tools for the PLC project planning and configuration, e.g. "PLC Engineering"
- Retrieving and acknowledging diagnostics

③ Side navigation



The side navigation is used to navigate in the ctrlX OS web interface and can be hidden via .

By installing apps, further entries can be added to the side navigation if the respective app contains an entry.

The following entries are included in the side navigation upon delivery:

- “Home”
Start page of the ctrlX OS web interface, see [↗ Documentation](#)
- “Diagnostics”
Contains the diagnostic system of the ctrlX CORE device:
 - “Logbook”
Contains all diagnostic messages of the ctrlX CORE device, see [↗ Documentation](#)
- “Settings”
Contains submenus with general system information and centralized setting and configuration options, see
- “About ctrlX OS”
General information:
 - Showing installed apps and their versions (link to the “Apps” window)
 - Information on the manufacturer
 - Included open source components
 - General terms and conditions
 - Option to create a system report

13.2 Windows

13.2.1 Window – “Home”

The window is the start page of the ctrlX OS web interface and provides a configurable dashboard to quickly display relevant information to the user and enable quick navigation in the web interface.

The information displayed is shown as widgets in the dashboard provided by the ctrlX system as well as by additionally installed apps (if supported).

The dashboard initially displays all available widgets by the ctrlX system by default upon delivery. You can configure the dashboard by removing, adding or moving widgets.

Configuring dashboard

1. ➤ To configure the dashboard, click on  at the bottom of the dashboard.
 - ➔ The dashboard then switches to editing mode.
2. ➤ In editing mode, widgets can be moved, removed from the dashboard or more widgets can be added.
3. ➤ To add more widgets to the dashboard, click on [Add widgets] at the bottom of the dashboard.
 - ➔ A dialog to select the widgets opens.
4. ➤ Select the widgets to be displayed in the dashboard and confirm the dialog.
 - ➔ The widgets are added to the dashboard and can be positioned as desired.
5. ➤ Complete the configuration by exiting the editing mode with “Save”.

Window call

In the side navigation the ctrlX OS web interface: “Home”

Further topics

➔ [ctrlX OS web interface – Overview](#)

➔ [Apps – Basics](#)

➔ [Configurations – Introduction](#)

13.2.2 Window – Logbook

The window is used to display pending alarms and the history of all logbook messages accumulated up to now.

The header is located at the top of the window and can be used to configure logbook settings or a remote logging server, see description below.

Below the header, the logbook messages are displayed as table via the tabs “Alarms” and “History”:

- Tab “Alarms”
Displays currently pending alarms, see:
➔ [Tab – “Alarms”](#)
- Tab “History”
Displays the history of all log messages accumulated up to now, see:
➔ [Tab – “History”](#)

Standard/expert view

The logbook has two views to customize the displayed information content for different user groups:

- Standard view (e.g. suitable for device commissioning)
- Developer view (display of extended diagnostics and filters, trace messages, ...)

The view can be configured in the “Settings” dialog, see:

➔ [Dialog – “Settings” \(Logbook\)](#)

Call:ctrlX OS side navigation “*Diagnostics → Logbook*”**Window elements**

GUI element	Description
“Header”	Buttons: • “Remote Logging” To configure a remote logging server, see: ↗ Window – Remote Logging •  Opens the “Settings” dialog, see: ↗ Dialog – “Settings” (Logbook) Setting options: - Enabling/disabling developer view - Setting logbook to persistent (log entries are saved permanently) - Configuring time zone settings
Tab “Alarms”	Display of currently pending messages, see: ↗ Tab – “Alarms”
Tab “History”	Display of the history of all messages accumulated up to now, see: ↗ Tab – “History”

Related topics[↗ Logbook – Overview](#)[↗ Logbook – Exporting](#)[↗ Remote Logging – Configuring external server](#)**13.2.3 Window – Remote Logging**

To configure connections to the remote Syslog servers (Syslog), go to the window. "Syslog" is a protocol to transmit logbook entries to one or multiple Syslog logging servers.

Copies of all logbook entries are made for each configured and activated server connection. Copies are made according to the configuration of the message filter. The copy of a logbook entry is transmitted to the active server connections immediately after the entry into the logbook of the ctrlX device.

The configured connections and message filters on the remote logging server are saved in the app data in the folder "remote-logging".

For more information on the "Syslog" topic, refer to: [↗ https://wikipedia.org/wiki/Syslog](https://wikipedia.org/wiki/Syslog)

Related topics[↗ Remote Logging – Configuring external server](#)[↗ Logbook – Overview](#)[↗ Dialog – “Add remote logging server”](#)**Call:**

In the ctrlX OS web interface:

ctrlX OS “side navigation → Diagnostics → Logbook → In the header, click on “Remote Logging””

Window elements

No server connection has been set up upon delivery. To create a new server connection, select the  button.

Once one or multiple server connections have been configured on the ctrlX device, the representation is modified in the window and the server connections are listed in a table. To add more server connections, select the  button in the upper window section.

GUI element	Description
Name	Symbolic name of the server connection
Host / IP address / domain	IP address of the server or the host PC
Port	Port of the server or the host PC
Protocol	Set network protocol: • TCP • UDP • TLS
Enabled	 : Server connection disabled  : Server connection enabled
State	Graphic display of the connection status to the Syslog server
Actions	→ Only available if the server connection is disabled  : Opens the editor to modify the server settings and to configure the message filter, see: ➔ Editor – “Remote Logging”  : Removes the server connection

13.2.4 Window – “Manage app data”

The active settings of the ctrlX device are displayed and managed in the window “Manage app data”. The settings are also called “App Data”.

Use cases:

- **Backup**
Saving active app data to the archive of the ctrlX device.
- **Restoration**
Loading the saved configuration from the archive to the ctrlX device.
- **Download**
Downloading active app data or configuration from the archive of the ctrlX device to an engineering PC (zip file).
- **Upload**
Uploading a configuration saved on the engineering PC to the current app data or to the archive of the ctrlX device.

Related topics:

➔ [Configurations – Introduction](#)

Window design

The window is divided into two sections:

- Upper section “Active”
Shows the settings enabled in the apps.
- Lower section “Archive”
Contains backed up configurations kept in the file system of the ctrlX device.



For a better overview, the sections can be expanded and collapsed via buttons in the table headers:

- Collapse [^]
- Expand [v]

“Active section”

The “Active” section provides an overview on the active app data of the ctrlX device. The overview consists of list entries containing specific directory structures that depend on the category. The number of list entries depends on the configuration of the ctrlX device and changes dynamically by installing additional apps that implement specific settings for example. Changes and supplements in the settings resulting during the project planning and configuration are updated and saved automatically.

The section provides the following actions:

- ➔ Viewing active app data on file level
- ➔ Comparing active app data with configurations from the archive
- ➔ Saving active app data to the archive of the ctrlX device
- ➔ Transferring active app data to the engineering PC
- ➔ Loading app data from the engineering PC
- ➔ Resetting active app data
- ➔ Editing description of a configuration
- ➔ ViewingLoading report

“Archive section”

The “Archive” section shows app data saved as configuration in the archive of the ctrlX device (empty upon delivery).

The section provides the following actions:

- ➔ Viewing configuration in the archive
- ➔ Comparing configuration with other configurations from the archive or with active app data
- ➔ Loading configuration from the archive to the ctrlX device
- ➔ Transferring configuration from archive to engineering PC
- ➔ Uploading from the engineering PC to the archive of the ctrlX device
- ➔ Copying configuration to the archive
- ➔ Editing description of a configuration
- ➔ Deleting configuration in the archive



When loading saved configurations to the ctrlX device, the active app data is replaced by settings of the loaded configuration.

Configurations can only be loaded or saved completely!

Call:

There are two options to open the “Manage app data” window:

- Via the  button in the header of the ctrlX OS web interface
- Via the ctrlX OS side navigation in the window *"Home"* → *tile "Manage app data"*

Elements of the "Manage app data" window

GUI element	Description
Section "Active " (Display of the active app settings)	The "Active" section contains a command bar and lists the configuration elements in the subsection "Content" Functions of the command bar •  "Save to archive" Saves the active settings in the apps to a new or an existing configuration (the content of an existing configuration is replaced while saving). The saved configuration is shown in the "Archive" section. •  ... Opens the Context menu using the following commands: -  Upload Uploading a configuration from the engineering PC to the ctrlX device. Active app data is overwritten. -  Download Saves active app data as configuration to the local engineering PC. All data is zipped to a .zip archive. -  File view App data is grouped in directories. In the file view, you can navigate between the directories and view and download the files, see: ➡ -  Compare Compares active app data with a configuration from the archive. Differences are displayed in a table. It can be navigated in subfolders and files of the configurations. -  Loading report Display of the last loading results The display includes: - Loading date and loading time - Name of the loaded configuration - Link to logbook - Successfully loaded data - Data that could not be loaded The menu item "Loading report" is only displayed if a configuration was loaded to the apps. -  Edit Opens the editor to change the "Description" -  Reset Causes a reset of the current app settings by loading an empty configuration to the ctrlX device.

GUI element	Description
	“Content” section Lists all elements included in the configuration. The displayed elements can be opened to reach specific settings. The settings are grouped in directories. You can navigate between the directories as well as view and download the files.
Section “Archive” (Display of the saved configurations)	The “Archive” section consists of a command bar and a tabular list of all saved configurations. Function of the command bar:  Upload Uploads a configuration from the local engineering PC. It is uploaded to a new or an existing configuration. Elements of the table: “Name” Name of the configuration. “Description” Description on the configuration (optional). “Modified” Date and time of the last change. “Actions”  Loads the configuration Thus, the configuration is loaded to the apps. Acknowledge a confirmation prompt before the execution. The loading result is shown in the loading report in the “Active” section.  Downloads the configuration Saves the configuration to the local engineering PC. All files of the configuration are zipped to a .zip archive.  More actions Opens the Context menu using the following subcommands:  File view Display of the content of the archived configuration, see:   Compare Compares the configuration with another configuration from the archive or with the active configuration. Differences are displayed in a table. It can be navigated in subfolders and files of the configurations.  Edit Opens the editor to change the “Name” and the “Description”.  Copy Copies the selected configuration to another saved configuration or to a new configuration.  Delete Deletes the selected configuration Acknowledge a confirmation prompt before the execution.  Info Shows which user saved the configuration on which device, which apps were saved in which version and which app versions are installed.

Windows – Manage app data > File view

The "File view" in the "Manage app data" window provides an extended display of the directories and files of active app data or in archived configurations:

- **Display of active app data**

Call the File view in the header of the window section Active via: [...] → []

The file view of the active app data changes dynamically by installing additional apps that implement specific settings for example.

Changes and supplements in the settings resulting during the project planning and configuration are updated and saved automatically.

- **Display of app data in archived configurations**

Call the File view in the window section Archive in the respective configuration column via: [...] → []

Use cases:

- **View**

Displaying the content of directories, subdirectories and files.

- **Upload**

Uploading a file saved on the engineering PC to the displayed configuration of the ctrlX device.

If it is a zip file, it can be uploaded as file or unzipped.

- **Create directory or file**

Adding empty directory or empty file.

- **Download**

Downloading a configuration directory or file to an engineering PC.

- **Move / Copy**

Inserting individual files and directories elsewhere.

- **Compare**

Comparing individual files or directories with each other (e.g. axis settings).

The File view consists of a command bar, a navigation bar and a tabular listing of directories and files that contain specific directory structures, depending on the category of app data to be displayed. The number of list entries depends on the configuration of the ctrlX device..

Elements of the File view

GUI element	Description
Command bar	• Name of the displayed configuration "Active" when the content of active app data is displayed. • [] Loads the configuration to the ctrlX device (only for archived configurations) • [] Compares the configuration with other configurations. • [] Shows details > Column "Mode" is displayed. [] Hides details from > Column "Mode" is hidden. • [] Uploads a file or file content from the engineering PC to the open directory. • [] Adds an empty directory or an empty file to the open directory. • [] View is refreshed.

GUI element	Description
Navigation bar	 (Root directory) Depending on the navigation, the respective path is displayed. To navigate back, click on the path.
Table	• “Name” Name of the directory or the file. • “Mode” Mode of the directory or file (hidden by default, see "Show details" above). • “Modified” Date and time of the last change of a file. • “Size” Size of a file. • “Actions” –  "Download" Downloads the directory content or the file to the engineering PC –  More actions Opens the context menu using the following subcommands: ■ Open ■ Compare (only for files) ■ Rename ■ Copy ■ Move ■ Delete

Windows – Manage app data > Configuration comparison

The differences between two configurations from the archive or between an archived configuration and the active app data are displayed in the window. The comparison can be used to compare files and directories of two configurations (e.g. axis settings).

Window structure

The window consists of a command bar, a display of the configuration names and a tabular list of the differences.

GUI element	Description
Command bar “Configuration comparison”	•  View is refreshed.

GUI element	Description
Configuration comparison	The top section of the configuration comparison shows which data sets are compared: • []: Name of the configuration from which the comparison was started (grayed out = entry cannot be changed). • []: Name of the second configuration (click on  to change the entry). • []: Closes the configuration comparison. • [...] differences: Number of differences found. • []: Navigation bar / root directory Depending on the navigation, the respective path is displayed. To navigate back, click on the path. • Table column [Left]: Directories and files of the first configuration. Click on an entry to display the corresponding comparison or opens a subdirectory. The symbol [] indicates whether an element is equal or unequal (unequal elements are highlighted in color). • Table column [Right]: Directories and files of the second configuration. Click on an entry to display the corresponding comparison. To show the differences between the directories or files, click on the comparison line. The relevant file path is displayed in the navigation bar.

13.2.5 Window – “Apps”

The “Apps” window manages apps on the ctrlX device and provides a better overview. New apps can be installed on the ctrlX device, installed apps can be updated, uninstalled or disabled.

The app overview of the window provides support during these actions. The app overview shows all apps installed on the ctrlX device or saved as "stock" in the internal app storage of the device (not installed).

If the ctrlX device is connected to the ctrlX Store, the app shows additionally those apps provided as web download from the ctrlX Store, see [➔ ctrlX Store in the web](#)

For an overview on the available apps:

[➔ Licenses – Overview](#)

Possible actions in the window

- ➔ Uploading apps to the app memory of the ctrlX device
- ➔ Installing apps on the ctrlX device from the device-internal app storage
- ➔ Installing apps from the engineering PC on the ctrlX device
- ➔ Installing apps on the ctrlX device from the ctrlX Store
- ➔ Updating apps
- ➔ Changing app version
- ➔ Disabling apps
- ➔ Enabling apps
- ➔ Downloading apps from the ctrlX device
- ➔ Uninstalling apps
- ➔ Reinstalling apps

[↪ Deleting apps from app storage](#)**Dependency indicator**

When installing apps on a ctrlX device, so-called dependency conflicts can result that can lead to the installation being canceled or to limited operation of an installed app. Such a conflict can be caused due to a missing license, missing required apps or unfulfilled system requirements for example. In this case, the integrated dependency indicator of the ctrlX device diagnoses the cause and provides clear information on the existing dependency.



Dependencies of already installed apps are displayed in the app detail view in the “Dependencies” tab, see: [↪ Extended app overview](#)

Call

There are two options to open the “Apps” window in the side navigation:

- In the window “Home → click on [+ install an app]”
- In the window “Settings → click on [Apps]”

Window description

The window “Apps” is divided into two sections:

- **Command bar** (below the status bar)
- **App overview** (below the command bar)

Command bar

The command bar provides the following functions and commands to use apps:

GUI element	Description
	The symbol indicates that the ctrlX device is not connected to the ctrlX Store. In this case, no apps are provided in the app overview for download from the ctrlX Store.
	Search field for keyword search in the app overview. Useful if multiple apps are shown, e.g. if no display filter was set and apps from all sources are shown in the overview (installed apps, saved apps, apps in the ctrlX Store).
	Display filter with the following options • “All”: All available apps are shown in this setting (installed apps, saved apps, apps in the ctrlX Store) • “Installed apps”: Only apps installed on the ctrlX device are shown. • “App storage”: Only apps saved in the app storage on the ctrlX device are shown (not yet installed). • “ctrlX OS Store”: Only apps for the device provided for download in the ctrlX Store are displayed. • “Unlicensed apps”: Only apps without available license are shown.

GUI element	Description
 Show system apps Alternatively  Hide system apps	Buttons to show/hide system apps (hidden in the default setting). System apps are part of the ctrlX OS Runtime and cannot be uninstalled or disabled. This is also symbolized by the "lock symbol" in front of the app name. The display of the system apps can for example also be used to check whether updates are available. System apps can be updated with the respective user rights.
 Install from file	Apps can be installed directly as file from the engineering PC to the connected ctrlX device, e.g. if the ctrlX device and the ctrlX Store are not connected. The button starts the initialization. First, the file browser opens to select the app files on the engineering PC. Advantage: Multiple apps can be installed at the same time.
 App storage	Apps can be saved in the device-internal app storage on the ctrlX device without installing the app on the device. This can be reasonable, but note that memory space is allocated on the internal ctrlX device memory. Select the button to show all apps in the app storage of the ctrlX device (empty upon delivery). Use cases: • To prepare for the commissioning of the ctrlX device. • Storing apps on the ctrlX device and installing them when required, e.g. to enable optional machine functions.
 Update all apps	The system checks whether there are updates for the installed apps via the button.
 Update UEFI Bios	Perform UEFI BIOS update, see: ➔ Updating the UEFI BIOS of the ctrlX device Only displayed if a BIOS update is available for the ctrlX device.
 Upgrade ctrlX OS	Upgrade the ctrlX OS basis system. Only available if multiple prerequisites are met, see: ➔ Upgrading the ctrlX OS basis system
 Settings	The button opens the "Settings" dialog to configure the options for the app installation and the app storage, see: ➔ Chapter 13.4.9 Dialog – "Settings" on page 348

App overview

The app overview shows all apps either installed on the ctrlX device or ready for installation.

The app overview is divided into two categories:

- **Installed apps**
Shows all apps already installed on the ctrlX device
- **Available apps**

Shows apps provided for the ctrlX device for the installation.

The following sources are considered:

- Apps saved in the device-internal app storage of the ctrlX device
- Apps provided in the ctrlX Store to be downloaded for the ctrlX device (only if there is an active connection to the ctrlX Store).

App representation in the app overview

Apps are shown as tiles in the app overview including additional information and interactive buttons:

- App version
- App name
- Brief app description
- Apps that are not yet licensed have a license symbol:
 - **Red icon**
The required license to operate the instance is not available
 - **Yellow symbol**
Indicates that the app provides internal options which are not yet licensed, see:
[↗ Chapter 6.3.1 Licenses – Overview on page 53](#)
- System apps are labeled with a lock symbol (locked for uninstalling and disabling)
- Interactive buttons (depending on the app status):
 - Installing app
 - Uninstalling app
 - Enabling app

Extended app overview

To show an extended app view with additional information and actions, click on one of the app tiles in the overview:

- **Tab “Description”**
Brief description on the app and its function (if supported by the app).
- **Tab “Dependencies”**
Shows requirements and permissions required to operate the app, as well as dependencies from or to other apps on the ctrlX device indicating the required app software version (min/max).
- **Tab “Available licenses”**
Shows the current licensing status on the ctrlX device:
 - License to operate the app if required
(licenses already installed on the ctrlX device have a green check)
 - If available, optional "In-app licenses" for the functional extension
(unlicensed in-app options are labeled with a yellow symbol)

Possible actions in the extended app overview (depending on the app status):

- Disabling app
- Enabling app
- Restarting app (the action is only available if the app contains services that can be restarted)
- Uninstalling app
- Updating app
- Reinstalling app
- Downloading app
- Licensing

The displayed button opens the “Licenses” window. In this window, use the Rexroth licensing center to manage the app licenses and to transfer them to the target devices, see:

➔ [Window – “Licenses”](#)

To close the extended view, select the [X] button.

Updating the UEFI BIOS of the ctrlX device



Depending on the hardware architecture, this functionality may not be available, for example on ctrlX devices with ARM architecture that do not have a UEFI BIOS.

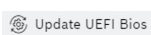
The UEFI BIOS of the ctrlX device can be updated if required.

New versions of the UEFI BIOS are published in the Hardware Support app.

The current version of the UEFI BIOS of the ctrlX device is displayed in the web interface in the information on the electronic type plate, see:

“side navigation → Settings → window System Information → widget Typeplate”

Performing UEFI BIOS update

1. ➔ Switch the ctrlX device to the "Service" operating state
2. ➔ Install the Hardware Support app with the new UEFI BIOS
3. ➔ In the header of the window “Apps”, click on  .
 - ➔ You are logged out and the update is started.
 - ➔ The update takes a few minutes. Switch on the device during the update!
4. ➔ After completing the upgrade, you are automatically redirected to the login screen.

Upgrading the ctrlX OS basis system



The ctrlX OS basis system is the part of ctrlX OS responsible for the installation and execution of apps and services. It provides libraries, runtime environments and other dependencies required to execute apps. It is provided by the "coreXX" app (e.g. "core20"). This is a system app and can therefore not be uninstalled.

The basis system can be updated. The following prerequisites have to be met:

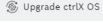
- The new basis system app version to be used as future basis system has to be installed on the ctrlX device before the upgrade.
- All system apps have to be installed in the latest version.

System apps:

- Automation Core
- bare
- Device Admin
- Hardware Support
- Realtime Kernel
- Setup
- snapd (min 2.61.2)
- Solutions
- Version Guard

Performing OS basis system upgrade

1. ➔ Switch the ctrlX device to the "Service" operating state

2.  In the header of the window “Apps”, click on .
 - ➔ You are logged out and the upgrade is started.
 - ➔ The upgrade takes a few minutes. Do not switch off the device during the upgrade!
3.  After completing the upgrade, you are automatically redirected to the login screen.
4.  The new OS basis system app is now active.
 - ➔ The old OS basis system app can be uninstalled from the ctrlX device.

13.2.6 Window – “Licenses”

Operating apps on ctrlX devices is usually subject to license conditions.

The “Licenses” window is used to manage licenses on the ctrlX device or on a license dongle.

To detect a license dongle, the ctrlX device has to be restarted after plugging in the dongle.

Related topics

➔ [Licenses - Overview](#)

➔ [Licenses in the ctrlX OS STORE](#)

Call:

ctrlX OS side navigation “*Settings* → *Licenses*”

Further information

- ➔ [Chapter Tab – “Licenses” on page 284](#)
- ➔ [Chapter Tab – “Sources” on page 288](#)

Tab – “Licenses”

Call:

ctrlX OS side navigation “*Settings* → *Licenses*” ➔ Tab “Licenses”

Function:

In the tab “Licenses”, licenses available on the license sources are listed in a table. The table can be sorted according to the columns “Product”, “App”, “Source” and “Expires (UTC)”.

If there are not yet any licenses or no license dongle on the device, the following buttons are shown instead of the overview table:

- : Link to open the license center
- : Uploading the license file to the ctrlX device or the plugged-in license dongle
- : Opens the settings (enabling temporary usage rights)

Functions in the window

• License overview

If licenses are included on the device or if there is a plugged-in license dongle, they are displayed in the “Licenses” tab in an overview table with the license source.

Additional information on the assignment to the apps and on the period of validity is provided.

If a license dongle is plugged-in, a banner informs on the site.

If the dongle is removed at runtime, it is shown with a warning icon on a banner on the site. Additionally, a warning is output immediately (E0550) followed by a multi-level error output. The licenses on the dongle can then not be used anymore.

