Nexeed Automation | Engineering Guide

Setting up Cycle Time Assist for ctrlX

V1.0

1 Content

2 Foreword

This technical documentation is an integral part of the software. This document contains important information regarding safety, application, parameterization and programming.

The document must be read and understood before application to eliminate potential errors and to ensure smooth operation.

3 General Information

Explanation of Icons

Safety Icons and Signal Words Used

This documentation uses the following safety icons and signal words. They provide information about potential hazards and include instructions on how to avoid these hazards.

DANGER **Description of a high-risk hazard which, if disregarded, will lead to death or serious injury.**
The consequences of the hazard are specified.
- Steps for preventing the hazard from occurring are specified.

WARNING **Description of a medium-risk hazard which, if disregarded, may lead to death or serious injury.**
The consequences of the hazard are specified.
- Steps for preventing the hazard from occurring are specified.

CAUTION **Description of a low-risk to medium-risk hazard which, if disregarded, may lead to minor or moderate injury.**
The consequences of the hazard are specified.
- Steps for preventing the hazard from occurring are specified.

NOTICE **Description of a risk to property and the environment. The note excludes injuries.**
The consequences of the hazard are specified.
- Steps for preventing the hazard from occurring are specified.

General Icons and Tags

In addition to the safety notices, the following information is formatted and tagged accordingly:

Information unit	Description	Example
Operating instruction step	Each operating instruction step is numbered.	1. In the File menu, click Open.
Interface text	Text that has been extracted from the interface of a software tool is shown in italics.	1. Click on the *Parameter* tab.
Code	Programming examples for the structured text.	`Example of a code.`
Result of an operating instruction	An arrow is used to show the result at the end of the operating instruction.	The object is parameterized.

Contains important information, references to other documents and/or tips and tricks.

Copyright

Disclaimer

This document has been produced taking into account the applicable standards and regulations, and according to the latest technology. The content of the documentation has been checked for accuracy, but discrepancies may still occur. No liability is assumed for these discrepancies. Changes and additions may be recorded in the next version of the document. Robert Bosch Manufacturing Solutions GmbH assumes no liability for damages or accidents that occur as a result of the following:

- Non-compliance with this document
- Use by untrained or unqualified personnel
- Use contrary to intended use
- Non-approved modification of the software/hardware

4 IT security and data protection

Intended use

The basic prerequisites required for using Nexeed Automation software products in compliance with IT security are as follows:

- The software specified for operating the machine is not intended for operation on open networks or wireless networks.
- Nexeed Automation software products are not intended for direct or unprotected operation on the Internet.
- Installations implemented using Nexeed Automation software products are not intended for operation in public areas, i.e. they should not be accessible to the public.

To ensure safe operation of Nexeed Automation software products, the following measures must be taken:

- Continuous training of operating and maintenance personnel regarding IT security and current IT security threats.
- The development of a suitable network zone concept. This includes network segmentation and controlled data flow between zones.
- Installation of an authorization management system and secure administration. IT components accessed and managed only by authorized personnel.
- Restrictive use of removable media and mobile devices.

If service is required:

- The operator must ensure that the internal and external service personnel with access to installations implemented using Nexeed Automation software products have been trained according to the operator's IT security guidelines.

Qualified personnel

All tasks and work listed must only be performed by qualified and authorized personnel.

- Nexeed Automation software products are designed to be operated by qualified personnel and maintained by technical personnel with the appropriate IT security training.
  Such personnel must work in accordance with an appropriate authorization and security management system.

Data protection

The protection of personal data and compliance with legal requirements regarding data protection are important to Robert Bosch Manufacturing Solutions GmbH.
As the developer of the Nexeed Automation software solution, we take the requirements of the General Data Protection Regulation (GDPR) seriously and abide by them with our
products by providing the technical capability to implement data protection legislation. The technical possibilities Nexeed Automation provides for this purpose can be found in the product documentation.
Compliance with data protection regulations is the responsibility of the operator of the software (responsible authority).

5 Cyle Time Assist für ctrlX

The product "Cycle Time Assist" allows the user to monitor and compare the cycle time of the machine as well as the runtimes of the movements and sequences of individual components (in the following commands). Cycle Time Assist Snap exists to collect and display this information. Snap is a service that receives and processes information. The information is displayed on a user interface. The source of this information is the PLC in which the machine processes have been programmed.

Cycle Time Assist offers libraries for the PLC. tThe application engineer must integrate these libraries in the project. The CycleTimeAssist_FB function block is located in the libraries. This FB is instantiated by the application engineer in the PLC project. The application engineer can program the FB in order to inform it of the start and end times of the commands, among other things. The FB itself has the ability to communicate this and other information to the service.

Create configuration for Cycle Time Assist in the Cycle Time Assist Studio. (Cycle Time Assist Studio see page 8)

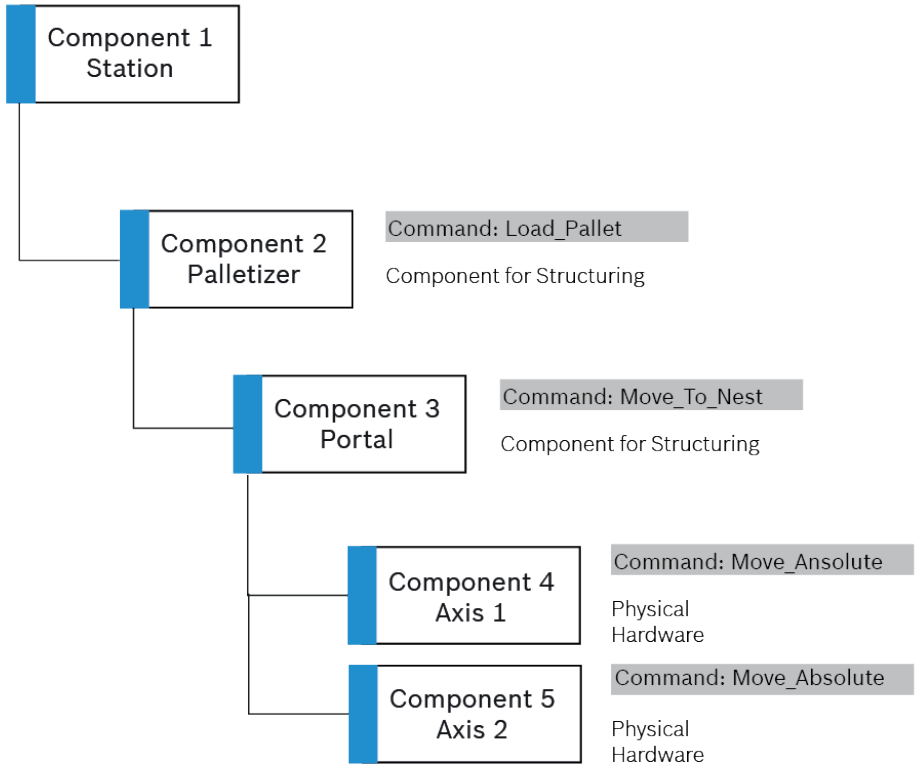
Install and set up Cycle Time Assit ctrlX App (Cycle-Time-Assist-App see page 22)

Call up and parameterize Cycle Time Assist with the help of the FB in the PLC (Cycle Time Assist FB see page 47)

5.1 Cycle Time Assist Studio

The Cycle Time Assist Studio makes it possible to work with Cycle Time Assist outside the Control-plus-Studio environment. With the help of the studio, the machine components and their commands can be created in a suitable control plus structure. The components and commands appear later with the assigned names in the PLC development environment, as well as in the Cycle Time Assist and the Cycle Time Analysis.

The relevant machine components and appropriate commands must be created in a corresponding structure in the studio. The Control-plus environment follows a hierarchical approach. Commands are passed through from top to bottom:

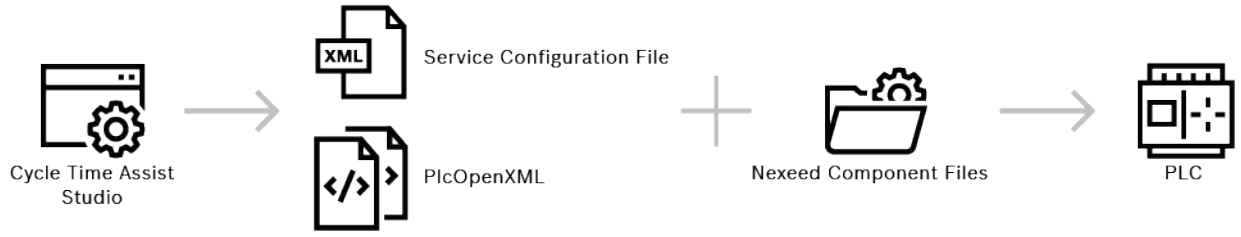


It is also possible to display the machine structure in a flat hierarchy. The machine structure is later presented in this form in the Cycle Time Analysis Tool.

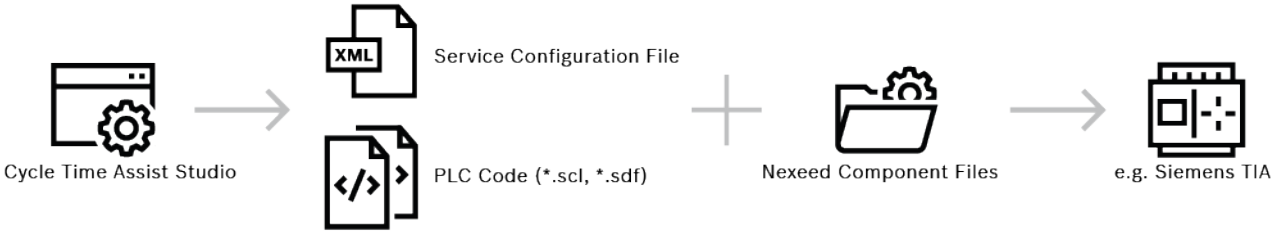
For Cycle Time Assist, only the components have to be created as a machine structure, which are also later relevant for Cycle Time Assist. The complete machine structure does not have to be reconstructed. After the structure is created, a configuration file and the appropriate SPS code can be exported. After the files have to be imported into the development environment with the files of the respective Nexeed component.

Depending on the selected target system, different files are output:

Any Target System



Siemens Target System



5.1.1 Installing Cycle Time Assist Studio

To install Cycle Time Assist Studio, the Cycle Time Assist Studio installation folder must be present.

1. Open the Cycle Time Assist Studio folder.
2. Double-click the *Bosch.Nexeed.Automation.CTAStudio.exe*.

The Cycle Time Assist Studio starts.

5.1.2 Creating a Project

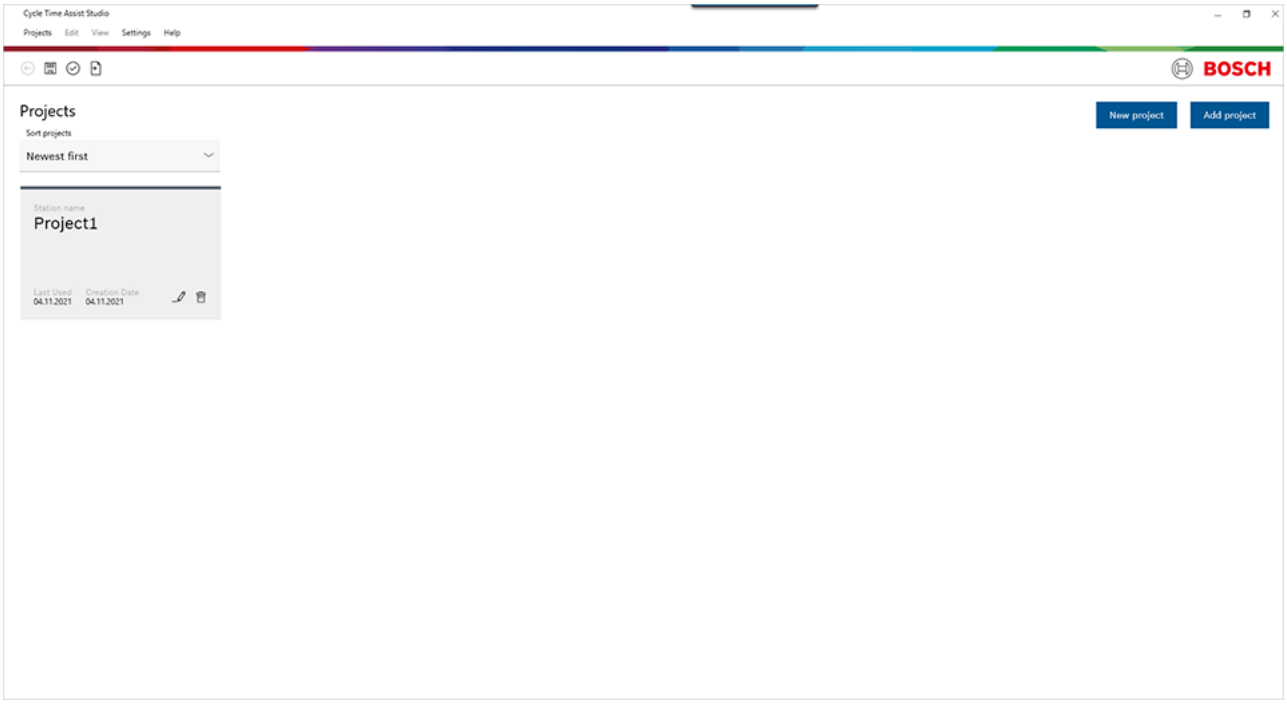
For each Cycle Time Assist application, a project must be created that contains the structure of the machine on which the Cycle Time Assist application runs.

1. Click *+New project*.

   The Save dialog opens.