If the ctrlX device is restarted without the license dongle being plugged-in, only the licenses locally available on the device are shown in the overview table in the “Licenses” tab and only these can be used.

- **Connection to the “licensing center”**

Use the  button to access the “licensing center”.

The “licensing center” is used to manage and to assign licenses to target devices.

Sequence:

After purchasing licenses in the ctrlX OS STORE, the licenses are provided in the “licensing center” and can be assigned to the respective target device.

After the assignment, the license file can be transferred to the engineering PC and subsequently to the target device.

It is also possible to assign several licenses to one operation.

- **License upload to the target device or the license dongle**

After the license file has been generated in the “licensing center” and after download to the PC, the license file can be transferred to the target device with . If a license dongle is plugged in at the ctrlX device, the license file created for the dongle was transferred to this dongle.

If it is tried to upload an older license file with licenses differing from those currently on the license source, an HTTP client error 080E0200/0C7A0202 message is displayed. When uploading a current license file, a message window displays the changed licenses compared to the current state on the license source.

- **Licenses with expiry date**

The user is automatically notified at set intervals when a license expires. There are notification intervals of 1 month, 1 week and 1 day before the license expires.

The diagnostic bell is set to active ("A message is pending icon") and the device LED flashes yellow. The respective warning is entered into the log-book. In the license overview list, a warning icon is displayed next to the license in the “Expires (UTC)” column.

The warning can be confirmed (diagnostic bell). Thus, the icon next to the diagnostic bell disappears (a message about an active alarm is not pending anymore) and the device LED changes back to its previous state and remains.

Exceptions:

- If, for example, the "1-month warning" is pending and confirmed, it is no longer displayed. The "1-week-warning" is displayed again before expiry.
- This also applies to the "1-week warning", which is no longer displayed after confirmation; only the "1-day warning" is displayed again as active warning.
- After switching off/on the device, a confirmed warning corresponding to the period until the license expires appears again after 15 minutes at the latest. Confirming the warning is therefore not persistent.
- The warning icon next to the license in the “Expires (UTC)” column of the license overview page continues to be displayed even after the warnings have been confirmed.



Licenses that have expired, are automatically deleted from the table.

License response in a ctrlX CORE Virtual

There is a license **4 hours Engineering license** in the virtual control after startup. The runtime is four hours.

When starting a logbook entry, the following warning is issued in the logbook:
Main diagnostics: 080E05FF , "Device Administration: Warning", detailed diagnostics: 00641F00 , "The system was started in test mode"

After the runtime, ctrlX CORE Virtual is shut down.

Temporary usage rights

The temporary usage rights allow to operate the ctrlX device without the required licenses.

A use case for example is the device exchange to minimize the standstill time of the machine if the hardware is defective. After exchanging the device, the machine can still be operated without the required licenses and the licenses can be assigned and loaded subsequently to the device via the "licensing center".



Temporary usage rights can only be enabled once for 10 days per device.

To enable the temporary usage rights, click on the  button to open the license options in the command bar of the "Licenses" tab. The remaining time of the temporary usage rights is shown in a banner in the window.

Error reaction when removing the license dongle at runtime

Outputs in the diagnostic logbook:

- **Immediately after removing the dongle:**

An E0 warning is triggered.

Main diagnostics: 080E0550, "License manager warning"; detailed diagnostics: 007A0091, "License dongle removed"

The warning is acknowledged automatically when the dongle is plugged in again.

- **After 60 s.:**

An F0 error is triggered.

Main diagnostics: 080F9550, "Error in license management", detailed diagnostics: 007A0091, "License dongle removed"

The error is not automatically acknowledged when plugging in the dongle again. It has to be reset actively.

- **After 10 min.:**

An F6 error is triggered. Axes and kinematics are stopped.

Main diagnostics: 080F9550, "Critical error in the license management", detailed diagnostics: 007A0091, "License dongle removed"

The error is not automatically acknowledged when plugging in the dongle again. It has to be reset actively. Axes and kinematics are not restarted automatically.

- **After 20 min.:**

An F9 error is triggered. The ctrlX device is switched off.

Main diagnostics: 080F9550, "Fatal error in the license management", detailed diagnostics: 007A0091, "License dongle removed"

Also refer to "ctrlX CORE Controls", DOK-XCORE*-*****-ITRS-EN-P ([➔R911405645](#)), Chapter "Device description → Status displays".

Elements of the tab “Licenses”

GUI element	Description
Command bar	“[x] items”: Shows the number of listed licenses
	 Link to open the license center. In the “licensing center”, previously purchased licenses can be assigned to the target device or dongle and the respective license file can be downloaded to the engineering PC
	 Command: “Uploads license file (Capability Response).” Opens the File Explorer to transfer the assigned license file from the Engineering PC to the license source for which it was created. If it is tried to upload an older license file with licenses differing from those currently on the license source, an HTTP client error 080E0200 / 0C7A0202 message is displayed. When uploading a current license file, a message window displays the changed licenses compared to the current state on the license source.
	 Opens the settings (enabling temporary usage rights)
Table	<u>Column entries</u> • “Product” License type designation • “Quantity” Number of licenses of this product • “App” Name of the app belonging to the license. The app name is also the link to the app description under Settings → Apps • “Description” What a license allows • “Source” Shows whether the license is on a plugged-in dongle or locally on the device • “Expires (UTC)” Expiry of the license validity. Contains only one value designed as mm-dd-yyyy hh:mm:ss if it is not a permanent license The table can be sorted according to “Product” “App”, “Source” and “Expires (UTC)”.

GUI element	Description
	Icons in the “Product” column refer to the status of the license: • License is OK • No respective app installed (information in the column “App”) • License expiring soon (information in the column “Expires (UTC)”)

Further information

- [↗ Chapter 13.2.6 Window – “Licenses” on page 284](#)

Tab – “Sources”**Call:**

ctrlX OS side navigation “*Settings* → *Licenses*” → Tab “Sources”

Function:

In the tab “Sources”, existing license sources are shown in a table.

Elements of the tab “Sources”

GUI element	Description
Command bar	“[x] items”: Shows the number of listed licenses
Table	<u>Column entries</u> • “Type” Type of license source • “Serial number” Serial number of license source • “Available” An icon shows whether the license source is available

Further information

- [↗ Chapter 13.2.6 Window – “Licenses” on page 284](#)

13.2.7 Window – “Data Layer”**Function:**

In the window “Data Layer”, process data of all components registered at the Data Layer can be accessed. Process data can be of any data type, e.g. drive parameters, PLC symbol variables or commands.

The Data Layer window visualizes all information on a node and allows to modify the values of writable nodes.

The values of the nodes you specified are shown in a monitoring window.

The window is split into three segments:

- Data Layer tree on the left side (see [↗ Data Layer tree on page 290](#))
- Command bar above the Data Layer tree (see [↗ Command bar on page 289](#))
- Detailed view on the right, next to the Data Layer tree (see [↗ Detailed view on page 292](#))

Call:

ctrlX OS side navigation *"Settings → Data Layer"*

Navigation:

In the Data Layer, it is primarily navigated in the Data Layer tree. Select the buttons > and ^ to open or close the node and thus to navigate through the Data Layer structure to the individual nodes.

More options to navigate to the nodes:

- Command bar
- Child node
Allows to navigate to a child node by clicking on the respective line in the table
- Metadata
Allows to navigate to a referenced node by clicking on the link in the attribute "References" if a reference is available

Command bar

GUI element	Description
← "Back to..."	Navigates to the previously selected node
↑ "Up to..."	Navigates to the parent node
↻ "Refreshes selected node"	Reads the data of the currently selected node in the tree again from the Data Layer
✦ "Top level node"	Navigates to the top node in the tree
datalayer > nodesrt "Breadcrumb"	Navigates to the node selected in the "Breadcrumb"
📄 "Copies path"	Copies the URL from the selected node
🔍 "Navigate to path"	Opens a modal dialog to enter a path. Press the button "Navigate to" and the dialog is closed and navigation to the specified node is started.
👁 "Opens watchlist"	Opens the watch window (monitoring window) in the sidebar
⚙ "Shows settings"	Allows different settings to represent the interface: • Showing last tree node while opening • Using last tree node while opening • Showing incomplete configuration from Data Layer • Animating dynamic changes of node values

Data Layer tree

The Data Layer tree shows all nodes from the Data Layer in a tree structure.

Each node has a name and is assigned to a defined node class represented with a respective symbol.

Table 36: Node classes

GUI element	Description
 "Variable" read only: Variable value is read-only read/write: Variable value can be read or written	This node represents a status value or a configurable value
 "Resource"	This node represents an object encapsulating an asset with multiple variables as child node
 "Collection"	This node represents a parent object of a number of resources of the same type or at least the basic type
 "Folder"	This node improves the readability of the tree by grouping nodes
 "Program"	This node stands for an action with a long runtime
 "Method"	This node stands for a short running action that returns a result immediately
 "Type"	A type definition or a schema of another node

All nodes with one or multiple child nodes can be expanded via  and closed via . When the node is expanded for the first time, the Data Layer reads its child nodes and shows them in the tree.

The nodes without child nodes (called leave nodes) have no symbol for expanding or closing.

If a node is selected in the tree by clicking on the name or on the symbol of the node, its detailed information is shown in the detailed view.

The tree can be freely enlarged or reduced in its width. A dividing line is used between the Data Layer tree and the detailed view.

BigData node to handle a large volume of data

It is possible that individual nodes are provided with a high number of child nodes (from multiple thousand up to hundred thousand nodes).

The representation and operation of a large volume of data is not user-friendly and the interface components used are not designed to handle large volume of data. Thus, the interface might not respond anymore and the display requires a long time to load all nodes or to display only a part of the nodes.

Thus, there is a special handling of nodes with a big selection of child nodes.

These nodes are called BigData nodes.

Definition of BigData nodes

In the "Settings" dialog, specify the minimum number of child nodes a node has to be provided with to be treated as BigData node.

The user can set the limit between 10 and 100 child nodes.

The default setting is 50.

Visualizing the BigData node in the tree

All nodes exceeding the number of child nodes previously defined in the "Settings" dialog are shown in blue in the tree.

Additionally, these nodes are provided with a  button showing the total number of child nodes.

If a BigData node expands, the first block (first node up to the maximum number of nodes) is shown.

Click on the  button to open a dialog to select a range of child nodes of the BigData node to be displayed.

After selecting the child node pressing the “Apply” button, all previously displayed child nodes are deleted in the tree first. The newly selected child nodes are then read from the Data Layer and displayed. This can always be repeated.

Elements of the selection dialog for BigData nodes

GUI element	Description
Command bar	
 x nodes	Visualizes the number of child nodes considering the set filter
 Filter function	Used to filter the currently displayed child nodes
 Previous range	Selects the previous range
 Next range	Selects the next range
Table	
	The table contains all child nodes from the selected BigData node. The nodes highlighted in blue are the currently selected nodes displayed in the tree. The number of selected nodes corresponds to the number of child nodes defined in the settings. Click on an entry in the list to automatically determine a new range starting from the selected node up to the maximum number of nodes defined. The number of nodes currently selected from the total number of child nodes is shown below the table.

Detailed view

The detailed view visualizes information on the currently selected node in the tree.

Depending on specific node properties, the respective areas are shown or hidden in the detailed view.

There are the following areas in the detailed view:

- **Header Area**

The header area is always shown for all nodes in the top area of the detailed view. The header area contains the symbol for the node classes, the names and a brief node description. The brief description is determined from the metadata. If the metadata for the node does not contain any brief description, the text “No description available” is shown.

Additional functions can be displayed in the header area of certain nodes. The node visualizes these functions depending on the following “Operations” attributes of the metadata.

Table 37: Header Area

GUI element	Description
 Adds to watchlist	This function enters/removes the path of the selected node into/from a specific watchlist. This function is only available for nodes with a set "read" operation.
+ “Create”	A child node is created with “Create”. If “createType” is set in the “References” attribute of the metadata, make an entry in the modal dialog to create a child node. The entry has to correspond to the referenced “createType”. Otherwise, an error message results.
 “Delete”	This function deletes the selected node.

- **Input area**

The input area of the node is directly below the header area. This input area is only displayed if the node has a value. This is the case if the “Operations” attribute is set to “read” in the metadata. Otherwise, this area is not shown.

Depending on the value type shown in the top line of the input area, the input area of the representation varies (also refer to [JSON object editor on page 294](#)).

If the “Operations” attribute is set to “read” and not to “write” in the metadata, the node value is read-only. No entry into the field possible.

If the “Operations” attribute is set to “read” and to “write” in the metadata, the node value is read and write. In this case, the two additional buttons “Write” and “Discard” are directly shown below the input area.

These two buttons can only be selected if a change is made in the input area.

- **Child nodes Area**

The child node is located below the input area. It is only shown if the node has a child node (in the metadata, “browse” has to be set in the “Operations” attribute).

Open this area with  or close it with . It contains a list of all child nodes.

The following information is included for each child node:

- Symbol
Represents the node class of the child node
- “Name”
Name of the child node

- “Value”
Values of the child node
- “Type”
Value type

Click on an entry in the list (on a child node) to navigate automatically to this child node.

- **Metadata Area**

The lower area in the detailed view is the metadata area shown for all nodes.

Open this area with  or close it with . It contains a list of the following metadata information:

Name of the attribute	Value	Meaning
“Node Class”		Refer to ↗ Table 36 Node classes on page 290
“Operations”	read	Node contains a value
	write	The value can be written to the node
	create	Node has a "create" function used to create child nodes
	delete	Node can be deleted
	browse	Node has child nodes
“Description URL”	<i>Currently not used</i>	
“Display name”	<i>Currently not used</i>	
“Display format”	Auto	Display in default font
“Unit”		Unit of a value (if available). It is also shown in the input area
“References”	readType	Type when reading a value (input/output type are identical)
	readInType	Input type when reading a value
	readOutType	Output type when reading a value
	writeType	Type when writing a value (input/output type are identical)
	writeInType	Input type when writing a value
	writeOutType	Output type when writing a value
	createType	Type when creating a child node

To automatically navigate to the specified URL, click on the link given in the “References” attribute.

Special handling for the node class "Method"

A method stands for a short running action that returns a result immediately. The method can include input and output data of different types.

Types of methods:

- Data is read-only with methods of the type "Read". These methods are labeled with the "Read" button in the GUI
The data types for the input ("readInType") and for the output ("readOutType") are described in the "References" attribute of the metadata.

Remark:

There are methods of the type "Read" that do not require an entry. In this case, the input area (Input parameter) is not shown. One characteristic of the method is that no "readInType" is defined in the "References" of the metadata.

- Use methods of the type "Write" to write data to the Data Layer. These methods are labeled with the "Write" button in the GUI.
The data type for the input and for the output are described in the "References" attribute of the metadata.

Special handling for the node class "Program"

A program stands for an action with a long runtime. The program is processed asynchronously and does not return a result immediately. The request returns a unique task ID instead that can be used to follow the progress and the status of the started program.

In a program, the "Operations" attribute "create" has to be set in the metadata. This ensures that this node is provided with the "create" function used to start the program.

If the "Create" button is pressed, a dialog to enter the input parameter is shown first. The program can subsequently be started via "Create".

JSON object editor

To enter values of the data type "object" (in the Data Layer these are always JSON objects), a JSON editor is used, which provides various additional functions compared to the other editors and thus makes input much easier for the user.

The following additional functions are available:

- **Line number**
Displays the line number at the left editor margin.
- **Syntax highlighting**
Displays the JSON object-specific words and character combinations in different colors depending on their meaning.
- **Syntax validation**
Checks the input for its syntactical correctness for JSON objects and marks the errors directly in the input. For each detected error, an error text is generated and displayed.
- **Autocomplete**
It is a function that reasonably supplements a user input.
- **JSON language support with schemes**
Extends the syntax validation and autocompletion to include the schema definition specific to the node. Furthermore, parameter descriptions (if provided in the schemes) can be displayed as tooltips.

Further information

- ➔ [Chapter 13.4.24 Dialog – “Watchlist” on page 362](#)
- ➔ [Chapter 10.2 Data exchange via Data Layer on page 184](#)

13.2.8 Window – “Date & Time”

The window “Date & Time” is used to set time, date and time zone on the ctrlX device.

Setting options:

- Manual input in the window segment “ctrlX OS”
- Application of browser settings in the window segment “Your browser”
- Automatic synchronization with an NTP server

Call:

ctrlX OS side navigation “*Settings → Date & Time*”

NOTICE

Note that time-based licenses lose their validity if you set the time back on the ctrlX device by more than 24 hours.

To activate the licenses again, please contact the Rexroth Support.

Elements of the “Date & Time” window

GUI element	Description
Drop-down menu “Date”	Date specification (required) Input schema: mm/dd/yyyy
Input field “Time”	Time specification (required) Input schema: hh:mm:ss
Drop-down menu “Timezone”	Time zone selection
Segment “Your browser”	Display of the current date and time settings of the web browser: • Date: mm/dd/yyyy • Time: hh:mm:ss • Time zone
Button “Assign to ctrlX OS”	to apply the time settings of the web browser to the ctrlX device. Note: The settings are only applied after saving.
Options switch “Time synchronization”	Selection: • Manual time input • Time synchronization via NTP

GUI element	Description
Button 	Opens a dialog to set the NTP polling interval (min/max) with predefined values. Please note that the value of the maximum polling interval always has to be greater than the value of the minimum polling interval. - Minimum polling interval (default: 32 s) can be set to: 16s / 32s / 1min / 2min / 4min / 9min / 17min / 34min / 1hour / 2hours / 5hours / 9hours / 18hours - Maximum polling interval (default: 34min) can be set to: 32s / 1min / 2min / 4min / 9min / 17min / 32min / 1hour / 2hours / 5hours / 9hours / 18hours / 36hours
Input box "NTP servers"	To enter the NTP server address to synchronize the time (the option "Time synchronization" has to be enabled). Available commands:  Adding more NTP servers to the list  Deleting selected NTP server from the list
Button "Check connection"	Allows to check which of the entered NTP servers are available and whether they are actually NTP servers.
Button "Save"	Click on the Button "Save" to apply the changes to the window "Date & Time" and to load them to the control

13.2.9 Window – "System Performance"

The window contains the following tabs that provide information on the performance data of the ctrlX device:

- Tab "Resources"
Contains values for CPU and memory load, see:
[↗ Tab – "Resources"](#)
- Tab "Processes"
Contains a tabular overview on all active processes on the ctrlX device, see:
[↗ Tab – "Processes"](#)
- Tab "Threads"
Contains a tabular overview on all active threads on the ctrlX device, see:
[↗ Tab – "Threads"](#)

Call:

ctrlX OS side navigation *"Settings → System Performance"*



Reserving processor cores and memory volume for real-time processes

The processor cores to process realtime processes is usually reserved automatically by the system when realtime data processed on the ctrlX device. In addition, further processor cores and working memory volumes can be reserved for realtime data processing to achieve a more homogeneous workload if required.

Note the following points:

- Changes to the settings can only be made in the device operating state "Service".
- When reserving processors, at least one processor has to remain free to process non-real-time processes.
- Changes in the processor settings are applied after device restart.

To open a dialog with two setting areas, click on the [⚙] button in the header of the window:

- **Cores**

Here you can reserve additional processor cores to process realtime processes.

Reserved processor cores are displayed in the "Resources" tab with the icon [⚙], see:

➔ [Tab - "Resources"](#)

Processor cores responsible to process system processes cannot be reserved for real-time processing and are grayed out in the selection.

- **Memory**

Here, enable the option to reserve memory volume for real-time processes and set the working memory volume to be reserved using the slider.

The available memory area depends on the device and is limited by the system.

13.2.10 Window – "System Information"

The window contains the following widgets that provide information on the ctrlX device:

- **Widget "General"**

Contains general information on the ctrlX device, see:

➔ [Widget - "General"](#)

- **Widget "Hardware"**

Contains information on the device hardware, see:

➔ [Widget - "Hardware"](#)

- **Widget "Typeplate"**

Contains information on the electronic type plate of the ctrlX device, see:

➔ [Widget - "Typeplate"](#)

Call:

ctrlX OS side navigation *"Settings → System Information"*

13.2.11 Window – "Scheduler"

Function:

Tasks can be created, edited and deleted in the "Scheduler" window. An overview on their current state is also provided.

Call:

ctrlX OS side navigation *"Settings → Scheduler"*

Elements of the command bar

The command bar contains the name of the application and a button to open the dialog for the settings.

If the “Debug” mode is set,  is shown. Disable the “Debug” mode again via .

GUI element	Description
<Name>	Application name: Scheduler
	The icon is displayed when the Debug mode is set. Click on  to start the “Scheduler settings” dialog (see Chapter 13.4.19 Dialog – “Scheduler settings” on page 357).
	The icon starts the “Scheduler settings” dialog (see Chapter 13.4.19 Dialog – “Scheduler settings” on page 357)

Tabs

The “Scheduler” window has the following tabs:

- Tab Overview
The tab contains an overview on all tasks and their states (see [Chapter 13.3.7 Tab – “Overview” on page 327](#)).
- Tab configuration
In this tab, parameters of the Scheduler can be changed and tasks can be created or edited (see [Chapter 13.3.8 Tab – “Configuration” on page 328](#)).

Further information

- [Chapter 13.3.7 Tab – “Overview” on page 327](#)
- [Chapter 13.3.8 Tab – “Configuration” on page 328](#)
- [Chapter 13.4.19 Dialog – “Scheduler settings” on page 357](#)
- [Chapter 13.3.9 Tab – “Startup” on page 331](#)
- [Chapter 13.3.10 Tab – “Debug” on page 332](#)
- [Chapter 13.4.20 Dialog – “Add task” on page 358](#)
- [Chapter 13.4.21 Dialog – “Delete task” on page 359](#)
- [Chapter 13.4.22 Dialog – “Add callable to <taskname>” on page 359](#)
- [Chapter 13.4.23 Dialog – “Delete callable” on page 361](#)
- [Chapter 13.5.7 Sidebar – “schedMain” \(Parameter\) on page 373](#)
- [Chapter 13.5.8 Sidebar – “Task: <name>” \(Task parameter\) on page 373](#)
- [Chapter 13.5.9 Sidebar – “<name_callable>” \(Callable parameter\) on page 374](#)
- [Chapter 9.2 ctrlX CORE Scheduler on page 151](#)

13.2.12 Window – “Backup & Restore”

The window “Backup & Restore” is used to back up and restore a ctrlX device (backup).

The window has two tabs:

- “Back up”
To save the settings of the ctrlX device in a backup file, see [Tab – “Back up”](#)
- “Restore”

To apply existing settings to the current ctrlX device, see [➔ Tab – “Restore”](#)
When creating the backup, parts of the setup can be selected or deselected.

Related topics

[➔ Backup – Creating from ctrlX device](#)

[➔ Backup – Applying to ctrlX device](#)

Call:

ctrlX OS side navigation *“Settings → Backup & Restore”*

13.2.13 Window – “Storage”

related topics:

[➔ Using SD cards and USB devices](#)

[➔ Mounting external storage medium](#)

[➔ Formatting external storage medium](#)

[➔ Removing external storage medium](#)

Function:

The window “Storage” provides an overview on the internal ctrlX CORE storage and all available external storage media. Each storage medium is shown as tile displaying the current memory load. Click on the tile to show all detailed information on the storage medium.

The actions listed above can be executed for the external storage media.

Call:

ctrlX CORE side navigation *“Settings → Storage”*

Window elements:

GUI element	Description
Available storage media in tile representation	Upon delivery, the internal storage of the ctrlX device is shown including the current memory load.

13.2.14 Window – “Firmware Update”

Use this functionality to install a new system image on the ctrlX device.



All changes made by the user are lost when the firmware is updated!

- All device settings are reset to their default values.
- All apps installed by the user are removed.
- Certificates and keys in the „Permanent Storage“ remain.
- Installed licenses remain.



Backing up and restoring device settings and app data

Use the backup functionality to back up the device settings and app data before a firmware update to automatically initialize the ctrlX device with the backed up settings and resources after the update.

Please refer to: [➔ Web documentation](#)

Prerequisites

- The firmware update is not supported by all ctrlX devices.

If the ctrlX device does not support this function, a corresponding message is displayed in the window, e.g. for virtual ctrlX device instances.

- A temporary storage medium is required for the update, see: [↗ Web documentation](#)

Note that the temporary storage medium is repartitioned during the update and that all data on the medium will be lost!

In addition, there may be device-specific restrictions with regard to the media or slots/connections that can be used.

Example: For the device type ctrlX CORE X7, the temporary storage medium has to be connected to the USB port XF01C, as this device cannot boot from other ports.

Operating principle

The following steps are carried out automatically during the update:

- The temporary storage medium is repartitioned.
- The system image and the backup file (if available) are copied to the temporary storage medium.
- The ctrlX device is restarted and installs the system image.
- The settings and resources from the backup file are applied (if available).



The complete operation can take up to 30 minutes. Do not switch off the device during the operation!

Procedure

1. Set up a temporary storage medium on the ctrlX device, see [↗ Mounting external storage medium](#)
2. Specify the system image to be installed in the window.
3. Optionally, specify a backup file is that to be initialized on the ctrlX device after the firmware update.
4. Select the temporary storage medium that was set up on the ctrlX device in step 1.
5. Start the firmware update
 - ➔ This operation can take up to 30 minutes!
 - ➔ The ctrlX device is not accessible during the process!
 - ➔ Do not switch off the device during the operation!
6. After the update, the login window of the ctrlX device opens in the browser.
 - ➔ If this is not the case, try to reach the ctrlX device using the default IP address: **192.168.1.1**

Call

ctrlX OS side navigation “*Settings → Firmware Update*”

13.2.15 Window – “Reset Device Data”

The window provides a function to reset the ctrlX device to the original factory settings.

Prerequisites

To be able to reset, the following requirements have to be met:

- ctrlX CORE basis image version 1.18 or higher.
- System administrator rights on the ctrlX device, see: [➔ Window – “Permissions”](#)
- External storage medium in FAT32 formatting, with a free space of at least approx. 450 MB.

Operating principle

The following actions are performed during reset:

- All settings (e.g. network, SSH settings) are reset to their default values.
- All apps - newly installed by the user - are removed.
- All preinstalled apps remain in the currently installed version.
- Certificates and keys in the „Permanent Storage“ remain.
- Installed licenses remain.

To reset to factory settings, a FAT32-formatted storage medium has to be mounted on the ctrlX device to exchange data to allow that all necessary data can be temporarily stored on it. The target state and the versions of the applications installed after the reset are displayed.

This operation can take up to 30 minutes!



Procedure

1. ➔ Mounting the external storage medium (FAT32) on the ctrlX device, see [➔ Mounting external storage medium](#)
2. ➔ Select target medium in the window (external storage medium from step 1)
3. ➔ If all requirements are met, the reset process can be started via the button “Reset device”
4. ➔ First, all required data is loaded to the external storage medium
 - ➔ During this time, the process can still be canceled
5. ➔ After saving the data, the reset process starts
 - ➔ This operation can take up to 30 minutes!
 - ➔ The ctrlX device is not accessible during the process!
6. ➔ After restoration, the login window of the ctrlX device opens in the browser
 - ➔ If this is not the case, try to reach the ctrlX device using the default IP address: **192.168.1.1**

Call

In the ctrlX CORE web interface:

ctrlX OS side navigation “*Settings → Reset Device Data*”

13.2.16 Window – “Certificates & Keys”

Function:

The window “Certificates & Keys” lists applications and services installed on the ctrlX device with name and description for which certificates and keys can be managed.

A command bar is located above the table. The number of listed services is shown on the left of the table. To open the “PKIs”, select the “Manage PKIs” button on the right side of the command bar. The connections to the PKI servers managed there are used to automatically renew the validity of the certificates on the ctrlX device.

Call:ctrlX OS side navigation “*Settings → Certificates & Keys*”**Elements of the “Certificates & Keys” window**

GUI element	Description
Command bar	“[x] item(s)” Number of listed services
	 Managing PKIs Opens the “PKIs” window to create connections to the PKI servers and to edit them, see ↗ Chapter 13.2.17 Window – “PKIs” on page 302
Table	“Service”: Name of the installed applications and services Click on the name to open the administration site of the respective service or the application to edit them, see ↗ Chapter 13.5.5 Editor – Certificates & Keys on page 368
	“Description” Description of installed applications and services

Further information

- [↗ Chapter 6.4.1 Introduction on page 63](#)
- [↗ Chapter 13.5.5 Editor – Certificates & Keys on page 368](#)
- [↗ Chapter 13.2.17 Window – “PKIs” on page 302](#)

13.2.17 Window – “PKIs”**Function:**

The created connections to the PKI servers are listed in a table in the “PKIs” window. The connection to the PKI can be secured via TLS by referencing a server certificate when establishing the connection.

To edit and delete connections, use the buttons in the “Actions” column.

If no connection has been set up, no command bar and no table are displayed in the window, but a line with the text “No PKIs available” and the + “Add PKI” button.

Call:ctrlX OS side navigation “*Settings → Certificates & Keys*” → command bar → button  “Manage PKIs”**Elements of the “PKIs” window**

GUI element	Description
Command bar	“[x] item(s)” Number of listed connections
	 Managing Certificates & Keys The “Certificates & Keys” window opens

GUI element	Description
	 Setting up a connection to a PKI server The “Add PKI” dialog opens. The endpoint of the PKI server has to be specified in the “URI *” field. In the “Server certificate” field “none, skip verification” or a server certificate already uploaded under "Network Security" can be selected. Via the button  "Upload Server certificate", a certificate can also be uploaded directly in the dialog. Select the protocol used for the certificate renewal in the drop-down selection field “Protocol *”. • “CMP” Certificate management protocol • “SCEP” Simple certificate enrollment protocol When selecting the “SCEP” protocol, the following “SCEP protocol options” are displayed. The selection of options is determined, among other things, by the requirements of the PKI. • “Generate new key on renewal” A new key is generated when the certificate is renewed • “Sign request by new key on renewal” When selected, the “Generate new key on renewal” option is selected automatically. • “Use PKCS request on renewal” The renewal request to the PKI is made in PKCS format If “Challenge” is selected in the “Authentication *” field, a field for hidden input of the challenge is displayed. The challenge is defined by the PKI server and the requester. In the “Authentication *” field, the type of authentication for the PKI “None” or “Challenge” is specified. “None” is fixedly set for protocol CMP.
Table	“URI” PKI server endpoint “Protocol” Protocol used: “Actions” Includes buttons to edit or delete a connection  Editing a connection The editor to edit the PKI sever connection is shown on the right of the “PKIs” window.  Deleting a connection

Further information

- ➔ [Chapter 6.4.1 Introduction on page 63](#)
- ➔ [Chapter 13.4.7 Dialog – “Certificate renewal” on page 345](#)
- ➔ [Chapter 13.3.11 Tab – “Certificates” on page 332](#)
- ➔ [Chapter 13.3.15 Tab – “Certificate enrollments” on page 339](#)

13.2.18 Window – “Users & Permissions”**Call:**

ctrlX OS side navigation “*Settings → Users & Permissions*”

Function:

The window “Users & Permissions” is the user management of the control to manage users and user groups and to configure the individual permissions. The user management provides setting options for the password and session policies.



To go to the user management, a connection has to be set up to the control, as device-specific users, groups and permissions are not stored in the project.

Elements of the “Users & Permissions” window

GUI element	Description
Tile “Users”	Creating, editing and deleting users, see ➔ Chapter 13.2.19 Window – “Users” on page 305
Tile “Groups”	Creating, editing and deleting groups, see ➔ Chapter 13.2.20 Window – “Groups” on page 306
Tile “Permissions”	Displaying and configuring individual permissions for users and groups, refer to ➔ Chapter 13.2.21 Window – “Permissions” on page 306
Tile “Sessions”	Displaying and closing sessions, refer to ➔ Chapter 13.2.22 Window – “Sessions” on page 307
Tile “More settings”	Hides a window with links on the configuration dialogs for: • “Password Policies” Opens a dialog to set the password policies. The following conventions apply while entering: ➔ Chapter 6.1.12 Conventions for login and session monitoring on page 46 • “Session Policies” Opens a dialog to set the login and session policies. The following conventions apply while entering: ➔ Chapter 6.1.13 Username and password conventions on page 46 • “Login banner” Opens an editor to enter a text to be shown in the banner of the “Login” dialog, see ➔ Chapter 6.1.6 Login banner on page 44 • “Remote authentication”

GUI element	Description
	Opens a dialog to enable the authentication using a remote server and settings of the connection to the server, see ↗ Chapter 13.4.15 Dialog – “Remote authentication” on page 353 • “Easy service access” Opens a dialog to enable the option “Easy service access”, see ↗ Chapter 13.4.16 Dialog – “Easy service access” on page 355

Further information

- [↗ Chapter 6.1.1 Introduction on page 42](#)

13.2.19 Window – “Users”**Call:**

ctrlX OS side navigation “*Settings → Users & Permissions → Users*”

Function:

The window “Users” manages users. Upon delivery, the user “boschrexroth” has already been configured on the device. The configured users are displayed in a table with command bar. To create users, select the buttons in the command bar.

Elements of the “Users” window

GUI element	Description
Command bar “Users”	“[x] items”: Number of listed users
	 Creating a new user, see ↗ Chapter 13.4.10 Dialog – “Add user” on page 348 (Naming convention, see ↗ Chapter 6.1.13 Username and password conventions on page 46)
	 Loading “System user assertion” file to the device. Use the “System user assertion” file to create a user with access rights to the operating system of the ctrlX device, see ↗ Chapter 6.1.10 Creating a system user on page 45
Table “Users”	“Name”: Name of the user
	“State” Use the switch “State” (enabling state) to enable or disable the selected user. A disabled user cannot be logged into the device and cannot execute any actions on the device
	“Actions”  Editing selected user, refer to ↗ Chapter 13.5.4 Editor – Users on page 367  Deleting selected user

Further information

- ➔ [Chapter 6.1.1 Introduction on page 42](#)
- ➔ [Chapter 13.2.18 Window – “Users & Permissions” on page 304](#)

13.2.20 Window – “Groups”**Call:**

ctrlX OS side navigation “*Settings → Users & Permissions → Groups*”

Function:

The window “Groups” manages user groups and their permissions.

Upon delivery, the user group “sshuser” is already configured on the control. Only "System User" assigned to the group implicitly obtain the right to connect to the ctrlX OS device via SSH, see ➔ [Chapter 6.1.10 Creating a system user on page 45](#).

The configured user groups are displayed in a table with command bar. To create user groups, select the buttons in the command bar.

Elements of the “Groups” window

GUI element	Description
Command bar “Groups”	“[x] items”: Number of listed user groups  Creating new user group (Naming convention, see ➔ Chapter 6.1.13 Username and password conventions on page 46)
Table “Groups”	“Name”: Name of the user group “Actions” Includes buttons to edit and delete the user group  Editing the selected user group, refer to ➔ Chapter 13.5.3 Editor – Groups on page 366  Deleting selected user group

Further information

- ➔ [Chapter 6.1.1 Introduction on page 42](#)
- ➔ [Chapter 13.2.18 Window – “Users & Permissions” on page 304](#)

13.2.21 Window – “Permissions”**Call:**

ctrlX OS side navigation “*Settings → Users & Permissions → Permissions*”

Function:

The window “Permissions” lists all permissions of the available apps in a table. Click on the name and each permission can be edited. In the editor, configure for which user or user group the permissions are granted or revoked, see ➔ [Chapter 13.5.2 Editor – Permissions on page 365](#).



Users to whom read/write permissions were assigned for an app can implicitly receive permissions for the certificate and keystore of this app if the app supports this function, see ➔ <https://boschrexroth.github.io/ctrlx-automation-sdk/package-assets.html#certificate-stores>

Elements of the “Permissions” window

GUI element	Description
Command bar	“[x] item(s)” Number of listed elements in the “Category” column
	 /  Showing or hiding the “Scope identifier” column in the table
Table	Table columns: • “Category” Specification of the category to which the permission is applied to • “Permission” Name of the permission that is also the link to open the permission editor • “Scope identifier” The column that can be shown is displayed for each permission of the respective scope ID. Scope IDs are shown in error messages if the user does not have the required permission

Further information

➔ Chapter 6.1.1 Introduction on page 42

➔ Chapter 13.2.18 Window – “Users & Permissions” on page 304

➔ Chapter 13.5.2 Editor – Permissions on page 365

13.2.22 Window – “Sessions”

Call:

ctrlX OS side navigation “Settings → Users & Permissions → Sessions”

Function:

The “Sessions” window lists all active sessions of the control in a table. The individual entries in the table show the most important information on the session. Furthermore, individual sessions can be exited with the button .

Session details:

Click on one of the table entries and another window opens. This window shows all available session details. The following information supplements the entries in the table:

- ID
- Type
- Type details
- User name
- Creation date
- Expiration date
- Timeout duration
- Last access date

- Client IP
- Permissions

Elements of the “Sessions” window

GUI element	Description
Table “Sessions”	“Username” User that started the session
	“Client IP” IP address of the client
	“Creation Date (UTC)” Starting date of the session/time
	“Expiration Date (UTC)” Expiry date of the session/time
	“Type” Session type
	“Actions” 
	Press the button  to close the selected session. A prompt is shown after the execution asking whether you are sure about exiting the session: •  ends the session and the respective user is logged off •  does not end the session. You return to the “Sessions” site

Further information

- ➔ [Chapter 6.1.1 Introduction on page 42](#)
- ➔ [Chapter 13.2.18 Window – “Users & Permissions” on page 304](#)

13.2.23 Window – “Hosts”

The host table of the control can be edited in the “Hosts” window. The displayed entries correspond to the content of the system file: `</etc/hosts>`.

The host table is used to link IP addresses with host names. As the DNS servers are normally used for linking, it is not required to make changes in the settings.

NOTICE

The ctrlX device might not be reachable anymore if changes are made in the host setting!

Each entry in the host table consists of an IP address, the respective host name and an optional alias. Using an alias can for example be useful to shorten long host names or reduce effort when changing the host name, as the alias can remain unchanged.

Window call

ctrlX OS side navigation “*Settings* → *Hosts*”

Window elements

Element	Description
“IP address”	IP address of the host
“Host name”	Host name

Element	Description
"Alias"	Host alias
"Edit"	Button to switch the host table to editing mode Note: The ctrlX device might not be reachable anymore if changes are made in the host setting!
"Save"	Saves changes in the host table
"Cancel"	Closes the editing mode without saving changes

13.2.24 Window – "Network Diagnostics"

The window contains network diagnostic tools to check whether a specific host is reachable in an IP network. For example, pinging the host name and host IP address can determine whether there is a problem with name resolution. The diagnostic tools are grouped in the following tabs:

Tab "Ping"

Ping is a network tool that checks the connection between two devices by sending a small data package and measuring the response time. No response means that the IP address is currently unavailable. However, some server administrators block the ping response mechanism to prevent the devices from being flooded with ping commands. In this case, it cannot be concluded from a missing ping response that the remote station is not reachable.

1. To perform the ping test, enter the IP address first.
2. Set the interval and the number of data packages.
Default setting: Interval 1 second / 4 data packages
3. Click on "Ping" to start the test.

➔ The test result is displayed after the test:

Example: Ping result (positive)

```

PING google.com (142.250.185.174) 56(84) bytes of data.
64 bytes from fra16s51-in-f14.1e100.net
(142.250.185.174): icmp_seq=1 ttl=119 time=9.16 ms
64 bytes from fra16s51-in-f14.1e100.net
(142.250.185.174): icmp_seq=2 ttl=119 time=7.64 ms
64 bytes from fra16s51-in-f14.1e100.net
(142.250.185.174): icmp_seq=3 ttl=119 time=7.07 ms
64 bytes from fra16s51-in-f14.1e100.net
(142.250.185.174): icmp_seq=4 ttl=119 time=7.23 ms

```

```

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time
3003ms
rtt min/avg/max/mdev = 7.071/7.774/9.157/0.824 ms

```

Example: Ping result (negative)

```

PING google.com (142.250.185.174) 56(84) bytes of data.

```

```

--- google.com ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3123ms

```

Tab "ARP"

The "ARP" is used as a diagnostic tool to check the reachability of a device in the network. The check uses the "Address Resolution Protocol" (ARP) that assigns an IP address to the corresponding physical MAC address in a local

network. In some cases, multiple MAC addresses can be assigned to one IP address, which can occur in complex network environments such as systems with load sharing.

1. ➤ For the ARP test, enter the IP address.
2. ➤ Click on “ARP” to start the check.
 - ➔ The test result shows the determined MAC addresses if the specified IP address is reachable in the network.

Tab “Check connection”

The tab allows to check a connection via the set proxy or without a proxy. To configure the proxy settings, go to the following window: ➔ [Window – “Proxy”](#)

1. ➤ To check, enter the target URL first.
2. ➤ Select whether the validity of the SSL certificate should be verified.
3. ➤ Click on “Check” to start the connection test.
 - ➔ The result of the connection test is displayed.

Calling the window

In the ctrlX CORE web interface:

ctrlX OS side navigation “*Settings → Network Diagnostics*”

Related topics

➔ [Configuring IP settings](#)

13.2.25 Window – “Network discovery”

In the “Network discovery” window, you can disable or enable the services LLMNR, MDNS, MDNS-RESOLVER and UPNP.

Call:

ctrlX OS side navigation “*Settings → Network discovery*”

Elements of the “Network discovery” window

GUI element	Description
“Service”	Name of the service The following services are available: GRATUITOUS-ARP Gratuitous Address Resolution Protocol (GRATUITOUS-ARP) is a service. A device sends an ARP request to refresh the ARP cache of other devices on the network. It helps to maintain efficient and accurate communication within a network. There are two settings with Gratuitous ARP: • Interval The interval refers to the time between successive Gratuitous ARP messages. The unit of the interval is measured in minutes. A smaller interval means that the Gratuitous ARP messages are sent more frequently, while a larger interval results in less frequent transmissions. • Count The setting defines the number of Gratuitous ARP messages that the ctrlX device sends. A higher count means that more Gratuitous ARP messages are sent, while a lower count results in fewer messages. LLMNR Link-Local Multicast Name Resolution (LLMNR) is a service to enable the name resolution in scenarios when a conventional DNS name resolution is not possible. MDNS Multicast DNS (MDNS) is a service to resolve host names in IP addresses within small networks that do not include a local name server. MDNS-RESOLVER Multicast DNS Resolver (MDNS-RESOLVER) is a service that assumes a domain name (.local) as input and resolves it to an IP address using MDNS. UPNP Universal Plug and Play (UPNP) is a service that allows devices in the same local network to discover each other and to automatically connect through standard networking protocols
“State”	State of the service
“Enable” (toggle switch)	Enabling/disabling the service : Disabled : Enabled Disabling: Stopping the service and disabling autostart. Enabling: Starting the service and enabling autostart
Description	Describing the service (see above)

13.2.26 Window – “Network interfaces”

The window is used to manage and configure network connections of the ctrlX device.

Available interfaces are displayed as tiles, grouped by category:

- Ethernet
- WiFi
- Bridge
- VLAN

Further topics

- ➔ [Configuring IP settings](#)
- ➔ [Automatic/manual IP configuration](#)
- ➔ [Defining routes](#)
- ➔ [Bridging interfaces](#)
- ➔ [Limiting network bandwidth](#)
- ➔ [Connecting to WiFi network](#)
- ➔ [Setting up WLAN hotspot](#)
- ➔ [EoE configuration \(Ethernet over EtherCAT\)](#)
- ➔ [Adding and configuring VLAN interface](#)
- ➔ [Configuring security settings for wired interfaces](#)
- ➔ [Link local](#)

Call

In the ctrlX CORE web interface:

“side navigation → Settings → Network interfaces”

Window description

GUI element	
Header	Button [ Opens a sub-selection: • “Add bridge”, see: ➔ Bridging interfaces • “Add VLAN interface”, see: ➔ Adding and configuring VLAN interface
	Button [ Opens the “Network settings” dialog, see: ➔ Dialog – “Network settings” Setting options: • Configuration or limitation of the network bandwidth. • Option for dynamic determination of the maximum package size (PMTUD).
List with configured network interfaces/ editor	The lower window area shows all configured network interfaces of the ctrlX device as tiles. Click on one of the tiles and an editor opens containing detailed information on the interface configuration. Network settings can be made, see: ➔ Editor – “Network interfaces”

13.2.27 Window – “Proxy”

To configure a proxy server, go to the “Proxy” window. The proxy server is a gateway between the ctrlX CORE control and the Internet.

The proxy settings are disabled upon delivery. To enable the proxy settings, at least one of the two proxy fields has to contain a valid IPv4, IPv6 or domain address. Refer to the table below.

If the network to which the ctrlX device is connected to has a proxy server, the ctrlX device has to be configured to use the proxy server of the network.

For general information on the proxy server, refer to the following web site:

➔ https://wikipedia.org/wiki/Proxy_server



Call:

ctrlX OS side navigation “Settings → Proxy”

Window elements

GUI element	Description
“Enable proxy” (toggle switch)	Enabling/disabling the proxy server : Disabled : Enabled
“HTTP proxy” (input field)	Enter an IPv4, Ipv6 or domain address and the port of the HTTP proxy server, e.g.: • IPv4: http://<Target_URL> • IPv6: http://<Target_URL>
“HTTPS proxy” (input field)	Enter an IPv4, Ipv6 or domain address and the port of the HTTPS proxy server, e.g.: • IPv4: https://<Target_URL> • IPv6: https://<Target_URL>
“No proxy” (input field)	Enter web addresses (host names) that should not be accessed via the proxy server. Multiple entries - separated by a comma - can be entered, e.g.: • localhost, *.<Target_URL>.com, 192.168.0.0/24, 2001:0db8::7344
“Enable authentication” (toggle switch)	: Proxy authentication disabled : Proxy authentication enabled Enter username and password to authenticate the proxy server. Both proxy servers (HTTP and HTTPS) are always used for the authentication.
“Check connection”	Use this button to check a connection via the set proxy or with the proxy disabled. Enter the target URL and select whether the validity of the SSL certificate should be verified.
“Save” Command to save the proxy settings	Changes in the proxy settings are applied after saving.

13.2.28 Window – “SSH”

Function:

To enable or disable the SSH communication protocol on the ctrlX OS device, go to the window “SSH”. Further connection parameters of the SSH communication service can be configured.

**Enabling/disabling the SSH communication service**

The SSH protocol can be enabled and disabled by an administrative user or a user with the respective permissions.

Only users created with "system-user-assertion" can log into the ctrlX OS device via SSH. You have to be a member of the "sshuser" group (default).