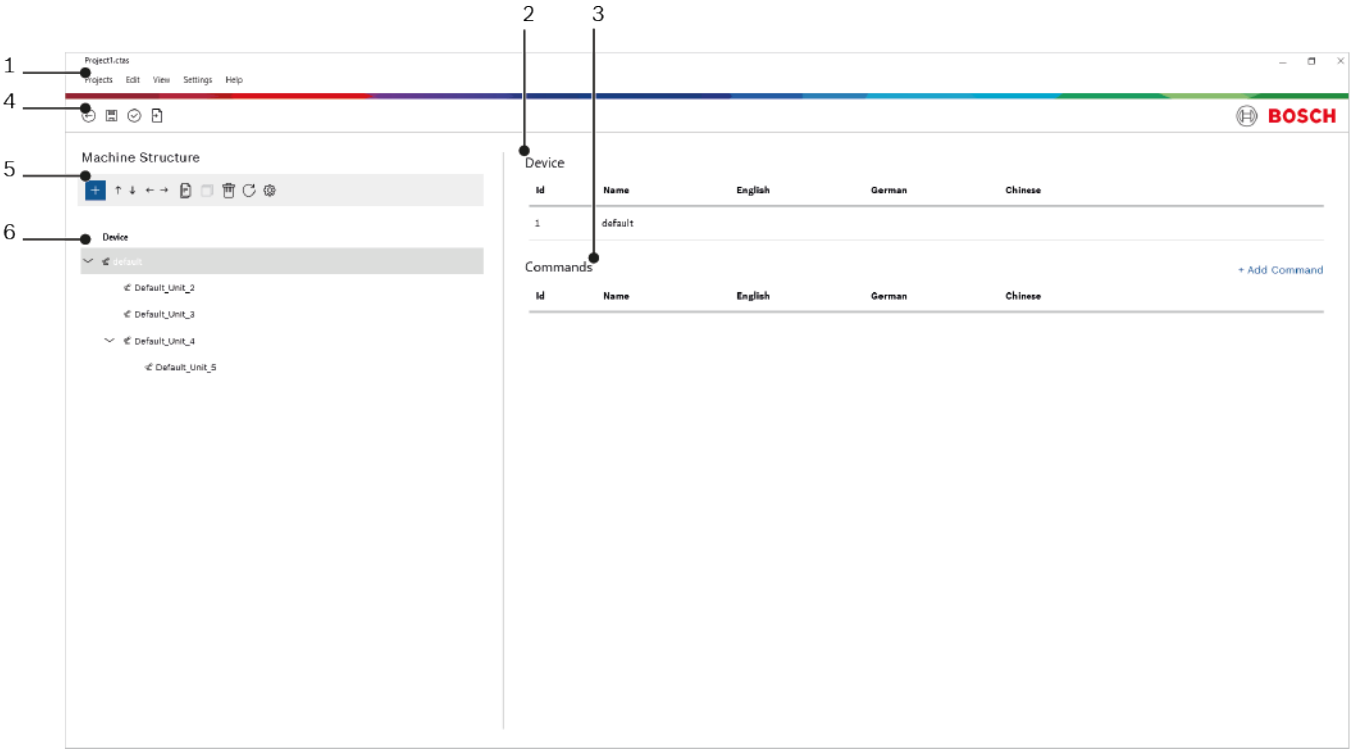
2. Assign a name for the project.

3. Click *Save*.



The project is created and can be configured.

5.1.3 Overview – Cycle Time Assist Studio



No.	Description
1	Menu
2	Device properties
3	Command properties
4	▪ Back to project overview ▪ Save project configuration ▪ Validate project ▪ Export project
5	▪ Add device. ▪ Move device up in the existing level. ▪ Move device down in the existing level. ▪ Move device up one level. ▪ Move device down one level. ▪ Delete device. ▪ Copy device. ▪ Insert device. ▪ Regenerate all command and device IDs. ▪ Additional project settings for Language, Export and Connection

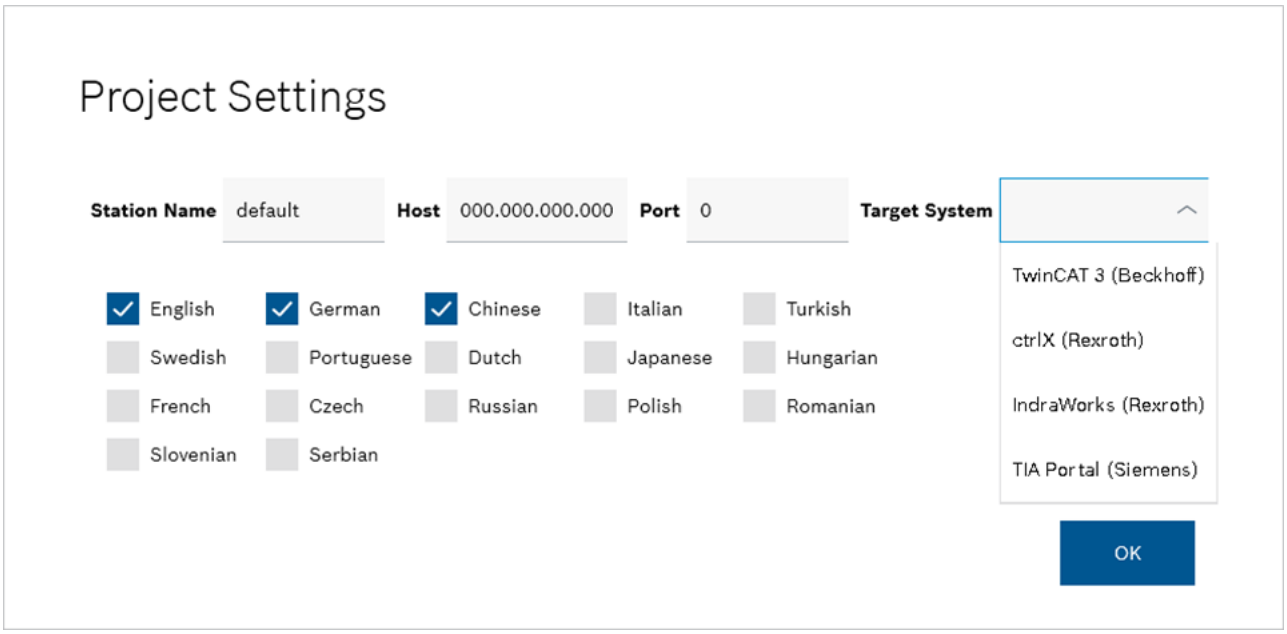
6	Machine hierarchy

5.1.4 Selecting a Target System

In order to export the appropriate PLC code files after the project has been completed, the target system must be selected in the Cycle Time Assist Studio.

1. In the toolbar, click Settings .

   The *Project Settings* window opens.



2. Select the appropriate target system from the Target System drop-down menu:
3. Confirm the selection with *OK*.

When the project is exported, the corresponding PLC code files will now be output.

5.1.5 Applying Language Settings

The model tree and machine structure can be maintained and exported in different languages.

Each language has an ID. The respective language entry is managed under this ID in the service configuration file (*.xml).

1. In the toolbar, click Settings .

   The *Project Settings* window opens.

2. Activate the checkboxes to select the language. The following languages can be selected:
   - English
   - German
   - Chinese
   - Italian
   - Turkish
   - Swedish
   - Portuguese
   - Dutch
   - Japanese
   - Hungarian
   - French
   - Czech
   - Russian
   - Polish
   - Romanian
   - Slovenian
   - Serbian

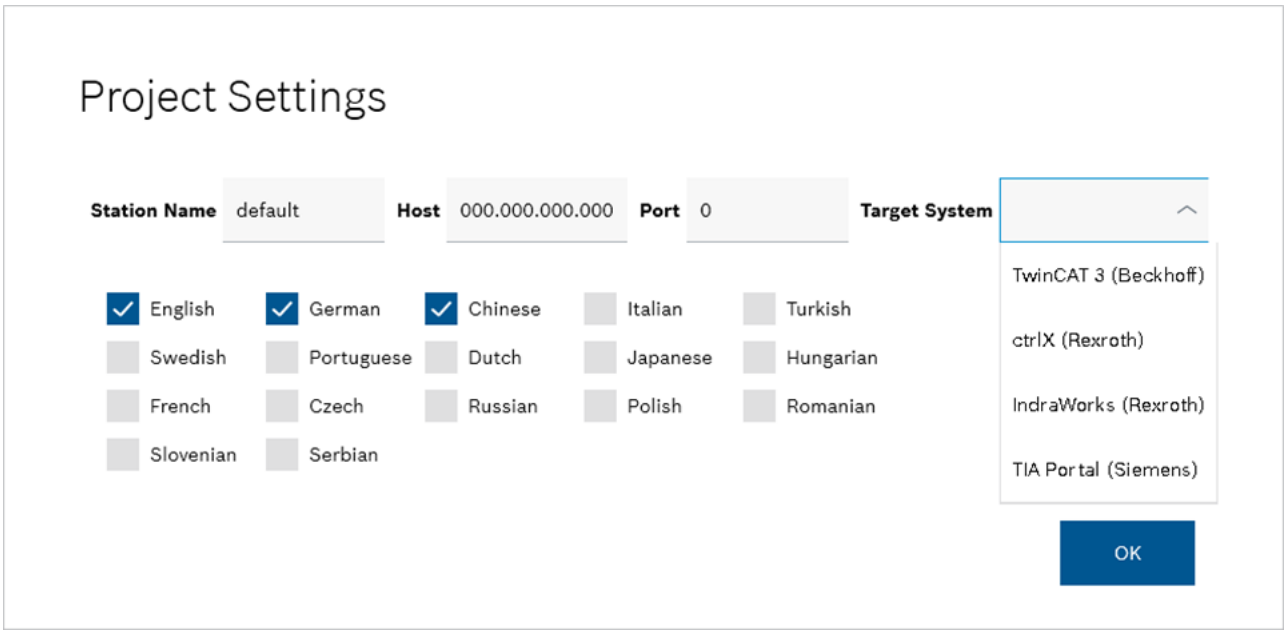
The selected languages are activated and appear as a text field for each node and command.

5.1.6 Applying Connection Settings

The Cycle Time Assist service can or sometimes must run on other systems. In order to ensure that the service can be subsequently actuated, the corresponding connection settings must be applied to the service.

1. In the toolbar, click Settings .

   The *Project Settings* window appears.



2. Implement the following settings:
   - **Station Name:** Enter the name of the station.
   - **Host:** Enter the IP address of the machine. If nothing is specified, it will correspond to the same system
   - **Port:** Specify the port (standard 56090).
3. Click *OK*.

The settings are applied to the active project.

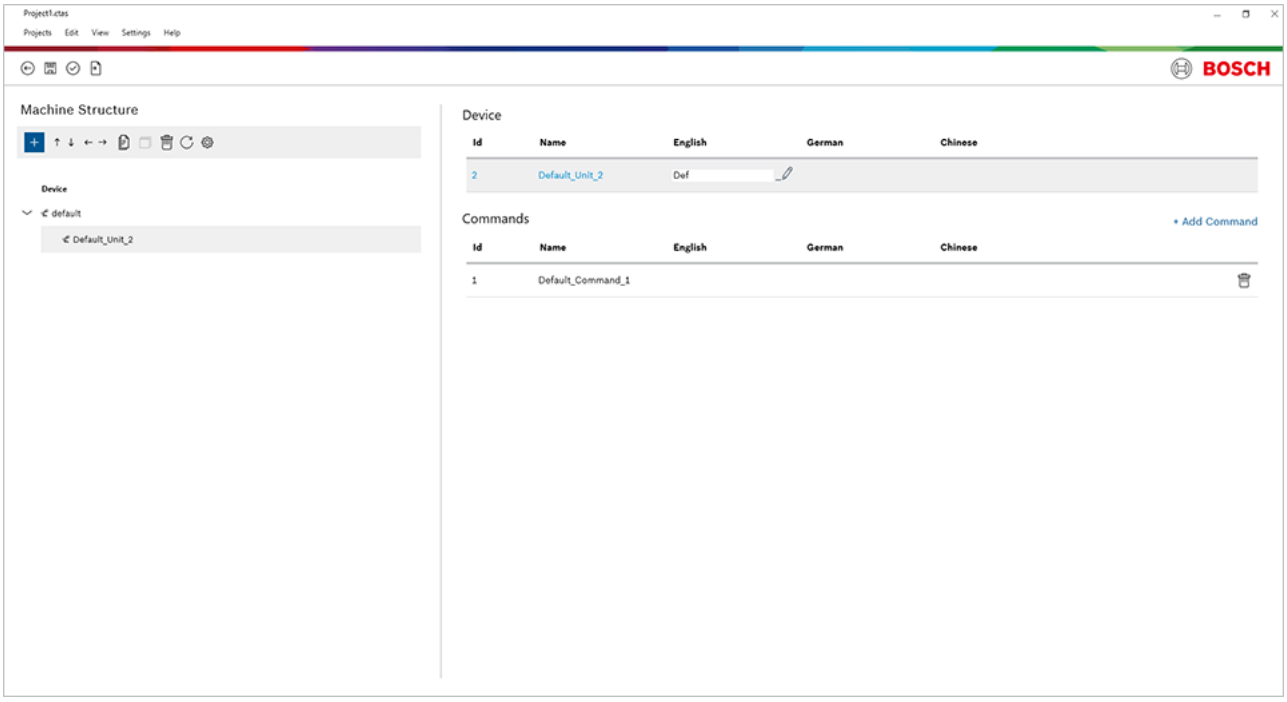
5.1.7 Creating a Component in the Machine Structure

The actual machine structure is mapped in the Cycle Time Assist Studio. To do this, the corresponding components must be added to the machine structure with their commands.

The machine structure can be nested as desired, but the first component or node is always set as the parent element. All other components are organized below this on a second level.

1. In the toolbar, click Add  .

   The component is created.



2. In the Device table, in the *Name* text field, enter a name for the created component.

   An automatically generated ID appears in the InstanceID text field. This ID is incremented with each created element (component or command). The ID can be changed, if necessary.

3. Create the respective language values.

The name is transferred to the machine structure. Commands can now be added to the created component.

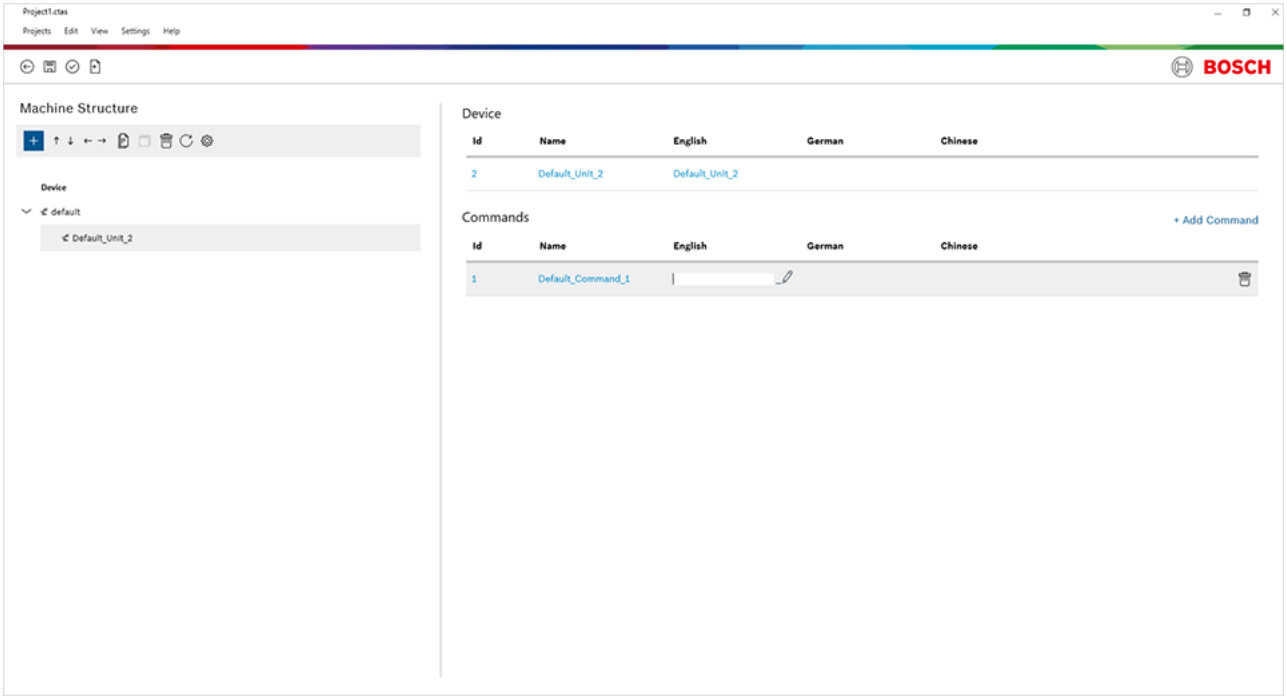
5.1.8 Creating Commands for a Component

For each created component, corresponding commands can be created that describe the functions of the component.

1. In the machine structure, under *Device*, select the component for which a command is to be created.
2. Click *+Add Command* in the lower-right corner of the window.

   A new entry appears in the *Commands* table.

   An automatically generated ID appears in the ID column. This ID is incremented with each created element (component or command). The ID can be changed, if necessary.



3. In the Name column, enter a name for the command.
4. Enter data for the respective languages.
5. To delete the command, click on the trash can after the entry.

The command is created.

5.1.9 Saving the Cycle Time Assist Studio Project

The project can be saved as a *.ctas file so that the project can be edited again at a later time in the Cycle Time Assist Studio.

1. Click Save to save the project file.

   A message window appears stating that the save process was successful.

2. Confirm the message with *OK*.

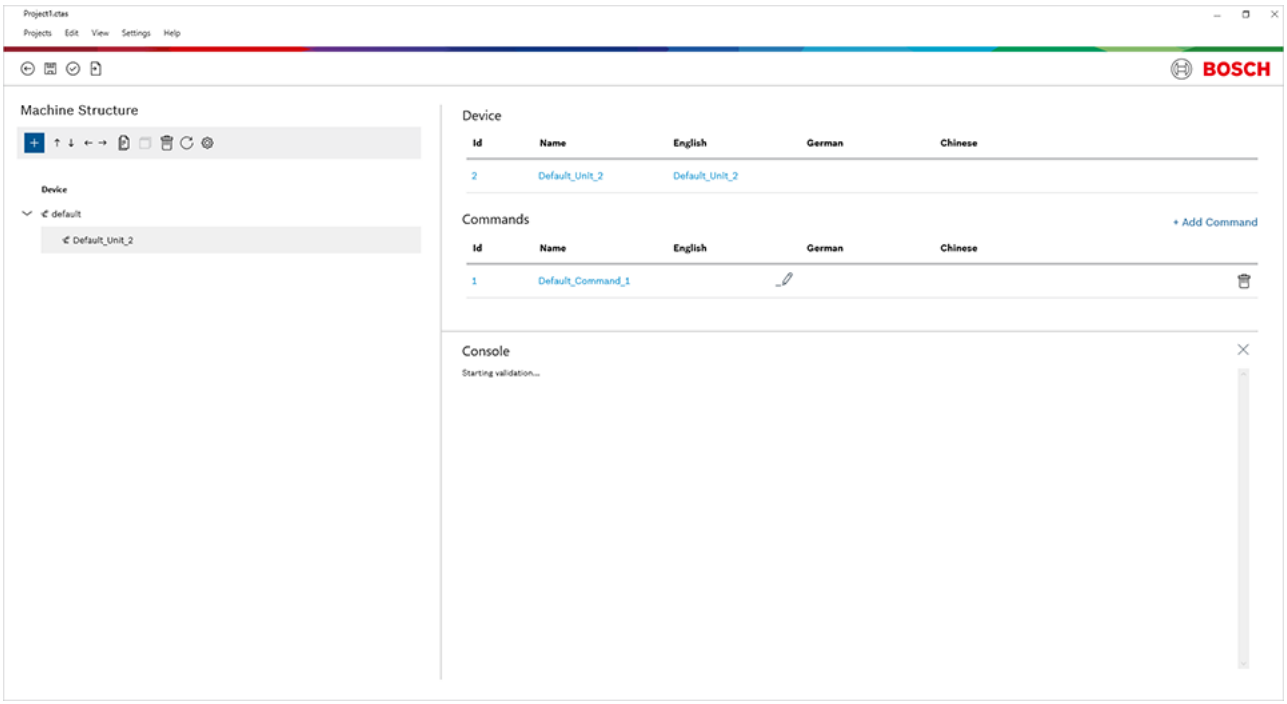
The project is saved under the path that was selected for the project.

5.1.10 Validating the Cycle Time Assist Studio Project

The machine structure can be validated to ensure that it has been created correctly with the components and commands.

1. Click Validate to validate the project file.

   A console appears.



Possible errors or success messages appear in the console. If the project is valid, it can be exported as a PLCOpenXML file.

5.1.11 Exporting the Cycle Time Assist Studio Project

When you export the Cycle Time Assist Studio project, two files are created: The CTA service configuration file and a file with the corresponding PLC code.

The xxxCtaService.xml must be exported to the Cycle Time Assist app and the PLC code file exported to the PLC development environment. The PLC code can be programmed using the Cycle Time Assist FB.

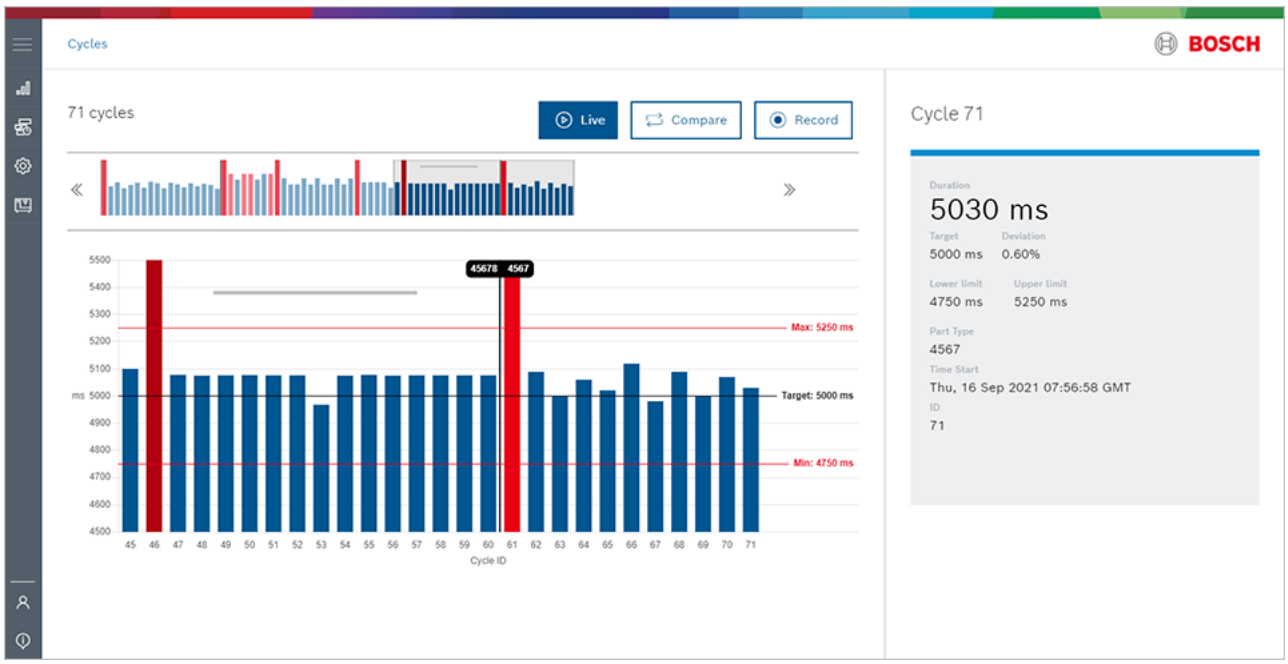
1. Click Export 
2. Select a folder to save the export files in or create a new one.

   The console appears. The project is validated and then exported.

After a successful export, the storage path appears in Windows Explorer.

5.2 Cycle-Time-Assist-App

In the cycle time assist app, the part cycles of the machine are visualized according to the implementation by the CycleTimeAssist add-on in the PLC. In the add-on, the PLC specifies the part cycle, the target cycle, and the permissible deviations and displays them in the app. Recordings for the part cycle can be started via the PLC as well as via the app, in which all individual movements of the station are recorded. Cycles for which a recording is available can be compared directly in the app. The individual movements of the cycles with the largest deviations are shown in a chart. For a more detailed analysis of the recordings, the *Cycle Time Analysis* Office Tool is available. Recordings from the app can be exported as .xml and .json files and read into the Cycle Time Analysis Tool.



5.2.1 System Requirements

The following preconditions must be met in order to use the product:

- Existing Internet connection
- Supported browsers: Google Chrome, Microsoft Edge or Mozilla Firefox
- Existing Bosch ID for each user (Creating a Bosch ID (see page 24))
- Cycle Time Assist Configurator installed
- Cycle Time Assist configuration file (xml)
- Nexeed app installed (Cycle Time Assist) with the appropriate license

5.2.2 Creating a Bosch ID

In order to be able to use the online services offered by Bosch and Bosch Rexroth, users must complete a registration process, during which a Bosch ID is created.

The online services include, for example, the ability to download and use software components and apps.

Preconditions

- Existing Internet connection
- Email address

1. Open the App Zone on the Rexroth website.
2. Click on *my Rexroth*.

   The Bosch ID login page appears.
3. In the login window, click *Not yet registered*.

   The registration window appears.
4. Read the General Terms and Conditions and Privacy Policy.

   A Bosch ID can only be created once you have accepted the General Terms and Conditions and Privacy Policy.
5. Enter an email address and password for the Bosch ID.
6. Accept the General Terms and Conditions and Privacy Policy.
7. Click *Register*.

   An automatically generated email is sent to the specified email address.
8. Confirm the registration via the email.

The Bosch ID has been created and can now be used for the Bosch online services.

5.2.3 Downloading the Nexeed App

The Nexeed apps are located in the App Zone for Bosch Rexroth ctrlX products.

Preconditions:

- Valid license for the relevant Nexeed app.

1. Go to the App Zone.
2. Use the search function to go to the corresponding Nexeed app.
3. Download the app.

The app is now downloaded to the PC and must now be installed in the ctrlX environment.

5.2.4 Installing the Nexeed App in the ctrlX Environment

To use the app in the ctrlX environment, it must be transferred from the PC to the ctrlX controller and installed there.

Preconditions

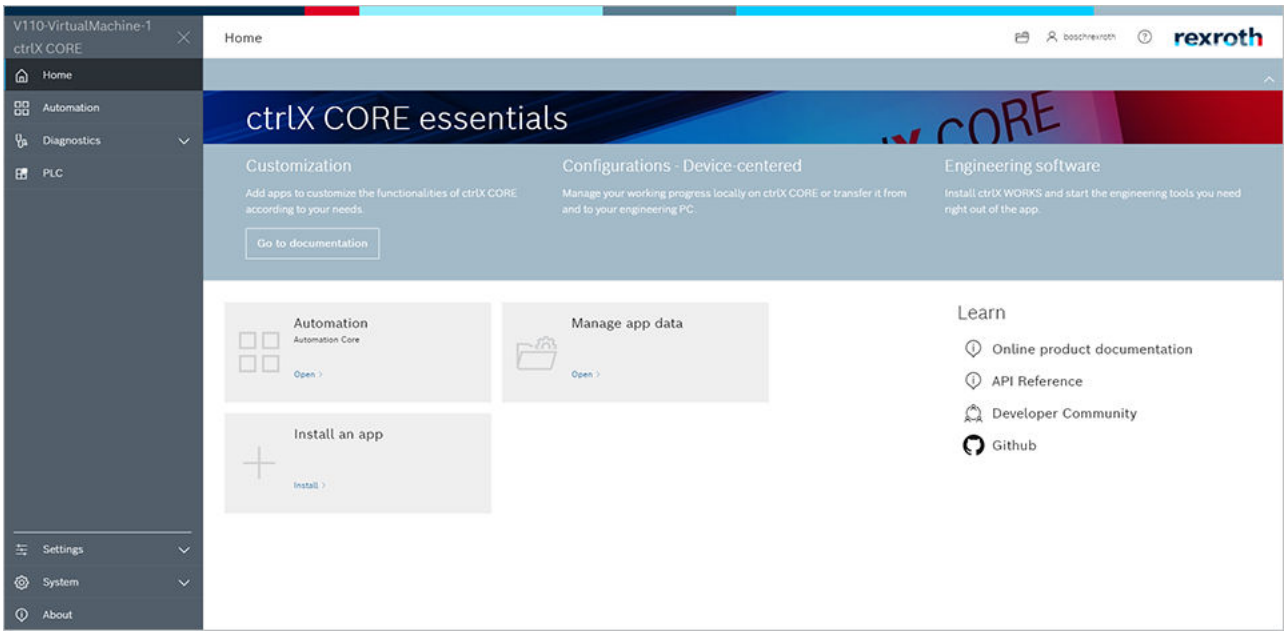
- App downloaded from the ctrlX App Store (*.snap file)
- Access to the ctrlX controller
- Login for the ctrlX homepage
- Permissions

1. Open the ctrlX homepage.

   Enter the IP address of the controller into the browser (web server for the controller). It is not necessary to establish another connection to the controller.