Call:

ctrlX OS side navigation *"Settings → SSH"*

Elements of the "SSH" window

GUI element	Description
Selection box "SSH protocol status"	Disabling and enabling the SSH protocol on the ctrlX OS. Default value = inactive
Configuration field "Maximum number of open sessions per connection *"	Maximum number of open sessions per SSH connection. Default value = 10
Configuration field "Maximum number of authentication attempts per connection *"	Maximum number of login attempts per SSH connection. Default value = 6
Configuration field "Maximum number of concurrent unauthenticated connections *"	Maximum number of simultaneous SSH connections without authentication. Default value = 10
Configuration field "Logout after session length in minutes (0 for unlimited) *"	Automatic logout after a certain time of inactivity, in minutes. Default value = 0 (disabled)
Input field Banner text	Free text output in the SSH client after entering the username. For a better overview in the shell, complete the text with a new line after the last character

Further information

- [↗ General information on page 45](#)
- [↗ Call: on page 305](#)
- [↗ Chapter 6.1.1 Introduction on page 42](#)

13.2.29 Window – "Open Source Components"**Call:**

ctrlX OS side navigation *"About ctrlX OS → Open Source Components"*

Function:

The window "Open Source Components" provides information on open source components implemented into the ctrlX CORE software instance. The components are shown in a table.

After the table, the "Written offer for source code" is shown, see [↗ Chapter 13.2.30 Written offer for source code on page 315](#)

Elements of the “Open Source Components” window

GUI element	Description
Table	“Name” Name of the open source component
	“Version” Version of the open source component
	“License(s)” License type
“Written offer for source code”	Information on the reference of the implemented source code

13.2.30 Written offer for source code**Call:**

ctrlX OS side navigation *“About ctrlX OS → Open Source Components → Written offer for source code”*

License information

This product contains software components that are licensed as "Free software" or "Open source software" under one or several of the below-mentioned licenses and which thus require that the source code is made available. The source code of these software components is not provided together with this product. Instead, Bosch Rexroth provides the licenses upon request. To obtain the source code, email a request to ➔ open.source@boschrexroth.de or post it to the following address:

Bosch Rexroth AG
Open Source Office
Zum Eisengießer 1
97816 Lohr am Main
Germany

a.

For software components licensed under the GNU General Public License Version 2 and/or 3, the GNU Library General Public License Version 2 and/or the GNU Lesser General Public License Version 2.1 and/or 3.0, the Affero General Public License Version 1,2 and/or 3, the 7-Zip license or the eCos License 2.0, the following applies:

- Everybody has the right to obtain the source code of the software components from Bosch Rexroth.
- This offer is valid until three years after the last distribution of the object code by Bosch Rexroth; notwithstanding the foregoing, in the case of licensing under the GNU General Public License Version 3, the Affero General Public License Version 3 or the GNU Lesser General Public License Version 3.0, this offer shall stand as long as Bosch Rexroth provides spare parts or support services for the product.
- The corresponding source code also comprises the source code required to generate, install, execute and modify the object code if required by the relevant license.

- For the circulation of the source code, Bosch Rexroth reserves the right to charge incidental costs for preparing the data carrier (CD ROM, DVD or USB flash drive) as well as for postage.
- Please let us know where to send the corresponding source code to. If possible, please provide additional information about the affected product (e.g. product ID, serial number). This will help in finding the corresponding source code.

b.

For software components or corresponding source code licensed either under Mozilla Public License (MPL) version 1.0, 1.1 or 2.0, der Common Development and Distribution License (CDDL) version 1.0, of the Nokia Open Source License (Nokia or NOKOS) version 1.0a, Common Public Attribution License v.1.0 or which are part of the exception of Modified GPLv2 FreeRTOS License (Exception), the following applies:

- If you received such software components by Bosch Rexroth, upon request, Bosch Rexroth provides you with the source code of the software components, complying with the applicable previously mentioned license. Depending on the data volume, the code is attached to an email or is provided via a file hosting service.
- If Bosch Rexroth has modified pre-existing source code, you may obtain the corresponding source code for that modification (licensed under the terms of the applicable license above) for at least 12 months after it is first provided to a third party, but at least 6 months after a subsequent version of the modification is provided to a third party.
- Please provide the product information (e.g. product ID, serial number) containing the software components to help us find the correct corresponding source code.

Further information

- ➔ [Chapter 13.2.29 Window – “Open Source Components”](#) on page 314

13.2.31 Window – “About ctrlX OS”

Function:

The window “About ctrlX OS” provides product information on the present ctrlX OS software instance:

- Software and version
- Software publisher (copyright)
- Open source components
- General conditions of use

Call:

ctrlX OS side navigation *“About ctrlX OS”*

13.3 Tabs

13.3.1 Tab – “Threads”

The tab displays all threads currently running on the ctrlX device in a table. The content is refreshed every 5 seconds.

Command bar

The command bar above the table provides the following functions:

- Display/total number of active threads
- Input field for the text search for threads

Call:

ctrlX OS side navigation *“Settings → window System Performance → Tab → Threads”*

Description

Information	Description
“Process ID”	Process ID number
“Name”	Thread name
“App”	App from which the process was started
“Memory (MB)”	Memory space in the internal device memory used by the process
“CPU (%)”	CPU load caused by the process
“Priority”	Priority specified by the operating system
“CPU Core”	Shows on which CPU core the thread was processed last

Further information

➔ [Chapter 13.2.10 Window – “System Information” on page 297](#)

13.3.2 Tab – “Alarms”

The tab displays all currently pending errors and warnings. Already trouble-shot errors are also shown if they still have to be confirmed by the user.

The messages are displayed in a table with an additional command bar. If there are no errors, an empty table is shown. The command bar shows the number of pending messages and has buttons used to enable various functions, see description of the tab elements below.

The table is updated automatically as soon as new errors are pending or current errors were fixed and acknowledged.

Standard/developer view

The display of the logbook can be customized for different user types with regard to the contents displayed, see:

➔ [Dialog – “Settings” \(Logbook\)](#)

The standard view shows all required data important to commission the machine.

The developer view provides the full range of data columns and is aimed more at app developers.

Call:

ctrlX OS side navigation *“Diagnostics → Logbook → tab “Alarms””*

Elements of the tab in the standard view

GUI element	Description
"Command bar"	• [x] items Number of listed entries •  Button to reset and delete all pending diagnostics. Errors and warnings that cannot be reset, are still shown in the list. •  Button to download all displayed diagnostics, see: Logbook – Exporting •  Buttons to "expand" and "collapse" the lines to display further information on a diagnostic message.
"Table"	• "Level" Symbolic representation of the message category: -  = Warning to be fixed -  = Error to be acknowledged -  = Error to be fixed • "State" - To be fixed = the error/warning is still pending and has to be fixed. - To be acknowledged = the error was fixed, but still needs to be acknowledged by the user • "Date (UTC+/-xxxx)" Date and time of the log message in the format: "DD.MM.YYYY, hh:mm:ss.ms" • "Code" Main diagnostic code and detailed diagnostic code (if available) • "Description" Depending on the log message, the description can contain the following attributes: - Line 1: Main text - Line 2: Detailed text - Line 3: Dynamic description (only visible when expanded) • "Source" Depending on the log message, the source can contain the following attributes: - Line 1: Entity - Line 2: Origin • "Actions" This column displays buttons to perform actions for the selected diagnostic message: -  = Navigate to the alarm in the history overview. -  = Acknowledge error. -  = Expand the message line. -  = Expand the message line.

Elements of the tab in the developer view

GUI element	Description
"Command bar"	• [x] items Number of listed entries •  Button to reset and delete all pending diagnostics. Errors and warnings that cannot be reset, are still shown in the list. •  Button to download all displayed diagnostics, see: Logbook - Exporting
"Table"	• "Level" Symbolic representation of the message category: -  = Warning to be fixed -  = Error to be acknowledged -  = Error to be fixed • "State" - To be fixed = the error/warning is still pending and has to be fixed. - To be acknowledged = the error was fixed, but still needs to be acknowledged by the user • "Date (UTC+/-xxxx)" Date and time of the log message in the format: "DD.MM.YYYY, hh:mm:ss.ms" • "Code" Main diagnostic code and detailed diagnostic code (if available) • "Description" Depending on the log message, the description can contain the following attributes: - Line 1: Main text - Line 2: Detailed text - Line 3: Dynamic description • "Source" Depending on the log message, the source can contain the following attributes: - Line 1: Entity - Line 2: Origin - Line 3: Unit • "Developer information" Depending on the log message, the developer information can contain the following attributes: - Line 1: Filename - Line 2: function name - Line 3: Line number • "User ID" If the log message has a user ID, it is displayed here. • "Actions"

GUI element	Description
	This column displays buttons to perform actions for the selected diagnostic message: -  = Navigate to the alarm in the history overview. -  = Acknowledge error.

Related topics

➔ [Window - Logbook](#)

➔ [Tab - "History"](#)

13.3.3 Tab - "History"

The tab displays present information, warnings and error messages of the system. The tabular representation of the messages can be customized using the buttons in the command bar, see description below. The header of the table contains a filter symbol next to the column names, which can be used to enable the filter function (see "Filter function" below). The command bar contains the number of pending log messages and buttons that can be used to enable various functions.

Use the arrow buttons or the slider on the right-hand side of the table to navigate through the message table. The messages are reloaded automatically when the currently loaded log messages are approached and if there are still more log messages in the logbook.

Standard/developer view

The display of the logbook can be customized for different user types with regard to the contents displayed, see:

➔ [Dialog - "Settings"](#) (Logbook)

The standard view shows all required data important to commission the machine.

The developer view provides the full range of data columns and is aimed more at app developers.

Filter

As the number of log messages in the history overview can be very high, filters are provided to restrict the number of log messages to be displayed to improve the overview. Open the filters via the filter icons in the header line of the table.

The following filter types are provided to enter filters:

- **Level filter**

The level filter is only available in the "Level" column. All levels possible for the currently selected message type are displayed:

- **Message type "Diagnostics and trace":**
Error, warning, information
 - **Message type system:**
Emergency, Alarm, Critical, Error, Warning, Note, Information, Debug
- Any combination of levels can be selected.

- **Date filter**

With the date filter (only available in the "Date" column), four different time ranges can be defined:

- **Is equal to:**
Selection of a date (day, month, year).

All log messages from this day are displayed.

- **Before:**

Selection of a date and time.

All log messages that occurred before the defined day and time are displayed.

- **To:**

Selection of a date and time.

All log messages that occurred after the defined day and time are displayed.

- **Between:**

Selection of a date range, start time and end time.

All log messages that occurred within the specified date range and end time are displayed.

- **Text filter**

In the standard view, the text filter is provided for several columns in the table. A text can be entered as a search criterion. It is searched for it in all attributes displayed in the respective column.

Example: In the text filter of the "Description" column, the search text "Error" is entered. As this column visualizes the three attributes "Main text" (line 1), "Detailed text" (line 2) and "Dynamic description" (line 3), every log message containing the substring "Error" in at least one of the three attributes is displayed.

- **Complex filters**

In the developer view, complex filters are provided for multiple columns of the table. The difference to the text filter is that with the complex filters, a separate input option for a search text is displayed for each attribute of the column.

Example: In the complex filter of the "Description" column, search text "Text1" is entered for "Main text" (input field 1), search text "Text2" is entered for "Detailed text" (input field 2) and search text "Text3" is entered for "Dynamic description" (input field 3). All log messages that contain at least one of the 3 terms in their respective attributes are displayed, i.e. the search query is linked with "OR" (default setting).

The complex filters also allow to adjust the linking of input fields and to switch between "OR" and "AND". If the link is set to "AND", only the log messages that contain the set search text in the respective attributes are displayed in the example above.

Remark: If no search text is entered into an input field, this field is not included in the search!

- **Extension for complex filters**

The two columns "Source" and "User ID" have an extension in their input fields that allows the user to select one or more search texts from a list of all possible search texts for the respective attribute. The input field also allows to enter any text. All search texts in an input field are always linked with "OR" during the search. An "AND" link is not possible for the search texts of an attribute.

Call:

ctrlX OSide navigation "*Diagnostics* → *Logbook* → tab *"History"*"

Elements of the tab in the standard view

GUI element	Description
"Command bar"	• [x] items Number of listed entries. •  Button to delete all set bookmarks (only available if at least one bookmark was set). •  Button to jump to the previous bookmark (only available if at least one bookmark was set). •  Button to jump to the next bookmark (only available if at least one bookmark was set). •  Button to delete all set filters (only available if at least one filter was set). •  Button to reset and delete all pending diagnostics. •  Button to update the log message display. •  Button to download all displayed diagnostics, see: Logbook – Exporting •  Buttons to "expand" and "collapse" the lines to display further information on a diagnostic message.
"Table"	• "Level" Symbolic representation of the message category: -  = Information -  = Warning -  = Error • Additional symbolic representation of alarms -  = Warning to be fixed -  = Error to be acknowledged -  = Error to be fixed • "State" - To be fixed = the error/warning is still pending and has to be fixed. - To be acknowledged = the error was fixed, but still needs to be acknowledged by the user • "Date (UTC+/-xxxx)" Date and time of the log message in the format: "DD.MM.YYYY, hh:mm:ss.ms" • "Code" Main diagnostic code and detailed diagnostic code (if available) • "Description"

GUI element	Description
	Depending on the log message, the description can contain the following attributes: - Line 1: Main text - Line 2: Detailed text - Line 3: Dynamic description (only visible when expanded) • “Source” Depending on the log message, the source can contain the following attributes: - Line 1: Entity - Line 2: Origin • “Actions” This column displays buttons to perform actions for the selected diagnostic message: -  = Set bookmark. -  = Delete bookmark. -  = Acknowledge error.

Elements of the tab in the developer view

GUI element	Description
“Command bar”	• [x] items Number of listed entries. • “Additional message types” Selection of the message types to be displayed: - “None” Only diagnostic messages are displayed. - “Trace” Diagnostic and trace messages are displayed. - “System and trace” Diagnostic, trace and system messages are displayed. •  Button to delete all set bookmarks (only available if at least one bookmark was set). •  Button to jump to the previous bookmark (only available if at least one bookmark was set). •  Button to jump to the next bookmark (only available if at least one bookmark was set). •  Button to delete all set filters (only available if at least one filter was set). •  Button to reset and delete all pending diagnostics. •  Button to update the log message display. •  Button to download all displayed diagnostics, see:

GUI element	Description
	➔ Logbook – Exporting  Opens a dialog to configure the data columns (shown/hidden), see: ➔ Dialog – “Select columns” (Logbook)
“Table”	“Level” Symbolic representation of the message category: 🕒 = Information ⚠ = Warning ❌ = Error Additional symbolic representation of alarms ⚠ = Warning to be fixed ❌ = Error to be acknowledged ❌ = Error to be fixed “State” - To be fixed = the error/warning is still pending and has to be fixed. - To be acknowledged = the error was fixed, but still needs to be acknowledged by the user “Date (UTC+/-xxxx)” Date and time of the log message in the format: "DD.MM.YYYY, hh:mm:ss.ms" “Code” Main diagnostic code and detailed diagnostic code (if available) “Description” Depending on the log message, the description can contain the following attributes: - Line 1: Main text - Line 2: Detailed text - Line 3: Dynamic description (only visible when expanded) “Source” Depending on the log message, the source can contain the following attributes: - Line 1: Entity - Line 2: Origin - Line 3: Unit “Developer information” - Line 1: File name - Line 2: Function name - Line 3: Line number “User ID” Specification of the logged-in user whose action generated a logbook entry, for example when changing the user administration. “Actions”

GUI element	Description
	This column displays buttons to perform actions for the selected diagnostic message: -  = Set bookmark. -  = Delete bookmark (only available if a bookmark is set). -  = Acknowledge error (only available if there is an error to be acknowledged).

Related topics

➔ [Window – Logbook](#)

➔ [Tab – “Alarms”](#)

13.3.4 Tab – “Resources”

The tab “Resources” contains the following displays with performance information on the ctrlX device:

CPU usage

Graphical display of the load of the available CPU cores of the ctrlX device. Processor cores reserved either for system processes or for real-time processes are identified by icons:

Symbol	Description
	Processor core reserved by the user to process real-time processes.
	Processor core reserved by the system.

Use the filter symbol to select the CPU to be displayed in the performance diagram. All available CPUs are displayed in the factory setting.

The refresh rate provides the following settings:

- Each second
- Every 5 seconds
- Every 10 seconds
- Paused (CPU load not displayed)

RAM usage

Graphical display of RAM load on the ctrlX device:

- Total memory size
- Memory used
- Memory for buffer/cache
- Free memory



The names of the displayed partitions vary depending on the ctrlX device type used (real device / virtual device / alternative device platforms).

Call:

ctrlX OS side navigation “*Settings* → *window System Performance* → *tab Resources*”

Further information

➔ [Chapter 13.2.10 Window – “System Information” on page 297](#)

13.3.5 Tab – “Performance”

The tab “Performance” shows the current load of the available CPU cores and the internal device memory graphically in two XY-diagrams for the previous 60 seconds (continuously refreshed).

- **Diagram “CPU”**

Performance display of the available CPU cores in percent.

Select the  button to set the refresh rate of the performance display in three levels:

- High (1 second)
- Normal (5 seconds)
- Low (10 seconds)

Note:

The number of available CPU cores can vary depending on the ctrlX CORE device type:

- Real ctrlX CORE device
- Virtual ctrlX CORE device
- Alternative CORE hardware platform (e.g. Raspberry Pi)

Example: For a real ctrlX device of the types M3 or M4, the performance values of the CPU cores (CORE 0 to CORE 3) are shown.

- **Diagram “Memory”**

Shows the memory load (GB) of the device-internal memory in the following categories:

- “Used”
Assigned memory space (app/payload)
- “Buffer/Cache”
Assigned memory space (buffer and cache)
- “Free”
Free memory space
- “Total”
Total size of the internal device memory

Call:

ctrlX OS side navigation “Settings → window System Information → Tab → Performance”

Further information

→ [Chapter 13.2.10 Window – “System Information” on page 297](#)

13.3.6 Tab – “Processes”

The tab “Processes” displays all processes currently running on the ctrlX device in a table. The content is refreshed every 5 seconds.

Command bar

The command bar above the table provides the following functions:

- Display/total number of active processes
- Input field for the text search for processes

Call:

ctrlX OS side navigation “Settings → window System Performance → Tab → Processes”

Description

Information	Description
“App”	App from which the process was started
“Memory (MB)”	Memory space in the internal device memory used by the process
“CPU (%)”	CPU load caused by the process

Further information

→ Chapter 13.2.10 Window – “System Information” on page 297

13.3.7 Tab – “Overview”**Function:**

This tab contains an overview on all tasks and their current states.

An overview on all task time measurements is provided and these time measurements can be started, stopped and reset automatically.

Call:

ctrlX OS side navigation “Settings → Scheduler → Tab Overview”

Elements of the “Overview” tab

Table 38: Command bar of the “Overview” tab

GUI element	Description
“<x> items”	Number of Scheduler tasks
 Reset all timers	Resets all timers of all tasks
 Stop all timers	Stops all timers of all tasks
	Collapses all timers
	Expands all timers

Table 39: Table of the “Overview” tab

Column	Description
Task type (icon)	Shows the task type as icon  System task - There is always only one system task  PLC task - Optional  Automation task - Depends on the installed apps  Time measurement - Each task (except PLC task) has six measurements The time measurement is indented for better readability and the line is highlighted in blue
“Name”	Task name
“State”	Task state • OPERATING • SETUP • SERVICE

Column	Description
“Wdg error counter”	Shows the number of pending watchdog errors If no watchdog is configured for this task, this field is empty
“Cycle time”	Specifies in which cycle time the task should be called
“Minimum”	Minimum value of the time measurement
“Maximum”	Maximum value of the time measurement
“Average”	Average value of the time measurement
“Actions”	●  In the task: “Resets all timers from task” In the time measurement: “Resets timer” ●  In the task: “Starts all timers from task” In the time measurement: “Starts timer” ●  In the task: “Stops all timers from task” In the time measurement: “Stops timer” ●  Expands the timers of the task and shows them ●  Collapses the timers of the task

Further information

- ➔ [Chapter 13.3.8 Tab – “Configuration” on page 328](#)
- ➔ [Chapter 13.4.19 Dialog – “Scheduler settings” on page 357](#)
- ➔ [Chapter 13.3.9 Tab – “Startup” on page 331](#)
- ➔ [Chapter 13.3.10 Tab – “Debug” on page 332](#)
- ➔ [Chapter 13.2.11 Window – “Scheduler” on page 297](#)
- ➔ [Chapter 9.2 ctrlX CORE Scheduler on page 151](#)

13.3.8 Tab – “Configuration”**Function:**

In this tab, different Scheduler parameters can be modified.

Existing tasks can also be edited or deleted.

More tasks can also be created, edited or deleted for the Scheduler.

One or multiple callables can be added to or deleted from each task.

Call:

ctrlX OS side navigation “*Settings → Scheduler → Tab Configuration*”

Elements of the “Configuration” tab

A note on “Automatic program configuration” is displayed. The user can close it via the  symbol.

Attention: When the note is closed, this note is no longer displayed when the interface is restarted.

Table 40: Command bar of the “Configuration” tab

GUI element	Description
“<x> items”	Number of Scheduler tasks

GUI element	Description
	Switches on or off the “Automatic program configuration”. When switching off and switching on “Automatic program configuration”, a warning is displayed informing the user on the consequences of his/her actions. Switching is only possible either in the “Service” mode or in the “Setup” mode. In the “Operating” mode, the switch is inactive.
	Resets the Scheduler configuration. When this function is enabled, a warning notice is displayed informing the user that the current configuration is lost. This function is only available either in the “Service” mode or in the “Setup” mode. In the “Operating” mode, the switch is inactive.
	Collapses all tasks
	Expands all tasks
	Adds a task The modal dialog “Add task” opens (see Chapter 13.4.20 Dialog – “Add task” on page 358). If the “Automatic program configuration” is active, no task can be added manually. In this case, the button is inactive.

Table 41: Table of the “Configuration” tab

Column	Description
Task type (icon)	Shows the task type as icon  System task - There is always only one system task  PLC task - Optional  Automation task - Depends on the installed apps  Callable Icon is indented for better readability and the line is highlighted in blue
“Name”	Name of the task or callable
“Cycle time”	Only for task: Specifies in which cycle time the task should be called
“Priority”	Only for task: Priority of the task
“CPU Core”	Only for task: Specifies the CPU on which the task is to be executed
“Wdg error class”	Only for task and if watchdog is enabled: - Minor error (F0) - Critical error (F6)
“Wdg max. errors”	Only for task and if watchdog is enabled: Maximum number of subsequent errors

Column	Description
"Wdg task behavior"	Only for task and if watchdog is enabled: • DISABLE The task is not triggered anymore after triggering the watchdog • CONTINUE After triggering the watchdog, the task is still called until the maximum number of subsequent errors is reached
"Alias"	Only for callables: Alias name of the callable
"ID"	Only for callables: Unique ID specified by the system
"Actions " (for task)	•  Edits task parameters The "Task: <name>" sidebar of the task parameters opens (see Chapter 13.5.8 Sidebar – "Task: <name>" (Task parameter) on page 373). The name of the task cannot be changed in automatic mode. •  Adds a callable to the program The modal dialog "Add callable to <taskname>" opens (see Chapter 13.4.22 Dialog – "Add callable to <taskname>" on page 359). No callables can be added in automatic mode. In this case, the button is inactive. •  Deletes task The modal dialog "Delete task" is shown before the task is deleted (see Chapter 13.4.21 Dialog – "Delete task" on page 359). A task cannot be deleted in automatic mode. In this case, the button is inactive. •  Expands the task and shows all callables of the task •  Collapses the task. Callables are no longer displayed
"Actions" (for callable)	•  Shows the parameters of the callable (see Chapter 13.5.9 Sidebar – "<name_callable>" (Callable parameter) on page 374) •  Deletes a callable from the program (see Chapter 13.4.23 Dialog – "Delete callable" on page 361) A callable cannot be deleted in automatic mode. In this case, the button is inactive.