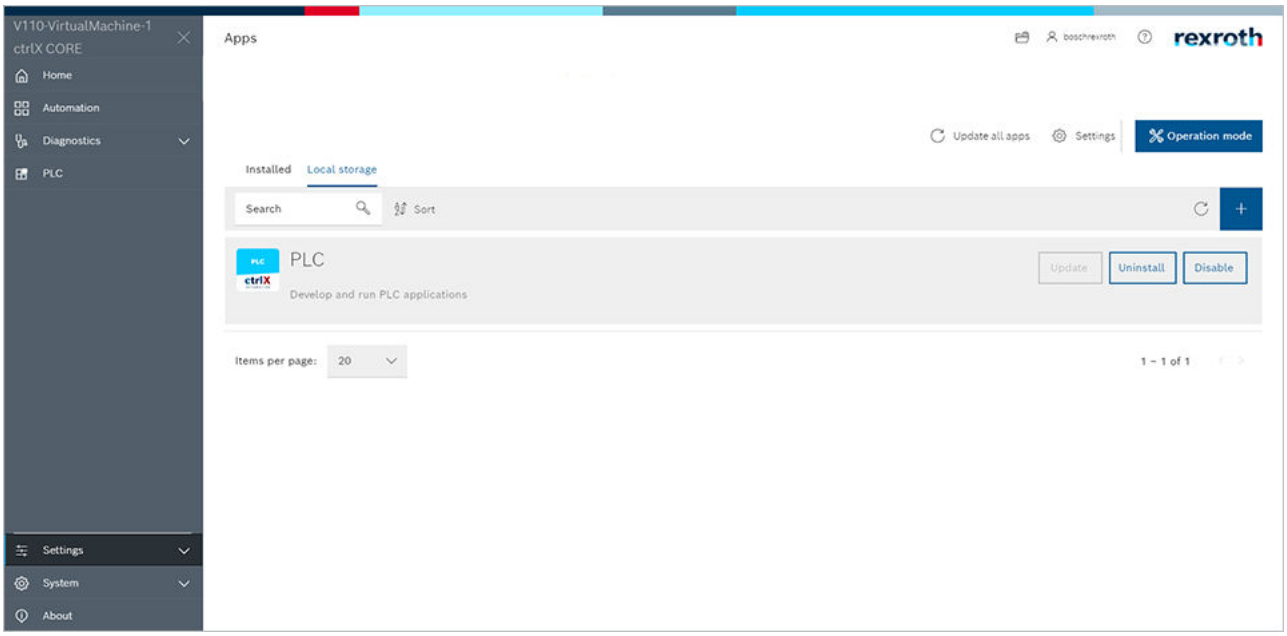
2. Select *Home* in the menu.

3. Click on *Install an app*.



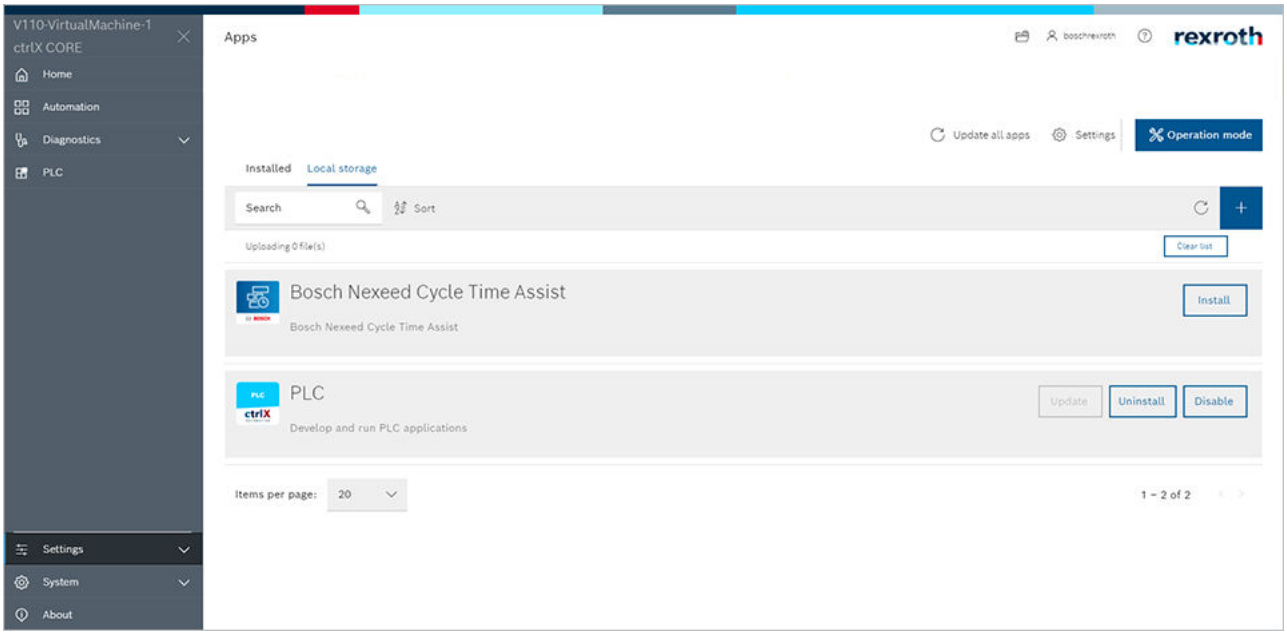
   The app overview appears.

4. Select the *Local storage* tab.



5. Click *Add*  to add the app stored locally on the PC.

   The app appears in the list under Local storage.



6. Click on *Install*.

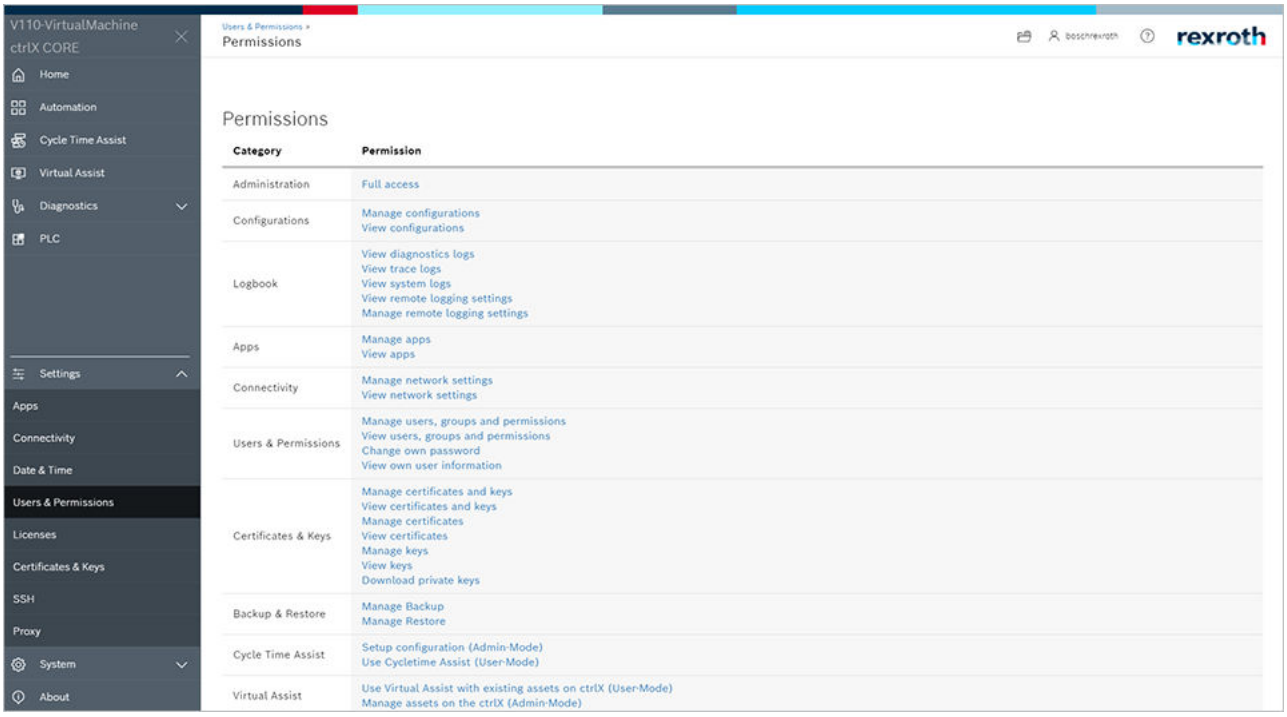
   The app is installed on the controller.

The app is installed and a corresponding confirmation message appears. The app can now be configured.

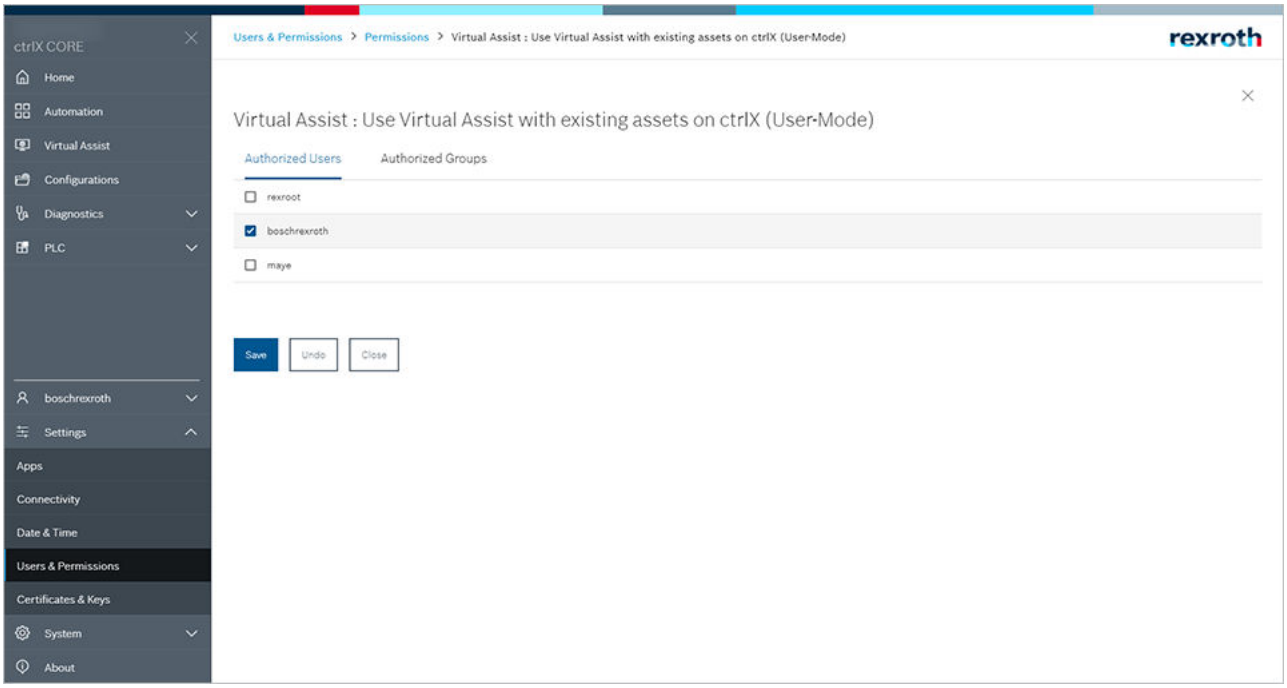
5.2.5 Managing Users and Permissions

In this view, the permissions can be assigned for each of the individual areas, components and/or apps.

1. In the menu, navigate to *Settings>Users & Permissions>Permissions*.



2. Select the permissions for the *User Mode* of the app (Cycle Time Assist or Virtual Assist) in the *Permission* column of the respective app.

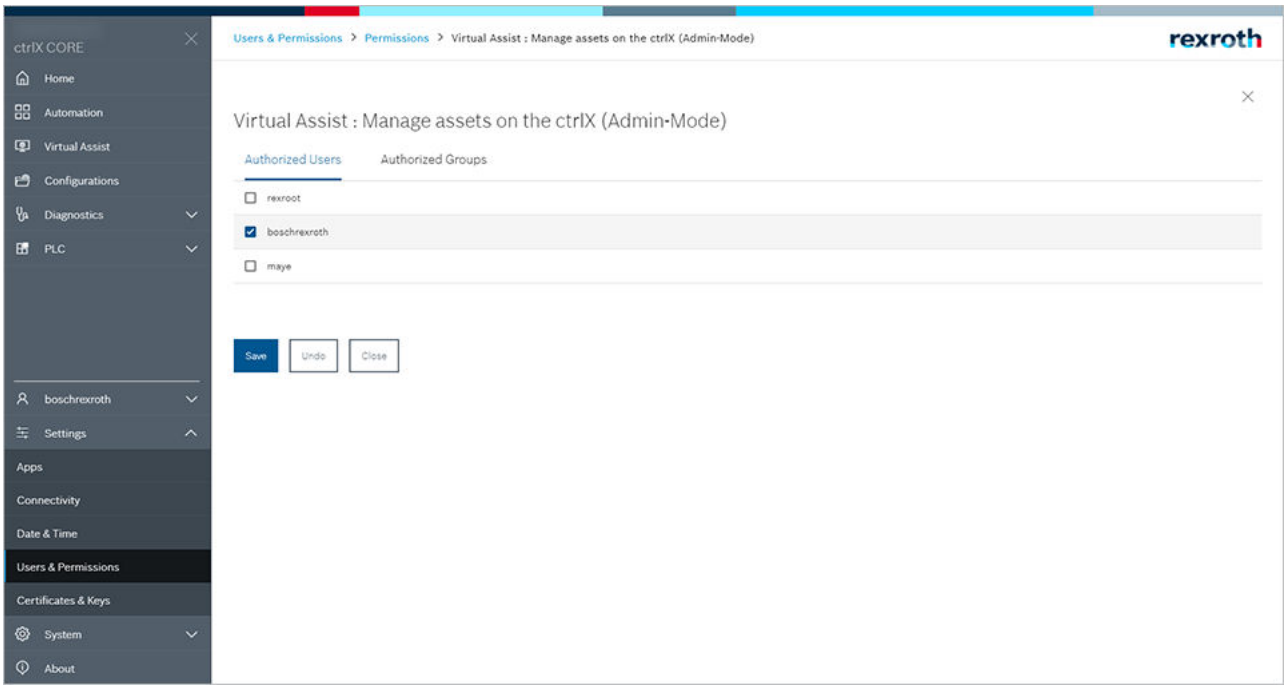


3. Assign the permissions for the *User Mode* to the individual users and/or groups. Take into account the recommended information for users and rights.

4. To save the settings, click on *Save*.

   The *Users & Permissions* overview appears again.

5. Select the permissions for the *Admin Mode* of the app (Cycle Time Assist or Virtual Assist) in the *Permission* column of the respective app.



6. Assign the permissions for the *Admin Mode* to the individual users and/or groups. Take into account the recommended information for users and rights.

7. To save the settings, click on *Save*.

The created users and assigned permissions are up-to-date.

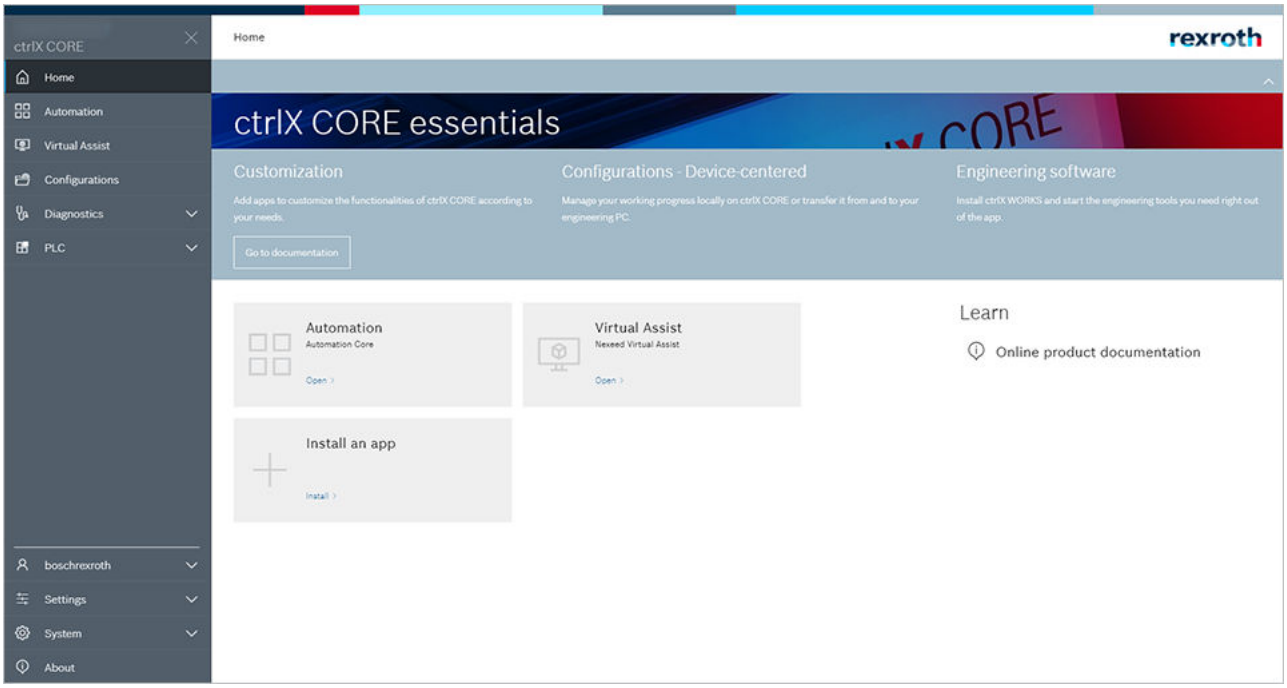
5.2.6 Uninstalling the Nexeed App from the ctrlX Environment

To uninstall an app from the ctrlX environment, it must first be uninstalled from the controller. After that, the app will still be available locally.

Preconditions:

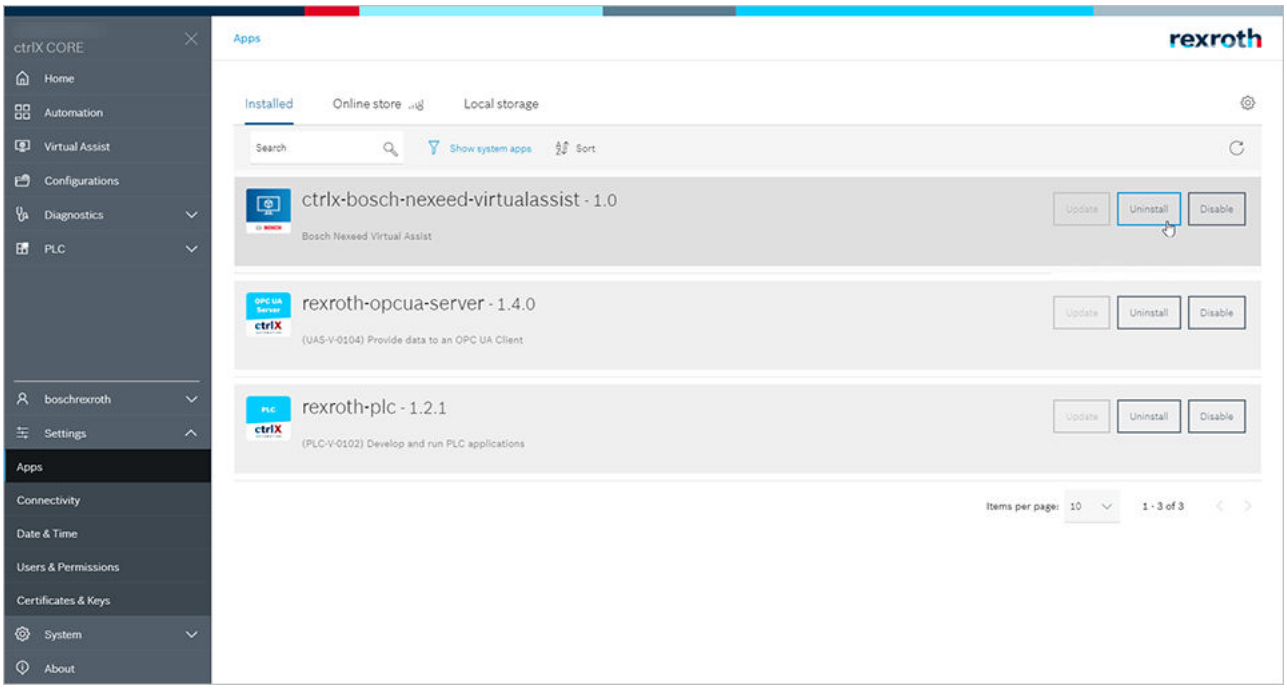
- Access to the ctrlX controller
- Login for the ctrlX homepage

1. Open the ctrlX homepage.
2. Establish a connection to the ctrlX controller.



3. In the menu, navigate to *Settings>Apps*.

   The app overview is displayed.



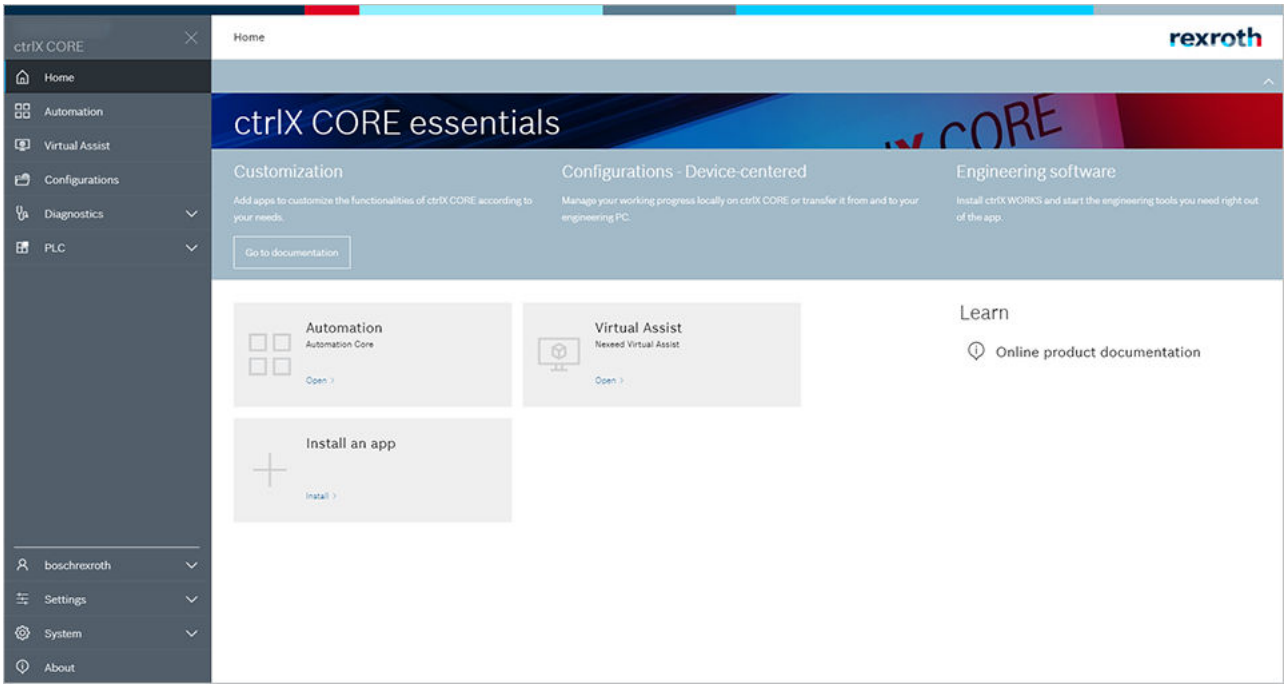
4. Click on *Uninstall* in the row for the app in question.

The app is uninstalled from the controller. Since the app is still available locally, the *Install* button is still active in the portal. This button can be used to reinstall the app on the controller.

5.2.7 Opening the Nexeed App in the ctrlX Environment

In order to open a Nexeed app in the ctrlX environment, the app must be installed on the ctrlX controller.

1. Open the ctrlX portal.



2. Click on the tile for the app you want to open.

The selected app opens in the browser.

5.2.8 Live-Ansicht

The app starts in the Live view. The Live view shows the current part cycles.

The scale of the view is fixed and depends on the target cycle and permitted deviations. The values of the Y-axis are linear and do not start at 0.The values depend on the set min and max values in the PLC. Therefore, the set values should be chosen meaningfully.

The target cycle is indicated by a black line.

All part cycles that are in the target value range are displayed in blue. Part cycles that exceed the range of values become dark-red.

The Live view shows the last 1000 part cycles.

No.	Description
1	Collapse and expand the menu
2	Recorded measurements (Overview of All Recordings see page 44) The icon shows the number of measurements recorded so far.
3	Menu ▪ Re-set the chart ▪ Information about the app
4	Settings
5	ctrlX homepage
6	User information
7	Version information
8	Part cycle that exceeds the values of the scale

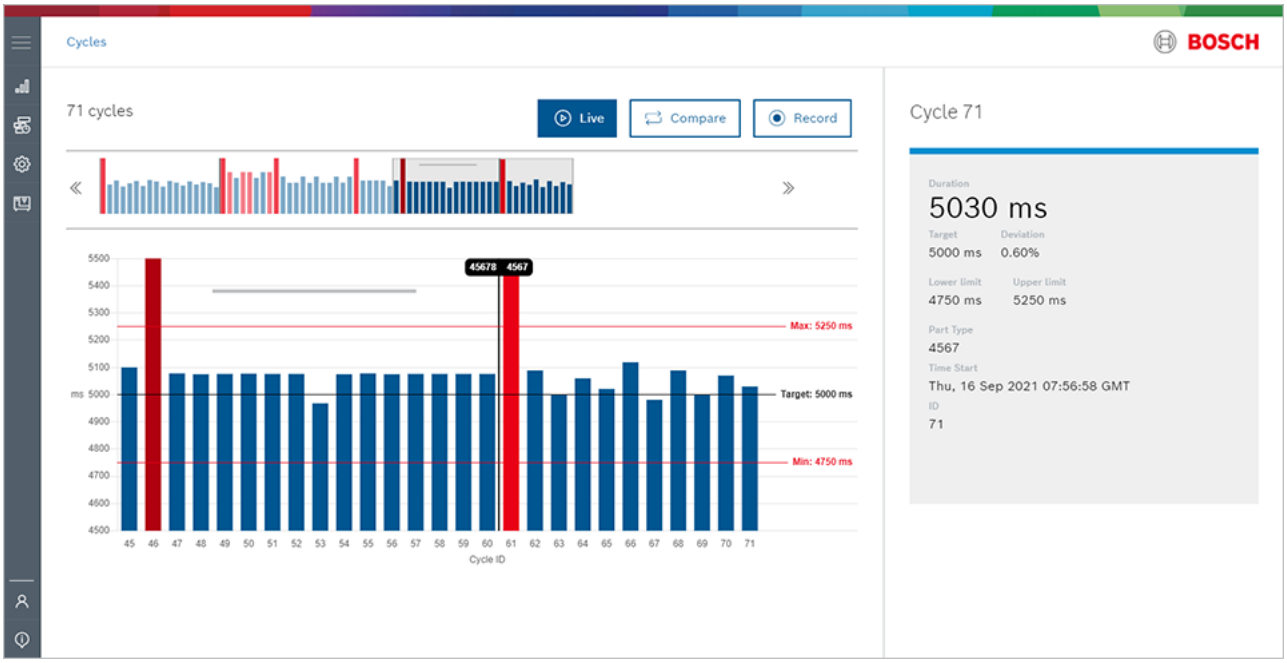
9	Complete overview of live values. The displayed part can only be adjusted if the Live view is deactivated.
10	Live view The Live view can be switched on and off via the button. ▪ If the button is blue: Live view is on, the current cycles are displayed. ▪ If the button is white: Live view is switched off. The view is frozen.
11	Comparison View (see page 39)
12	Record measurements (Starting a Recording see page 36)
13	Part cycle information Info window is displayed when a part cycle is selected.
14	Displays the part cycle measured up to this point.
15	Part cycle
16	Scale The scale is fixed.

5.2.8.1 Freezing the Live View

The function can be used, for example, to look at individual cycles more closely or to switch to older values in the overview display.

Only the view is frozen – the measurements continue to run in the background. When the user returns to the Live view, the view jumps to the most recent cycle.

1. Click on the blue *Live* button.

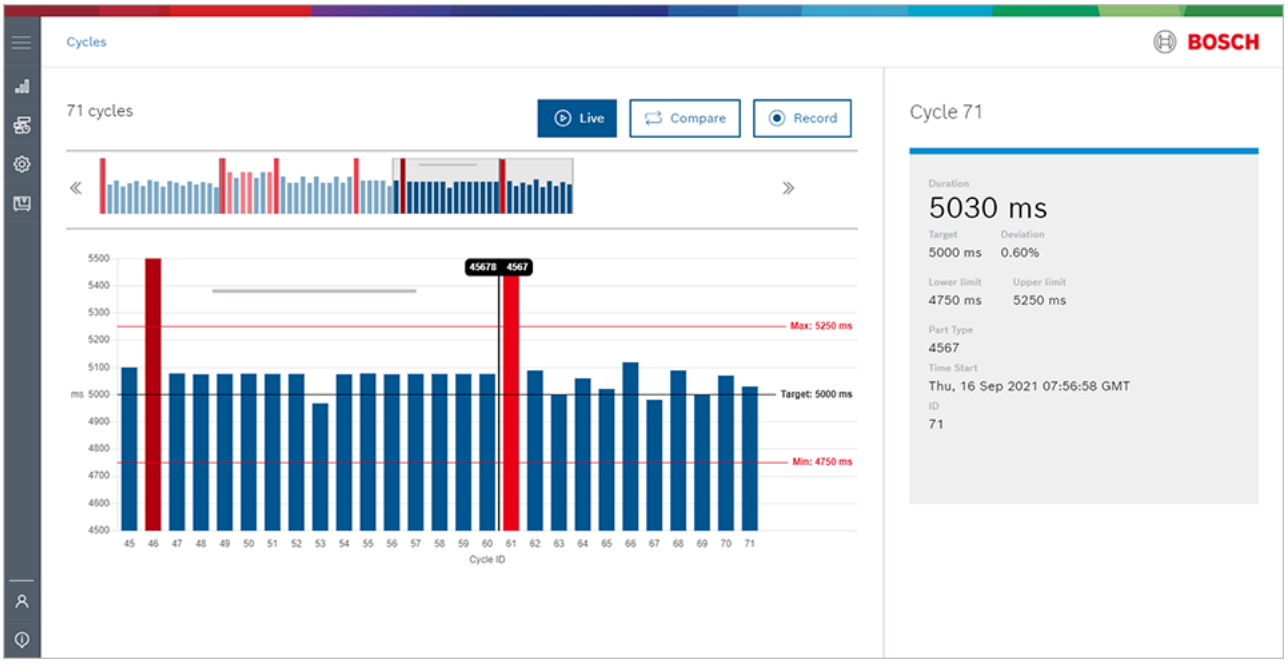


The button turns white and the view is frozen. Click on the button again to continue the view.