Further information

- ➔ Chapter 13.4.19 Dialog – “Scheduler settings” on page 357
- ➔ Chapter 13.3.9 Tab – “Startup” on page 331
- ➔ Chapter 13.3.10 Tab – “Debug” on page 332
- ➔ Chapter 13.3.7 Tab – “Overview” on page 327
- ➔ Chapter 13.3.8 Tab – “Configuration” on page 328
- ➔ Chapter 13.4.20 Dialog – “Add task” on page 358
- ➔ Chapter 13.4.21 Dialog – “Delete task” on page 359
- ➔ Chapter 13.4.22 Dialog – “Add callable to <taskname>” on page 359
- ➔ Chapter 13.4.23 Dialog – “Delete callable” on page 361
- ➔ Chapter 13.5.7 Sidebar – “schedMain” (Parameter) on page 373
- ➔ Chapter 13.5.8 Sidebar – “Task: <name>” (Task parameter) on page 373
- ➔ Chapter 13.5.9 Sidebar – “<name_callable>” (Callable parameter) on page 374
- ➔ Chapter 13.2.11 Window – “Scheduler” on page 297
- ➔ Chapter 9.2 ctrlX CORE Scheduler on page 151

13.3.9 Tab – “Startup”

Function:

In this tab, settings can be made for the entire system.

Call:

ctrlX OS side navigation “Settings → Scheduler” → ⚙ → Tab “Startup”

Elements of the “Startup” tab

GUI element	Description
“State”	Specifies to which target state the Scheduler should switch to after switching on • OPERATING Tasks are called and callables are ready-to-operate • SETUP Tasks are called and callables can be configured • SERVICE Tasks are not called, callables are not active and apps can be installed
“Timeout”	Specifies the time how long the Scheduler waits for additional components before beginning the startup
“Error reaction”	Specifies how the Scheduler responds on a timeout or missing configured components during startup • Stop The Scheduler stops the startup • Continue The Scheduler proceeds with the startup without the missing components

Further information

- ➔ Chapter 13.2.11 Window – “Scheduler” on page 297
- ➔ Chapter 13.4.19 Dialog – “Scheduler settings” on page 357

- ➔ [Chapter 13.3.10 Tab – “Debug” on page 332](#)
- ➔ [Chapter 9.2 ctrlX CORE Scheduler on page 151](#)

13.3.10 Tab – “Debug”

Function:

Enable or disable the Debug mode in this tab.

Call:

ctrlX OS side navigation “*Settings* → *Scheduler*” → ⚙ → Tab “Debug”

Elements of the “Debug” tab

GUI element	Description
“Debug mode”	Select  to enable the Debug mode and  to disable the Debug mode. The hardware watchdog is disabled when the Debug mode is active.

Further information

- ➔ [Chapter 13.2.11 Window – “Scheduler” on page 297](#)
- ➔ [Chapter 13.4.19 Dialog – “Scheduler settings” on page 357](#)
- ➔ [Chapter 13.3.9 Tab – “Startup” on page 331](#)
- ➔ [Chapter 9.2 ctrlX CORE Scheduler on page 151](#)

13.3.11 Tab – “Certificates”

Function:

The tab “Certificates” lists available certificates on the ctrlX OS device in a table.

If no certificate was loaded for a service on the ctrlX OS device, no command bar and no table, but the buttons  “Upload certificate file” and are displayed in the tab.

After a certificate upload, the command bar and the table with the entry of the uploaded certificate are shown in the tab.

Use a command in the command bar to load further certificates to the device. To execute different actions for the certificates, go to the selection menu in the “Actions” column.

Certificate files with multiple certificates

If a file with multiple certificates is uploaded to the device, the respective number of certificates, all with the same name (that of the certificate file), is displayed in the table after the upload. The entry in the table is made as per the order in the file. The name is displayed with the extension (n / m). 'n' stands for the number of certificates of the total number 'm' of certificates in the file. The “Category” is set to "Own" for everyone. The action “Rename” on one of the certificates affects all certificates in the certificate file.

Call:

ctrlX OS side navigation “*Settings* → *Certificates & Keys*” → Click on the service name (e.g. SSH) in the column Service → tab Certificates

Elements of the tab "Certificates"

GUI element	Description
Command bar	"[x] items" Number of listed certificates
	 "Upload certificate file" Select another certificate and upload it to the device
Table	"Name" Certificate name
	"Category" Certificate category
	"Issued for" User name to whom the certificate relates to
	"Valid from (UTC)" Validity period specifying from in UTC
	"Valid until (UTC)" Validity period specifying until in UTC
	"Renewal" Status of the automatic certificate renewal
	"Actions" ...
	"More settings" Open the selection menu with the possible actions for the respective certificate. Press ESC or click outside the menu to close it again
	 " Show details" More details of the certificate are shown
	 "Rename" Renaming the certificate on the ctrlX OS device. The name may not exceed 64 characters. No special characters except "-" and "_"
	 "Renewal" Calling the dialog to configure the automatic certificate renewal
	 "Download" Downloading the certificate from the ctrlX OS device to the engineering PC and saving it in ".pem" format. Browsers allowing to open the file directly in an editor have to save the file from the opening editor if the respective option is selected
	 "Delete" Deleting certificate from device

GUI element	Description
	If a certificate is either assigned to the category “Trusted” or “Rejected”, either the entry “Reject” or “Trust” is also shown. It is used to assign the certificate to the respective category in the “Actions” column Only for certificate files with multiple certificates:  “Extract” Extracting certificate from a certificate file with multiple certificates. Opens the “Extract certificate” dialog that removes the certificate from the original file and saves it to a separate file with a new name under the respective category. In the dialog, the current name of the certificate to be extracted is displayed under “Certificate to extract”. In the “Category of the extracted certificate *” field, select the matching category for the certificate under which it is to be saved. A new name that does not yet exist under the category has to be entered into the “Name of the extracted certificate *” field. The original certificate "<name>.pem (n /m)" is deleted from the list and the extracted certificate is displayed in the list of certificates of the service under "<new name>.pem".

Further information

- ➔ [Chapter 6.4.1 Introduction on page 63](#)
- ➔ [Chapter Certificates on page 64](#)
- ➔ [Chapter 13.2.16 Window – “Certificates & Keys” on page 301](#)
- ➔ [Chapter 13.4.7 Dialog – “Certificate renewal” on page 345](#)
- ➔ [Chapter 13.2.17 Window – “PKIs” on page 302](#)

13.3.12 Tab – “Keys”**Function:**

The “Keys” tab lists the available keys on the ctrlX OS device in a table.

If no key was loaded for a service on the ctrlX OS device, no command bar and no table are shown in the tab, but the buttons  “Upload key file” and  “Generate key”.

After a key upload, the command bar and the table with the entry of the key are shown in the tab.

To load more keys on the device and to generate private keys on the device, use the commands in the command bar.

To execute different actions for the keys (as per their type and category), go to the selection menu in the “Actions” column.

Call:

ctrlX OSside navigation “*Settings* → *Certificates & Keys*” ➔ Click on the name (e.g. SSH) in the list Service ➔ Tab Keys

Elements of the tab “Keys”

GUI element	Description
Command bar	“[x] items”: Number of listed keys
	 “Upload key file ” Loading a key to the device
	 “ Generate key” Generating a key of the type "private" on the device
Tab	The existing keys are shown in a table. Shown are: Name of the key, category, type, algorithm used, key length and information whether the key was created using the Hardware Security Module (HSM) of the control and the key status
	“Actions” ... “More settings ” Open the selection menu with the actions intended for the respective key. Press ESC or click outside the menu to close it again. Elements in the selection menu:  “Rename” Renaming the key on the ctrlX OS device. The name may not exceed 64 characters. No special characters except [-] and [_].  “ Download as *.pem” Download a key of the type “Public” or a key of the type “Private” that is not secured using HSM from the ctrlX OS device to the engineering PC and save it in ".pem" format.  “ Download as *.der” Download a key of the type “Public” or a key of the type “Private” that is not secured using HSM from the ctrlX OS device to the engineering PC and save it in ".der" format.  Delete Deleting key from device. If a key is assigned to the category “Trusted” or “Rejected”, the entry “Reject” or “Trust” is also shown used to assign the key to the respective category in the “Actions” column. The following actions can be executed for keys of the type “Private”:  “Generate self signed certificate” The “Generate self signed certificate” dialog opens.

GUI element	Description
	Make valid entries into the mandatory fields "Name **", "Validity (days) **" and "Common name or fully qualified domain name **" to ensure that - when selecting the "Generate" button, the certificate is generated and displayed in the "Certificates" tab of the service below the "Own" category. All other dialog fields are optional. Once the certificate creation is complete, the dialog is closed. Even self-signed certificates can only be used for test purposes or in protected networks.  "Enroll certificate" <i>Currently only supported for PKIs that use the "simple certificate enrollment protocol (SCEP)".</i> The "Enroll certificate" dialog opens. Make valid entries into the mandatory fields "PKI **" and "Common name or fully qualified domain name **" to ensure that - when selecting the "Enroll" button - the enrollment request is sent to the selected PKI that has to support the SCEP (simple certificate enrollment protocol). The other dialog fields are optional. The request is entered into the "Certificate enrollments" tab and displayed with its current status, see ➔ Chapter 13.3.15 Tab – "Certificate enrollments" on page 339.  "Generate certificate signing request" Creating and downloading a certification request to get the key signed from a certification authority.  "Show certificate signing requests" Showing the certification requests created with the key  "Downloading public part as *.pem" Downloading public part of a private key from the ctrlX OS to the engineering PC and saving it in ".pem" format.  "Downloading public part as *.der" Downloading public part of a private key from the ctrlX OS to the engineering PC and saving it in ".der" format. Browsers allowing to open the file directly in an editor have to save the file from the opening editor if the respective option is selected. To use the keys with the "encrypted" status, select the entry from the selection menu:  "Decrypt"

GUI element	Description
	The dialog to enter the key password and the new key name opens. A name is already suggested in the dialog. Click on the “Decrypt” button to store it without encryption and with the new name on the control and to show it in the list

Further information

- ➔ [Chapter 6.4.1 Introduction on page 63](#)
- ➔ [Chapter Keys on page 64](#)
- ➔ [Chapter 13.2.16 Window – “Certificates & Keys” on page 301](#)
- ➔ [Chapter “Generate certificate signing request” on page 65](#)
- ➔ [Chapter 13.4.17 Dialog – “Generate key” on page 356](#)
- ➔ [Chapter 13.3.15 Tab – “Certificate enrollments” on page 339](#)
- ➔ [Chapter 13.2.17 Window – “PKIs” on page 302](#)

13.3.13 Tab – “Revocation lists”

Function:

The “Revocation lists” tab lists the available revocation lists on the ctrlX OS device in a table.

If no revocation list was loaded for a service to the ctrlX OS device, no command bar and no table are displayed on the tab. Only the button  “Upload revocation list file” is displayed.

After uploading a revocation list file, the command bar and table with the entry of the uploaded revocation list are displayed in the tab.

Select the  “Upload revocation list file” in the command bar to load additional revocation lists to the device. The selection menu in the “Actions” column can be used to display detailed information for a revocation list or to delete it.

Call:

ctrlX OS side navigation “*Settings* → *Certificates & Keys*” → click on the name of the service (e.g. SSH) in the column Service → tab Revocation lists

Elements of the “Revocation lists” tab

GUI element	Description
Command bar	“[x] items” Number of listed revocation lists
	 “Upload revocation list file” Select another revocation list file and upload it to the device
Table	“Name” Name of the revocation list
	“Issued by” Common name of the issuing body
	“Valid since (UTC)” Date in UTC format since validity of the revocation list

GUI element	Description
	“Next update (UTC)” Date of the next update of the revocation list in UTC format “Actions” “More settings” Open the selection menu with the possible actions for the respective revocation list. Press ESC or click outside the menu to close it again  “Show details” Further details of the revocation list are shown  “Delete” Delete revocation list from device

Further information

- ➔ [Chapter 6.4.1 Introduction on page 63](#)
- ➔ [Chapter 13.5.5 Editor – Certificates & Keys on page 368](#)

13.3.14 Tab – “PKCS12 containers”**Function:**

In the tab “PKCS12 containers” the containers available on the ctrlX OS device for a service are listed in a table in .pfx format.

If no container has yet been loaded or created for a service on the ctrlX OS device, no command bar or table is displayed on the tab, only the buttons  “Upload PKCS12 container file” and  “Create PKCS12 container” are displayed.

After uploading or creating a container file, the command bar and table with the entry of the uploaded PKCS12 file are displayed in the tab.

Select the  “Upload PKCS12 container file” button in the command bar to load additional container files to the device.

Select the  “Create PKCS12 container” button in the command bar to open the dialog to create a container.

Go to the selection menu in the “Actions” column of the table to import the keys and certificates in a PKCS12 container for the service using the  “Import content” button.

The  “Download” button is used to download a container file to the local engineering PC. To delete the container file from the ctrlX OS device, select the  button.

Call:

ctrlX OSside navigation “*Settings* → *Certificates & Keys*” ➔ Click on the service name (e.g. SSH) in the column Service ➔ tab PKCS12 containers

Elements of the tab “PKCS12 containers”

GUI element	Description
Command bar	“[x] items” Number of container files listed

GUI element	Description
	 "Upload PKCS12 container file" Select additional PKCS12 containers and load them to the device
	 "Create PKCS12 container"
Table	"Name" Name of the PKCS12 container
	"Actions" ...
	"More settings" Open the selection menu with the possible actions for the respective container. Press ESC or click outside the menu to close it again:
	 "Import content" Importing the contents of the container into Certificates & Keys
	 "Download" Downloading the PKCS12 container to the local engineering PC
	 "Delete" Deleting PKCS12 from device

Further information

- ➔ [Chapter 6.4.1 Introduction on page 63](#)
- ➔ [Chapter 13.4.18 Dialog – "Create PKCS12 container" on page 357](#)

13.3.15 Tab – "Certificate enrollments"**Function:**

In the "Certificate enrollments" tab, the requests generated by the ctrlX OS device with private keys are listed as table. The "Enroll certificate" button can also be used to generate further requests with private keys.

If no certificate enrollment request has yet been generated for a service by the ctrlX OS device, no command bar and no table are displayed in the tab, only the note "No certificate enrollments available" and the button  "Enroll certificate" are displayed.

If - via the button  "Enroll certificate" or in the "Keys" tab with a private key using actions ➔ "Enroll certificate" - a certificate request was sent to a PKI (has to support SCEP), the command bar and table including information on the requests in the columns "Requested for", "Requested by PKI" and "Status" are shown in the tab.



After switching the device off/on, the certificate enrollment requests are deleted. This also applies to requests that were still in the "pending" state. In the tab, only the note "No certificate enrollments available" and the button  "Enroll certificate" are displayed.

Call:

ctrlX OS side navigation “*Settings → Certificates & Keys*” → Click on the name (e.g. SSH) in the column Service → Tab Certificate enrollments

Elements of the “Certificate enrollments” tab

GUI element	Description
Command bar	“[x] items”: Number of listed requirements  “Enroll certificate” The “Enroll certificate” dialog to create a certificate enrollment request to a PKI opens. In the mandatory field “Keys” *, select the private key used to make the request. Make valid entries into the mandatory fields “PKI” and “Common name or fully qualified domain name *” to ensure that - when selecting the “Enroll” button - the enrollment request is sent to the selected PKI that has to support the SCEP (simple certificate enrollment protocol). The other dialog fields are optional
Table	The existing requests are shown in a table. Requested for Common name of the requester Requested from PKI PKI (has to support SCEP) to which the request was sent Status Status of the request • running Transmitted to the PKI for processing • pending (Manual) processing is still pending for the PKI • done Certificate was enrolled and transmitted to the ctrlX device • failed, see logbook for details Certificate enrollment request failed. The reason is in the logbook message

Further information

- ➔ [Chapter 6.4.1 Introduction on page 63](#)
- ➔ [Chapter Keys on page 64](#)
- ➔ [Chapter 13.2.17 Window – “PKIs” on page 302](#)

13.3.16 Tab – “Restore”**Call:**

ctrlX OS side navigation “*Settings → Backup & Restore → Tab Restore*”

Function

Go to the tab “Restore” to restore a previously created backup on a ctrlX device.

The following data is loaded from the backup file to the ctrlX device:

- Active app data
- Communication settings
- Date and time
- Configuration of users and permissions
- Certificates and keys (only if a password is specified when the backup was created)

Restoring a backup on the ctrlX device

Refer to [↗ Backup – Applying to ctrlX device](#)

13.3.17 Tab – “Back up”**Call:**

ctrlX OS side navigation “*Settings → Backup & Restore → Tab Back up*”

Function

Go to the tab “Back up” to create a backup of the ctrlX device.

The following data is read from the ctrlX device and written to a backup file:

- Active app data
- Communication settings
- Date and time
- Configuration of users and permissions
- Certificates and keys (only if a password is specified when the backup is created)

Creating backup

Refer to [↗ Chapter 6.6.2 Backup – Creating from ctrlX device on page 79](#)

13.4 Dialogs**13.4.1 Dialog – “Add VLAN interface”**

The dialog is used to create a virtual network.

Virtual networks are used to isolate data traffic, improve network security and optimize network performance, see also:

[↗ Adding and configuring VLAN interface](#)

Related topics

[↗ Configuring IP settings](#)

[↗ Window – “Network interfaces”](#)

Call

In the ctrlX CORE web interface:

ctrlX OS side navigation “*Settings → Network interfaces*” Click on [\[+\]](#) in the header of the window and click on “Add VLAN interface”

Setting options

	Description
“Parent interface”	Selection of the real interface on which the virtual network is to be operated.
“VLAN ID”	ID number of the VLAN network to be created.

13.4.2 Dialog – “Add a bandwidth”

The dialog is used to configure and to limit the network bandwidth.

By limiting the network bandwidth, data flow can be regulated in a network by defining limits on how much data can be transmitted to avoid overloading and ensure a fair distribution of bandwidth among users or applications.



Note that an interface can only be assigned to one bandwidth!

Related topics

- ➔ [Limiting network bandwidth](#)
- ➔ [Configuring IP settings](#)
- ➔ [Window – “Network interfaces”](#)

Call

ctrlX OS side navigation “*Settings → Network interfaces*”

To open the network settings, click on [🔗] in the header

Open “*Bandwidth management tab*” in the network settings. Click on [⛶]

Setting options

	Description
“Name”	Name of the bandwidth limitation (alias) One bandwidth limit can be configured for each interface. Thus, assign a unique name.
“Limit”	Bandwidth value in MB/s to which the assigned network interfaces are limited.
“Burst”	Amount of data in KB that may temporarily exceed the allocated bandwidth limit. This is useful when there is a sudden, increased need for data transmission, such as when loading a large web page or downloading a file.
“Description”	Field for individual description of the database limitation (optional)
List of available network interfaces	Select the network interfaces whose bandwidth is to be limited. Only one bandwidth limit can be assigned to each network interface.

13.4.3 Dialog – “Network settings”

The dialog is used to configure the following network settings:

- Configuration or limitation of the network bandwidth.
- Option for dynamic determination of the maximum package size (PMTUD).

Related topics

- ➔ [Configuring IP settings](#)
- ➔ [Window – “Network interfaces”](#)

Call

In the ctrlX CORE web interface:

ctrlX OS side navigation “*Settings* → *Network interfaces*” and click in the header on [⚙]

Setting options

	Description
“Bandwith management”	The data flow in a network can be regulated by limiting the network bandwidth. Thus, limits can define how much data can be transferred to avoid overloading and ensure a fair distribution of bandwidth among users or applications, see also: ↗ Limiting network bandwidth
“Path MTU Discovery”	Path MTU Discovery (PMTUD) dynamically determines the maximum package size that a network path can process. When this option is enabled, the sender starts with an assumed MTU, adapts based on feedback (ICMP "Fragmentation Needed" messages) and iterates until it finds the optimum data package size to minimize data fragmentation for an efficient data transmission, also refer to: ↗ Option for dynamic determination of the maximum package size (PMTUD)

13.4.4 Dialog – “Select columns” (Logbook)

The dialog is used to configure the data columns in the History tab. Data columns can be shown or hidden as required except for "Level" and "Date". They are always displayed.

The dialog provides two opposing lists to select/deselect data columns:

- “Available columns”
Data columns currently deselected and not displayed in the log history are listed here.
- “Selected columns”
Data columns displayed in the log history are listed here.

Selecting/deselecting columns

In the factory setting of the ctrlX device, all available data columns are always displayed in the log history.

Columns are selected/deselected by moving column entries to the respective list (shown or hidden).

Thus, either individual column entries or all column entries in a list can be selected and moved to the opposite list using the arrow buttons.

Saving applies the setting to the log history.

Restriction

The column selection is only available in the History tab in the developer view, see:

[↗ Dialog – “Settings” \(Logbook\)](#)

Call

ctrlX OS side navigation “*Diagnostics → Logbook → tab “History”*” click in the header on []

Related topics

↪ [Tab – “History”](#)

13.4.5 Dialog – “Settings” (Logbook)

The dialog is used to configure the following logbook settings:

- “Developer view”
The logbook has two views to customize the displayed information content for different user groups. If the developer view is enabled, extended diagnostics, trace messages and filter options are also displayed.
- “Set logbook to persistent”
If this option is enabled, all logbook entries are saved permanently and are not lost even after restarting the ctrlX device.
- “Time zone setting”
Selection of the source for the logbook time zone (time stamp of the logbook messages).

Call

ctrlX OS side navigation “*Diagnostics → Logbook*” click in the header on []

Settings	Description
“Developer view”	 : Developer view disabled = reduced information content / suitable for commissioning.  : Developer view enabled = extended diagnostics, trace messages and filter options.
“Set logbook to persistent”	 : Persistent storage disabled = logbook entries are lost, e.g. if the ctrlX device is restarted.  : Persistent storage enabled = logbook entries are saved permanently and remain even after a restart of the ctrlX device.
“Time zone setting”	Selection of the source for the time zone setting of the logbook. Available sources: • “Browser”: The time zone is applied from the web browser on which the CORE web interface is currently displayed. • “Device”: The time zone is applied from the ctrlX device from which the logbook messages originate.

Related topics

↪ [Window – Logbook](#)

13.4.6 Dialog – “DCHP settings”



The dialog is used to configure the advanced settings for IPv4 and IPv6. The settings made in the dialog equally affect IPv4 and IPv6!

Related topics

➔ [Configuring IP settings](#)

➔ [Window – “Network interfaces”](#)

Call

In the ctrlX CORE web interface:

ctrlX OS side navigation “*Settings* → *Network interfaces* → Click on the respective network tile to open the editor → Open the tab IPv4 or IPv6 in the editor” and click on [⚙]

Setting options

	Description
“Use DNS”	If enabled: DNS servers received from the DHCP server are used (higher priority than all statically configured DNS servers)
“Use NTP”	If enabled: NTP servers received from the DHCP server is used for the time synchronization (higher priority than all statically configured NTP servers)
“Use MTU”	If enabled: MTU received from the DHCP server is set as MTU of the network interface
“Route metric”	The entered value is used as the default metric for automatically added routes
“DCHP client identifier”	Specifies how the interface identifies itself at the DHCP server: • “MAC address” To be used for a Windows DHCP server with IPv4 address reservation • “RFC 4361” For all other use cases

13.4.7 Dialog – “Certificate renewal”

Function:

In the “Certificate renewal” dialog, it can be configured from which PKI server and during which period before the validity ends the certificate can be renewed automatically. This applies to a certificate used on the ctrlX device. Use the dialog to manually trigger the renewal immediately.

Call:

ctrlX OS side navigation “*Settings* → *Certificates & Keys*”, click on the name (e.g. SSH) in the list “Service” → Tab Certificates → “Actions” → ... → “Renewal”

Enabling automatic certificate renewal

Open the dialog via the “Certificate renewal” of the certificate. Select the server endpoint in the “PKI” field. This is the server endpoint from which the certificate renewal is retrieved. Select the key in the “Private key” used to create the certificate signing request. In the “Renewal period before end of



validity (days) ** field, specify the number of days the automatic certificate renewal is triggered before the certificate expires. If the fields are filled with valid values in the dialog, click on the button  to save the configuration and to close the dialog. The status of the certificate renewal for the certificate is refreshed in the “Renewal” column of the table.

To pause the automatic certificate renewal, deselect “Enable automatic certificate renewal” before saving.

Select the “Renew now” button in the dialog and the created configuration can then trigger the certificate renewal manually

Status of the certificate renewal

- not configured
The connection to the PKI server was not configured for the certificate. No automatic renewal is active
- paused
A connection to the PKI server was configured for the certificate. No automatic renewal is active
- automatically
A connection to the PKI server was configured for the certificate. The automatic renewal is active

Elements of the “Certificate renewal” dialog

Dialog element	Description
“Certificate”	Display of the certificate name
“Valid until:”	Display of the expiry of the certificate as: yyy-mm-dd hh:mm:ss
“PKI”	Selection field of the PKI server requesting the certificate renewal, see ↪ Chapter 13.2.17 Window – “PKIs” on page 302
“Private key”	Selection field of the private key used to create the certificate signing request
“Renewal period before end of validity (days) **”	Number of days the renewal is triggered automatically before the certificate becomes invalid (default: 7 days)
“Enable automatic certificate renewal”	If the checkbox is selected, the certificate is renewed automatically at the specified PKI server at the given point in time (default: selected)
“Renew now”	Button to manually trigger the certificate renewal with the connection set irrespective of whether the automatic certificate renewal is enabled

Dialog element	Description
buttons 	Buttons to save/cancel the configuration. Click on the button  to save changes and to close the dialog only if valid entries were made in the fields "PKI", "Private key" and "Renewal period before end of validity (days) *". Otherwise, no change is saved and the dialog remains open

Further information

- [↪ Chapter 13.2.17 Window – "PKIs" on page 302](#)
- [↪ Chapter 6.4.1 Introduction on page 63](#)
- [↪ Chapter 13.2.16 Window – "Certificates & Keys" on page 301](#)

13.4.8 Dialog – "Add remote logging server"

The dialog is used to configure an external server connection and to save logbook entries of the ctrlX device on a Syslog server.

Related topics

- [↪ Remote Logging – Configuring external server](#)
- [↪ Window – Remote Logging](#)

Call

In the ctrlX CORE web interface:

ctrlX OS side navigation *"Diagnostics → Logbook → Click in the header on "Remote Logging" → Click on "Add server"*

Configuring the connection

The following entries are required in the dialog to configure the connection to the logging server:

Connection data	Description
Name	Symbolic name of the server connection
Host / IP address / domain	IP address of the server or the host PC
Port	Port of the server or the host PC
Protocol	Network protocol to be used: • TCP • UDP • TLS
Message format	Syslog protocol to be used: • RFC3164 • RFC5424

13.4.9 Dialog – “Settings”

Dialog call

Via the side navigation in the window “*Settings → Apps → command bar → Click on [Settings]*”



In smaller browser windows, the call is in the extended options [⋮]

The following settings can be made in the dialog:

Allow installation from unknown source

Due to IT security, only apps can be installed and operated on the ctrlX device in the default setting if their source is known and approved by Bosch Rexroth. However, the default setting can be changed to allow the installation and operation of third-party apps or development states as *.snap on your ctrlX device.



The installation and operation of third-party apps are at own risk!

Activate app storage on the ctrlX CORE

Apps can be saved in the internal app memory of the ctrlX device without an installation on the device.

Thus, the "stock" of apps can directly be saved on the ctrlX device and installed when required.



Saving apps in the app storage reduces the device-internal memory capacity. Thus, remove unused apps from the app storage.

13.4.10 Dialog – “Add user”

Function:

To create a new user on the control, go to the dialog “Add user”.

Call:

ctrlX OS side navigation “*Settings → Users & Permissions → Users → button*”


Elements of the Add user dialog

	Function
Username	Field to enter the new user name
Password policy	Use the drop-down field to assign a valid password policy for the user. The field is pre-assigned with the default password policy "Default", see Chapter 13.4.12 Dialog – “Password policies” on page 349
Password	Input field for the new password
“Confirm password”	Input field to confirm the new password (has to be identical to “Password”)
“Add user”	Command to confirm the dialog and to create the new user
	Command to cancel the dialog

Further information

- ➔ [Chapter 6.1.1 Introduction on page 42](#)
- ➔ [Chapter 6.1.9 Creating a new user on page 45](#)
- ➔ [Chapter 6.1.13 Username and password conventions on page 46](#)

13.4.11 Dialog – “Change password”**Call:**

ctrlX OS ctrlX CORE side navigation “<user name> → *Change password*”

Function:

To change the password of the logged in user, go to the dialog “Change password”.

Dialog elements

	Function
“Username”	Shows the current username
“Old password”	Input field for the currently valid password (Required to change the password)
“New password”	Input field for the new password
“Confirm new password”	Input field to confirm the new password (has to be identical to “New password”)
“Change password”	Command to confirm the dialog and to change the password Prerequisites: • The current password is correct • The entered password in “New password” and “Confirm new password” match.
“Cancel”	Command to cancel the dialog without password change

Further information

- ➔ [Chapter 6.1.1 Introduction on page 42](#)
- ➔ [Chapter 6.1.7 Changing the user password on page 44](#)

13.4.12 Dialog – “Password policies”**Elements of the Password policies dialog**

The header of the editor is the command line with the number of listed elements on the left-hand side and the  button to create a new password policy. Below is the list of existing password policies.

Function:

To change the password policies and to create more password policies, go to the "Password policies" dialog.

To create more password policies, click on the  button in the command line of the dialog. The editor for the password policy opens. Except for the fields "Name *" and "Identifier", all fields are pre-assigned with the values of the "default policy". Alternatively, copy from an existing policy using the button . Open the editor with the fields pre-assigned from this policy, modify them and save the password policy under a new name with a new automatically generated or self-assigned identifier. Except for the "Default password policy", a policy can be deleted from the list via the  Delete button if it is not used.

To change a password policy, click on the  Edit button next to its name. The ID of a password policy is used for the backup functionality. It can no longer be changed after creating the policy.



Valid characters for a password policy ID

A self-assigned ID may contain lower and upper case letters and numbers. Special characters other than hyphens and underscores are not permitted. The identifier must not begin or end with a hyphen or underscore.



Password policy "Default"

This is the default policy and should not be weakened, but only adapted so that it is always the one with the strongest rules of all password policies.

Call:

ctrlX OS side navigation *"Settings → Users & Permissions → More settings → Password policies"*

Elements of the editor "Password policy"

	Function
"Name *"	Name of the password policy
"Identifier (autogenerated if not specified)"	ID of the password policy Assign a "descriptive" name (permitted characters: Valid characters for a password policy ID on page 350) or leave the field blank to have an ID generated automatically. After creating the password policy, the ID can no longer be changed
"Minimum password length"	Minimum password length (default value is 12)
"Minimum lowercase characters (a-z)"	Minimum number of lower case letters (default value is 1)
"Minimum uppercase characters (A-Z)"	Minimum number of upper case letters (default value is 1)
"Minimum digits (0-9)"	Minimum number of numerical digits (default value is 1)
"Minimum special characters (!\$%&/()=? etc.)"	Minimum number of special characters
"Password must not contain username"	This option determines whether the username may be part of the password: ☒ Enabled by default = (user name must <u>not</u> be part of the password)

	Function
“Password history length”	The option specifies the number of passwords that can be saved in the password history: • Default value is 5 (if 0, the password history is disabled) • Up to 10 passwords can be saved in the password history
“Password expiration time in days (0 for unlimited)”	Specifies the validity period of the password and the date when a new password has to be specified. It is specified in days. • The password validity does not expire when the value is 0
Button “OK”	Use the command to apply changes in the password policies to the control
Button “Cancel”	This command closes the dialog. Changes are not applied to the control

Further information

- ➔ [Chapter 6.1.1 Introduction on page 42](#)
- ➔ [Chapter 6.1.7 Changing the user password on page 44](#)

13.4.13 Dialog – “Session policies”

Call:

ctrlX OS side navigation “*Settings → Users & Permissions → More settings → Session policies*”

Function:

To change the session policies, go to the dialog “Session policies”.

Dialog elements

	Function
“Logout after inactivity in minutes (0 for unlimited)”	Logout after a certain time of inactivity, in minutes (Default value is 0 = Function disabled)
“Logout after session duration in minutes (0 for unlimited)”	Logout after a certain time without further condition, in minutes (Default value is 480)
“Maximum number of sessions per user (0 for unlimited)”	Maximum number of sessions per user (Default value is 0 = Unlimited number of sessions)
“Number of login attempts without subsequent waiting period (0 for unlimited)”	Maximum number of subsequent failed logins without waiting time (Default value is 0 = Unlimited number of failed login attempts)
“Waiting time after login failed (seconds)”	Waiting time in seconds after a specified number of subsequent logins (Default value is 300)
Button “OK”	Use the command to apply changes in the session policies to the control.
Button “Cancel”	This command closes the dialog. Changes are not applied to the control.

Further information

- ➔ [Chapter 6.1.1 Introduction on page 42](#)
- ➔ [Chapter 6.1.3 Users, groups, permissions, sessions on page 43](#)

13.4.14 Dialog – Login banner**Function:**

Enter a text to be shown in the banner of the “Login” dialog via the dialog “Login banner”.

Call:

ctrlX OS side navigation “*Settings → Users & Permissions → More settings → Login banner*”

Elements of the “Login banner” editor

GUI element	Description
“Input field”	“Login banner” Field to enter the text to be shown in the banner of the “Login” dialog
“Buttons”	 Any changes made are saved on the ctrlX OS device. If no changes were made in the input field, the button remains disabled.  Editing is aborted and the editor is closed without saving

Entering text

There is an automatic line break at the outer right of the editor window. This does not correspond to the display in the Login banner. There is a line break after approximately 60 characters.

13.4.15 Dialog – “Remote authentication”

Function:

Go to dialog “Remote authentication” to activate the authentication via a remote server and to configure the connection to this server.



Connections to the authentication server are supported with the RADIUS protocol and the LDAP protocol.

To load the latest saved configuration and to enable the respective configuration fields, select "Authentication service type".

If the remote authentication is activated, the authentication takes place via the server during login. The connection settings of the server have been configured in the dialog. To save them, click on the Button OK. Use the "Force local authentication" option in the login dialog to change it for the current login process. Subsequently, the settings on ctrlX OS device are used.

Call:

ctrlX OS side navigation *“Settings → Users & Permissions → More settings → Remote authentication”*

Elements of the dialog Remote authentication for the RADIUS server

	Function
Checkbox “Enable remote authentication”	Enables the authentication function via a remote server and adds an option to the login dialog. If this option is enabled, the local settings can be used for the current login
Drop-down menu “Authentication service”	Authentication service selection on the remote server. Select “RADIUS” for servers with radius protocol
Input field “Server IP address or host name”	IP address of the RADIUS server
Input field “Server port”	Port on the RADIUS server cannot be reached via the service. (Default value is 1812)
Drop-down menu “Transport layer protocol”	Selection whether the communication to the RADIUS server is established via the UDP or TCP protocol. (Default value is UDP)
Input field “Client secret”	"Shared secret" to communicate with the RADIUS server. This is specified by the server admin and has to be obtained by the server admin
Drop-down menu “Authentication protocol”	Selection of the authentication protocol “PAP” used or “MS-CHAPv2” to connect to the RADIUS server. (Default value is MS-CHAPv2)
Input field “Vendor ID”	ID of the RADIUS server operator. The ID is specified by the server admin and has to be obtained by the server. (Default value is 0)

	Function
Input field "Groups attribute ID"	The "Groups attribute ID" is used to assign the user to one of the local permission groups. The assignment takes place on the RADIUS server. The ID is specified by the server admin and has to be obtained by the server admin. (Default value is 0)
Button 	Use the command to apply changes made in the Remote authentication to the control
Button 	This command closes the dialog. Changes are not applied to the control

Elements of the Remote authentication dialog for the LDAP server

	Function
Checkbox "Enable remote authentication"	Enables the authentication function via a remote server and adds an option to the login dialog. If this option is enabled, the local settings can be used for the current login
Drop-down menu "Authentication service"	Authentication service selection on the remote server. Select "LDAP" for the directory server with LDAP protocol
Input field "Server IP address or host name"	IP address of the LDAP server
Input field "Server port"	Port on the LDAP server cannot be reached via the service. • Default port for unsecure connections is 389 • Default port for TLS connections is 636
Checkbox "Use transport layer security (TLS)"	Selection whether the communication to the LDAP server should be encrypted. When enabled, three drop-down selection fields are displayed. The certificates and the key for the TLS connection are selected in the fields. For this purpose, load them before to the ctrlX device under <i>"Settings → Certificates & Key → Network security"</i> . To directly upload the certificates and keys from the Remote authentication dialog, select the  button next to the selection fields. It can be set additionally via the "Skip verification of server certificate" checkbox whether checking the server certificate should be omitted
Drop-down menu "Server type"	Selection of the server type used to connect to the LDAP server. "Open LDAP" and "Active Directory" are supported
Input field "Base DN"	Distinguished name (DN) of the starting directory for the prompt on the server. The DN is written from the child to the parent hierarchy levels from left to right. Each hierarchy level is written as follows: keyword=object
Input field "Read-only user"	User with the permission on this LDAP server to read from these groups and persons
Input field "Read-only password"	Password of the read-only user on the LDAP server

	Function
Input field "User filter"	LDAP filter for users that should get permissions on the ctrlX device. When setting the filter, ensure that the query result does not exceed 500
Input field "Group filter"	LDAP filter for groups that should get permissions on the ctrlX device. When setting the filter, ensure that the query result does not exceed 500
Input field "Username attribute"	Attribute of the usernames on the LDAP server
Input field "Groupmembers attribute"	Attribute of the group members on the LDAP server
Button 	Use the command to apply changes made in the Remote authentication to the control
Button 	This command closes the dialog. Changes are not applied to the control

Further information

- [↪ Chapter 6.1.1 Introduction on page 42](#)
- [↪ Chapter 6.1.3 Users, groups, permissions, sessions on page 43](#)

13.4.16 Dialog – "Easy service access"

Call:

ctrlX OS side navigation *"Settings → Users & Permissions → More settings → Easy service access"*

Function:

To enable the option, go to the dialog "Easy service access" (respective right has to be assigned to the user). This allows the customer service to create a "System user" on the ctrlX OS device before previous authentication and to access the operating system level of the device.

Elements of the "Easy service access" dialog

GUI element	Description
"Checkbox"	"Allow easy service access" Checkbox to enable the option
"Buttons"	"OK" Any changes made are saved on the ctrlX OS device. If no change was made, the button remains disabled. "Cancel" The process is canceled and the editor is closed without saving

Further information

- [↪ General information on page 45](#)

13.4.17 Dialog – “Generate key”

Function:

To generate a private key for an installed service on the control, go to the dialog “Generate key”.

Different algorithms can be selected to generate the key.

Keys with the RSA algorithm and a key length of 2048 bit as well as the algorithm P-256 (based on the elliptic curve) can be created optionally supported by the Hardware Security Module (HSM) of ctrlX CORE.



Option “Secure key by HSM”

The generated keys are bound to the device on which they were created. They cannot be transferred.

Call:

ctrlX OS side navigation “*Settings* → *Certificates & Keys*”, click on the name (e.g. SSH) in the list “Service” → Tab Keys →

Generating the key

Click on the button to close the dialog and the key is automatically entered into the tab “Keys” in the table if the fields are filled in correctly. It is displayed there with the “Category: Own” and the “Type: Private”. While the key generation is running, “Generating” is shown with a progress bar in the “Status” column of the key above. After the key generation, the state “Available” is shown in the “Status” column.

Dialog elements

	Function
“Name”	Key name to be generated. The name must not contain any spaces. It has to start with a lower-case letter or an underscore and may only contain small letters (a-z), digits (0-9) and the special characters [-] and [_]
“Extension”	Drop-down menu for the file format of the created key. Currently, only “.pem” is available and pre-assigned to the field
“Algorithm”	Selecting the algorithm used for the key generation. Default is "RSA 4096" Further selectable algorithms are "RSA 2048", "RSA 8192" as well as "ED25519", "P-256", "P-384" and "P-521" that are based on elliptic curves
“Option Secure key by HSM”	When selecting the option, the key is generated with the HSM of the ctrlX CORE. The option can currently only be selected if either "RSA 2048" or "P-256" is selected in the field “Algorithm”
Button 	Use the command to generate the key as shown in the dialog on the control
Button 	This command closes the dialog. No key is generated on the control

Further information

- ➔ [Chapter 13.2.16 Window – “Certificates & Keys” on page 301](#)
- ➔ [Chapter 6.4.1 Introduction on page 63](#)
- ➔ [Chapter 13.3.12 Tab – “Keys” on page 334](#)

13.4.18 Dialog – “Create PKCS12 container”**Function:**

Go to the dialog “Create PKCS12 container” to create a new container with an existing certificate on the device and the corresponding private key as well as CA certificates in .pfx format.

To create a container, select at least one client certificate or one CA certificate in the dialog.

Call:

ctrlX OS side navigation “*Settings* → *Certificates & Keys*” → Click on the name of the service (e.g. SSH) in the column Service → tab “PKCS12 containers” → Click on the button + or  “Create PKCS12 container”

Elements of the “Create PKCS12 container” dialog

Field	Function
“Container name *”	Mandatory input field to enter a name for the new PKCS12 container. The .pfx file format is fixedly set by default
“Client certificate to be added (optional)”	Drop-down selection box to add one of the certificates for this service on the control to the container or select “Do not add a client certificate”
“Private key to be added (optional)”	Drop-down selection box to add the private key belonging to the selected certificate to the container or select “Do not add a private key”
“CA certificates to be added (optional)”	List of CA certificates on the control for this service, which can be added to the container by selecting the checkbox (multiple selection possible)
“Password *”	Mandatory input field for the password to be able to import the container content (on another device)
“Confirm password *”	Mandatory input field to confirm the password (has to be identical to “Password *”)
	Button to create the new PKCS12 container. The dialog is closed if the fields are filled in correctly. The container is created and displayed in the table
	Button to cancel the creation of the PKCS12 container

Further information

- ➔ [Chapter 13.3.14 Tab – “PKCS12 containers” on page 338](#)
- ➔ [Chapter 13.3.11 Tab – “Certificates” on page 332](#)

13.4.19 Dialog – “Scheduler settings”**Function:**

In this dialog, you can make settings for the entire system.

The dialog consists of two tabs

- “Startup”

refer to [↗ Chapter 13.3.9 Tab – “Startup” on page 331](#)

- “Debug”

refer to [↗ Chapter 13.3.9 Tab – “Startup” on page 331](#)

Call:

ctrlX OS side navigation “Settings → Scheduler → Command bar Scheduler” → 

Further information

- [↗ Chapter 13.3.9 Tab – “Startup” on page 331](#)
- [↗ Chapter 13.3.10 Tab – “Debug” on page 332](#)
- [↗ Chapter 13.2.11 Window – “Scheduler” on page 297](#)
- [↗ Chapter 9.2 ctrlX CORE Scheduler on page 151](#)

13.4.20 Dialog – “Add task”

Function:

To create a new task in the Scheduler, go to the “Add task” dialog.

Call:

ctrlX OS side navigation “Settings → Scheduler → Tab Configuration” → 

Dialog elements

GUI element	Function
“Name”	Field to enter a name
“Extended mode”	The dialog can be extended optionally. The following attributes can be adjusted: • “CPU Core”: – 0 - non-realtime – 1 - non-realtime – 2 - realtime (default) – 3 - realtime • “Event”: “cyclic” - Attribute cannot be changed • “Priority”: Value between 11 (high) and 139 (low) Default value: 40 • “Cycletime”: 500 µs or integral multiple of 1000 µs
“Add”	Task is created
“Cancel”	Process is aborted

Further information

- [↗ Chapter 13.5.8 Sidebar – “Task: <name>” \(Task parameter\) on page 373](#)
- [↗ Chapter 13.4.22 Dialog – “Add callable to <taskname>” on page 359](#)
- [↗ Chapter 13.4.21 Dialog – “Delete task” on page 359](#)
- [↗ Chapter 13.3.8 Tab – “Configuration” on page 328](#)
- [↗ Chapter 13.2.11 Window – “Scheduler” on page 297](#)
- [↗ Chapter 9.2 ctrlX CORE Scheduler on page 151](#)

13.4.21 Dialog – “Delete task”

Function:

The “Delete task” dialog is used to delete a task.

Before the task is deleted, you are explicitly prompted whether you really want to delete this task.

Call:

ctrlX OS side navigation “Settings → Scheduler → Tab Configuration → Table → Task” → 

Elements of the “Delete task” dialog

GUI element	Function
“Delete”	Task is deleted
“Cancel”	Deletion is aborted

Further information

- ➔ [Chapter 13.4.20 Dialog – “Add task” on page 358](#)
- ➔ [Chapter 13.3.8 Tab – “Configuration” on page 328](#)
- ➔ [Chapter 13.2.11 Window – “Scheduler” on page 297](#)
- ➔ [Chapter 9.2 ctrlX CORE Scheduler on page 151](#)

13.4.22 Dialog – “Add callable to <taskname>”

Function:

Add a callable to a program from a task via the “Add callable to <taskname>” dialog.

The call sequence defines the order in which the callables are to be processed by the program.