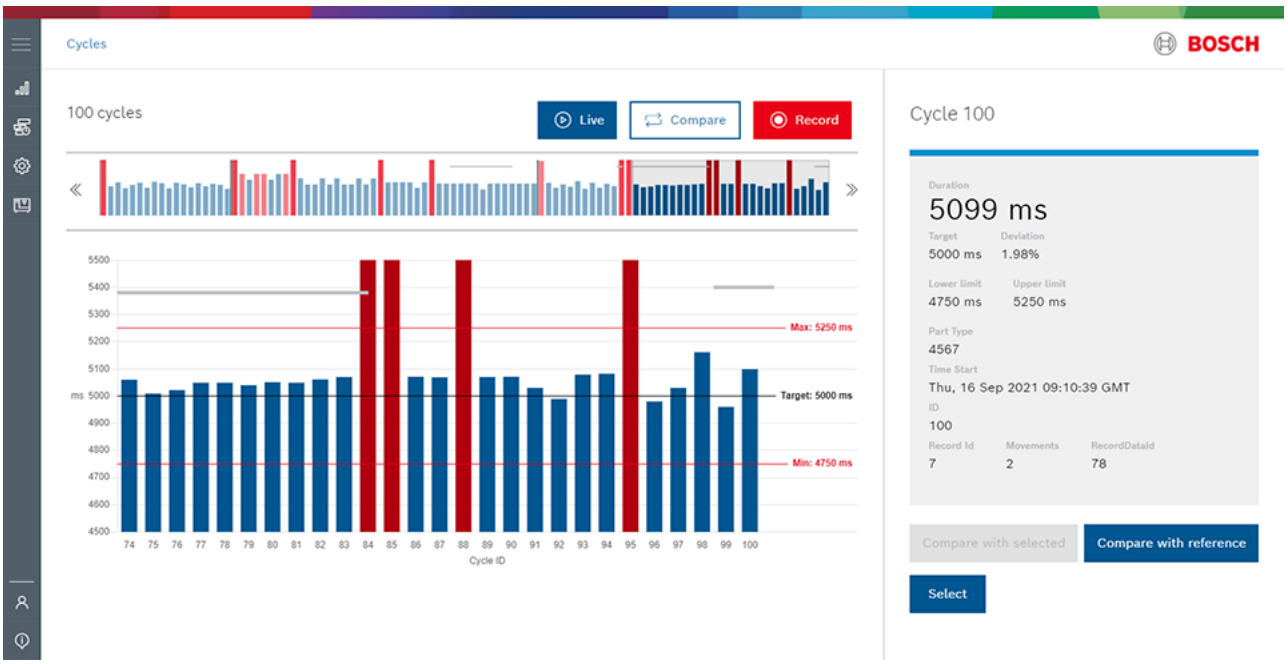
5.2.8.2 Starting a Recording

In the Live view, recordings can be started and stopped. Recordings contain all the individual movements of the station and enable a detailed evaluation of the part cycles.

1. Click on the *Recording* button.



The button turns red and the recording starts.



2. Click on the *Recording* button again to stop recording.

The measurement is recorded and appears in the measurement overview. Overview of recorded measurements (Overview of All Recordings see page 44)

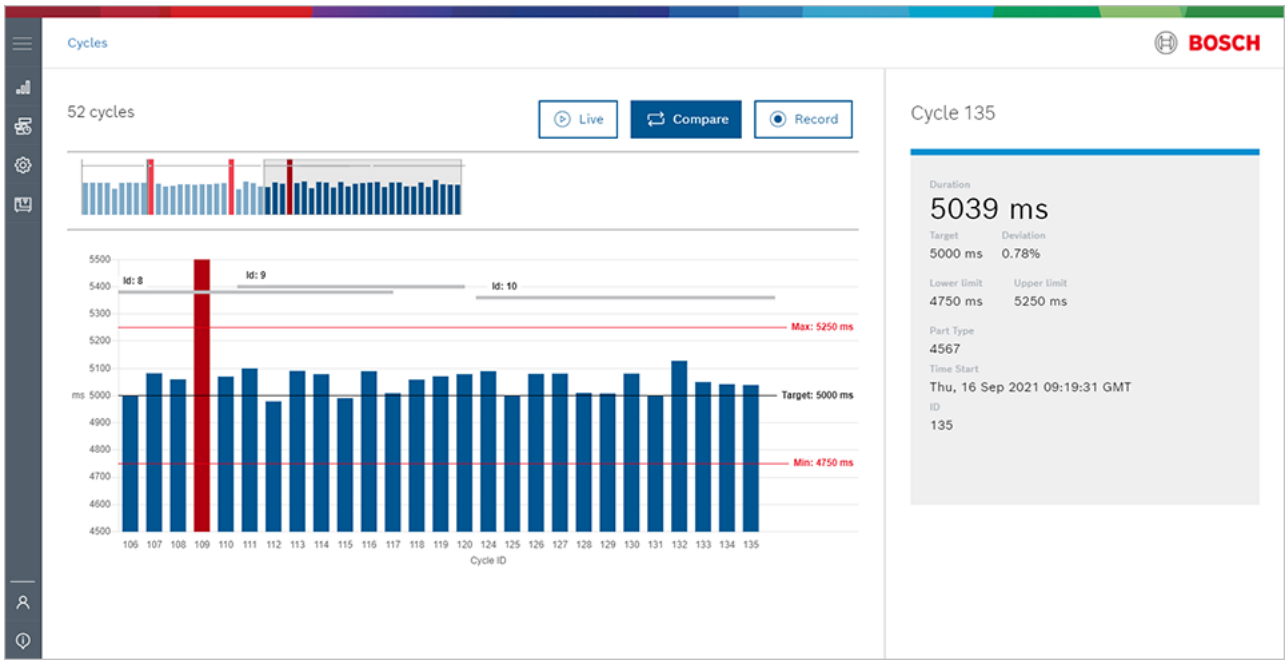
5.2.8.3 Invalid Part Cycle

Errors, changeover processes or production breaks always result in situations in which production is no longer continued and therefore no new part cycles are available. The PLC programmer can take these cases into account for the Cycle Time Assist app by setting notifications for the app in its workflow. In the app, the interruptions to the standard part cycle sequence are represented by a gray area. The gray area is a placeholder and is always the same width. The width of the area is not related to the time span.

5.2.9 Comparison View

In the Comparison view, all recorded measurements are displayed with the respective part cycles. A gray bar summarizes all the part cycles of a measurement. Click on the bar to display an info window for the measurement.

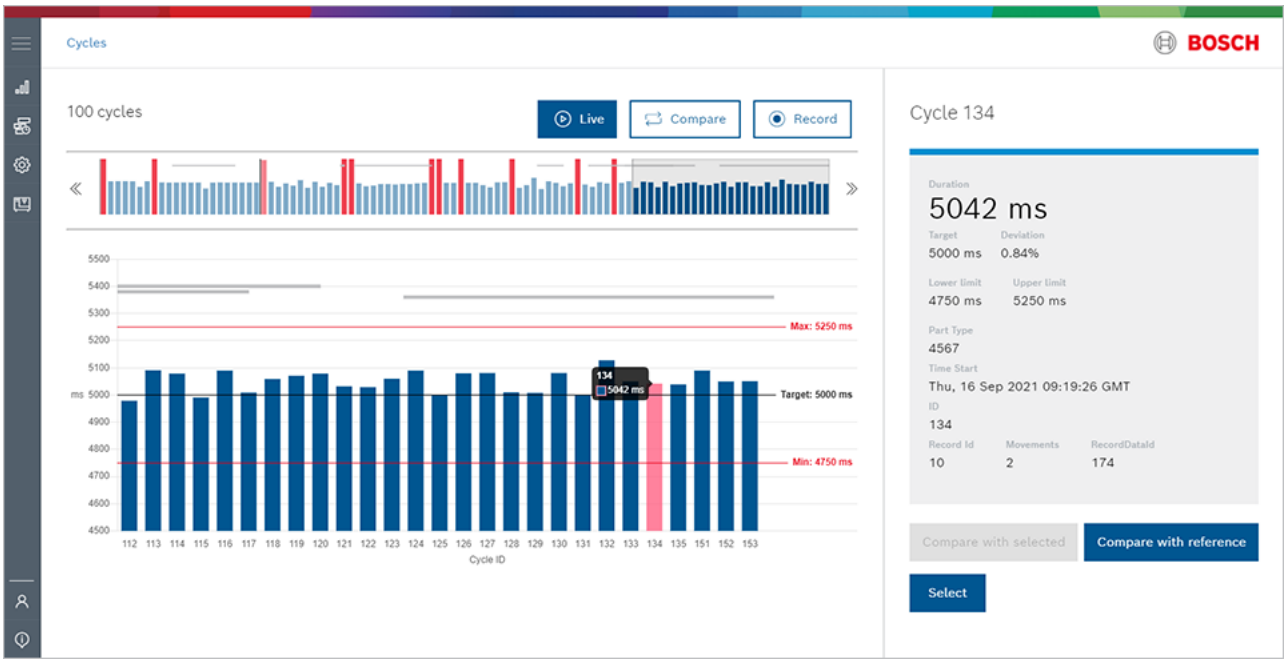
In the view, cycles of individual measurements can be compared to one another and reference cycles can be selected for comparison. Comparing cycles with one another (Comparing Part Cycles With One Another see page 40)



5.2.9.1 Comparing Part Cycles With One Another

Individual part cycles of a recorded measurement can be compared with one another.

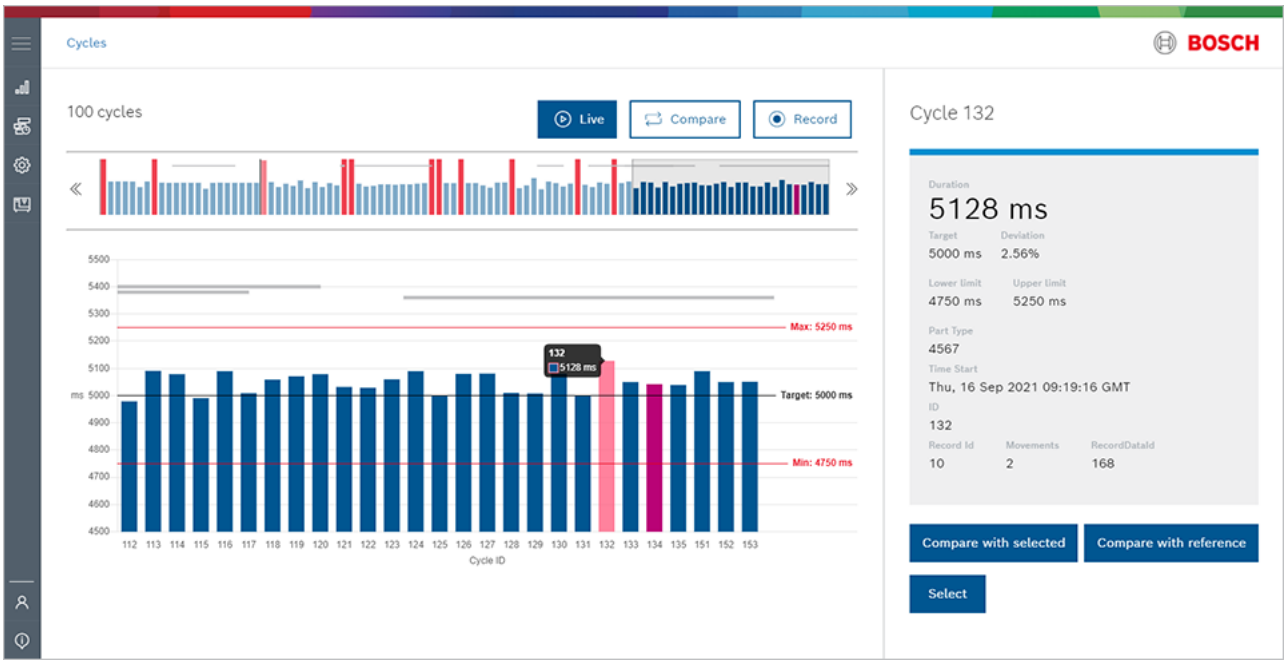
1. Select a part cycle to compare



   The info window appears.

2. Click *Select*.

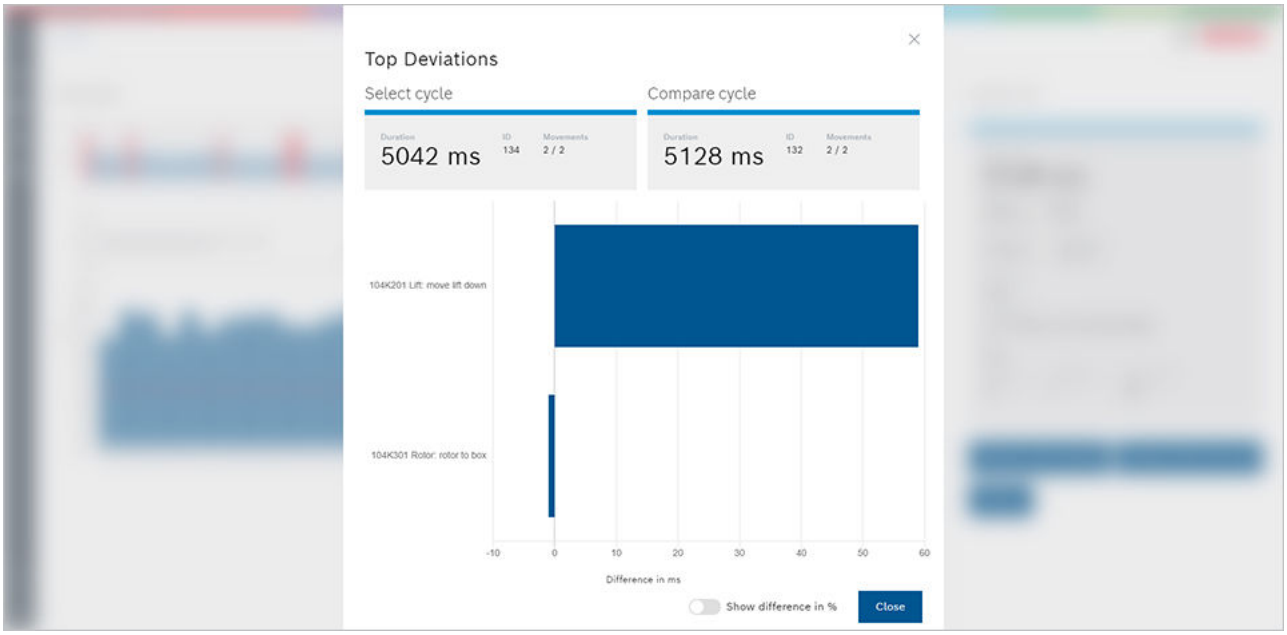
   The selected part cycle is highlighted violet.