When creating a callable, there are two options that affect the call sequence:

- Determining the call sequence by setting synchronization points
- Determining the call sequence by the Run-Index

Call:

ctrlX OS side navigation “Settings → Scheduler → Tab Configuration → Table → Task” → 

Elements of the “Define callable” site

GUI element	Function
Selecting the definition for the call sequence	• “SyncPoints (recommended)” That specifies that the call sequence is defined by setting the SyncPoints • “Run-Index” That specifies that the call sequence is defined by setting the Run-Index
Drop-down menu “Callable”	Callable type selection
“Alias”	Alias name specification (optional)
“Arguments”	Arguments can be defined Check which arguments are allowed for which callable types.
Button “Continue”	Callable is created

Elements of the “After SyncPoints” site

GUI element	Function
“Available SyncPoints”	List with all SyncPoints available for the task
“After SyncPoints”	List of defined “After SyncPoints”
Switch between the two lists	• > “Add selection” • < “Remove selection” • » “Add all” • « “Remove all”
“Select all”	Deselecting all SyncPoints in the list
“Deselect all”	Selecting all SyncPoints in the list
“Sync-point”	Input field to detect new SyncPoints
	Applies the input to the list of available SyncPoints
Button “Continue”	Navigates to the next “Before SyncPoints” site
Button “Back”	Navigates to the previous “Define callable” site

Elements of the “Before SyncPoints” site

GUI element	Function
“Available SyncPoints”	List with all SyncPoints available for the task
“Before SyncPoints”	List of defined “Before SyncPoints”
Switch between the two lists	• > “Add selection” • < “Remove selection” • » “Add all” • « “Remove all”
“Select all”	Deselecting all SyncPoints in the list
“Deselect all”	Selecting all SyncPoints in the list
“Sync-point”	Input field to detect new SyncPoints
	Applies the input to the list of available SyncPoints
Button “Continue”	Navigates to the last “Finish” site
Button “Back”	Navigates to the previous “After SyncPoints” site

Elements of the “Set Run-Index” site

GUI element	Function
“Run-Index”	Input field to enter the Run-Index
Currently active call sequence	Visualizes the current call sequence of the callables. It is used to determine the Run-Index
Button “Continue”	Navigates to the last “Finish” site
Button “Back”	Navigates to the previous “Define callable” site

Elements of the “Finish” site

GUI element	Function
Button “Finish”	Applies the data to the firmware and closes the dialog If an error occurs in the firmware when creating the callable, the dialog remains open and the firmware error is displayed in the dialog
Button “Back”	Navigates to the previous site

Further information

- ➔ [Chapter 13.5.9 Sidebar – “<name_callable>” \(Callable parameter\) on page 374](#)
- ➔ [Chapter 13.4.23 Dialog – “Delete callable” on page 361](#)
- ➔ [Chapter 13.2.11 Window – “Scheduler” on page 297](#)
- ➔ [Chapter 9.2 ctrlX CORE Scheduler on page 151](#)

13.4.23 Dialog – “Delete callable”**Function:**

The “Delete callable” dialog is used to delete a callable.

Before the callable is deleted, you are explicitly prompted whether you really want to delete this callable.

Call:

ctrlX OS side navigation “Settings → Scheduler → Tab Configuration → Table → Callable” → 

Elements of the “Delete callable” dialog

GUI element	Function
Button “Delete”	Callable is deleted from the program of the task
Button “Cancel”	Deletion is aborted

Further information

- ➔ [Chapter 13.4.22 Dialog – “Add callable to <taskname>” on page 359](#)
- ➔ [Chapter 13.2.11 Window – “Scheduler” on page 297](#)
- ➔ [Chapter 9.2 ctrlX CORE Scheduler on page 151](#)

13.4.24 Dialog – “Watchlist”**Call:**

ctrlX OS side navigation “Settings → Data Layer → Command bar ” 

Function:

The watch window visualizes the watchlists with elements defined by the user. To create a watchlist, the “ Create” function is available in the collection *datalayer/watchlists*.

To delete the watchlist, use the function “Delete”.

A watchlist contains a list of paths to the nodes in the Data Layer. Its values are to be shown in the watchlist. This path list can either be edited directly in the watchlist or it can be modified using the “Adds to watchlist” function.

Elements of the command bar

GUI element	Function
	Filter function to filter the currently displayed watchlist
	Selection list of all watchlists to select the watchlist to be displayed
	Showing/hiding of the action column in the table (with additional functions)
	Creating a new element at the bottom position in the current list

Columns of the table

GUI element	Function
“Name”	Name of the node (part of the path)
“Path”	Path of the node in the Data Layer
“Value”	Value of the node (this value is refreshed automatically if it changes)

GUI element	Function
"Action"	This column is shown/hidden and contains the following functions: •  Copy entry: The currently selected element is copied and directly inserted below the current element in the list •  Edit entry: The currently selected element is edited. •  Delete entry: The currently selected element is deleted from the list.

More functions in the table

Function	Description
Sorting	To sort the list alphabetically in ascending or in descending order, click either on the column header "Name" or "Path"
Navigation to the node	To navigate to the respective node in the Data Layer tree view, double-click on an element in the table

Further information

- ➔ [Chapter 13.2.7 Window – "Data Layer" on page 288](#)

13.5 Editors

13.5.1 Editor – "Remote Logging"

After a remote logging server was added to the ctrlX device, the settings for communication and message filters can be configured in the editor.

Also refer to:

➔ [Remote Logging – Configuring external server](#)



To open the editor, select the  button. However, the connection status between the ctrlX device and the logging server is "not connected". This is shown with the icon  in the status display, see: ➔ [Window – Remote Logging](#).

Call

ctrlX OS side navigation "*Diagnostics → Logbook → Remote Logging*" Click on the icon  in the table entry of the logging server

The editor has two tabs for connections with TCP and UDP protocol:

- "Server settings"
- "Message filter"

The following tab is also available for connections with TLS protocol:

- "TLS encryption"

Tab Server settings

Communication settings from the ctrlX device to the remote logging server

GUI element	Description
Name	Symbolic name of the server connection

GUI element	Description
Host / IP address / domain	IP address of the server or the host PC
Port	Port of the server or the host PC
Protocol	Set network protocol: • TCP • UDP • TLS
Message format	Syslog protocol to be used: • RFC3164 • RFC5424

Tab Message filter

Use message filters to define which logbook messages are to be sent to the remote logging server.

Required entries:

- Define a name for the message filter
- Select the filter criteria in the subordinate filter entries (drop-down menu/ multiple selection possible)

Filter logic (mode of action of the filter):

- The set filters are linked with "OR" within one filter entry
- Set field entries are "AND"-operated



Only when the filter criteria are met, a logbook message is sent to the remote logging server.

Field entries	Description
Name	Symbolic name of the server connection
Unit	Program unit issuing the error
Log level	Logging level, e.g. warning, error, info,...
Facility	System part issuing the error, e.g. kernel,...
Program	Program issuing the error
Message type	Message type: Trace, diagnostics, Empty (includes all others)

Tab TLS encryption

In the tab, certificates and keys can be uploaded to the ctrlX device that are required for TLS-encrypted message transmission to the syslog server. If the option "Skip verification of server certificate" is disabled, a server or CA certificate has to be available and selected on the ctrlX device.

Tab elements	Description
Server or CA certificate	Selection field displaying the server or CA certificates located on the ctrlX device. The field is empty upon delivery. To upload a certificate to the ctrlX device, click on the [] button and select the certificate to be uploaded from the engineering PC to the ctrlX device.

Tab elements	Description
Client certificate	Selection field displaying the client certificates on the ctrlX device. The field is empty upon delivery. To upload a certificate to the ctrlX device, click on the [] button and select the certificate to be uploaded from the engineering PC to the ctrlX device.
Client key	Selection field displaying the client key on the ctrlX device. The field is empty upon delivery. To upload a client key to the ctrlX device, click on the [] button and select the key to be uploaded from the engineering PC to the ctrlX device.
“Skip verification of server certificate”	Option to enable/disable the verification of the server or CA certificate. The option [] is enabled upon delivery. If this option is disabled, a server or CA certificate has to be selected in the “Server or CA certificate”.

13.5.2 Editor – Permissions

Call:

ctrlX OS side navigation “Settings → Users & Permissions → Permissions → Click on a permission”

Elements of the “Permissions” editor

GUI element	Description
“Tab”	“Authorized users” Users with the permission are shown under the “Authorized users” tab. The permissions can also be assigned or revoked here.
	“Authorized groups” Groups with the permission are shown under the “Authorized groups” tab. The permissions can also be assigned or revoked here.



“Full access” permission

- The “Full access” right cannot be assigned to any group
- The “Full access” right can only be assigned to other users by users that already have the “Full access” right
- The properties of users that have the “Full access” right can only be changed by users that have the “Full access” right

Editing authorizations

The tabs “Authorized users” and “Authorized groups” are located on the opening editing site..

At the bottom of each tab on the left side, you can find the  button to write the changes to the control and the  button to exit the window and to return to the window of the “Permissions” tile.

If there is information in the editor that is not yet written to the control, a window opens prompting whether this information is to be backed up on the control with the  button or discarded with the  button or whether to return to the window using the  button.

13.5.3 Editor – Groups

Call:

ctrlX OS side navigation “Settings → Users & Permissions → Groups” 

Elements of the “Groups” editor

GUI element	Description
“Tab”	“Users” To select the users to be assigned to the group, go to the “Users” tab. Groups from a remote authentication server are shown as <Group Name>(remote), see Chapter 13.4.15 Dialog – “Remote authentication” on page 353 . No users can be assigned to these groups.
	“Permissions” To assign rights to the group from the applications listed in the “Category” column, go to the “Permissions” tab. The rights are listed next to the name of the respective application (e.g. Users & Permissions, Connectivity) in the “Permissions” column and assigned to or revoked from a group by selecting or deselecting the checkbox. The “Scope identifier” column that can be shown, is displayed for each permission of the respective scope ID. Scope IDs are shown in error messages if the user does not have the required permission.



“Full access” permission

- The “Full access” right cannot be assigned to any group
- The “Full access” right can only be assigned to other users by users that already have the “Full access” right
- The properties of users that have the “Full access” right can only be changed by users that have the “Full access” right

Editing a group

Editing is possible for a group from the list. Select  in the “Actions” column of the element list.

The tabs “Users” and “Permissions” are located on the opening editing site.

To show the number of elements in the “Category” column, go to tab “Permissions” and then to the command bar on the left. To show and hide the column “Scope identifier” in the table, select / on the right.

At the bottom of each tab on the left side, there is the  button to write the changes to the control and the  button to exit the window and to return to the window of the “Groups” tile.

If there is information in the editor that is not yet written to the control, a window opens prompting whether this information is to be backed up on the control with the  button or discarded with the  button or whether to return to the window using the  button.

13.5.4 Editor – Users

Call:

ctrlX OS side navigation “Settings → Users & Permissions → Users” 

Elements of the “Users” editor

GUI element	Description
“Tab”	“Details” To disable/enable the user, go to the “State” field in the “Details” tab In the drop-down selection field “Password policy”, a previously created individual password policy can be assigned to the user instead of the default password policy, see ↗ Chapter 13.4.12 Dialog – “Password policies” on page 349. By default, the field is pre-assigned with the "Default" password policy. A new password can be set for the user via the two fields “Password” and “Confirm password” below. If the checkbox “Force password change on next login” is selected, the user has to change the password upon its next login. When creating a user, the function is selected upon default. If the user changed the password, the function is disabled Use the selection box “Individual session settings” to select user-defined settings overwriting the general "Session Policies", see ↗ Chapter 13.4.13 Dialog – “Session policies” on page 351
	“Groups” To select the groups to which the user should be assigned to, go to this tab. Users can be members of several groups. When assigned to a group, the user also gets the group permissions.
	“Individual permissions” To assign rights from the individual applications to the user, go to this tab. The rights are listed in the “Permission” column next to the name of the respective application (e.g. Users & Permissions, Connectivity) in the “Category” column. The “Scope identifier” column that can be shown, is displayed for each permission of the respective scope ID. Scope IDs are shown in error messages if the user does not have the required permission.
	“Overall permissions” All user rights are displayed under this tab. Information whether the user rights are individually assigned or via group assignment is also provided under this tab. The rights are displayed as list with the following columns: “Category”, “Permission” and “Obtained from”. Editing rights is not possible



Remote user

Users from a remote authentication server are shown as <UserName>(remote), see ➔ [Chapter 13.4.15 Dialog – “Remote authentication” on page 353](#). No settings can be made in the tabs “Details” and “Groups”.



“Full access” permission

- The “Full access” right cannot be assigned to any group
- The “Full access” right can only be assigned to other users by users that already have the “Full access” right
- The properties of users that have the “Full access” right can only be changed by users that have the “Full access” right

Editing a user

Only a user from the list can open the editor via  in the “Actions” column of the element list.

The tabs “Details”, “Groups”, “Individual permissions” and “Overall permissions” are located on the opening editing site

To show the number of elements in the “Category” column, go to the tab “Individual permissions” and then to the command bar on the left. To show and hide the column “Scope identifier” in the table, select / on the right.

At the bottom of each tab on the left side, there is the button  to write the changes to the control and the  button to exit the window and to return to the window of the “Users” tile.

If there is information in the editor that is not yet written to the control, a window opens prompting whether this information is to be backed up on the control with the  button or discarded with the  button or whether to return to the window of the tile “Users” using the  button.

Further information

- ➔ [Chapter 13.4.12 Dialog – “Password policies” on page 349](#)
- ➔ [Chapter 13.4.13 Dialog – “Session policies” on page 351](#)
- ➔ [Chapter 13.4.15 Dialog – “Remote authentication” on page 353](#)
- ➔ [Chapter 13.2.18 Window – “Users & Permissions” on page 304](#)

13.5.5 Editor – Certificates & Keys

Function:

In the “Certificates & Keys” editor, the management site of the respective application or service is displayed. Here are the tabs “Certificates”, “Keys”, “Revocation lists”, PKCS12 containers and Certificate enrollments, on which the existing certificates and keys, the certificate revocation lists, the PKCS12 containers and existing certificate enrollments are listed in a table. Use the buttons in the command bar of the respective tab to load certificates, keys, revocation lists, PKCS12 containers to the device and to store private keys on the ctrlX OS device, generate private keys on the ctrlX OS device and register certificates.

The editor contains the following tabs:

- Tab “Certificates”
Certificates available on the ctrlX OS device are listed in a table, see:
➔ Tab – “Certificates”
- Tab “Keys”
Keys available on the ctrlX OS device are listed in a table, see:
➔ Tab – “Keys”
- Tab “Revocation lists”
Certificate revocation lists available on the ctrlX OS device are listed in a table, see:
➔ Tab – “Revocation lists”
- Tab “PKCS12 containers”
Containers available on the ctrlX OS device in .pfx format for a service are listed in a table, see:
➔ Tab – “PKCS12 containers”
- Tab “Certificate enrollments”
Requests generated on the ctrlX OS device with private keys are listed in a table, see:
➔ Tab – “Certificate enrollments”

Call:

ctrlX OS side navigation “*Settings* → *Certificates & Keys*” → Click on the name (e.g. SSH) in the list Service

13.5.6 Editor – “Network interfaces”

To make detailed settings of the network connection, go to the editor.

Call:

ctrlX OS side navigation “*Settings* → *Network interfaces* → *Double-click on one of the shown network connections*”

Description

The editor is divided into the following tabs:

Tab “General”

The tab shows general information of an interface. The interface can also be enabled and disabled here.

NOTICE

When the interface is disabled, the connection to the control is lost via this interface.

- Connection state
- Connection speed
- Sent and received data

Click on “Advanced settings” to display further information and option settings:

- “MAC address”
- “IPv4 gateway”
- “IPv6 gateway”
- “DNS addresses”
- “IP addresses”

- Option settings:
 - “MTU”: Value of the "Maximum Transmission Unit"
 - “Use external settings”
If the option is enabled, another instance (e.g. an app) is responsible for the interface configuration.
 - “Enable IP forwarding”

Note:

IP forwarding is disabled by default.

If the IP forwarding is enabled, the ctrlX device becomes an unsecure network device!

Enabling thus requires an additional acknowledgement of the dialog/confirmation prompt.



IP forwarding is only required if an external standard Ethernet communication is to be forwarded to the EtherCAT network via the "virtual Ethernet switch" of the ctrlX EtherCAT master. This is for example the case if EtherCAT slaves are to be configured via EoE using third-party engineering programs. The IP forwarding allows the ctrlX operating system to forward data packages like a router. Also refer to the EtherCAT Master documentation: ➔ [Web documentation](#)

Tab“ IPv4” and “IPv6”

Setting options:

- DHCP (on/off)
Obtaining automatic IP address automatically via Dynamic Host Configuration Protocol
- Configuring IP addresses
- Configuring DNS addresses
 - Configuring DNS suffixes:

To open a dialog to configure DNS suffixes, click on the [🔗] button.

The order of the entries under DNS suffixes results in the search order for name resolution.

A DNS domain suffix search list can be configured for DNS clients. This list extends or revises DNS search functions. Suffixes can be added to the list to search for short, unqualified computer names in multiple specified DNS domains. If a DNS query fails, the DNS client service can use this list to append other name suffix endings to your original name and retry DNS queries for these alternative fully-qualified domain names.

If the suffix search list is empty or not specified, the primary DNS suffix of the computer is appended to short, unqualified names and a DNS query is used to resolve the resulting fully-qualified domain name. If this query fails, the computer can try additional queries for alternative fully-qualified domain names by appending a connection-specific DNS suffix configured for network connections.

The term "DNS suffix" is essentially used in a Windows environment in the context of a Windows domain. The DNS suffix is basically the domain name. The name of the computer (host name) together with the DNS suffix results in the Fully-Qualified Domain Name.

Example:

DNS suffix: mydomain.com

Fully-Qualified Domain Name: mycomputer.mydomain.com

The DNS suffix is entered into the computer when it joins the domain. Multiple DNS suffixes can be entered. The order of the DNS suffixes is relevant

- Configuring gateway connections



Detailed information on IPv4 and IPv6

- The IP addresses are specified in the numerical format Classless Inter-Domain Routing (CIDR). For this information, place the cursor on the IP address field. 192.168.1.1/16 for example
- The individual DNS addresses point to the server of the multi-level domain-name system. This system resolves domain names in IP addresses and vice versa. The request is routed to the superimposed DNS server until the domain name can be resolved, the DNS server responsible for the top-level domain (e.g. .com) was reached and the domain name cannot be resolved anymore
- The response is returned to the original requester who can use this response for further network communication. The correct configuration syntax provides more information on when the cursor is placed above a DNS address field
- "Gateway" signals the endpoint used to route the network communication to another subnet. Enter the required endpoint

Tab "Security"

Select Network security to allow only authenticated connections to the Ethernet interface.



Wireless interfaces do not provide this option!

To enable the Network security, use the sliding switch.

Check whether the configuration contains valid entries.

Only IEEE 802.1X with the authentication method EAP-TLS is currently supported as access control.

Required settings for the enabled Network security:

- **Access control**
IEEE 802.1X (preset)
- **Authentication method**
EAP-TLS (preset)
- **Identity**
Selection of an account registered on the remote server

Supported certificates (upload to the ctrlX CORE control before, see above)

- **CA certificate**
Certificate selection (category "CA certificate")
- **Client certificate**
Certificate selection (category "Trusted" or "Own")
- **Client private key**
Certificate selection (category "Trusted" or "Own")
- **Password for client private key**
If "Client private key" is password-protected, enter the password into the input field.

Tab "Wi-Fi" (only available for WiFi Interface)

Wireless interfaces can be configured and radio networks to be connected to the control can be added.



Radio networks are not supported if the special character "/" is in the network name.

Supported encryption methods for radio networks:

- **Without encryption**

Value to be set:

- Security mode: "None"

- **Encryption WPA2/PreShared Key**

Values to be set:

- Security mode: WPA/WPA2 Personal
- Password: <password>

- **Encryption WPA2/EAP-TLS**

Values to be set:

- Security mode: WPA/WPA2 Enterprise (TLS)
- Identity: Selection of an account registered on the remote server

Supported certificates

(upload to the ctrlX CORE control before, see above)

- **CA certificate:**

Certificate selection: Category "CA certificate"

- **Client certificate:**

Certificate selection: Category "Trusted" or "Own"

- **Client private key:**

Certificate selection: Category "Trusted" or "Own"

- **Password for client private key:**

If "Client private key" is password-protected, enter the password into the input field.

- **Encryption WPA2/EAP-PEAP**

Values to be set:

- Security mode: WPA/WPA2 Enterprise (PEAP)
- Identity: Selection of an account registered on the remote server
- Password: The respective password
- CA certificate (optional): WiFi network certificate
- Phase 2 authentication
- Anonymous identity (optional)



The control always connects to the strongest network!

To connect a control to a specific network, delete all entered networks before.

The status "Connected" is shown in the list of known networks behind the SSID if the control is connected to this radio network.

Tab "VLAN"

Only displayed if a VLAN interface is selected in the window.

The network priority of the VLAN interface can be configured in the tab, see here:

➔ [Adding and configuring VLAN interface](#)

Tab "Bridge"

Only displayed if a network bridge is selected in the window.

The interfaces to be bridged can be selected in the tab, see here:

➔ [Bridging interfaces](#)

13.5.7 Sidebar – “schedMain” (Parameter)

Function:

A parameter for the “schedMain” task can be changed in the “schedMain” sidebar.

Call:

ctrlX OS side navigation “Settings → Scheduler → Tab Configuration → Table → Task schedMain” → 

Sidebar “schedMain”

GUI element	Description
“Trigger”	Specifying the trigger source for the scheduler: • “TIMER”: Timers of the operating system are used to start the Scheduler cycle • “INTERRUPT”: An internal interrupt is used to trigger the Scheduler

Further information

- [↩ Chapter 13.3.8 Tab – “Configuration” on page 328](#)
- [↩ Chapter 13.2.11 Window – “Scheduler” on page 297](#)
- [↩ Chapter 9.2 ctrlX CORE Scheduler on page 151](#)

13.5.8 Sidebar – “Task: <name>” (Task parameter)

Function:

The “Task: <name>” sidebar of the task parameters consists of two tabs to make different task settings.

Call:

ctrlX OS side navigation “Settings → Scheduler → Tab Configuration → Table → Task” → 

Elements of the “Properties” site

GUI element	Description
“Name”	Task name
“Priority”	Priority of the task: Between 0 and 139
“Cycletime”	Specifies in which cycle time (in microseconds) the task should be called
“CPU Core”	Specifies the CPU on which the task is to be executed
“Stacksize”	Specifies the stack size of the task

Elements of the “Watchdog” site

To enable or disable the watchdog, there is a switch at the top of the tab.
If a watchdog is enabled, different parameters can be configured:

GUI element	Description
Radio button “Minor error (F0)”	Informs the system that a watchdog was triggered with a minor error

GUI element	Description
Radio button “Critical error (F6)”	Informs the system that a watchdog was triggered with a critical error
“Task behavior”	• “Disable” The task is not triggered anymore after starting the watchdog • “Continue” After triggering the watchdog, the task is still called until the maximum number of subsequent errors is reached
“Max. consecutive errors”	Maximum number of subsequent errors: The task is disabled when the maximum number of subsequent errors is reached

Further information

- ➔ [Chapter 13.3.8 Tab – “Configuration” on page 328](#)
- ➔ [Chapter 13.2.11 Window – “Scheduler” on page 297](#)
- ➔ [Chapter 9.2 ctrlX CORE Scheduler on page 151](#)

13.5.9 Sidebar – “<name_callable>” (Callable parameter)**Function:**

Parameters of a callable are shown in the sidebar “<name_callable>”.
The parameters cannot be changed.

Call:

ctrlX OS side navigation “*Settings → Scheduler → Tab Configuration → Table → Callable*” → ⓘ

GUI element	Description
“ID”	ID from callable Is uniquely specified by the system
“Callable”	Name of the callable type
Alias	Alias name of the callable (optional)
Run-Index	Defined run index
“After SyncPoints”	List of defined “After SyncPoints” (optional)
“Before SyncPoints”	List of defined “Before SyncPoints” (optional)
Arguments	List of arguments (optional)

Further information

- ➔ [Chapter 13.3.8 Tab – “Configuration” on page 328](#)
- ➔ [Chapter 13.2.11 Window – “Scheduler” on page 297](#)
- ➔ [Chapter 9.2 ctrlX CORE Scheduler on page 151](#)

13.6 Widgets**13.6.1 Widget – “Hardware”**

The widget provides information on the ctrlX device hardware.

Call

ctrlX OS side navigation “*Settings → window System Information → widget Hardware*”

Displayed information

The information displayed can vary depending on the device variant.

- Total memory size.
- Swap memory size.
- Process architecture of the ctrlX device.
- Number of CPU cores.
- Information on the TPM (Trusted Platform Module) used.
- Information on the available network interfaces.
- Information on the system memory load.

Further information

→ [Window](#) – “System Information”

13.6.2 Widget – “Typeplate”

ctrlX devices have an electronic type plate with all relevant data on the device identification.

The widget displays the electronic type plate information listed below.

Call

ctrlX OS side navigation “*Settings → window System Information → widget Typeplate*”

Displayed information

The information displayed can vary depending on the device type.

Information	Description
“Manufacturer”	ctrlX device type manufacturer.
“Product name”	Product name of the ctrlX device.
“UUID”	Universally unique identifier of the ctrlX device.
“Version”	Installed system version.
“BIOS version”	Installed BIOS version.
“Serial number”	Serial number ID of the ctrlX device.

Further information

→ [Window](#) – “System Information”

13.6.3 Widget – “General”

The widget provides general information on the ctrlX device.

Call

ctrlX OS side navigation “*Settings → window System Information → widget General*”

Displayed information

The information displayed can vary depending on the device variant.

- Name of the ctrlX device can be edited via the button [✎].
- Operating time since the last device start.
- Operating system variant/version of the ctrlX device.
- Information on system integrity.
- Information on real-time capability.
- Information on device type (real or virtual)
- Display of device boot mode (Secure boot)
If "true", the current security status includes fully secure system integrity ensuring a secure boot chain and the use of full disk encryption.
- Information on kernel modules (can be expanded).

Further information

➔ [Window](#) – “System Information”

14 Related documentation

14.1 Overview

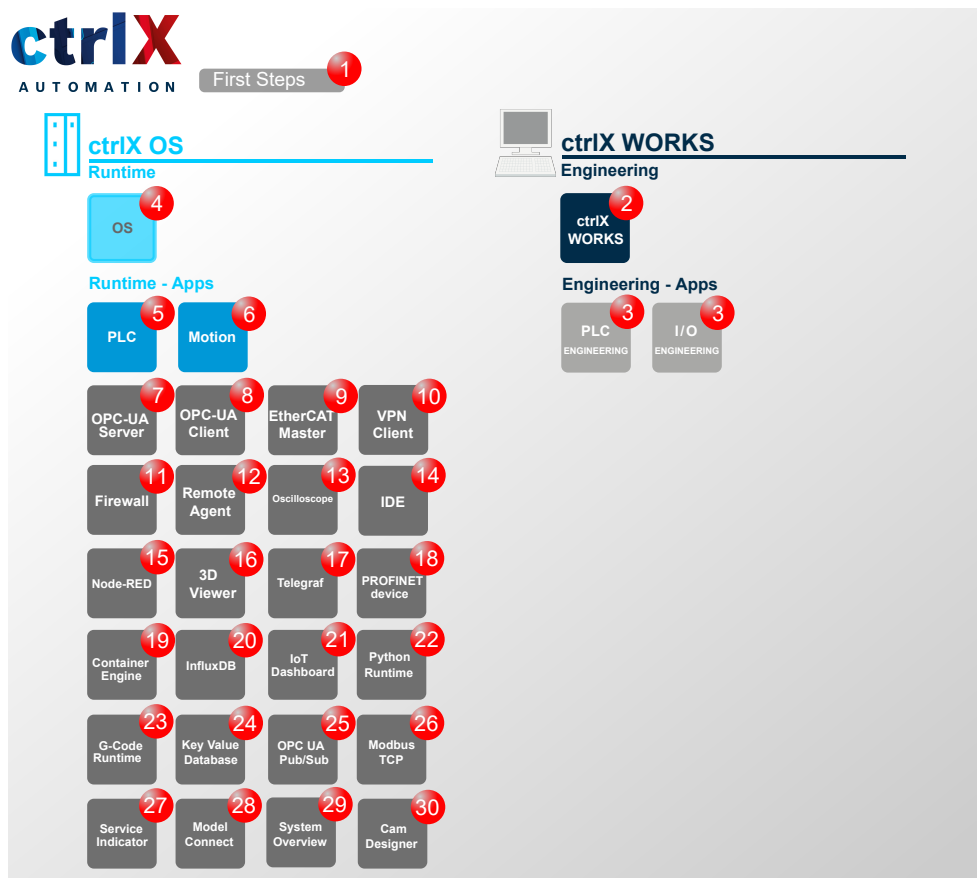


Fig. 71: Overview on further documentations

14.2 ctrlX AUTOMATION

No.	Documentation
1	ctrlX WORKS First Steps Quick Start Guide ↗ Web documentation link Ordering information: • DOK-XWORKS-F*STEP*****-QU01-EN-P • R911403760

14.3 ctrlX WORKS

No.	Documentation
2	ctrlX WORKS Basic System 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XWORKS-WRK***V02**-APRS-EN-P • R911421576
3	ctrlX PLC Engineering - PLC Programming System 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XPLC**-ENG*****V02-APRS-EN-P • R911421578
3	ctrlX PLC Engineering - PLC Libraries 02VRS Reference Book ↗ Web documentation link Ordering information: • DOK-XPLC**-LIB***V02**-RERS-EN-P • R911421580

14.4 ctrlX OS

No.	Documentation
4	ctrlX OS - Operating System for ctrlX CORE Control Devices 02VRS Application Manual ↪ Web documentation link Ordering information: • DOK-XCORE*-XCR***V02**-APRS-EN-P • R911421590
	ctrlX OS - Data Layer nodes 02VRS Reference Book ↪ Web documentation link Ordering information: • DOK-XCORE*-DL****V02**-RERS-EN-P • R911421592
	ctrlX OS - Diagnostics 02VRS Reference Book ↪ Web documentation link Ordering information: • DOK-XCORE*-DIAG**V02**-RERS-EN-P • R911421594

14.5 ctrlX OS apps

No.	Documentation
5	PLC App - PLC Runtime Environment for ctrlX OS 02VRS Application Manual ↪ Web documentation link Ordering information: • DOK-XCORE*-PLC***V02**-APRS-EN-P • R911421584
6	Motion App - Motion Runtime Environment for ctrlX CORE 02VRS Application Manual ↪ Web documentation link Ordering information: • DOK-XCORE*-MOT***V02**-APRS-EN-P • R911421610
7	OPC UA Server App - OPC UA Server for ctrlX OS 02VRS Application Manual ↪ Web documentation link Ordering information: • DOK-XCORE*-UAS***V02**-APRS-EN-P • R911421598

No.	Documentation
8	OPC UA Client App - OPC UA Client for ctrlX OS 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-UAC***V02**-APRS-EN-P • R911421600
9	EtherCAT Master App - EtherCAT master for ctrlX OS 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-ECM***V02**-APRS-EN-P • R911421604
10	VPN Client App - Remote Support Software for ctrlX OS 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-VPN***V02**-APRS-EN-P • R911421596
11	Firewall App - Security Functions for ctrlX OS 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-FRW***V02**-APRS-EN-P • R911421606
12	Remote Agent App - ctrlX Device Portal Connection for ctrlX OS Devices 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-RMA***V02**-APRS-EN-P • R911421608
13	Oscilloscope App - Oscilloscope Function for ctrlX OS Devices 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-OSC***V02**-APRS-EN-P • R911421587
14	IDE App - Integrated Development Environment 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-IEN***V02**-APRS-EN-P • R911421612

No.	Documentation
15	Node RED App - Graphic Programming for ctrlX OS 02VRS Application Manual ↪ Web documentation link Ordering information: • DOK-XCORE*-NOENRED*V02-APRS-EN-P • R911421582
16	3D Viewer App - Browser-based 3D Kinematic Simulation for ctrlX OS 02VRS Application Manual ↪ Web documentation link Ordering information: • DOK-XCORE*-3DV***V02**-APRS-EN-P • R911421615
17	Telegraf App - Server Agent for Collecting Data in the Data Layer 02VRS Application Manual ↪ Web documentation link Ordering information: • DOK-XCORE*-TSA***V02**-APRS-EN-P • R911421623
18	PROFINET Device App - PROFINET Device for ctrlX OS 02VRS Application Manual ↪ Web documentation link Ordering information: • DOK-XCORE*-PROFINETV02-APRS-EN-P • R911421617
19	Container Engine App - Using Docker® Images on the ctrlX OS 02VRS Application Manual ↪ Web documentation link Ordering information: • DOK-XCORE*-DOE***V02**-APRS-EN-P • R911421619
20	InfluxDB App - Influx Database Connection for ctrlX OS 02VRS Application Manual ↪ Web documentation link Ordering information: • DOK-XCORE*-IDB***V02**-APRS-EN-P • R911421625
21	IoT Dashboard App - Data Visualization in Dynamic, Interactive Dashboards 02VRS Application Manual ↪ Web documentation link Ordering information: • DOK-XCORE*-GDB***V02**-APRS-EN-P • R911421633

No.	Documentation
22	Python Runtime App - Python Runtime App Environment for ctrlX CORE 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-PYR***V02**-APRS-EN-P • R911421629
23	G-Code Runtime App - G-Code Interpreter for ctrlX OS 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-GCO***V02**-APRS-EN-P • R911421631
24	Key Value Database App - Data Layer Management 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-KVD*****V02-APRS-EN-P • R911418735
25	OPC UA Pub/Sub App - OPC UA Pub/Sub for ctrlX OS 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-UAP***V02**-APRS-EN-P • R911421602
26	Modbus TCP App - Modbus TCP Communication over ctrlX OS 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-MOD*TCP*V02-APRS-EN-P • R911421621
27	Service Indicator App Service Indicator for ctrlX OS 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-SIN*****V02-APRS-EN-P • R911421627
28	Model Connect App Target for Model-based Development and Simulation for ctrlX OS 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-MOC***V02**-APRS-EN-P • R911421631

No.	Documentation
29	System Overview App - System Topology and System Information 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XCORE*-SOV***V02**-APRS-EN-P • R911424409
30	Cam Designer - Configuring ctrlX Motion Cams 02VRS Application Manual ↗ Web documentation link Ordering information: • DOK-XWORKS-CAM***V02**-APRS-EN-P • R911424390

15 Service and support

Our worldwide service network provides an optimized and efficient support. Our experts provide you with advice and assistance. You can contact us **24/7**.

Service Germany

Our technology-oriented Competence Center in Lohr, Germany, is responsible for all your service-related queries for electric drive and controls.

Contact the **Service Hotline** and **Service Helpdesk** under:

Phone: **+49 9352 40 5060**
Fax: **+49 9352 18 4941**
Email: ↗ service.svc@boschrexroth.de
Internet: ↗ <http://www.boschrexroth.com>

Additional information on service, repair (e.g. delivery addresses) and training can be found on our internet sites.

Service worldwide

Outside Germany, please contact your local service office first. For hotline numbers, refer to the sales office addresses on the internet.

Preparing information

To be able to help you more quickly and efficiently, please have the following information ready:

- Detailed description of malfunction and circumstances
- Type plate specifications of the affected products, in particular type codes and serial numbers
- Your contact data (phone and fax number as well as your e-mail address)

16 Index

1, 2, 3 ...

(nodes) = datalayer.browse (<path>) 182

(nodes) = datalayer.read (<path>) 182

A

A JSON file is used to register and unregister main and detailed diagnostics

General information 256

About ctrlX OS 316

Add remote logging server 347

Address 185

Administration 171

Apps

Disable apps on the control 279

Install apps on the control 279

Show installed apps 279

Uninstall apps from the control 279

Apps - Handling

Change app version 37

Delete apps 40

Disable app 37

Download apps from the ctrlX device ... 38

Enable app 38

Install apps from the engineering PC on the ctrlX device 33

Install apps on the ctrlX device from the ctrlX Store 32

Install apps on the ctrlX device from the device-internal app memory 34

Reinstall apps 39

Uninstall apps 39

Update apps 35

Upload apps to the app memory of the ctrlX device 33

Asynchronous response 208

Automation 128

Automation app

Basics 32

B

Backing up and restoring the configuration 174

Backup

Active configuration and configuration archive 88

Automatic application upon system startup 109

Certificates and keys 87

Command line 105

Connectivity 90

Date and time 99

Delete apps 104

Diagnostics 105

FAQ 117

Hosts 99

Introduction 79

Manual modification 105

Options during creation 80

Package management (apps) 102

Proxy 104

Show report 82

SSH 104

Structure of the backup file 84

System 104

System information 105

Type plate information 105

User management 99

Backup License

License Information 101

Browse requests

Special features 206

C

Callables 153, 157

Certificates 64

Changing password of the control

Dialog Change password 349

Dialog Password policies 349

Dialog Session policies 351

Compliance Checker 234

Configurations

Back up active app data 77

Compare configurations 71

Copy configuration to the archive 73

Delete configuration in the archive 74

Display information on a configuration in the archive 74

Download a configuration 77

Edit description of a configuration 75

Load app data from engineering PC 67

Load configuration from the archive to the ctrlX device 71

Loading app data 77

Manage configurations 75

Rename configuration in archive 74

Reset active app data 68

Save active app data to the archive of the ctrlX device 69

Transfer active app data to the engineering PC 67

Transfer configuration from archive to engineering PC 72

Upload from the engineering PC to the archive of the ctrlX device 70

View active data on file level 69

View configuration in the archive 72

View Loading report 78

Configuring the startup 173

Connection to the Data Layer RT (real time) ..
..... 255

Connections at front side 141

Control monitoring 139

Conventions for login and session monitoring
..... 46

CPU core 175

Creating a new user on the control	
Dialog Add user	348
ctrlX AUTOMATION	
Related documentation	376
ctrlX CORE	
Apply backup	81
Create backup	79, 341
Show backup history	83
ctrlX CORE Scheduler	151
ctrlX device	
Enable Operating mode	40
Enable Service mode	40
ctrlX Device certificate	122
ctrlX device ID	122
ctrlX OS	
Create system report	138
Licensing notes	19
Start page	270
ctrlX OS device	
Licensing	128
Manual licensing	128
Registering a new device for licensing	128
Secure hardware anchor for licensing	128
Trial licenses for development	129
ctrlX OS Gerät	
Automation	128
Generating a device key	124
Generating certificate signing request	126
Requesting a device certificate for your device	124
Requesting the device certificate	127
Upload device certificate	128
ctrlX OS Licensing notes	19
ctrlX OS web interface	
Overview	268
D	
Data Layer	
Address	185
Asynchronous methods	202
Complex data types	191
Data storage and data access	184
Data transport NRT	200
Data type "Variant"	192
Description of the nodes	240
FlatC	191
Further information	184
Metadata	194
Overview	184
Scope of data	184
Security	236
Simple data types	190
Synchronous methods	201
Data Layer RT (real time)	255
Data storage and data access	184
Data transport NRT	200
Data type	190, 191, 192
datalayer.browse	182
datalayer.create	183
datalayer.create_json	183
datalayer.create_json(<path>, <value>)	183
datalayer.create(<path>, <value>)	183
datalayer.read	182
datalayer.read_json	183
datalayer.read_json(<path>)	183
datalayer.remove	183
datalayer.remove(<path>)	183
datalayer.write	182
datalayer.write_json	183
datalayer.write_json(<path>, <value>)	183
datalayer.write(<path>, <value>)	182
DCHP settings	345
Deploying your Secure Boot certificates to device NVRAM	131
Detecting a reboot loop of the Automation Core app	150
Device identification	122
Diagnostic handling summary	246
Diagnostic logbook	
Dialog Add user	343
Diagnostics	178
ExportLogbook	133
Logbook	132
Remote Logging - Configure external server	136
Use cases	244
Dialog	
Add a bandwidth	342
Add callable to <taskname>	359
Add remote logging server	347
Add task	358
Add user	348
Add VLAN interface	341
Certificate renewal	345
Change password	349
DCHP settings	345
Delete callable	361
Delete task	359
Easy service access	355
Generate key	356
Login banner - Editor	352
Network settings	342
Password policies	349
PKCS12 Container	357
Remote authentication	353
Scheduler settings	357
Select columns	343
Session policies	351
Settings	344, 348
Watchlist	362
Double wildcard	205

E		
Easy service access	355	
Editor		
Certificates & Keys	368	
Groups	366	
Network interfaces	369	
Permissions	365	
Remote Logging	363	
Users	367	
Enabling Secure Boot	129	
Default ctrlX OS vs. customized ctrlX OS	129	
Deploying your Secure Boot certificates to device NVRAM	131	
Managing your Secure Boot keys	130	
Prerequisites	129	
Revoking keys and hashes on your device ..	131	
Secure Boot for default ctrlX OS	129	
Signing EFI binaries for Secure Boot with a customized ctrlX OS	130	
Steps to own Secure Boot for customized ctrlX OS	129	
Target state for your device	131	
UEFI password protection	131	
Ensuring a safe machine state after a critical error	150	
Error categories	242	
Error deletion	242	
Errors		
Return errors	207	
Event diagnostics	246	
Examples	248	
Exception handling	176	
Exporting Logbook	133	
G		
Generate certificate signing request	65	
Generating a device key	124	
Generating certificate signing request	126	
Generating private key on the control		
Dialog Generate key	356	
H		
Hardware reset in case ctrlX CORE X3, ctrlX COREplus X3 and ctrlX DRIVEplus	118	
Hardware reset in case ctrlX COREplus X7	120	
Hardware reset in case of ctrlX CORE and ctrlX DRIVEplus with integrated ctrlX CORE	118	
Hardware watchdog	178	
Helpdesk	383	
Hotline	383	
I		
Image conversion	132	
Conversion	132	
Precondition	132	
Intended use		
Areas of application	20	
Areas of use	20	
Introduction	19	
Interface of the owner	212	
Interface of the user	213	
Introduction into the diagnostic system ...	240	
K		
Keys	64	
L		
LED diagnostic code	149	
LED states	137	
Licensing a ctrlX OS device	128	
Log file		
Storage	177	
Structure	177	
Log structure (message format) in journald and its metadata	259	
Message format	259	
Metadata	261	
Log types	242	
Logbook		
Memory size	132	
Overview	132	
Logging user into the control		
Initial commissioning	44	
M		
Main diagnostics	252	
Managing certificates & keys		
Overview	63	
Managing your Secure Boot keys	130	
Manual licensing	128	
Memory range		
Revision	212	
Structure	210, 216	
Types	209	
Memory type SharedRetain	215	
Metadata	194	
Method <i>check-operations</i>	239	
Methods		
Asynchronous methods	202	
Synchronous methods	201	
Mode "Debug"	175	
Mounting external storage medium	143	
N		
Network settings	341, 342	
Node methods	206	
Nodes		
Methods	206	
NRT	200	
NRT access to RT data	214	
NRT data exchange between controls		
Access to data of another control	220	
Configuration	219	
Overview	218	

O	
Operating states	147
Owner	
Interface	212
P	
Performing firmware update	299
PKCS12 containers	338
PKI	123
Program configuration mode	166
Public Key Infrastructure (PKI)	123
Python functions for Data Layer access	182
Q	
Quick start	
Apps - Basics	31
Licenses - Overview	53
R	
Registering a new device for licensing	128
Registering an unregistering main and detailed diagnostics	
Structure of the JSON file	256
Registering and unregistering main and detailed diagnostics	
Options to register and unregister main and detailed diagnostics	258
Release cycle of apps and engineering software	41
Remote authentication	353
Remote Logging	
Configure external server in the ctrlX device	136
Server system requirements	136
Removing external storage medium safely	145
Renewing certificate automatically	
Dialog Certificate renewal	345
Requesting a device certificate for your device	124
Requesting the device certificate	127
Resetting the task configuration	174
Response of a provider	207
Responsibility of nodes	205
example	205
Returning errors	207
Revision of a memory range	212
Revoking keys and hashes on your device	131
RT MemoryType trigger	216
Runtime diagnostics	245
S	
Safe machine state	150
Safe mode	148, 151
Safety instructions	20
Scheduler	
Introduction	151
Strict mode	175
Scheduler load	176
Schema	122
Schema validation	122
Scheme of the diagnostic numbers	249
Scope	237
Check	238
Definition	237
Scope of data	184
Script parser/interpreter	179
Initialization script	181
Search paths	181
User interface via Data Layer	180
Secure Boot for default ctrlX OS	129
Secure hardware anchor for licensing	128
Selecting an initial CPU core	175
Service hotline	383
Settings	
Window Licenses	284
SharedRetain	215
Sidebar	
Callable parameter	374
schedMain parameter	373
Task parameter	373
Signing EFI binaries for Secure Boot with a customized ctrlX OS	130
Signing request	65
Simple wildcard	204
Special case Safe Mode	148
State "Error"	147
State "Init"	147
State "Operating"	147
State "Partial Operating"	147
State "Service"	147
State "Setup"	147
State "Shutdown"	148
State "SysError"	147
State "Warning"	147
State machine	146
State machine of the user	213
State of the ctrlX OS Runtime components	148
Status	171
Steps to own Secure Boot for customized ctrlX OS	129
Strict mode	175
Structure of a memory range	210, 216
Structure of the diagnostic number	251
Subscription	
Client examples	230
Global settings	229
Lossless subscription to real-time data	233
Rule set	225
Support	383
Synchronous response	208
System Handler	145, 146
T	
Tab	
Alarms	317
Back up	341
Certificate enrollments	339
Certificates	332

Configuration	328	Using LAN adapter	142
Debug	332	Using SD cards	142
History	320	Using the diagnostic system	242
Keys	334	Using WiFi adapter	142
Licenses	284	W	
Overview	327	Widget	
Performance	326	General	375
PKCS12 containers	338	Hardware	374
Processes	326	Typeplate	375
Resources	325	Wildcard	204, 205
Restore	340	Window	
Revocation lists	337	About ctrlX OS	316
Sources	288	Apps	279
Startup	331	Backup & Restore	298
Threads	316	Certificates & Keys	301
Target state for your device	131	Data Layer	288
Task	153	Date & Time	295
Task "Priority"	156	Groups	306
Task configuration		Home	270
Reset	174	Hosts	308
Task properties	154	Licenses	284
Task response	153	Logbook	271
Task status	170	Manage app data	273
Time measurements	168	Network Diagnostics	309
Transitions	148	Network discovery	310
Trial licenses for development	129	Network interfaces	312
Trigger source	173	Open Source Components	314
Types of memory ranges	209	Permissions	306
U		PKIs	302
UEFI password protection	131	Proxy	313
Unintended use	20	Remote Logging	272
Consequences, disclaimer	19	Reset Device Data	300
Uploading the device certificate	128	Scheduler	297
Use cases	244	Sessions	307
User		SSH	313
Interface	213	Storage	299
State machine	213	System Information	297
User and rights management on the control		System Performance	296
Change user password	44	Users	305
Configure user management	46	Users & Permissions	304
Create new user	45	Working with configurations in the control	
Groups	43	Introduction	66
Login banner	44	Written offer for source code	315
Open user management	46		
Overview	42		
Password forgotten	44		
Permissions	43		
Sessions	43		
State upon delivery/default user	43		
User management – Functional principle	42		
Users	43, 44		
Users, groups, permissions, sessions	43		
User and rights management on the device			
Create system user	45		
User interface via Data Layer	256		
Username and password conventions	46		
Using external storage media	142		

Bosch Rexroth AG
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr a.Main
Germany
Tel. +49 9352 18 0
Fax +49 9352 18 8400
www.boschrexroth.com/electrics



R911421590 04