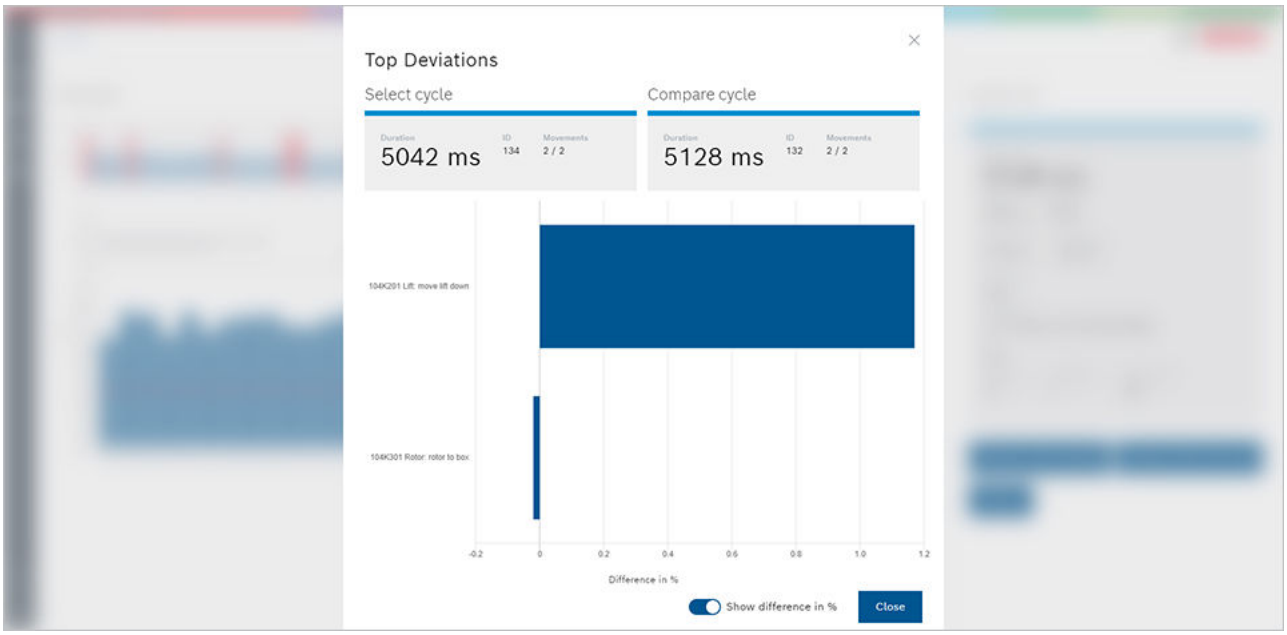
3. Select another part cylce and click *Compare with selected*, to add the part cycle to the comparison or

Click *Compare with reference* to compare the selected part cycle (violet) with the reference part cycle.

The deviations of the two part cycles are displayed.



4. Click the switch *Show difference in %* to the right for displaying the deviations in %.



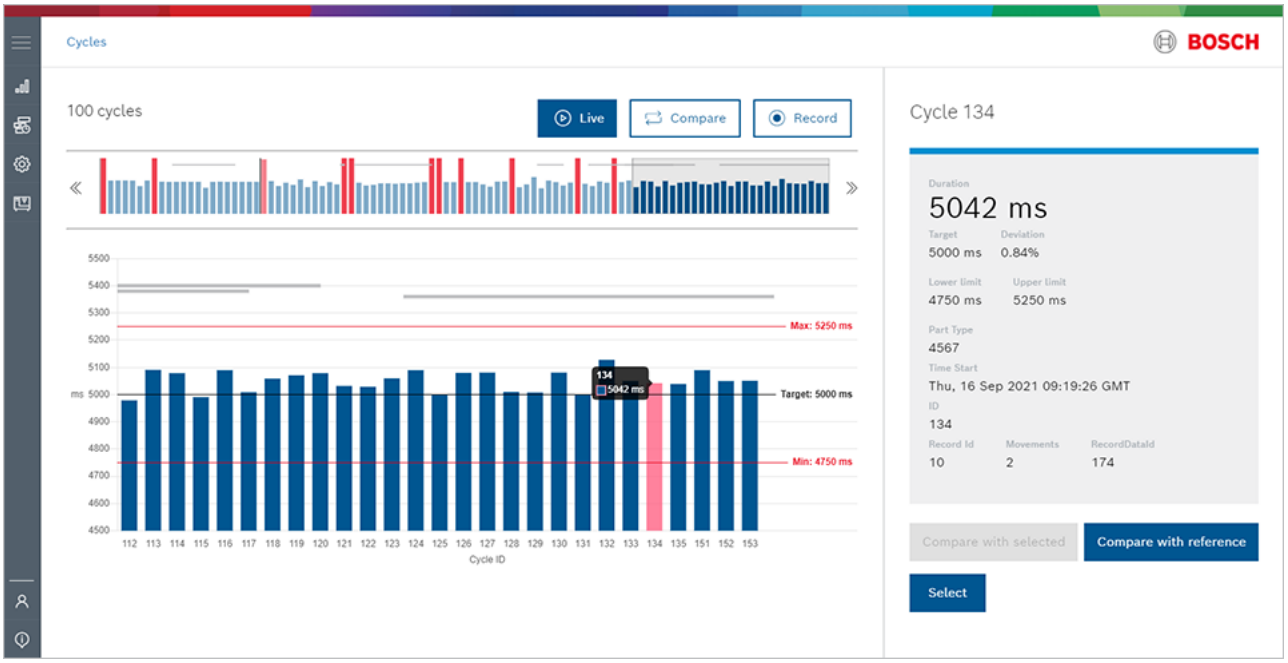
5. Click *Close* to close the deviation.

The part cycles can be compared with one another and exported to the Cycle Time Analysis Tool for closer examination. Export a measurement (Exporting a Recording see page 45)

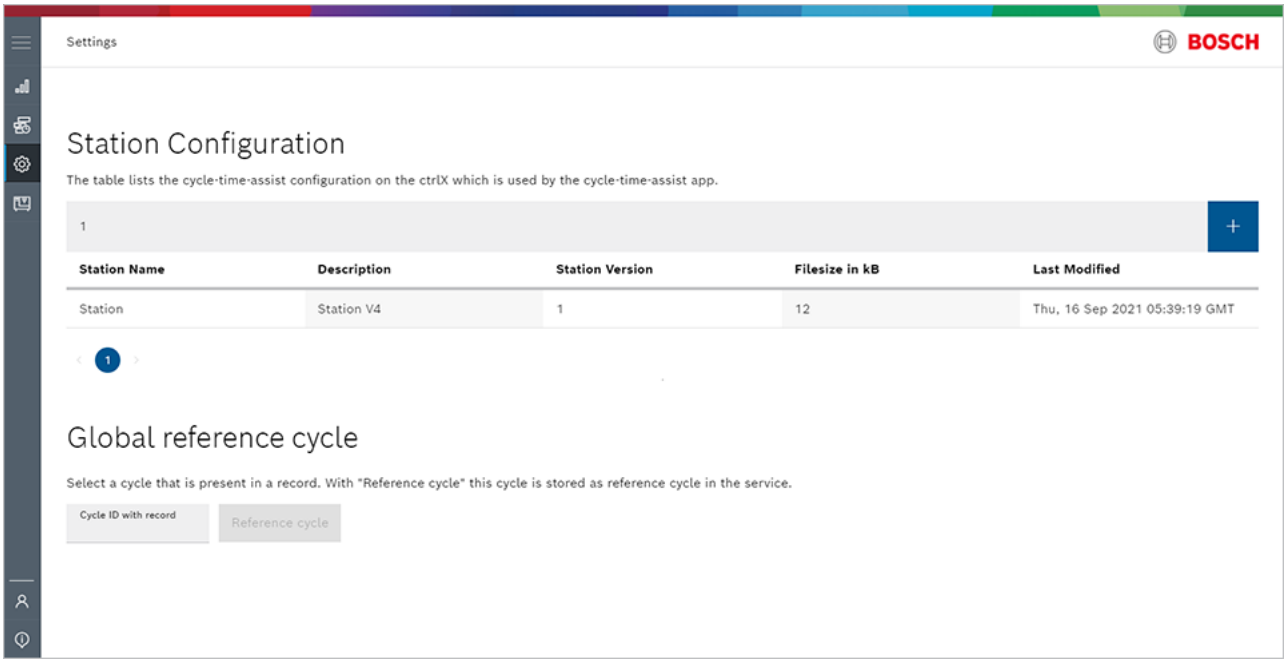
5.2.9.2 Setting a Reference Cycle

If the values of a specific part cycle were particularly good, the cycle can be set as a reference cycle (there must be a recording of this cycle). This reference cycle is always used for comparison purposes.

1. Click on a part cycle to be set as a reference.



2. In the menu open the settings  .



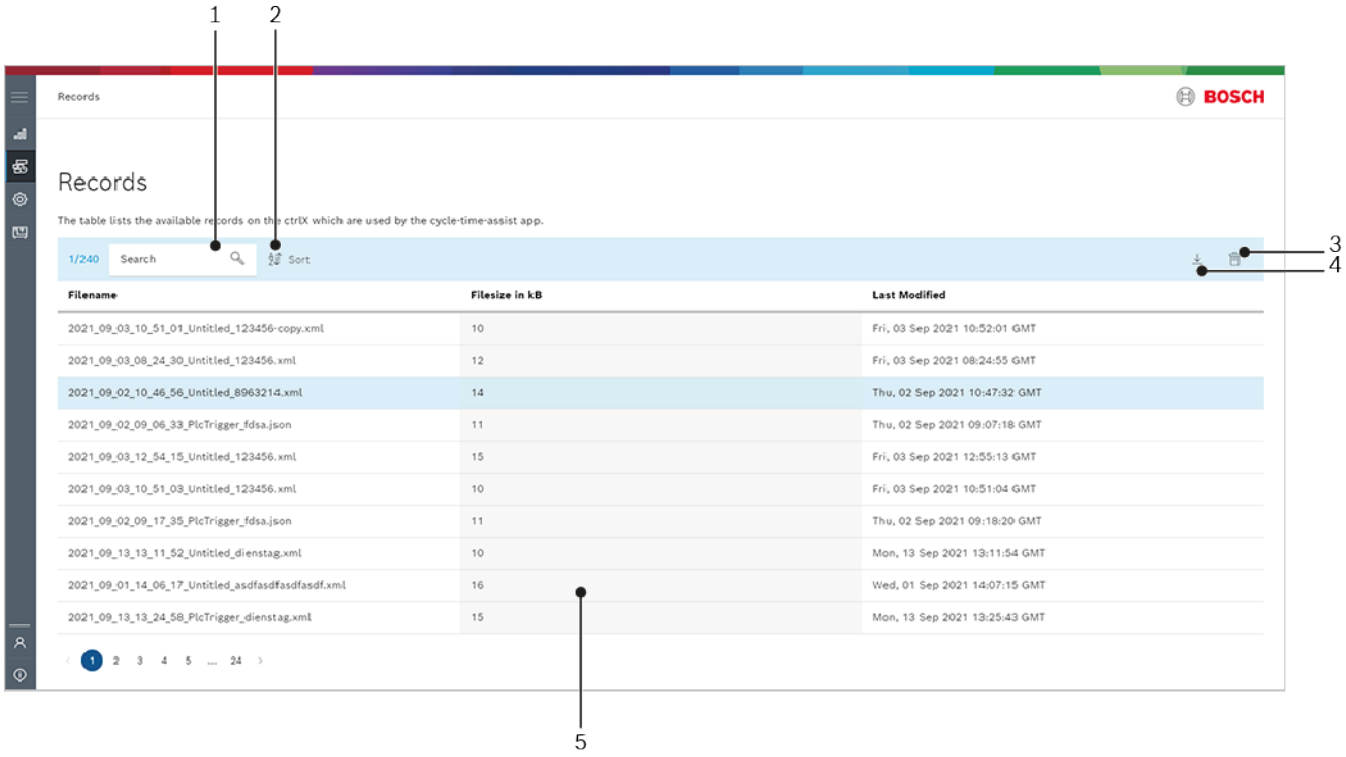
Under Global reference cycle, enter the ID of the selected part cycle. The screenshot corresponds to ID 134.

3. Click on *Reference cycle*.

A confirmation appears and the part cycle is set as a reference and can now be compared with another part cycle. Compare part cycles with one another (Comparing Part Cycles With One Another see page 40)

5.2.10 Overview of All Recordings

All previous recordings are displayed in the overview. The recordings can be edited in this overview.



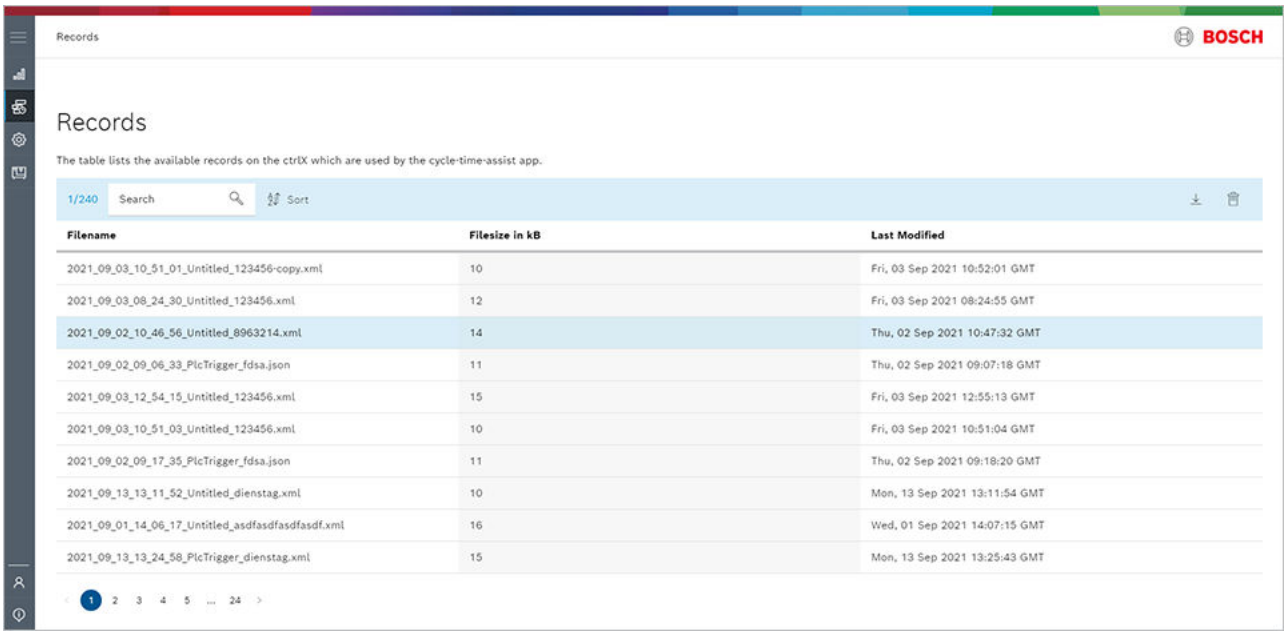
No.	Description
1	Close the view and return to the Live view (Live-Ansicht see page 33).
2	Entry of a recording
3	Reference cycle
4	Part cycle of a recording
5	Mark the recording Recordings can be marked for easy retrieval.
6	Edit the recording
7	Export the recording (Exporting a Recording see page 45)
8	Delete the recording (Deleting a Recording see page 46)
9	Open the recording so that all recorded part cycles are visible.

5.2.10.1 Exporting a Recording

Recordings can be exported for further analysis in the Cycle Time Analysis tool.

The recording can be exported in the Comparison view (see page 39) as well as in the Overview of recordings (Overview of All Recordings see page 44).

1. In the overview of recordings, select the recordings to be exported and click on .



   The Save dialog box appears.

2. Save the export file under any path.

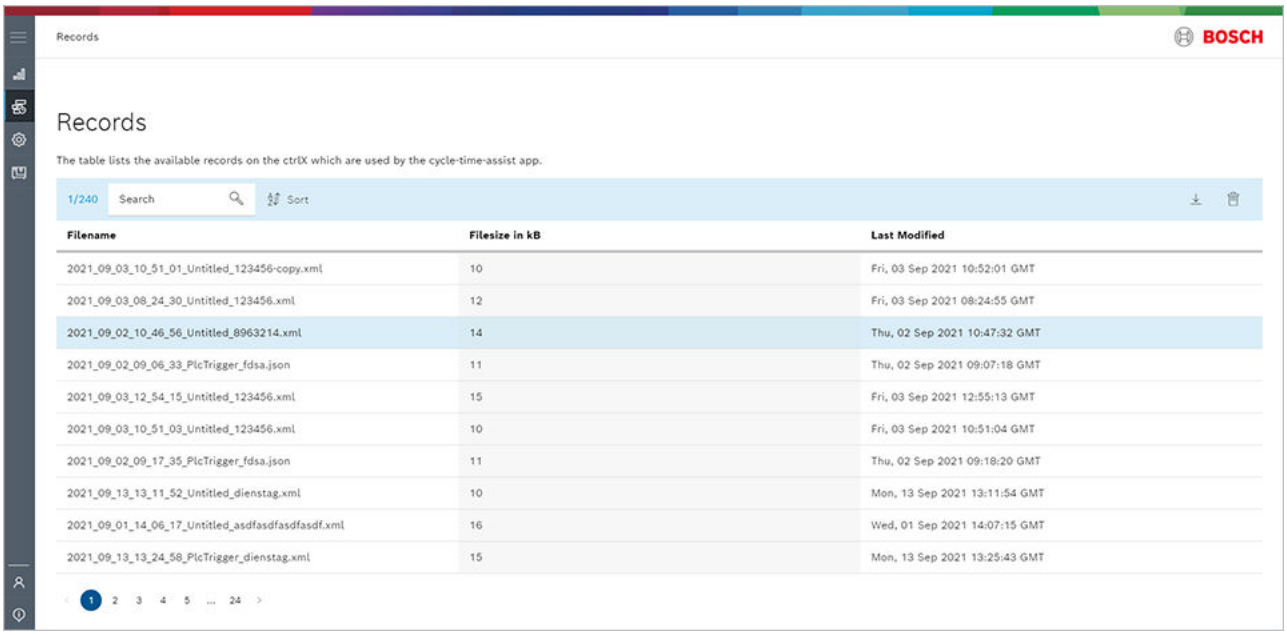
The recording is exported and can be opened in the Cycle Time Analysis Tool.

5.2.10.2 Deleting a Recording

Recordings can be deleted. However, deleted recordings cannot be recovered.

The recording can be exported in the Comparison view (see page 39) as well as the Overview of recordings (Overview of All Recordings see page 44).

1. In the overview of recordings, select the recordings to be deleted and click on .



   A security inquiry is displayed.

2. Confirm the inquiry.

The recording is deleted.

5.3 Cycle Time Assist FB

The product "Cycle Time Assist" allows the user to monitor and compare the cycle time of the machine as well as the runtimes of the movements and sequences of individual components (in the following commands). Cycle Time Assist Snap exists to collect and display this information. Snap is a service that receives and processes information. The information is displayed on a user interface. The source of this information is the PLC in which the machine processes have been programmed.

Cycle Time Assist offers libraries for the PLC. tThe application engineer must integrate these libraries in the project. The CycleTimeAssist_FB function block is located in the libraries. This FB is instantiated by the application engineer in the PLC project. The application engineer can program the FB in order to inform it of the start and end times of the commands, among other things. The FB itself has the ability to communicate this and other information to the service.

The following information can be processed by CycleTimeAssist_FB and forwarded to the Cycle Time Assist Snap service.

- Command start and command end times
- Part cycle signal
- Trigger to start a recording

5.3.1 Content of CycleTimeAssist_FB

This chapter describes the variables and methods of calling the CycleTimeAssist_FB.

5.3.1.1 Input Parameters

The Cycle Time Assist FB has various input parameters to describe.

If a parameter value changes, this is immediately transmitted to the service. However, changes to values do not apply until the next cycle interval. For the current cycle interval, the values that were available at the beginning of the clock apply.

Name	Type	Description
TargetPartCycleTime	TIME	Specification for the target part cycle
AllowedPosDeviation	TIME	Specification for the permissible positive deviation from the target part cycle, up to which duration the current cycle interval should be considered valid and should not be considered exceeded.
AllowedNegDeviation	TIME	Specification for the permissible negative deviation from the target part cycle, up to which duration the current cycle interval should be considered valid and should not be considered exceeded.
PartType	STRING(20)	Sets the name of the part type
AddText	STRING(255)	Appends additional text to the part type.

5.3.1.2 Output Parameters

The Cycle Time Assist FB has various output parameters to describe.

Name	Type	Description
Initialized	BOOL	Return value: TRUE = FB is initialized with communication parameters FALSE = FB is not initialized with the communication parameters.
ComState	CycleTimeAssistComStates	Current state of communication with the service This structure helps to identify communication problems. ▪ UNDEFINED : Value not set ▪ WAIT_FOR_FB_INITIALIZE : FB waits for the initialization method to be called. ▪ WAIT_FOR_CLIENT : FB is waiting for service or trying to connect to it. ▪ WAIT_FOR_CLIENT_IDENT : Connection to the service has been established. Waiting for service authentication. ▪ SEND_PLC_INFO : Send the PLC configuration to the service. ▪ WAIT_CLIENT_INFO : Waiting for service configuration information. ▪ READY_TO_SEND : Ready to send telegrams ▪ SEND_TELEGRAMS : Sends telegrams to the service ▪ WAIT_FOR_SEND_ACKN : Waiting for a confirmation of receipt from the service ▪ ERROR_TRY_RECONNECT : An error occurred. The FB attempts to re-establish the connection to the service.
RunningCommands	ARRAY[1..CYCLE_TIME_ASSIST_MAX_NUM_RUNNING_COMMANDS] OF CycleTimeAssistRunningCommands	Information about all currently running commands
NumRunningCommands	UDINT	Number of commands currently running

IsRecordRunning	BOOL	Return value: TRUE = at least one recording is running False = recording is not running

5.3.1.3 Method Calls

This section describes the methods of calling.

Some method calls have the return value DINT. In this case, call the method cyclically until it receives the return value 0.
If no return value exists, call the method only once.

Method name	Description	Parameter
CommandCancel	Tells the service that a specific component has canceled a command. If no command is running in the component, the method call has no effect.	▪ InstanceId : Instance number of the component in which the command was aborted.
CommandCancelAll	Informs the service that all running commands have been interrupted. For example, this method call can be used when canceling an automatic sequence when all components cancel their execution.	
CommandEnd	Tells the service that a specific component has terminated a command. If no command is running in the component, the method call has no effect.	▪ InstanceId : Instance number of the component in which the command was terminated.
CommandError	Tells the service that a particular component has terminated or canceled a command with an error. If no command is running in the component, the method call has no effect.	▪ InstanceId : Instance number of the component in which the command has an error and was terminated.
CommandStart	Tells the service that a specific component has started a command. Only one command at a time can run within a component. If another command is called via the CommandStart() method, this call is ignored.	▪ InstanceId : Instance number of the component in which the command was started. ▪ CommandNumber : Number of the command within the component.
EndTriggerRecord	Prematurely terminates a recording.	▪ RecordIndex : Index of the recording that is to be terminated. When triggering the recording using the StartTriggerRecord() method, the index is output.
SetPartCycle	Tells the service that a new part cycle interval is starting. The part cycle is the time difference between two method calls.	

StartTriggerRecord	Instructs the service to start a recording and save it in its database. Recordings can also be made retrospectively in time by setting method parameters. The maximum duration of a recording depends on the configuration of the service, default setting: 2 minutes.	▪ TimeBeforeTrigger : Specifies the length of time before the method call to include in the recording. ▪ TimeAfterTrigger : Specifies the length of recording time after the method call to be included. ▪ rRecordIndex : Index of the current recording. This is written after the method has been successfully executed. The index is used to identify a recording, which can be aborted prematurely with the method EndTriggerRecord(). Several recordings can be started in parallel.

5.3.2 Using CycleTimeAssist_FB

5.3.2.1 Required PLC Libraries

In addition to the ctrlX PLC libraries included, the following libraries must be installed and also integrated manually in the project:

- NxBaseSysDep
- NxCycleTimeAssistAddon
- NxSocketSysDep
- OpconBase
- OpconBaseCommonDef

The included libraries have dependencies to the base ctrlX libraries that need to be resolved.

5.3.2.2 Integrating the PlcOpenXML

After the station hierarchy has been created, the Cycle Time Assist Studio generates a PlcOpenXML file that can be read into the PLC development environment. Reading in the file adds three elements to the project.

- CycleTimeAssist_FB (type: FB): Function block from which a cyclically called instance must be created in the project. FB Instantiation and Cyclical Call (see page 57).
- CtaInstanceIds (type: GVL): This global variable list contains the instance IDs of all components configured in the Cycle Time Assist Studio. These entries can be used for method calls for the CycleTimeAssist_FB instance. Recording Command Run Times (Recording of Command Run Times see page 58).
  The names of the entries correspond to the assigned component names in upper case letters from the Cycle Time Assist Studio.
- CtaCommandIds (type: GVL): This global variable list contains the command IDs of all components configured in the Cycle Time Assist Studio. These entries can be used for method calls for the CycleTimeAssist_FB instance.Recording Command Run Times (Recording of Command Run Times see page 58).
  The names of the entries correspond to the assigned component names, followed by the assigned command names, in upper case letters, from the Cycle Time Assist Studio.

If changes are made to the project in the Cycle Time Assist Studio, such as changes to the hierarchy, instance IDs or command IDs, the listed elements must be deleted and the updated PlcOpenXML must be read in.

5.3.2.3 FB instantiation and cyclical call

Instantiation

To use the FB, it must be instantiated in the project. An instantiation can take place in a GVL, a program or FB.

Example:

```
PROGRAM PLC_PRG
VAR
_cta : CycleTimeAssist_FB;
END_VAR
```

cyclical call

Once the FB has been instantiated, it must be called cyclically during runtime. Ideally, the cyclical call is made at the beginning of the cycle.

Example:

```
_cta();
```

The CycleTimeAssist_FB is now ready to send information to the service regarding command start and command end times.

5.3.2.4 Recording of Command Run Times

The command start and command end times are passed on in CycleTimeAssist_FB by calling methods.

The methods are preceded by the Command prefix. The following methods are available. For more information, see the Method Calls section.

- All of these methods have no return value and therefore only need to be called once.
- Only one command at a time can run within a component.
  If another command is called via the CommandStart() method, it is ignored.

- CommandCancel()
- CommandCancelAll()
- CommandEnd()
- CommandError()
- CommandStart()

In addition to the CommandStart() command start method and the CommandEnd() command end method, the CommandCancel() and CommandError() methods can also be used to notify the service if a command has been terminated abnormally.

Example:

```
_cta.CommandStart(CtaInstanceIds.ROBOT,CtaCommandIds.ROBOT_MOVE);
```

The call informs the service that the MOVE command of the robot component has been started. The two transfer parameters result from the Import of the PlcOpenXML (Integrating the PlcOpenXML).

5.3.2.5 Triggering Recordings

The Cycle Time Assist FB has the option of triggering recordings from the PLC. Methods are called into the PLC, which internally give the service the command to start a recording.

Example:

```
_ret := _cta.StartTriggerRecord(TimeBeforeTrigger := T#5S, TimeAfterTrigger := T#10S, rRecordIndex := _recIndex);
```

This call triggers the recording. Call the method until the return value returns _ret = 0.

The recording includes the command runtimes. The command runtime includes the timespan of 5 seconds before the code line is called and 10 seconds after the code line is called.

This means that the total length of the recording is 15 seconds. The index of the recording is stored in _recIndex because several recordings can run in parallel.

The index is also relevant to terminating a recording with the *StopTriggerRecord()* method.

5.3.2.6 Setting the Part Cycle

The part cycle can be communicated to the service via a method call in the PLC. Do this by calling the method SetPartCycle(). The FB informs the service that a new part cycle has begun and the current one has ended.

6 Service – Helpdesk

If technical assistance is needed or if you require more information about our products and solutions, please contact the Help Desk via the following channels.

+49 (711) 811-8811
Monday to Friday 08:00–16:00 (GMT+1)

Nexeed.Helpdesk@bosch.com

www.bosch-connected-industry.com
Contact form

7 Index

Robert Bosch Manufacturing Solutions GmbH

Bosch Connected Industry (BCI)
P.O.Box 30 02 20
70442 Stuttgart
Germany
Phone +49 711 811 – 8811

www.bosch.com

www.bosch-connected-industry.com